



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
DIREKTORAT JENDERAL PENDIDIKAN VOKASI
DIREKTORAT SEKOLAH MENENGAH KEJURUAN

DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM



**SMK KELAS X
SEMESTER 1**

SMK KELAS X | i



Hak Cipta © 2021 pada Kementerian Pendidikan, Kebudayaan Riset dan Teknologi Dilindungi Undang- Undang

Katalog Dalam Terbitan (KDT)

Disklaimer:

Buku ini merupakan Buku yang disusun sebagai Dasar Kejuruan bagi Siswa SMK Program Keahlian Pengembangan Perangkat Lunak Dan Gim disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi. Buku ini merupakan "dokumen hidup" yang senantiasa diperbaiki, diperbarui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis dan laman <http://buku.kemdikbud.go.id> atau melalui email buku@kemdikbud.go.id diharapkan dapat meningkatkan kualitas buku ini.

Dasar-dasar Pengembangan Perangkat Lunak Dan Gim

Penulis : Yustiana Amita Utama
Esa Hergatama

Penelaah : Eko Subiyantoro

Editor : Tri Firmansyah

Desainer : Hendro soemarno

isi/Setter :

Penerbitan :
Pusat Kurikulum dan Perbukuan
Badan Penelitian dan Pengembangan dan Perbukuan
Kementerian Pendidikan, Kebudayaan Riset dan Teknologi
Jalan Gunung Sahari Raya No. 4 Jakarta Pusat

Cetakan ke-1, 2021

ISBN...

Buku ini menggunakan huruf Tahoma 12pt
hlm.



Kata Pengantar

Dalam menyediakan referensi materi pembelajaran bagi peserta didik di SMK, Direktorat Pembinaan SMK berupaya menyediakan buku siswa yang sesuai dengan kebutuhan pembelajaran di SMK bidang keahlian Teknologi Informasi pada mata pelajaran Dasar-Dasar Pengembangan Perangkat Lunak Dan Gim. Buku Siswa yang disusun pada tahun anggaran 2021 diharapkan dapat mewujudkan motivasi belajar bagi peserta didik di SMK bidang keahlian Teknologi Informasi. Karena buku siswa yang telah disusun ini selain menyajikan materi secara tertulis, juga dilengkapi dengan beberapa materi yang bersifat interaktif yang dapat memperluas pengetahuan dan keterampilan individu yang menggunakannya serta sikap sebagai makhluk yang mensyukuri anugerah alam semesta yang dikaruniakan kepadanya melalui pemanfaatan yang bertanggung jawab.

Buku siswa ini menjabarkan usaha minimal yang harus dilakukan siswa untuk mencapai kompetensi yang diharuskan. Sesuai dengan pendekatan yang digunakan dalam kurikulum merdeka belajar, siswa diberanikan untuk mencari dari sumber belajar lain yang tersedia dan terbentang luas di sekitarnya. Peran guru sangat penting untuk meningkatkan dan menyesuaikan daya serap siswa dengan ketersediaan kegiatan buku ini.

Buku siswa ini terus dilakukan perbaikan dan penyempurnaan. Untuk itu kritik, saran, dan masukan dari para pembaca untuk perbaikan dan penyempurnaan selalu kami harapkan. Atas kontribusi tersebut, kami ucapkan terima kasih. Semoga kita dapat memberikan yang terbaik bagi kemajuan dunia pendidikan di Indonesia.



Prakata

Puji syukur kami panjatkan ke hadirat Allah SWT. atas limpahan Rahmat dan Hidayah-Nya, sehingga kami dapat menyelesaikan penyusunan Buku Dasar-Dasar Pengembangan Perangkat Lunak Dan Gim Kelas X Semester 1 (satu) sebagai Dasar Kejuruan SMK Program Pengembangan Perangkat Lunak Dan Gim pada Bidang Keahlian Teknologi Informasi..

Sekolah Menengah Kejuruan memiliki peran yang sangat besar dalam menyiapkan peserta didik agar siap bekerja baik bekerja didalam negeri maupun keluar negeri. Oleh karena itu, arah pengembangan pendidikan menengah kejuruan diorientasikan pada permintaan industri.

Upaya untuk peningkatan mutu pendidikan melalui perbaikan mutu proses pembelajaran merupakan inovasi pendidikan yang harus terus dilakukan. Salah satu sarana yang paling penting adalah penyediaan buku pelajaran sebagai rujukan yang baik dan benar bagi siswa. Penyertaan buku ini sangat penting karena buku teks pelajaran merupakan salah satu sarana yang signifikan dalam menunjang proses kegiatan pembelajaran.

Berdasarkan hal tersebut diatas , kami berharap semoga buku ini dapat dijadikan sebagai referensi untuk pegangan siswa dalam menambah wawasan dan pengetahuan dalam kegiatan belajar .Kami menyadari masih banyak kekurangan dan kelemahan karena terbatasnya kemampuan kami dalam menyusun buku ini ,oleh karenanya kami menerima berbagai kritik dan saran yang sifatnya membangun untuk kesempurnaan buku ini di masa yang akan datang

Jakarta, Juni 2021

SMK KELAS X | iv



DAFTAR ISI

Cover	
Preliminari	
Halaman	
Judul	i
Hak Cipta	ii
Kata Pengantar	iii
Prakata	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	xvi
Petunjuk Penggunaan Buku	xvii
PENDAHULUAN	xx
A. Rasional	xx
B. Cakupan atau Ruang Lingkup	xxi
C. Tujuan Mata Pelajaran	xxi
D. Pendekatan/Strategi Pembelajaran	xxii
E. Media Pembelajaran	xxiii
F. Evaluasi Pembelajaran	xxiii
Konsep Dasar Pengembangan Perangkat Lunak Dan Gim	1
Tujuan Pembelajaran	2
Peta Konsep	3
Bab I Proses Bisnis, Perkembangan Dunia Kerja dan Profesi	
Pengembang Perangkat Lunak dan Gim	4
A. Konsep-Konsep Dasar Pengembangan Perangkat Lunak dan Gim	5
B. Budaya Mutu dan Manajemen Proyek Pengembangan Perangkat Lunak dan Gim	74
C. Perkembangan Teknologi Industri 4.0	93
D. Profil dan Profesi Bidang Pengembang Perangkat Lunak dan Gim	105
E. Refleksi	120
F. Rangkuman	121
G. Pengayaan	124



Aset Desain	125
Tujuan Pembelajaran	126
Peta Konsep	127
Bab II Aset Desain Pada Perangkat Lunak Dan Gim	128
A. Konsep dan Tools Pembuatan Aset Desain dalam Pengembangan Perangkat Lunak dan Gim	129
B. Dasar-Dasar Desain dalam Pembuatan Aset Desain	135
C. Aset Grafis Berbasis Vector	147
D. Aset Grafis Berbasis Bitmap	224
E. Refleksi	233
F. Pengayaan	233
 Pemrograman Dasar.....	234
Tujuan Pembelajaran	235
Peta Konsep	235
Bab III Pemrograman Dasar	236
A. Algoritma dan Pemrograman	237
B. Pengenalan C#	245
C. Struktur Kode C#	254
D. Membuat Program Sederhana	268
E. Operator Aritmatika dan Logika	270
F. Struktur Kontrol	275
G. Procedure.....	283
H. Mengenal Array	285
I. Aktifitas Pembelajaran	295
J. Refleksi	296
K. Pengayaan	297
 Daftar Rujukan	298
Indeks	300
Glosarium	302
Biodata Penulis	306
Biodata Penelaah	308
Biodata Editor	310
Biodata Desainer	311



DAFTAR GAMBAR

Gambar 1. Diagram Pencapaian Kompetensi	xix
Gambar 1.1 Perangkat Lunak Microsoft Word	5
Gambar 1.2 Game Super Mario Bros	7
Gambar 1.3 Pascalene The First Mechanical adding Machine.....	9
Gambar 1.4 ENIAC (Electronic Numerical Integrator And Computer)... Gambar 1.5 IBM 701 Defense with Calculator.....	9
Gambar 1.6 Apple Lisa	11
Gambar 1.7 Perkembangan Telepon Selular	12
Gambar 1.8 Permainan <i>duck shoot game</i> oleh Ray-O-Lite Rifle, Seeburg	13
Gambar 1.9 Permainan Simulator <i>Rudal Cathode Ray Tube Device Amusement</i>	14
Gambar 1.10 Permainan OXO <i>Game</i>	14
Gambar 1.11 Mesin <i>Pinball</i> SEGA	15
Gambar 1.1 Permainan Tennis For Two	15
Gambar 1.2 Permainan Space War	16
Gambar 1.3 Permainan The Brown Box	16
Gambar 1.4 Spacewars! Online	17
Gambar 1.5 PDP Space Wars!	18
Gambar 1.6 Magnov Odyssey	19
Gambar 1.7 Permainan Atari Pong	19
Gambar 1.8 Grand Track 10	20
Gambar 1.9 Permainan Pong Eksklusif	21
Gambar 1.10 Console Game Fairchild dan Apple I	21
Gambar 1.11 Atari 2600 dan Nintendo Color TV Game	22
Gambar 1.12 PDP-10 Minicomputer	22
Gambar 1.13 Tandy TRS80 dan Commodore PET	23
Gambar 1.14 Apple II	23
Gambar 1.15 Permainan Game Space Invaders dan Odyssey 2	23
Gambar 1.16 Handheld Console Cosmos oleh Atari	24
Gambar 1.17 Nintendo Game and watch	25
Gambar 1.29 PC Pertama IBM	25
Gambar 1.30 Atari 5200 dan Commodore 64	26
Gambar 1.31 Dragon's Lair menggunakan Teknologi Laserdisc	27



Gambar 1.32 Game Console Nintendo Famicom	27
Gambar 1.183 Permainan Karate Champ dan Apple Personal Computer Machintosh	28
Gambar 1.194 Windows 1.0 dan Quantumlink layanan public online pertama	28
Gambar 1.35 Nintendo NES	29
Gambar 1.36 Game Console Atari 7800 dan Sega Master System	29
Gambar 1.37 Game Console SEGA Genesis	29
Gambar 1.38 Game Console Nintendo Game Boy	30
Gambar 1.39 Game Console Turbo Express	30
Gambar 1.40 Game Console Nintendo SNES	31
Gambar 1.41 CD ROM PC	31
Gambar 1.42 Atari Jaguar	32
Gambar 1.43 Permainan Cyan Myst	32
Gambar 1.44 Sony Playstation 1	33
Gambar 1.20 Virtual Boy	33
Gambar 1.46 Nintendo 64	34
Gambar 1.47 Tamahotchi Original	34
Gambar 1.48 Nintendo Game Boy Color	35
Gambar 1.49 Sony Playstation 2	35
Gambar 1.50 Nintendo GBA	36
Gambar 1.51 Playstation Portable dan Nintendo DS	36
Gambar 1.52 Xbox 360 dan Xbox Live Membership	37
Gambar 1.53 Nintendo Wii dan Playstation 3	37
Gambar 1.54 Permainan Angry Bird	38
Gambar 1.55 Xbox 360 Kinect	38
Gambar 1.56 Nintendo 3ds	39
Gambar 1.57 Playstation 4 dan Xbox One	39
Gambar 1.58 Generasi Pertama PlayStation VR	40
Gambar 1.59 Nintendo Switch	40
Gambar 1.60 Nintendo Switch dan Labo	41
Gambar 1.61 Xbox Series X	42
Gambar 1.62 Playstation 5	42
Gambar 1.63 Aspek <i>Game</i>	44
Gambar 1.64 Perangkat Lunak Sistem (Sistem Operasi)	45



Gambar 1.65 Code Editor atau IDE Contoh Perangkat Lunak Pemrograman	46
Gambar 1.66 Dashboard Zahir Aplikasi Accounting	47
Gambar 1.67 Aplikasi GES (Alur Kerja Bisnis / ERP)	48
Gambar 1.68 Tiktok Perangkat Lunak Media dan Hiburan	48
Gambar 1.69 <i>Google Class Room</i> , Perangkat Lunak Pendidikan	49
Gambar 1.70 Corel Draw Suite 2021	49
Gambar 1.71 Tampilan Permainan Pacman	50
Gambar 1.72 Tampilan Permainan HIVE	51
Gambar 1.73 Tampilan Permainan Solitaire	51
Gambar 1.74 Battle Card Game Pokemon	52
Gambar 1.75 Quiz Game Who Wants To Be A Millionre	52
Gambar 1.76 Puzzle Game Tetris	53
Gambar 1.77 Shoot Them Up Space Impact	53
Gambar 1.78 Side Scroller Game Mario Bros 3D	54
Gambar 1.79 Street Fighter	54
Gambar 1.80 Racing Game Project Cars Go	55
Gambar 1.81 Turn Based Strategy Gear Tactics	55
Gambar 1.82 Real Time Strategy Warcraft III	56
Gambar 1.83 Euro Truck Simulator 2	56
Gambar 1.84 First Person Shooter Counterstrike	57
Gambar 1.85 First Person Shooter 3D Firestorm	57
Gambar 1.86 Third Person 3D Game	58
Gambar 1.87 Role Playing Game Beyond Divinity	58
Gambar 1.88 Adventure Beyond Good and Evil 2	59
Gambar 1.89 Educational and Edutainment	59
Gambar 1.90 Game Sport Super Mega Baseball 3	60
Gambar 1.91 Event BPMN	63
Gambar 1.92 Activity BPMN	64
Gambar 1.93 Gateway BPMN	64
Gambar 1.94 Sequence Flow	64
Gambar 1.95 Message Flow	65
Gambar 1.96 Association	65
Gambar 1.97 Pool	65
Gambar 1.98 Lane	66
Gambar 1.99 Data Object	66



Gambar 1.100 Group	67
Gambar 1.211 Manajemen Projek Tree	74
Gambar 1.102 Elemen TELOS	80
Gambar 1.103 Posisi duduk	88
Gambar 1.104 Cyber <i>Security</i>	98
Gambar 2. 1 Gambar Taman Bunga.....	128
Gambar 2. 2 Penerapan multimedia grafik, teks dan animasi dalam aplikasi starbucks	130
Gambar 2. 3 Peningkatan Kualitas gambar	130
Gambar 2. 4 Contoh Komponen Desain Titik	136
Gambar 2. 5 Contoh komponen desain garis	137
Gambar 2. 6 Bentuk – Bentuk Geometris	138
Gambar 2. 7 Bentuk – bentuk Organis	139
Gambar 2. 8 Tampilan Adobe Illustrator 2021	147
Gambar 2. 9 Memilih Template dari Kelompok Web	148
Gambar 2. 10 Width dan Height menentukan ukuran artboards	148
Gambar 2. 11 Pengaturan Bleed	149
Gambar 2. 12 Pengaturan Color Mode	149
Gambar 2. 13 Tampilan dokumen baru pada Adobe Illustrator.....	149
Gambar 2. 14 Memasukan Gambar pada Artboard	150
Gambar 2. 15 Toolbar Document Setup	151
Gambar 2. 16 Document Setup	151
Gambar 2. 17 Transparency Grid dalam posisi aktif	152
Gambar 2. 18 Document Setup tab (Type)	153
Gambar 2. 19 Memilih Document Color Mode	154
Gambar 2. 20 Simpan file ke dalam format .AI (Mac Os)	155
Gambar 2. 21 Tampilan pada saat membuka dokumen (Mac Os)	157
Gambar 2. 22 Pengaturan jumlah artboard saat pertama membuat dokumen	158
Gambar 2. 23 Icon Menu Artboards	159
Gambar 2. 24 Icon Untuk Menambah Artboards	159
Gambar 2. 25 Artboards baru telah ditambah	159
Gambar 2. 26 Tampilan Adobe Illustrator saat memiliki 2 Artboards ...	160
Gambar 2. 27 Tampilan Workspace Essential Classic.....	161
Gambar 2. 28 Tampilan Workspace Web	162
Gambar 2. 29 Memilih Menu Interface	162



Gambar 2. 30 Adobe Illustrator Light Mode	163
Gambar 2. 31 Toolbar Zoom Tool	164
Gambar 2. 32 Menentukan Zoom Level melalui Status Bar	164
Gambar 2. 33 Fit On Screen membantu melihat artboards secara utuh	165
Gambar 2. 34 Hand Tool (H)	165
Gambar 2. 35 Layers	167
Gambar 2. 36 Create New Layer.....	167
Gambar 2. 37 Tampilan saat Layer baru telah dibuat	167
Gambar 2. 38 Objek - objek di dalam dokumen	168
Gambar 2. 39 Tampilan Objek-objek pada masing-masing layers	168
Gambar 2. 40 Toolbar Adobe Illustrator	168
Gambar 2. 41 Memilih ukuran stroke yang diinginkan.....	169
Gambar 2. 42 Pilihan width profile untuk stroke.....	169
Gambar 2. 43 Objek dengan berbagai variable width profile di dalam dokumen	170
Gambar 2. 44 Brush Definition Properties	170
Gambar 2. 45 Berbagai objek dengan Brush Definition di dalam dokumen	170
Gambar 2. 46 Rectangle Tool (M).....	171
Gambar 2. 47 Objek persegi empat di dalam dokumen	171
Gambar 2. 48 Membuat persegi empat secara presisi	172
Gambar 2. 49 Rounded Rectangle Tool	172
Gambar 2. 50 Objek Rounded Rectangle di dalam dokumen	172
Gambar 2. 51 Toolbar Shape	173
Gambar 2. 52 Shape Properties	173
Gambar 2. 53 Persegi Empat dengan nilai corner yang tinggi.....	173
Gambar 2. 54 Ellipse Tool (L).....	174
Gambar 2. 55 Objek Ellipse di dalam dokumen	174
Gambar 2. 56 Polygon Tool.....	175
Gambar 2. 57 Objek Polygon di dalam dokumen.....	175
Gambar 2. 58 Toolbar Shape	175
Gambar 2. 59 Polygon Shape Properties	176
Gambar 2. 60 Objek Polygon dengan berbagai sisi di dalam dokumen	176
Gambar 2. 61 Star Tool	177
Gambar 2. 62 Objek Bintang di dalam dokumen	177



Gambar 2. 63 Line Segement Tool (/).....	177
Gambar 2. 64 Objek Line di dalam dokumen	178
Gambar 2. 65 Arc Tool	178
Gambar 2. 66 Objek Arc Tool di dalam dokumen	178
Gambar 2. 67 Spiral Tool	179
Gambar 2. 68 Objek Spiral di dalam dokumen	179
Gambar 2. 69 Rectangular Grid Tool	179
Gambar 2. 70 Objek Rectangular Grid Tool di dalam dokumen.....	180
Gambar 2. 71 Polar Grid Tool.....	180
Gambar 2. 72 Buat objek	180
Gambar 2. 73 Berbagai Objek Chart di Adobe Illustrator.....	181
Gambar 2. 74 Berbagai tool untuk menambahkan symbol	183
Gambar 2. 75 Panel Symbols	183
Gambar 2. 76 Objek Symbols di dalam dokumen	184
Gambar 2. 77 Symbol Libraries Menu	184
Gambar 2. 78 Pen Tool (P)	185
Gambar 2. 79 Pilihan Stroke Color	186
Gambar 2. 80 Membuat garis lurus menggunakan Pen Tool	186
Gambar 2. 81 Garis lengkung dibuat dengan metode klik-drag.....	187
Gambar 2. 82 Asset Vector yang dibuat dari Pen Tool	187
Gambar 2. 83 Direct Selection Tool (A)	188
Gambar 2. 84 Mengatur titik menggunakan Direct Selection Tool	188
Gambar 2. 85 Convert selected anchor points to smooth	189
Gambar 2. 86 Kanan adalah objek setelah di convert selected anchor points to smooth	189
Gambar 2. 87 Convert selected anchor points to corner	190
Gambar 2. 88 Kanan adalah objek setelah di convert anchor point to corner.....	190
Gambar 2. 89 Cut path at selected anchor points.....	190
Gambar 2. 90 Hasil object cut path at selected anchor points di dadalm dokumen	191
Gambar 2. 91 Pen Tool (P)	191
Gambar 2. 92 Add Anchor Point Tool (+)	192
Gambar 2. 93 Titik baru berhasil ditambahkan kepada garis pada objek bagian bawah.....	192
Gambar 2. 94 Delete Anchor Point Tool (-).....	192



Gambar 2. 95 Contoh objek yang telah di hapus titik nya ada di sebelah kanan	193
Gambar 2. 96 Anchor Point Tool (Shift+C)	193
Gambar 2. 97 Sebelah kanan adalah objek yang sudah di tambahkan anchor point.....	194
Gambar 2. 98 Curvature Tool (Shift + `).....	194
Gambar 2. 99 Object Curvature di dalam dokumen	195
Gambar 2. 100 Eraser Tool (Shift + E).....	195
Gambar 2. 101 Objek yang telah dihapus bagian tengahnya oleh eraser tool	196
Gambar 2. 102 Scissors Tool (C)	197
Gambar 2. 103 Perbandingan objek setelah dilakukan scissiors tool.....	197
Gambar 2. 104 Knife Tool	197
Gambar 2. 105 Hasil Objek yang dilakukan Knife Tool	198
Gambar 2. 106 Objek Lingkaran dan Bintang.....	198
Gambar 2. 107 Devide Objects Below	199
Gambar 2. 108 Hasil Devide Objects Below	199
Gambar 2. 109 Objek yang akan di reshape	200
Gambar 2. 110 Hasil Objek yang telah di reshape.....	200
Gambar 2. 111 Paintbrush Tool	201
Gambar 2. 112 Hasil Objek menggunakan paintbrush tool	201
Gambar 2. 113 Brush Definition	201
Gambar 2. 114 Blob Brush Tool (Shift+B)	202
Gambar 2. 115 Objek Hasil Bloob Brush Tool.....	202
Gambar 2. 116 Pencil Tool (N)	203
Gambar 2. 117 Objek Hasil Pencil Tool.....	203
Gambar 2. 118 Smooth Tool	204
Gambar 2. 119 Perbedaan Hasil Objek dengan smooth tool	204
Gambar 2. 120 Path Eraser Tool.....	205
Gambar 2. 121 Hasil Objek yang dihapus menggunakan Path Eraser Tool	205
Gambar 2. 122 Join Tool.....	206
Gambar 2. 123 Hasil Objek setelah di join tool.....	206
Gambar 2. 124 Contoh dua objek yang bersinggungan.....	206
Gambar 2. 125 Shape Builder Tool (Shift + M)	207
Gambar 2. 126 Objek dipisah sebelum menggunakan Shape Builder ..	207



Gambar 2. 127 Objek dipisah setelah menggunakan Shape Builder	207
Gambar 2. 128 Selection Tool	208
Gambar 2. 129 Tampilan saat berbagai objek di seleksi.....	208
Gambar 2. 130 Select Similiar Objects pada Toolbar	209
Gambar 2. 131 Lasso Tool (Q)	210
Gambar 2. 132 Tampilan saat menyeleksi menggunakan lasso tools...	211
Gambar 2. 133 Magic Wand Tool (Y)	211
Gambar 2. 134 Pilihan Menu Select.....	212
Gambar 2. 135 Tampilan Menu Layer	213
Gambar 2. 136 Objek berwarna hitam terseleksi.....	214
Gambar 2. 137 Mengatur ukuran objek menggunakan Selection Tools	214
Gambar 2. 138 Scale Tool (S)	215
Gambar 2. 139 Mengatur ukuran objek menggunakan scale tool.....	215
Gambar 2. 140 Kotak isian width dan height pengatur ukuran	216
Gambar 2. 141 Contoh Objek Merah yang diputar dengan Reference Point di tengah	217
Gambar 2. 142 Share Tool.....	217
Gambar 2. 143 Hasil objek menggunakan shear tool.....	218
Gambar 2. 144 Rotate Tool (R)	218
Gambar 2. 145 Reflect Tool (O).....	218
Gambar 2. 146 Objek dengan Reflect Tool	218
Gambar 2. 147 Free Transform (E)	219
Gambar 2. 148 Tampilan Objek dengan jarak beragam	220
Gambar 2. 149 Berbagai macam jenis distribute	221
Gambar 2. 150 Hasil Objek dengan Distribut Vertical	221
Gambar 2. 151 Berbagai jenis align	221
Gambar 2. 152 Hasil Penerapan Align Center.....	222
Gambar 2. 153 Perbandingan objek lingkaran ketika menggunakan Bring to Front	222
Gambar 3.1 Symbol Flowchart	239
Gambar 3.2 Contoh Flowchart Penjualan.....	240
Gambar 3.3 Contoh Flowchart Runtutan.....	241
Gambar 3.4 Contoh Flowchart Pemilihan	242
Gambar 3.5 Contoh Flowchart Pengulangan	243
Gambar 3.6 Download .NET SDK x64.....	246
Gambar 3.7 License Agreement VS Code.....	247



Gambar 3.8 Select Additional Tasks	247
Gambar 3.9 Ready to Install	248
Gambar 3.10 Proses Install	248
Gambar 3.11 Instalasi Selesai	249
Gambar 3.12 Pilih Extensions C#	250
Gambar 3.13 Install Extensions C#	250
Gambar 3.14 Hasil Instalasi C#	251
Gambar 3.15 Pembuatan Folder	251
Gambar 3.16 Tampilan Command Prompt	252
Gambar 3.17 Tampilan Perintah Dotnet New Console.....	252
Gambar 3.18 Tampilan Perintah Code.....	252
Gambar 3.19 Tampilan Required Assets	253
Gambar 3.20 Tampilan Folder .vscode	253
Gambar 3.21 Tampilan File Program .cs	253
Gambar 3.22 Output HelloWorld.....	254
Gambar 3.23 Lokasi Pembangunan.....	257
Gambar 3.24 Contoh Konstanta Bernama.....	266
Gambar 3.25 Contoh Menggunakan Input dan Output.....	268
Gambar 3.26 Contoh Penggunaan Operator Aritmatika.....	270
Gambar 3.27 Contoh Penggunaan And	272
Gambar 3.28 Contoh OR.....	273
Gambar 3.29 Contoh Percabangan If	275
Gambar 3.30 Contoh Percabangan If-Else	276
Gambar 3.31 Contoh If-Else-If	277
Gambar 3.32 Contoh Switch-Case.....	278
Gambar 3.33 Contoh Perintah For	279
Gambar 3.34 Contoh Perulangan While.....	279
Gambar 3.35 Contoh Perulangan Do-While.....	280
Gambar 3.36 Output Procedure.....	285
Gambar 3.37 Elemen Array	285
Gambar 3.38 Output Array Satu Dimensi dan Dua Dimensi	291



DAFTAR TABEL

Tabel 1 Daftar Ikon	xviii
Tabel 1.1 Elemen dan Objek BPMN	63
Tabel 1.2 Faktor Penentu Kualitas Perangkat Lunak	83
Tabel 3.1 Daftar Karakter Khusus C#	256
Tabel 3.2 Empat Kombinasi untuk Kondisi 2 bit	258
Tabel 3.3 Tabel Jenis Tipe Data	258
Tabel 3.4 Tabel Float dan Double	259
Tabel 3.5 Pembacaan Tipe Data	260
Tabel 3.6 Konstanta Heksadesimal	263
Tabel 3.7 Symbol Operator	264
Tabel 3.8 Operator Aritmatika	270
Tabel 3.9 Tabel Operasi Logika	271
Tabel 3.10 Tabel Kebenaran Operator And	271
Tabel 3.11 Tabel Kebenaran Operator OR	272
Tabel 3.12 Tabel Kelulusan Siswa	289



Petunjuk Penggunaan Buku

Buku Dasar-Dasar Pengembangan Perangkat Lunak ini disusun untuk siswa SMK Kelas X Semester 1 Bidang Keahlian Teknologi Informasi, sebagai mata pelajaran dasar kejuruan untuk Program Keahlian Pengembangan Perangkat Lunak Dan Gim.

Buku siswa ini disusun tidak hanya membahas mengenai konsep dasar dalam memulai proses bisnis pada Pengembangan Perangkat Lunak Dan Gim saja melainkan perkembangan teknologi yang digunakan, proses kerja dan isu-isu global yang terkait dengan perkembangan teknologi saat ini. Mapel ini sangat diperlukan sebagai bagian dari pembelajaran disekolah dan relevan dengan uraian tugas di DUDIKA yang terdiri dari berbagai ilmu dasar sebagai penentu dalam mempelajari mata pelajaran yang lain pada Program keahlian Pengembangan Perangkat Lunak Dan Gim.

Untuk mengerti dan memahami Buku Dasar-Dasar Pengembangan Perangkat Lunak Dan Gim untuk Semester 1 (satu) ini terbagi dalam tiga Bab yaitu pada Bab 1 dibahas mengenai proses bisnis, perkembangan dunia kerja dan profesi pengembang perangkat lunak dan gim, Bab 2 membahas mengenai aset desain pada perangkat lunak dan gim, Bab 3 membahas mengenai pemrograman dasar.

Buku siswa disusun dengan ilustrasi gambar, diagram, dan tabel sederhana untuk keperluan siswa SMK kelas X Semester 1 (satu) yang disesuaikan dengan karakteristik siswa.



PETUNJUK PENGGUNAAN

Didalam buku ini kalian akan menemukan gambar-gambar sebagai penanda kegiatan pembelajaran yang akan dilakukan, cermati gambar-gambar berikut berserta artinya:

Tabel 1. Daftar Ikon

	Tujuan Pembelajaran	Gambar ini menunjukkan tujuan pembelajaran pada setiap bab yang akan dipelajari
	Apersepsi	Gambar ini menunjukkan apersepsi untuk menunjukkan gambaran dari materi sebelum kalian mempelajari lebih dalam.
	Kata Kunci	Gambar ini menunjukkan saatnya memahami kata-kata utama dalam materi yang akan dipelajari
	Refleksi	Gambar ini menunjukkan saatnya menelaah kumpulan singkat dari materi yang sebelumnya dipelajari
	Aktifitas Pembelajaran	Gambar ini menunjukkan saatnya kalian mengerjakan test
	Pengayaan	Gambar ini menunjukkan saatnya mengukur kembali kemampuan yang didapat dalam mempelajari buku ini



DIAGRAM PENCAPAIAN KOMPETENSI



Gambar 1. Diagram Pencapaian Kompetensi



PENDAHULUAN

A. Rasional

Dasar-dasar Pengembangan Perangkat Lunak dan Gim adalah mata pelajaran yang berisi kompetensi-kompetensi yang mendasari penguasaan keahlian pengembangan perangkat lunak dan teknologi *game*. Mata pelajaran ini dilengkapi dengan kemampuan memahami kebutuhan dan keinginan pelanggan serta *User Experience (UX)* dalam proses desain sebagai penerapan prinsip *Customer Oriented*.

Mata pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim berfungsi membekali peserta didik dengan seperangkat pengetahuan, keterampilan, dan sikap agar memiliki dasar yang kuat dalam mempelajari mata pelajaran-mata pelajaran pada konsentrasi keahlian di kelas XI dan XII. Lingkup materi mata pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim meliputi wawasan di bidang pengembangan perangkat lunak, gim, dan pemrograman.

Peserta didik juga dibekali kemampuan melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran, melalui berbagai aktivitas proses saintifik sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah. Dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri serta memahami dan menerapkan aspek *digital consumer behaviour*.

Mata pelajaran ini berkontribusi dalam membentuk peserta didik memiliki keahlian pada bidang pengembangan perangkat lunak dan gim, meningkatkan lebih lanjut kemampuan logika dan teknologi digital (*computational thinking*), yaitu suatu cara berpikir yang memungkinkan



untuk menguraikan suatu masalah menjadi beberapa bagian yang lebih kecil dan sederhana, menemukan pola masalah, serta menyusun langkah-langkah solusi mengatasi masalah sebagai implementasi *customer oriented*. Penguasaan kemampuan dasar-dasar pengembangan perangkat lunak dan gim akan membiasakan peserta didik bernalar kritis dalam menghadapi permasalahan, bekerja mandiri, serta kreatif dalam menemukan solusi permasalahan kehidupan.

B. Cakupan atau Ruang Lingkup

Mata pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim memiliki elemen materi sebagai berikut: Proses bisnis di dunia kerja bidang perangkat lunak dan gim, wawasan perkembangan dunia kerja bidang perangkat lunak dan gim, *technopreneurship* dan *job-profile* di bidang industri perangkat lunak dan gim, orientasi pengembangan perangkat lunak dan gim, pemrograman terstruktur, dan pemrograman berorientasi objek.

Pengembangan *soft skills* pada mata pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim sangat penting sebagai pembekalan dasar di dalam membangun etos kerja, meliputi komunikasi, *critical thinking*, kolaborasi, dan kreativitas. Mata Pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim merupakan pondasi dalam pemodelan, implementasi, dokumentasi serta testing perangkat lunak.

C. Tujuan Mata Pelajaran

Mata pelajaran Dasar-dasar Pengembangan Perangkat Lunak dan Gim bertujuan membekali peserta didik dengan dasar-dasar



pengetahuan, keterampilan, dan sikap (*hard skills* dan *soft skills*) yang diarahkan untuk mengembangkan kemampuan sebagai berikut :

1. Memahami proses bisnis di bidang industri pengembangan perangkat lunak dan gim;
2. Mampu mengembangkan wawasan tentang perkembangan teknologi dan isu-isu global bidang perangkat lunak dan gim;
3. Memahami profesi dan kewirausahaan (*job profile* dan *technopreneurship*) serta peluang usaha di bidang industri perangkat lunak dan gim;
4. Memahami lingkup kerja bidang pengembangan perangkat lunak dan gim;
5. Memahami pemrograman terstruktur dan pemrograman berorientasi obyek.

D. Pendekatan/Strategi pembelajaran

Pada awal pembelajaran peserta didik dikenalkan kepada lapangan kerja, jabatan kerja yang dapat dimasuki setelah lulus, dan konsentrasi-konsentrasi keahlian yang dapat dipelajari pada kelas XI dan XII. Untuk menumbuhkan *passion* (renjana), *vision* (visi), imajinasi, dan kreativitas dapat dikembangkan berbagai aktivitas pembelajaran antara lain sebagai berikut:

1. Pembelajaran di kelas;
2. Pembelajaran di laboratorium;
3. Projek sederhana;
4. Berinteraksi dengan alumni yang sudah berkarir dan praktisi industri;
5. Berkunjung ke industri yang relevan;
6. Pencarian informasi melalui media digital.



E. Media Pembelajaran

Media pembelajaran merupakan alat bantu yang dipergunakan untuk menyampaikan materi ajar kepada peserta didik. Media ini bisa berupa media fisik atau non fisik.

F. Evaluasi Pembelajaran

Penilaian meliputi aspek pengetahuan melalui tes dan non-tes, aspek sikap melalui observasi, catatan kejadian menonjol (*anecdotal record*), penilaian antar teman, dan penilaian diri serta aspek keterampilan melalui penilaian proses, produk, portofolio dan studi kasus.



DAFTAR ISI

Cover	
Preliminari	
Halaman	
Judul	i
Hak Cipta	ii
Kata Pengantar	iii
Prakata	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	xvi
Petunjuk Penggunaan Buku	xvii
PENDAHULUAN	xx
A. Rasional	xx
B. Cakupan atau Ruang Lingkup	xi
C. Tujuan Mata Pelajaran	xi
D. Pendekatan/Strategi Pembelajaran	xxii
E. Media Pembelajaran	xxiii
F. Evaluasi Pembelajaran	xxiii
Konsep Dasar Pengembangan Perangkat Lunak Dan Gim	1
Tujuan Pembelajaran	2
Peta Konsep	3
Bab I Proses Bisnis, Perkembangan Dunia Kerja dan Profesi	
Pengembang Perangkat Lunak dan Gim	
4	
A. Konsep-Konsep Dasar Pengembangan Perangkat Lunak dan Gim	5
B. Budaya Mutu dan Manajemen Proyek Pengembangan Perangkat Lunak dan Gim	74
C. Perkembangan Teknologi Industri 4.0	93
D. Profil dan Profesi Bidang Pengembang Perangkat Lunak dan Gim	105
E. Refleksi	120
F. Rangkuman	121
G. Pengayaan	124

Aset Desain	125
Tujuan Pembelajaran	126
Peta Konsep	127
Bab II Aset Desain Pada Perangkat Lunak Dan Gim	128
A. Konsep dan Tools Pembuatan Aset Desain dalam Pengembangan Perangkat Lunak dan Gim	129
B. Dasar-Dasar Desain dalam Pembuatan Aset Desain	135
C. Aset Grafis Berbasis Vector	147
D. Aset Grafis Berbasis Bitmap	224
E. Refleksi	233
F. Pengayaan	233
 Pemrograman Dasar.....	234
Tujuan Pembelajaran	235
Peta Konsep	235
Bab III Pemrograman Dasar	236
A. Algoritma dan Pemrograman	237
B. Pengenalan C#	245
C. Struktur Kode C#	254
D. Membuat Program Sederhana	268
E. Operator Aritmatika dan Logika	270
F. Struktur Kontrol	275
G. Procedure.....	283
H. Mengenal Array.....	285
I. Aktifitas Pembelajaran	295
J. Refleksi	296
K. Pengayaan	297
 Daftar Rujukan	298
Indeks	300
Glosarium	302
Biodata Penulis	306
Biodata Penelaah	308
Biodata Editor	310
Biodata Desainer	311

DAFTAR ISI

Cover	
Preliminari	
Halaman	
Judul	i
Hak Cipta	ii
Kata Pengantar	iii
Prakata	iv
Daftar Isi	v
Daftar Gambar	vi
Daftar Tabel	vii
Petunjuk Penggunaan Buku	viii
Aset Audio Dan Video	1
Tujuan Pembelajaran	2
Peta Konsep	2
Bab I Aset Audio Dan Video Pada Perangkat Lunak Dan Gim	
3	
A. Format audio video, tools pengolah audio dan efek pada audio	3
B. Tools pengolah video dan efek pada video	30
C. Interaksi Pengguna	46
D. Muatan Aktivitas Peserta Didik	52
E. Refleksi	55
F. Pengayaan	56
Pemrograman Berorientasi Obyek.....	57
Tujuan Pembelajaran	58
Peta Konsep	59
Bab III Pemrograman Berorientasi Obyek	60
A. Paradigma	60
B. Dasar dan Aturan	67
C. Class Dan Obyek	98
D. Package	110
E. Enkapsulasi	118
F. Inheritas dan Polymorphism	129
G. Muatan Aktivitas Peserta Didik	133

H. Refleksi	135
I. Pengayaan	136
Index	137
Glorasarium	138
Referensi	139
Biodata Penulis	140
Biodata Penelaah	142
Biodata Editor	143
Biodata Desainer	144

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
DIREKTORAT JENDERAL PENDIDIKAN VOKASI
DIREKTORAT SEKOLAH MENENGAH KEJURUAN
Dasar - Dasar Pengembangan Perangkat Lunak dan Gim
SMK Kelas X
Penulis : Yustiana Amita Utama dan Esa Hergatama
ISBN : xxx-xxx-xxx-xxx-x

BAB I

PROFESI BISNIS, PERKEMBANGAN DUNIA KERJA DAN PROFESI PERANGKAT LUNAK DAN GIM



TUJUAN PEMBELAJARAN

Setelah mempelajari BAB I, Anda diharapkan untuk mampu:

- a. Mendeskripsikan konsep perangkat lunak dan gim
- b. Mendeskripsikan sejarah dan aspek pada rekayasa perangkat lunak dan gim.
- c. Mendeskripsikan jenis-jenis dan karakteristik perangkat lunak dan gim
- d. Mendeskripsikan model proses pengembangan (perencanaan, analisis, implementasi, integrasi dan pemeliharaan) perangkat lunak dan gim
- e. Mendeskripsikan penerapan budaya mutu dan K3LH dalam pengembangan perangkat lunak dan gim
- f. Mendeskripsikan penerapan manajemen proyek dalam perangkat lunak dan gim
- g. Mendeskripsikan kebutuhan, keinginan dan validasi pengembangan perangkat lunak dan gim.
- h. Mendeskripsikan teknik pemasaran dan distribusi perangkat lunak dan gim
- i. Menjelaskan perkembangan teknologi pada perangkat lunak dan gim
- j. Menganalisis isu-isu penting bidang pengembangan perangkat lunak dan gim antara lain : dampak positif dan negatif gim, IOT, *Cloud Computing*, *Information Security*, *Personal Branding*, dan permasalahan terkait HAKI (Hak Atas Kekayaan Intelektual)
- k. Menjelaskan profesi dan kewirausahaan (job profile dan technopreneurship) industri pengembangan perangkat lunak dan gim
- l. Menjelaskan peluang usaha dibidang industri pengembangan perangkat lunak dan gim



PETA KONSEP



BAB I

PROSES BISNIS, PERKEMBANGAN DUNIA KERJA DAN PROFESI PENGEMBANGAN PERANGKAT LUNAK DAN GIM



APERSEPSI



Besok, Fazli ingin pergi ke toko pakaian untuk membeli seragam sekolah. Malamnya ia merencanakan perjalanan menuju toko pakaian tersebut..Kira – kira ia akan menggunakan kendaraan apa agar ia bisa pergi ke toko pakaian dengan cepat, murah, efisien dan efektif? Apakah ia akan menggunakan sepeda, motor atau mobil?

Nah! Proses perencanaan ini sangatlah penting agar kegiatan yang dilakukan mendapatkan gambaran yang jelas. Pada BAB ini anda akan mempelajari materi tentang konsep – konsep dasar perangkat lunak dan gim yang materi tersebut dapat menjawab pertanyaan anda terhadap apa sebenarnya perangkat lunak dan gim itu? Bagaimana cara membuat perangkat lunak dan gim? Serta profesi dan peluang usaha apa yang ada dalam industry perangkat lunak dan gim? Sehingga anda mempunyai gambaran yang jelas terhadap profesi yang anda akan tekuni dan cara tercepat menggapai tujuan.





KATA KUNCI

Proses Bisnis, Model Pengembangan, Perangkat Lunak dan Gim, Budaya Mutu, Manajemen Proyek, Industri 4.0, Profil Profesi, Peluang Usaha, Perangkat Lunak dan Gim

A. Konsep dasar dan model pengembangan perangkat lunak dan gim

Perangkat lunak komputer terus menjadi satu teknologi paling dan akrab kita gunakan sehari – hari mulai dari belajar, bekerja, bermain hingga berbelanja kita bisa menggunakan perangkat lunak. Beberapa puluh tahun yang lalu mungkin kita tidak ada yang membayangkan bahwa perangkat lunak akan menjadi teknologi yang sangat diperlukan di berbagai lini seperti bisnis, *sains*, dan hiburan. Buku ini membahas konsep dasar pengembangan perangkat lunak dan gim yang dalam kehidupan sehari – hari sering kita temukan atau jumpai.

1. Definisi Perangkat Lunak dan Gim
 - a. Definisi Perangkat Lunak



Gambar 1.1 Perangkat Lunak *Microsoft Word*
Sumber Gambar: <https://tekno.kompas.com/read/2021/02/19/18220097/microsoft-umumkan-office-2021>



Dari beberapa sumber pada buku pengertian dari *game* adalah sebagai berikut:

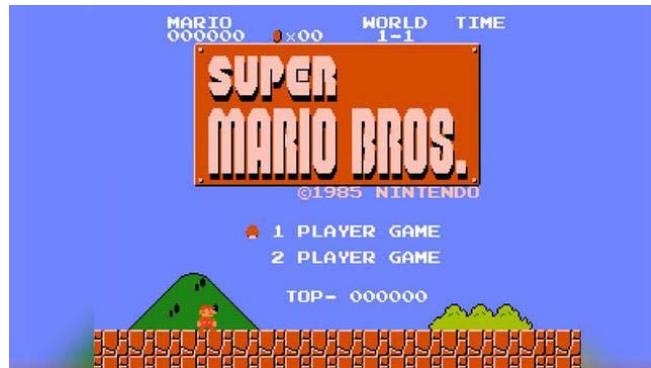
Perangkat lunak (*software*) adalah kumpulan instruksi atau program komputer (*syntaks/code* yang ditulis dalam bahasa pemrograman) yang ketika dijalankan menyediakan fungsi dan performa yang diinginkan. Perangkat lunak merupakan struktur data yang memungkinkan program memanipulasi informasi dan dokumen yang menggambarkan cara menggunakan dan mengoperasikan program tersebut (Pressman).

Menurut IEEE & ISO 1997, perangkat lunak adalah merupakan kumpulan program komputer (sintak/kode), prosedur, fungsi, modul (kumpulan program yang memiliki tujuan dan fungsi tertentu), dokumentasi, dan kumpulan data yang diperlukan untuk mengoperasikan suatu sistem perangkat lunak.

Sementara itu IEEE [IEE93] telah mengembangkan definisi yang lebih lengkap tentang rekayasa perangkat lunak (*Software Engineering*) yaitu merupakan aplikasi perangkat lunak yang bersifat sistematik, dalam pengembangannya menerapkan prinsip-prinsip rekayasa, metode pengembangan, operasi, dan *maintenance* dari *Software*; merupakan serangkaian studi yang menggunakan pendekatan rekayasa.”



b. Definisi Gim



Gambar 1. 2 *Game* Super Mario Bros

Sumber Gambar: <https://www.liputan6.com/tekno/read/4516053/fantastis-gim-super-mario-bros-lawas-dilelang-rp-45-miliar>

Dari beberapa sumber pada buku dan penelitian, pengertian dari game adalah sebagai berikut:

- 1) Menurut *International Game Development Association* (IGDA), *game* adalah suatu kegiatan dengan beberapa *rules* dan tujuan. *Game* merupakan bentuk permainan dimana sering kali (tetapi tidak selalu) melibatkan konflik, baik dengan pemain lain, dengan sistem permainan itu sendiri, atau dengan acak/nasib/ keberuntungan. Kebanyakan *game* memiliki tujuan tetapi tidak semua (Contoh: *The Sims* dan *Simcity*). Kebanyakan *game* juga telah menentukan titik mulai dan titik akhir, tetapi tidak semua (Contoh: *World of Warcraft*).
- 2) Pengertian *game* menurut J. Huizinga dalam karya klasiknya *Homo Ludens* (1938) memberikan definisi permainan adalah aktivitas yang berlangsung dalam beberapa batasan ruang dan waktu, dalam urutan yang terlihat, berdasarkan kepada urutan yang secara bebas diterima dan diluar lingkup kebutuhan.



3) Menurut Adams dalam bukunya yang berjudul "*Fundamentals of Game Design, Second Edition*" yang terbit pada tahun 2010. *Game* adalah :

- a) *Game* harus menghadirkan imajinatif, pengalaman koheren, sehingga desainer harus memiliki visi.
- b) *Game* harus menjual dengan baik, sehingga desainer harus mempertimbangkan selera konsumen.
- c) *Game* dengan lisensi berbayar harus kembali biaya lisensinya, sehingga desainer harus mengerti keuntungan lisensi dan memanfaatkannya sebagai keunggulan.
- d) *Game* harus menawarkan tantangan kecerdasan dan pengalaman yang mulus, sehingga desainer harus mengerti teknologi.

2. Sejarah Perangkat Lunak dan Gim

a. Sejarah Perangkat Lunak

Perangkat lunak awal yang diperkenalkan adalah dengan digunakannya Aljabar *Boolean* yang menggunakan bilangan biner 1 dan 0. Yang mana 1 adalah bentuk dari nilai benar dan 0 salah. Bilangan biner mulai dikelompokan menjadi 2 bit, 4 bit, *byte* yang terdiri dari 8 bit, 32 bit. Dari kumpulan bit tersebut mulai disusun ke dalam struktur kode instruksi seperti penyimpanan, transfer, operasi aritmatika, operasi logika yang kemudian disebut *assembler*. Namun karena bahasa *assembler* yang sangat rumit dan membingungkan kemudian lahir bahasa pemrograman tingkat tinggi yang mirip dengan bahasa manusia.

Perangkat lunak berevolusi selama empat era, yaitu:



1) Era Pioneer (1950-1960)



Gambar 1. 3 *Pascalene The First Mechanical adding Machine*
Sumber Gambar: <https://slideplayer.info/slide/12462754/>

Perangkat lunak pada era pioneer masih berupa sambungan-sambungan kabel yang dipindah-pindah antar bagian dalam komputer. Selain itu untuk mengakses komputer juga dapat menggunakan *punched card* yang berupa kartu yang diberi lubang.



Gambar 1. 4 *ENIAC (Electronic Numerical Integrator And Computer)*
Sumber Gambar: <https://slideplayer.info/slide/12462754/>

Penggunaan komputer dilakukan secara langsung dan komputer tak lain masih sebuah mesin untuk tujuan tertentu dan hasil yang selesai dikerjakan berupa *print out*. Perangkat lunak pada periode ini masih menjadi satu dengan perangkat keras dan proses yang di dalam komputer masih berupa garis instruksi yang diproses secara berurutan. Komputer pada era ini masih terbatas dan dibuat secara khusus.



2) Era Stabil (1960- 1980)



Gambar 1. 5 *IBM 701 Defense with Calculator*

Sumber Gambar:

https://www.ibm.com/ibm/history/exhibits/701/701_141508.html

Pada era stabil perusahaan perangkat lunak mulai bermunculan dan perangkat lunak dapat menjalankan beberapa fungsi. Baris perintah yang dijalankan tidak hanya satu baris, namun dapat menangani banyak proses yang bisa dilakukan secara bersama (*multi tasking*). Selain itu perangkat lunak sudah bisa menyelesaikan banyak pengguna (*multi user*) secara cepat atau langsung (*real time*). Karena perangkat lunak sudah dapat *multi tasking*, *multi user*, dan *realtime* penggunaan perangkat lunak mulai menyebar tidak hanya untuk kalangan peneliti dan akademi namun juga digunakan oleh kalangan industri. Pada era ini, sistem basis data mulai muncul dan perangkat lunak menjadi sebuah produk dari perusahaan.



3) Era Mikro (1980 sampai 1990)



Gambar 1. 6 *Apple Lisa*
Sumber Gambar: https://en.wikipedia.org/wiki/Apple_Lisa

Pada era ini komputer dan jaringan komputer mulai berkembang. Sehingga pada era ini perangkat lunak juga berkembang menjadi dua bagian yaitu sistem perangkat lunak internal dan sistem perangkat lunak berupa aplikasi. Sistem perangkat lunak berupa aplikasi ini yang digunakan secara langsung oleh pengguna untuk tujuan tertentu. Pada perangkat lunak aplikasi sudah disematkan sistem automatisasi yang merupakan suatu jenis kecerdasan buatan. Pada era ini penggunaan komputer berkembang tidak hanya digunakan perusahaan namun digunakan secara personal.

4) Era Modern (1990-sekarang)

Pada era modern perkembangan perangkat lunak sangat cepat. Bahkan sebuah telepon genggam sudah dilengkapi perangkat lunak yang tidak kalah dengan Personal Computer. Bahkan perangkat lunak yang terdapat di telepon genggam melebihi Personal Computer.





Gambar 1. 7 Perkembangan Telepon Seluler
Sumber Gambar: <https://www.soloposfm.com/perkembangan-telepon-seluler-sejak-1990-an-sampai-sekarang/9616/>

Bahkan perangkat lunak yang terdapat di telepon genggam melebihi komputer itu sendiri seperti bisa menjadi *remote AC*, *TV*, *printer* dan sebagainya. Tidak hanya komputer dan telepon genggam namun *TV*, *AC*, mesin cuci, dan *microwave* sudah ditanamkan sistem perangkat lunak untuk dalam model pengoperasian. Bahkan sekarang semua perangkat dapat saling terhubung dengan kecanggihan teknologi yang berkembang. Pembuatan perangkat lunak pada era ini tidak hanya profesi beberapa orang, namun sudah melibatkan banyak orang dengan disiplin ilmu yang berbeda dalam perancangannya. Tingkat kecerdasan perangkat lunak semakin ditingkatkan, tidak hanya mengetahui permasalahan teknis namun dapat mengenali suara dan gambar yang ditangkap.

b. Sejarah Gim

Dalam *paper*-nya Teressa Dellon yang berjudul "*Adventure Games for Learning and Storytelling*" membahas tentang sejarah *game* dari tahun ke tahun. Berikut adalah sejarah dari *game* :

- 1) *Pre-History* (Sebelum Tahun 1970)



a) Tahun 1889 (*The Pre-History*)

Fusajiro Yamauchi mendirikan *Koppai Nitendo* sebagai sebuah perusahaan kartu permainan di Jepang. Kemudian mereka menghilangkan kata “*Koppai*” dari namanya dan kini hanya di kenal sebagai “*Nitendo*”.

b) Tahun 1917 (Perusahaan Magnavox)

Edwin Pridhan dan Petter L. Jensen bersaudara membentuk perusahaan Magnavox, mereka menciptakan penemuan *moving-coil loudspeaker*.

c) Tahun 1936 (*Seeburg Ray O Lite*)

Jukebox Manufacturer Seeburg Corporation meluncurkan *Seeburg Ray-O-Lite*, *game* mekanik dimana pemain menembak bebek yang bergerak, mereka mengklaim *game arcade* pertama dengan *light gun*.



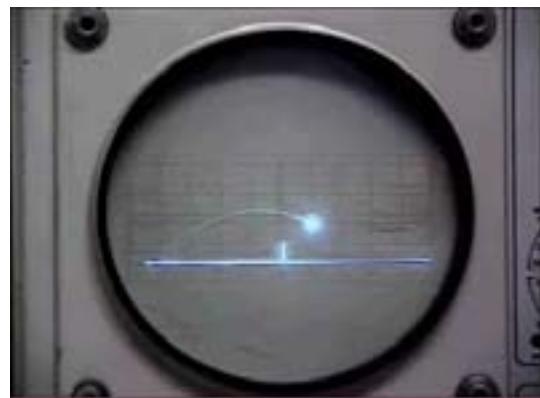
Gambar 1.8 Permainan *duck shoot game* oleh Ray-O-Lite Rifle, Seeburg
Sumber Gambar: <http://www.pinrepair.com/arcade/rayolit.htm>

d) Tahun 1947 (*The Age Discovery*)

Thomas T. Goldsmith Jr. dan Estle Ray Mann mengembangkan permainan elektronik pertama dengan menggunakan tabung sinar katoda menampilkan simulasi



mereka menembakkan rudal ke sasaran yang ditempatkan di layar *overlay*.

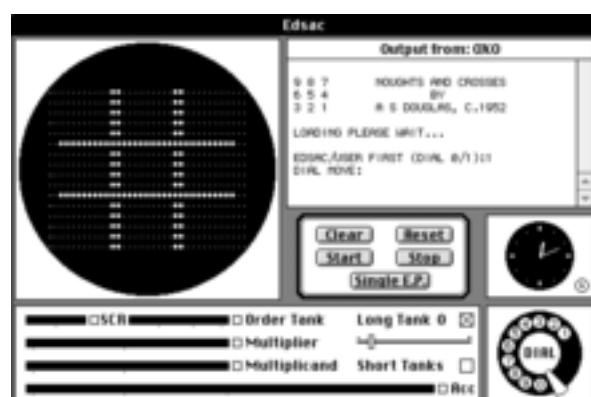


Gambar 1.9 Permainan Simulator *Rudal Cathode Ray Tube Device Amusement*.

Sumber Gambar: <http://prayoga-5150.blogspot.com/2018/03/history-of-video-game.html>

e) Tahun 1952 (*Oxo Game*)

Untuk tesisnya tentang interaksi manusia dan komputer, Alexander Sandy Douglas mengembangkan permainan komputer berbasis grafis pertama, ditambah dalam bidang kecerdasan buatan dengan permainan OXO (alias *Noughts* dan *Crosses*).



Gambar 1.10 Permainan OXO *Game*
Sumber Gambar: <https://en.wikipedia.org/wiki/OXO>



f) Tahun 1954 (*Service Game Company* / SEGA)

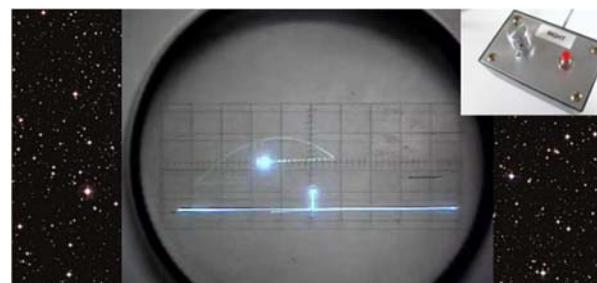
Pada tahun ini mulai di bangun *Service Games Company* (SEGA) dan mulai meng-import mesin *pinball* ke Jepang.



Gambar 1.11 Mesin *Pinball*/SEGA

Sumber Gambar: https://www.arcademuseum.com/game_detail.php?game_id=13254

g) Tahun 1958 (Permainan *Tennis For Two*)



Gambar 1. 8 Permainan *Tennis For Two*

Sumber Gambar:

<https://videogamehistorydevelopment.weebly.com/tennis-for-two.html>

Pada tahun 1958 di adakan demonstrasi pada hari kunjungan di *Brookhaven National Laboratory*, Kepala bagian *Instrumentation* *Willy Higinbotham* menciptakan *Tennis for Two*, permainan dua pemain yang terdiri dari



sebuah titik (bola) yang mengetuk bolak balik melintasi garis. Setiap pemain menggunakan tombol dan tombol untuk menembak bola dan menyesuaikan sudut untuk memukulnya.

h) Tahun 1961 – 1962 (Permainan *Space War*)



Gambar 1. 9 Permainan *Space War*
Sumber Gambar: <https://en.wikipedia.org/wiki/Spacewar!>

Teknisi komputer MIT, Stephen "Slug" Russell, Martin "Shag" Graetz, Wayne Witanen dan beberapa kontributor lain mengembangkan *Spacewar* sebagai demo untuk komputer PDP yang baru-1 (*Programmed Data Processor-1*). Permainan 1 *player* terdiri dari dua benda angkasa yang dikenal sebagai "*wedge*" dan "*pencil*" yang terbang dan saling menembakkan torpedo pada satu sama lain.

i) Tahun 1966 – 1967 (*The Brown Box*)



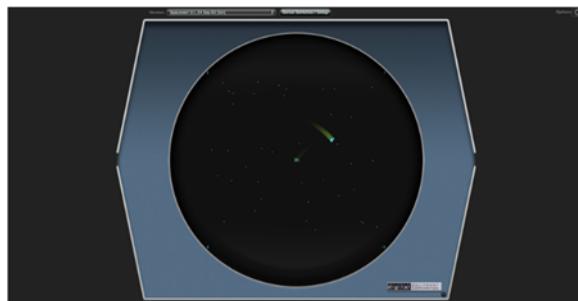
Gambar 1. 10 Permainan *The Brown Box*
Sumber Gambar:
https://americanhistory.si.edu/collections/search/object/nmah_1301997



Kontraktor pertahanan *Baer Associates* mempekerjakan Ralph Sanders untuk menciptakan permainan video untuk ditampilkan pada layar televisi. Permainan itu sendiri terdiri dari dua titik berkejaran. Dengan niat untuk mengembangkan teknologi sebagai sarana pelatihan bagi militer, pemerintah terus untuk mendanai proyek dengan label *top secret* Proyek *The Brown Box*. Baer dan timnya juga menciptakan permainan tenis untuk *Brown Box*.

j) Tahun 1969 (*Game Online*)

Rick Blomme menciptakan program *game online Spacewar (2 player)* untuk PLATO, sistem *sharing* komputer umum pertama kali yang menjadi kakek dari jaringan komputer dan internet melalui sistem PLATO dua pemain dapat bersaing dalam *Spacewar* dari komputer yang terpisah.



Gambar 1. 11 *Spacewars! Online*
Sumber Gambar: <https://www.masswerk.at/spacewar/>



2) Tahun 1971-1980

a) Tahun 1971 (*The Golden Age*)



Gambar 1. 12 PDP Space Wars!

Sumber Gambar:

https://www.gamasutra.com/view/feature/129861/down_the_hyperspatial_tunnel.php?page=4

Computer Space dan *Galaxy Game* merupakan video game arcade coin-op pertama yang pernah dibuat. Hanya selang dua bulan dirilis, keduanya berdasarkan *computer game PDP Spacewars*.

b) Tahun 1972 (*The First Generation Console – The Magnavox Odyssey dan Atari and Pong*)

(1) Magnavox Ralph Baer melisensikan teknologi permainan televisi dari *Sanders Associates* dan dirilis ke publik sebagai *Magnavox Odyssey*. Sistem ini menggunakan kartu permainan yang dapat ditukar (disebut kartu), sebuah *controller* dengan dua tombol (satu untuk gerakan horizontal, yang lain untuk vertikal).





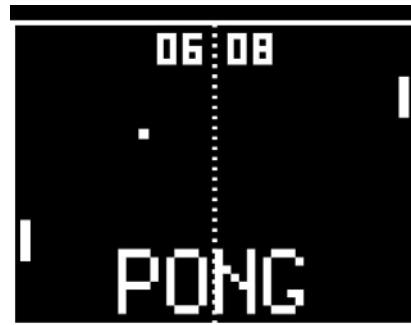
Gambar 1. 13 *Magnavox Odyssey*

Sumber Gambar:

<https://en.wikipedia.org/wiki/File:Magnavox-Odyssey-Console-Set.jpg>

(2) Nolan Bushnell dan Ted Dabney membentuk Atari dengan tujuan mengembangkan permainan video yang tersedia secara komersial.

(3) Al Alcorn yang disewa oleh Atari yang baru membuat program rilis resmi pertama mereka, *game arcade coin-op tenis* bernama Pong.



Gambar 1. 14 Permainan *Atari Pong*

Sumber Gambar:

<http://catatanrezapahlevi.blogspot.com/2013/07/perkembangan-teknologi-game-dari-masa.html>

c) Tahun 1973 (*The Arcade Game*)

(1) *Space Race – Game 2 player* dimana dua pesawat ruang angkasa berlomba sambil menghindari asteroid.



(2) Gotcha – permainan di mana satu pemain mengejar yang lain melalui sebuah labirin.

(3) Rebound – *game volley 2 player.*

d) Tahun 1974 (*Grand Trak 10*)



Gambar 1. 15 *Grand Track 10*
Sumber Gambar: https://www.arcade-museum.com/game_detail.php?game_id=7992

Gran Trak 10 (car racing game), Game arcade pertama berbasis ROM dirilis oleh Atari. Perusahaan baru bernama *Kee Games* mulai memproduksi *game arcade* salah satunya yang sukses adalah *Tank*.

e) Tahun 1975

Atari merilis versi pertama *game* versi rumahan “Pong-eksklusif” melalui *Sears Roebuck Adventure* (a.k.a *Colossal Cave Adventure*) - komputer berbasis *game* petualangan teks pertama yang diciptakan oleh Willie Crowther dan Don Woods. *General Instruments* mengembangkan chip AY-3-8500, sebuah kloning dari teknologi Atari's Pong. *Micro Instrumentation and Telemetry Systems* (MITS) merilis komputer *desktop* pertama, Altair 8800.





Gambar 1. 16 Permainan *Pong Eksklusif*
Sumber Gambar: <https://www.wikiwand.com/en/Pong>

f) Tahun 1976 (*The Second Generation*)



Gambar 1. 17 *Console Game Fairchild dan Apple I*
Sumber Gambar:
https://en.wikipedia.org/wiki/Fairchild_Channel_F_Videocarts
dan https://en.wikipedia.org/wiki/Apple_I

Game Historic Arcade dirilis: *Breakout*. Coleco merupakan perusahaan yang mengambil keuntungan dari instrumen chip AY-3-8500, bersama Telstar, membangun sistem yang berisi 3 game yang dapat dimainkan. *The Fairchild Camera and Instrument Corp.* merilis *home video game console Fairchild Channel F*. Steve Wozniak menciptakan *computer desktop Apple-1*.



g) Tahun 1977 (*Video Game Console*)



Gambar 1. 18 *Atari 2600* dan *Nintendo Color TV Game*

Sumber Gambar:

https://en.wikipedia.org/wiki/Fairchild_Channel_F_Videocarts
dan https://id.wikipedia.org/wiki/Color_TV_Game

Atari VCS (kemudian dikenal sebagai Atari 2600) membidik pasar rumahan. Bally-Midway masuk ke pasar *console* dengan Astrocade. Nintendo merilis sistem *console* rumah pertama mereka, Color TV Game 6.



Gambar 1. 19 *PDP-10 Minicomputer*

Sumber Gambar: <https://en.wikipedia.org/wiki/PDP-10>

Di MIT's *Laboratory for Computer Sciences*, Dave Lebling, Marc Blank, Tim Anderson, dan Bruce Daniels membuat *game* petualangan berbasis teks, Zork, untuk PDP-10 *minicomputer*. Tiga inovasi komputer *desktop* rumah dirilis secara simultan yaitu Tandy TRS80, Commodore PET, dan yang paling populer dari kelompok ini adalah Apple II.





Gambar 1. 20 *Tandy TRS80* dan *Commodore PET*
Sumber Gambar: <https://en.wikipedia.org/wiki/TRS-80> dan
https://id.wikipedia.org/wiki/Commodore_PET



Gambar 1. 21 *Apple II*
Sumber Gambar: https://en.wikipedia.org/wiki/Apple_II

h) Tahun 1978 (*Arcades and Consoles*)



Gambar 1. 22 Permainan *Game Space Invaders* dan *Odyssey 2*
Sumber Gambar: https://id.wikipedia.org/wiki/Space_Invaders
dan https://en.wikipedia.org/wiki/Magnavox_Odyssey_2

Game arcade dirilis : *Space Invaders*, game pertama yang menampilkan nilai tertinggi. Nintendo membuat game komputer Othello pertama. Carol Shaw menciptakan 3D Tic-Tac-Toe untuk Atari 2600.



Magnavox merilis konsol generasi kedua mereka, Odyssey 2. The PDP-10 *computer*, MUD (*multi-user dungeon*), *game text multiplayer* pertama yang dijalankan melalui ARPANET, yang berkembang dengan cepat dari kelompok sederhana dimana pemain dapat berpindah tempat dan melakukan chat, termasuk objek-objek, peristiwa, makhluk virtual, dan NPC (*non player character*). Ini adalah pendahulu *Massively Multiplayer Online Game* (MMOG).

- i) Tahun 1979 (*Arcades, Console and Computers*)



Gambar 1. 23 *Handheld Console Cosmos* oleh Atari
Sumber Gambar: <http://videogamekraken.com/cosmos-by-atari>

Historic Arcade Games dirilis : Asteroids, Basketball, Football - *The first game to use a trackball controller*, Galaxian, dan Lunar Lander. Sears merilis Tele-Games, sebuah *clone* dari Atari 2600. Mattel merilis *console system*, Intellivision. Perusahaan *software Computer game* Infocom mulai muncul. Atari mengembangkan *console handheld* menggunakan teknologi hologram yang disebut Cosmos. Milton Bradley merilis Microvision.



j) Tahun 1980 (*The Arcade Games*)



Gambar 1. 24 *Nintendo Game and watch*

Sumber Gambar:

https://thereaderwiki.com/en/Game_%26_Watch

Historic Arcade Games dirilis : Pac-Man – *Game* paling populer sepanjang waktu. Battlezone – *Game Arcade* pertama yang menggunakan gambar 3D. Defender – *Game* pertama yang menggunakan *virtual world*. Mattel merilis Intellivision ke pasar. Gunpei Yokoi creates the Nintendo Game & Watch. Kelton Flinn and John Taylor membuat Dungeons of Kesmai, *game RPG online* komersial pertama.

3) Tahun 1981-1990

a) Tahun 1981 (*The First Gaming Mag*)



Gambar 1.29 PC Pertama IBM

Sumber Gambar:

https://en.wikipedia.org/wiki/IBM_Personal_Computer

Historic Arcade Games dirilis : Centipede, Frogger, Galaga, Ms. Pac-Man, dan Donkey Kong - *the first*



game to feature Mario. Arnie Katz and Bill Kunkel mem-publish Electronic Games, majalah *video game* pertama. IBM me-launching Komputer, tanda awal dari era pc modern.

b) Tahun 1982 (*Arcade Game*)



Gambar 1. 30 Atari 5200 dan Commodore 64

Sumber Gambar: https://en.wikipedia.org/wiki/Atari_5200 dan https://id.wikipedia.org/wiki/Commodore_64

Historic Arcade Games dirilis : Burgertime, Buck Rogers - Planet of Zoom, Dig Dug, Donkey Kong Junior, Joust, Pole Position, Popeye, Q-bert, Robotron 2084, Star Trek, Tron, Xevious, Zaxxon Coleco merilis *The Colecovision Console Video Game* dengan prosesor PC. Atari merilis Atari 5200 *Super System*. Commodore 64 dirilis dan menjadi komputer rumah paling populer.

c) Tahun 1983 (*Arcade Game*)

Historic Arcade Games dirilis : Cliff Hanger, Congo Bongo, Dragon"s Lair, Mario Bros, Spy Hunter, Star Wars, Dragon's Lair, dan Cliff Hanger merupakan *game* pertama yang menggunakan teknologi *laserdisc*.





Gambar 1.31 *Dragon's Lair* menggunakan Teknologi *Laserdisc*
Sumber Gambar: https://en.wikipedia.org/wiki/Dragon%27s_Lair

d) Tahun 1983 (*The Crash of the Video Game Industry*)



Gambar 1.32 Game Console *Nintendo Famicom*
Sumber Gambar:

<https://www.liputan6.com/tekno/read/641610/30-tahun-nintendo-famicom-nes-sang-nenek-moyang-konsol-game>

Kompetisi antara Commodore, PC dan komputer Apple menyebabkan pasar *video game* di US merosot. Atari mengakhiri Atari 2600. Namun sangat beda keadaan pasar di Jepang dibanding dengan US. Nintendo merilis Famicom (*Nintendo Entertainment System*), *console game* rumahan paling canggih waktu itu di Jepang.

e) Tahun 1984 (*Arcade Games & Computer*)

Historic Arcade Games dirilis : Karate Champ, Marble Madness, Punch-out!, Tapper (a.k.a Root Beer Tapper), Space Ace. Apple meng-upgrade *desktop computer* mereka dan merilis Macintosh Komputer.





Gambar 1.33 Permainan *Karate Champ* dan *Apple Personal Computer Macintosh*

Sumber Gambar: https://en.wikipedia.org/wiki/Karate_Champ dan <https://id.wikipedia.org/wiki/Macintosh>

f) Tahun 1985 (*Arcade and Computer Gaming*)



Gambar 1.34 *Windows 1.0* dan *Quantumlink* layanan *public online* pertama

Sumber Gambar: https://id.wikipedia.org/wiki/Windows_1.0 dan <https://quantumcomputingtech.blogspot.com/2018/01/quantum-online-service.html>

Historic Arcade Games dirilis : Gauntlet, Indiana Jones and the Temple of Doom. Microsoft mengembangkan sistem operasi Windows sebagai add-on pada DOS. QuantumLink (a.k.a Q-Link), layanan publik *online* pertama yang meluncurkan komputer Commodore.

g) Tahun 1985 (*The Rebirth and Third Generation*)

Nintendo merilis Famicom di US dengan nama *Nintendo Entertainment System* (NES). Nintendo juga merilis game Super Mario Bros. Di *Soviet Academy of*



Sciences, programmer Alex Pajitnov menciptakan game puzzle, game Tetris.



Gambar 1.35 Nintendo NES
Sumber Gambar:
https://en.wikipedia.org/wiki/Nintendo_Entertainment_System

h) Tahun 1986 (*Atari's Return and SEGA's Launch*)



Gambar 1.36 Game Console Atari 7800 dan Sega Master System
Sumber Gambar: https://id.wikipedia.org/wiki/Atari_7800 dan https://id.wikipedia.org/wiki/Sega_Master_System

SEGA merilis SEGA *Master System* untuk bersaing dengan NES. Atari merilis Atari 7800, memperbaiki pendahulunya yaitu Atari 5200.

i) Tahun 1989 (*The Fourth Generation*)



Gambar 1.37 Game Console SEGA Genesis
Sumber Gambar: https://en.wikipedia.org/wiki/Sega_Genesis



SEGA meng-*upgrade console*-nya menjadi The Sega Genesis (a.k.a MegaDrive). The NEC Corporation merilis The Turbografx-16.

j) Tahun 1989 (*The Handheld Revolution*)



Gambar 1.38 *Game Console Nintendo Game Boy*
Sumber Gambar: https://id.wikipedia.org/wiki/Game_Boy

Nintendo merilis sistem *game handheld* paling populer Game Boy. Untuk menyaingi Nintendo, Atari merilis Atari Lynx.

k) Tahun 1990 (*Console and Computer Gaming*)



Gambar 1.39 *Game Console Turbo Express*
Sumber Gambar: <https://en.wikipedia.org/wiki/TurboExpress>

Nintendo merilis Super Mario Bros. 3 untuk NES. SNK merilis Neo-Geo. The Amiga CDTV merilis computer rumahan pertama yang menggunakan CD-Rom Drive. Turbo Express, versi portable dari Turbografx-16 dirilis. Sega merilis *handheld system Game Gear*.



4) Tahun 1991-2000

a) Tahun 1991 (*Arcade and Console Gaming*)



Gambar 1.40 *Game Console Nintendo SNES*

Sumber Gambar:

https://simple.wikipedia.org/wiki/Super_Nintendo_Entertainment_System

Historic Arcade Game dirilis : Street Fighter II. Nintendo merilis Super Nintendo (SNES), dengan *game* Super Mario World. Sega merilis *game* Sonic the Hedgehog. *Game Discworld MUD game* dirilis. America Online meluncurkan *game graphic* pertama, Neverwinter Nights. The ImagiNation Network (a.k.a Sierra Network) merilis *game multiplayer*.

b) Tahun 1992 (*Prelude to the CD-ROM Age*)



Gambar 1.41 *CD ROM PC*

Sumber Gambar: <https://www.jasareplikasicd.com/2015/11/10-tahapan-cara-membersihkan-cd-rom.html>

Historic Arcade Game dirilis : Mortal Kombat. SEGA merilis *console* dengan menggunakan CD-ROM lewat Sega CD.



c) Tahun 1993 (*The Fifth Generation*)



Gambar 1.42 Atari Jaguar

Sumber Gambar: https://id.wikipedia.org/wiki/Atari_Jaguar

Panasonic meluncurkan *console* CD-ROM pertama 3DO. Atari meng-*upgrade* *console*-nya yang bernama Jaguar. Selain menggunakan CD-ROM *system*, The Jaguar masih menyediakan slot untuk bermain menggunakan *cartridge games*. Doom dirilis dan dengan cepat menggantikan Wolfenstein 3D sebagai *game FPS* paling populer.

d) Tahun 1994 (*Sony Enters the Game*)



Gambar 1.43 Permainan *Cyan Myst*

Sumber Gambar: <https://en.wikipedia.org/wiki/Myst>

Historic Arcade Game dirilis : Tekken. SEGA Saturn dan *Sony PlayStation* dirilis di Jepang selang 1 bulan. Cyan merilis Myst dan menjadi *best selling computer game*.



e) Tahun 1995 (*Console and Computer Gaming*)



Gambar 1.44 *Sony Playstation 1*

Sumber Gambar:

[https://en.wikipedia.org/wiki/PlayStation_\(console\)](https://en.wikipedia.org/wiki/PlayStation_(console))

The SEGA Saturn dan Sony PlayStation dirilis ke United States. Microsoft merilis Window 95, ini menjadikan Windows sebagai sistem operasi paling populer.

f) Tahun 1995 (*The Virtual Boy*)

Untuk mencoba dan memanfaatkan Virtual Reality, Nintendo meluncurkan Virtual Boy.



Gambar 1.25 *Virtual Boy*

Sumber Gambar: <https://www.racketboy.com/retro/nintendo-virtual-boy-101-a-beginners-guide>

g) Tahun 1996 (*Console and Computer Gaming*)

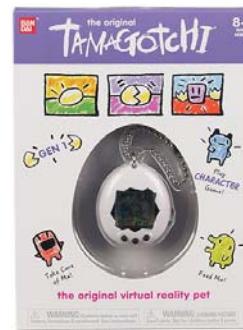
Nintendo merilis 64-Bit *console*, yaitu Nintendo 64 (N64). Tomb Raider diluncurkan untuk PlayStation, Saturn dan PC. Meridian 59, *game online* MMOG pertama dengan *graphic* 3D.





Gambar 1.46 *Nintendo 64*
Sumber Gambar: <https://www.racketboy.com/retro/nintendo-virtual-boy-101-a-beginners-guide>

h) Tahun 1996 (*Handheld and Novelty Gaming*)



Gambar 1.47 *Tamagotchi Original*
Sumber Gambar:
[https://tamagotchi.fandom.com/wiki/Tamagotchi_\(1996_Pet\)](https://tamagotchi.fandom.com/wiki/Tamagotchi_(1996_Pet))

Tiger Electronics meluncurkan Game.com untuk menyaingi *game boy. Rumble features* memperkenalkan *joysticks* dan *controllers*. Tamagotchi, *game virtual pet* pertama.

i) Tahun 1998 (*The Sixth Generation of Consoles Harnessing the Power of Computers*)
The Sixth Generation of Consoles Harnessing the Power of Computers, Sega meluncurkan Dreamcast di Jepang. Nintendo memberikan warna ke *game handhelds* lewat



Game Boy Color (GBC). SNK merilis *console* versi *handheld* dengan nama Neo-Geo Pocket.



Gambar 1.48 *Nintendo Game Boy Color*
Sumber Gambar: https://id.wikipedia.org/wiki/Game_Boy_Color

j) Tahun 1999 (*Dreamcast Fails and EverQuest Launches*)



Gambar 1.49 *Sony Playstation 2*
Sumber Gambar: https://en.wikipedia.org/wiki/PlayStation_2

Sega merilis Dreamcast di United States. Sony merilis PlayStation 2 pada 2001. Sony meluncurkan *game* MMOG populer yaitu EverQuest.

5) Tahun 2001-sekarang

a) Tahun 2001

The Third Generation of Handhelds Nintendo merilis *Game Boy Advance* (GBA).





Gambar 1.50 Nintendo GBA

Sumber Gambar:

https://en.wikipedia.org/wiki/List_of_Game_Boy_Advance_games

b) Tahun 2004



Gambar 1.51 Playstation Portable dan Nintendo DS

Sumber Gambar:

https://en.wikipedia.org/wiki/PlayStation_Portable dan

https://en.wikipedia.org/wiki/Nintendo_DS

Sony merilis *Playstation portable* (PSP) tahun 2004 di Jepang sebagai konsol yang bersaing dengan Nintendo DS yang juga diluncurkan pada tahun yang sama dengan PSP.

c) Tahun 2005

The Next-Gen Consoles Begin, Xbox meluncurkan Xbox Live Arcade, layanan *download* berbayar untuk Xbox dan Xbox 360 *systems*.





Gambar 1.52 *Xbox 360 dan Xbox Live Membership*
Sumber Gambar: <https://www.alzashop.com/xbox-360-500-gigabytes-reface-edition-forza-horizon-2-voucher-1-month-xbox-live-gold-d3028574.htm>

d) Tahun 2006



Gambar 1.53 *Nintendo Wii dan Playstation 3*
Sumber Gambar: <https://en.wikipedia.org/wiki/Wii> dan https://id.wikipedia.org/wiki/PlayStation_3

Next Gen Consoles Continue The Wii Virtual Console's
Wii Shop Channel menawarkan layanan *download* berbayar untuk *Wii console*. PlayStation Network meluncurkan layanan *download* berbayar untuk PlayStation 3.

e) Tahun 2009

Popularitas *game mobile* mulai melejit ketika Angry Birds muncul. Gim ini pertama kali dirilis pada



Desember 2009. Setelah sukses merajai pasar *Apps Store*, Rovio juga membawa Angry Birds ke *platform* Android.



Gambar 1.54 Permainan Angry Bird
Sumber Gambar: <https://www.angrybirds.com/10-years/>

f) Tahun 2010 (*Kinect*)



Gambar 1.55 Xbox 360 Kinect
Sumber Gambar: <https://en.wikipedia.org/wiki/Kinect>

Kinect adalah perangkat input penginderaan gerak yang diproduksi oleh Microsoft dan pertama kali dirilis pada tahun 2010. Teknologi ini mencakup perangkat keras yang awalnya dikembangkan oleh PrimeSense, dengan menggabungkan kamera RGB, proyektor inframerah, dan detektor yang memetakan kedalaman baik melalui cahaya terstruktur atau perhitungan waktu penerbangan, dan susunan mikrofon. Bersama dengan perangkat lunak dan kecerdasan buatan Microsoft memungkinkan perangkat melakukan pengenalan gerakan waktu nyata, pengenalan ucapan, dan deteksi



kerangka tubuh hingga empat orang. Hal ini memungkinkan Kinect digunakan sebagai perangkat *hands-free* untuk berinteraksi dengan sistem komputer.

g) Tahun 2011



Gambar 1.56 *Nintendo 3ds*

Sumber Gambar: https://en.wikipedia.org/wiki/Nintendo_3DS

Tahun 2011 melihat banyak video game baru kritis diakui permainan, seperti Ace Combat, Assassin Creed, Soccer game, Batman: Arkham, Banyak penghargaan pergi ke permainan seperti Batman. Kham City, Portal 2, The Elder Scrolls V: Skyrim, The Legend of Zelda: Skyward Sword dan Uncharted 3: Drake Deception. 2011 juga menandai rilis diseluruh dunia dari Nintendo 3DS.

h) Tahun 2013



Gambar 1.57 *Playstation 4 dan Xbox One*

Sumber Gambar: https://en.wikipedia.org/wiki/PlayStation_4 dan https://id.wikipedia.org/wiki/Xbox_One



Banyak game yang dirilis pada tahun 2013 seperti Ace Attorney, Army of Two, Asphalt, Assassin Creed, Batman: Arkham, Battlefield, BioShock, Call of Duty, Crysis, Dead Island, Dead Rising, Dead Space, Devil May Cry, final Fantasy, The Legend of Zelda: A Link Antara Worlds. Game konsol PlayStation 4 dari Sony Computer Entertainment dan Xbox One dari Microsoft juga dirilis pada tahun 2013.

i) Tahun 2016



Gambar 1.58 Generasi Pertama *PlayStation VR*
Sumber Gambar: https://en.wikipedia.org/wiki/PlayStation_VR

PlayStation VR adalah **headset** realitas virtual yang dikembangkan oleh *Sony Interactive Entertainment*, yang dirilis pada Oktober 2016.

j) Tahun 2017



Gambar 1.59 *Nintendo Switch*
Sumber Gambar:
https://id.wikipedia.org/wiki/Nintendo_Switch



Pada tahun 2017 nintendo meluncurkan *Nintendo Switch* sebagai *game handheld* pertama yang bisa dijadikan *game console* dengan menggunakan *system dock*.

k) Tahun 2018



Gambar 1.60 *Nintendo Switch dan Labo*

Sumber Gambar: https://en.wikipedia.org/wiki/Nintendo_Labo

Pada tahun 2018 nintendo meluncurkan Nintendo labo sebagai pengembangan produk Nintendo switch. Nintendo Labo terdiri dari 2 bagian, di mana satu bagian adalah permainan dan satu bagian adalah beberapa lembar karton (kardus). Permainan dengan potongan karton (kardus) dan bahan lain yang dirakit kemudian dikombinasikan dengan Nintendo Switch sehingga dapat berinteraksi dengan perangkat lunak permainan yang disertakan. Nintendo merancang Labo sebagai cara untuk mengajarkan prinsip-prinsip teknik, fisika, dan pemrograman dasar.

l) Tahun 2020

Xbox Series X dan Xbox Series S adalah konsol video game yang dikembangkan oleh Microsoft. Keduanya dirilis pada 10 November 2020 sebagai generasi



keempat dari keluarga konsol Xbox, menggantikan keluarga Xbox One. Seiring dengan Sony PlayStation 5, juga dirilis pada November 2020, Xbox Series X dan Series S adalah bagian dari generasi kesembilan dari konsol video game.



Gambar 1.61 *Xbox Series X*

Sumber Gambar:

https://watchdogs.fandom.com/wiki/Xbox_Series_X

Fitur yang dimiliki PlayStation 5 antara pengontrol DualSense dengan umpan balik haptic dan kompatibilitas mundur dengan sebagian besar game PlayStation 4 dan PlayStation VR.



Gambar 1.62 *Playstation 5*

Sumber Gambar:

https://playstation.fandom.com/wiki/PlayStation_5

3. Aspek dan Elemen pada Perangkat Lunak dan Gim

Secara garis besar fungsi, tujuan dan sejarah dapat mencirikan sebuah perangkat lunak dengan gim. Perangkat lunak digunakan sebagai perintah untuk melakukan sebuah aktivitas sedangkan gim



berfungsi sebagai sarana hiburan. Selain itu untuk membedakan perangkat lunak dan gim kita dapat mengetahui elemen-elemen yang terdapat di dalamnya. Berikut adalah elemen-elemen yang terdapat pada perangkat lunak dan gim.

a. Aspek pada Perangkat Lunak

Aspek-aspek pada perangkat lunak ini perlu diperhatikan setiap kali mengembangkan perangkat lunak. Berikut merupakan atribut-atribut pada perangkat lunak :

1) Efisien

Perangkat lunak disimpan dan berjalan pada perangkat keras. Performa perangkat keras ini pasti berbeda-beda, kita harus memastikan bahwa perangkat lunak yang dibuat harus seefisien mungkin. Perangkat lunak tidak boleh menggunakan sumber daya yang boros pada memori ataupun prosesor.

2) Mudah dirawat (*maintainable*)

Perangkat lunak tidak mengalami keausan dan perbaikannya tidaklah semudah mengganti suku cadang. Dengan demikian aspek kemudahan perawatan perangkat lunak perlu diperhatikan. Perangkat lunak harus dikembangkan sedemikian rupa sehingga dapat berevolusi untuk memenuhi perubahan kebutuhan pelanggannya. Jangan sampai setiap *patch* atau *upgrade* yang dibuat memerlukan tenaga, biaya yang lama seperti membuat perangkat lunak dari awal.

3) Dapat diterima

Terdapat dua jenis perangkat lunak: pertama yang diproduksi oleh pengembang perangkat lunak kemudian



dipasarkan dan yang kedua adalah perangkat lunak *on-demand* yang dikembangkan berdasarkan permintaan dari pengguna atau *stakeholders*. Keduanya harus dapat diterima oleh pengguna. Ini menandakan bahwa perangkat lunak harus mudah dimengerti, mudah digunakan, *kompatible* dengan sistem dan yang terpenting adalah sesuai dengan kebutuhan.

4) Dapat diandalkan

Perangkat lunak dikembangkan untuk mengerjakan berbagai fungsi spesifik. Untuk mencapai fungsi tersebut, aspek keandalan perlu diperhatikan. Perangkat lunak tidak boleh menyebabkan kerusakan fisik atau ekonomi jika terjadi kegagalan sistem atau error serta menjamin aspek kemanan dan keselamatan.

b. Aspek dan Elemen pada Gim

Menurut Fullerton dalam bukunya yang berjudul "*Game Design Workshop*" aspek-aspek yang membuat sebuah *game* menjadi *game* adalah terdapat pada *game* dapat digambarkan seperti gambar dibawah ini:



Gambar 1.63 Aspek *Game*



Menurut Teresa Dillon elemen-elemen dasar sebuah *game* adalah:

- 1) **Game rule** merupakan aturan perintah, cara menjalankan, fungsi objek dan karakter di dunia permainan. Dunia *game* bisa berupa pulau, dunia khayal, dan tempat-tempat lain yang sejenis yang dipakai sebagai *setting* tempat dalam permainan *game*.
- 2) **Plot**, biasanya berisi informasi tentang hal-hal yang akan dilakukan oleh *player* dalam *game* dan secara detail, perintah tentang hal yang harus dicapai dalam *game*.
- 3) **Tema**, biasanya ada pesan moral yang akan disampaikan *character* atau pemain sebagai karakter utama maupun karakter yang lain yang memiliki ciri dan sifat tertentu.
- 4) **Object** merupakan sebuah hal yang penting dan biasanya digunakan pemain untuk memecahkan masalah, ada kalanya pemain harus punya keahlian dan pengetahuan untuk bisa memainkannya.
- 5) **Text, grafik dan sound**, biasanya merupakan kombinasi dari media teks, grafik maupun suara, walaupun tidak harus semuanya ada dalam permainan *game*.
- 6) **Animasi** merupakan hal yang selalu melekat pada dunia *game*, khususnya untuk gerakan karakter-karakter yang ada dalam *game*, properti dari objek.
- 7) **User Interface** merupakan fitur-fitur yang mengkomunikasikan *user* dengan *game*.



4. Jenis – Jenis Perangkat Lunak dan Gim

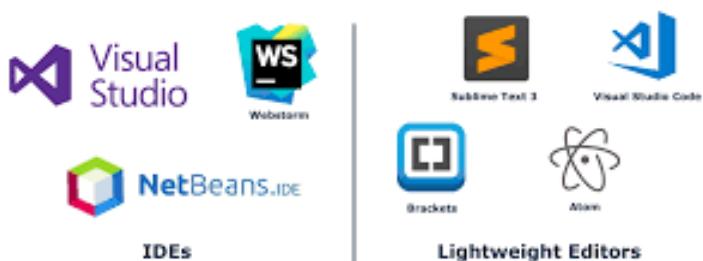
Perangkat lunak dan gim dapat dibedakan menjadi beberapa kategori ataupun jenis, yaitu:

a. Kategori Perangkat Lunak



Gambar 1.64 Perangkat Lunak Sistem (Sistem Operasi)
Sumber Gambar: https://itech.erickuha.com/book/ch1-computer-basics/4_operating_system.html

Sistem perangkat lunak komputer dapat diklasifikasikan menjadi tiga jenis utama, yaitu perangkat lunak sistem, perangkat lunak pemrograman, dan perangkat lunak aplikasi. **Perangkat lunak sistem** biasanya terdiri dari sistem operasi, perangkat *driver*, *server* dan komponen perangkat lunak. **Perangkat lunak pemrograman** sangat membantu dalam menulis kode program melalui alat-alat seperti editor, penghubung, *debugger*, *compiler* atau *interpreter*.



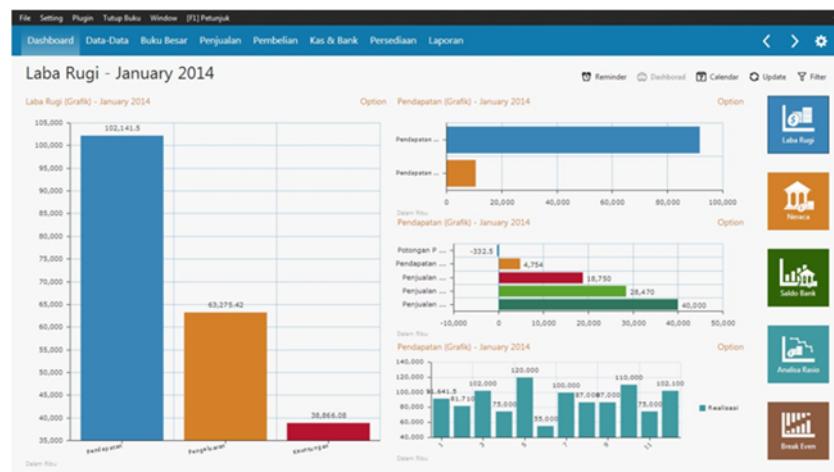
Gambar 1.65 *Code Editor* atau *IDE* Contoh Perangkat Lunak Pemrograman

Sumber Gambar: <https://codepumpkin.com/code-editors-for-web-development/>



Perangkat lunak aplikasi adalah perangkat lunak yang digunakan untuk mencapai atau mengerjakan tugas-tugas tertentu. Perangkat lunak aplikasi menggunakan kapasitas komputer secara langsung untuk tugas tertentu dan digunakan untuk mengetik dokumen, memanipulasi foto, merancang rumah, grafik, angka, dan lain-lain. Perangkat lunak aplikasi dapat digolongkan menjadi beberapa kelas, antara lain adalah sebagai berikut.

- 1) Perangkat lunak perusahaan (enterprise), meliputi perangkat lunak akuntansi, bagian pendukung (back office), perangkat lunak bisnis, dan manajemen sumber daya manusia.

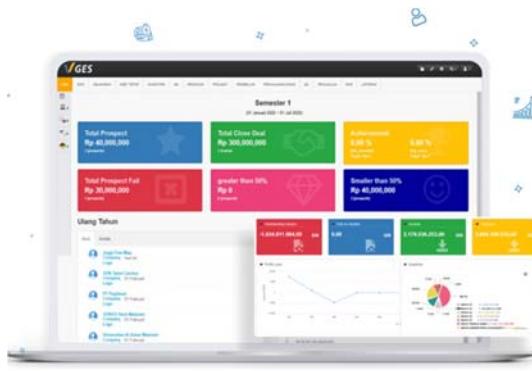


Gambar 1.66 *Dashboard Zahir Aplikasi Accounting*

Sumber Gambar: <https://zahiraccounting.com/id/blog/seberapa-penting-grafik-untuk-menilai-kinerja-perusahaan/dashboard-zahir/>

- 2) Perangkat lunak infrastruktur perusahaan, meliputi perangkat lunak alur kerja bisnis, sistem manajemen basis data, manajemen aset digital dan manajemen dokumen.

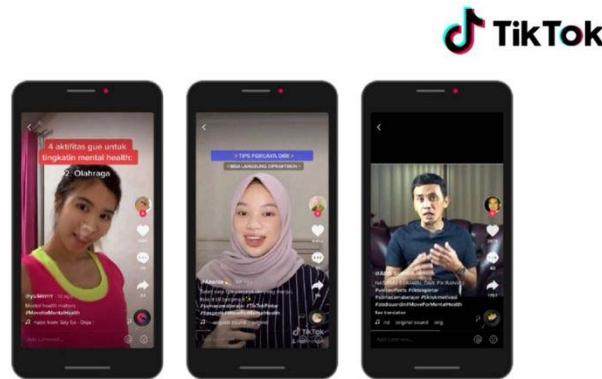




Gambar 1.67 Aplikasi *GES* (Alur Kerja Bisnis / *ERP*)

Sumber Gambar: <https://blog.gamatechno.com/8-software-erp-terbaik/>

- 3) Perangkat lunak informasi kerja meliputi manajemen waktu dan sumber daya, manajemen data, dokumentasi, perangkat lunak analitik, dan perangkat lunak kolaborasi.
- 4) Perangkat lunak media dan hiburan, meliputi media digital dan perangkat lunak hiburan.

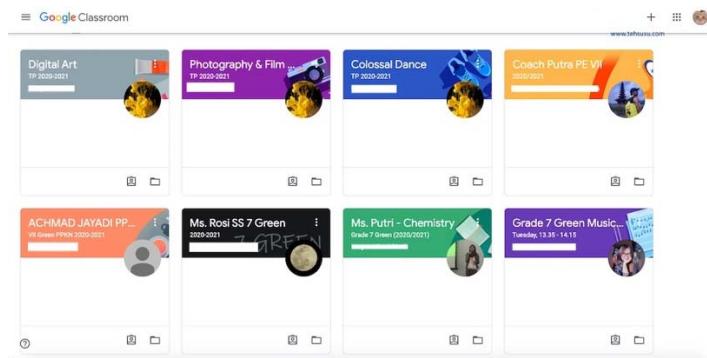


Gambar 1.68 Tiktok Perangkat Lunak Media dan Hiburan

Sumber Gambar: <https://nextren.grid.id/read/012454132/tiktok-udi-coba-durasi-video-sampai-3-menit-netizen-tidak-setuju?page=all>

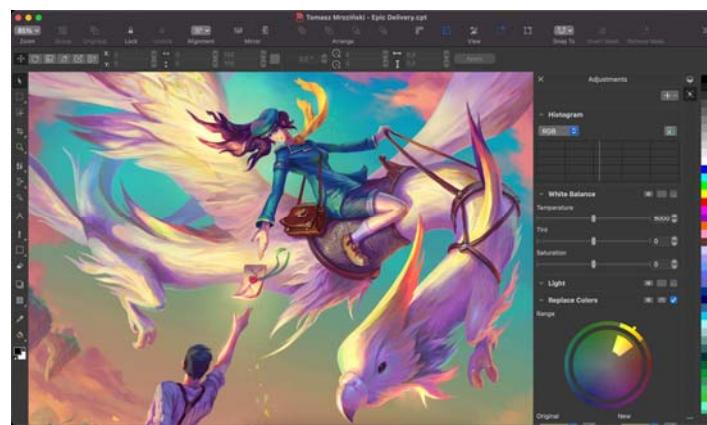
- 5) Perangkat lunak pendidikan, meliputi manajemen ruang kelas, manajemen survei, manajemen pelatihan, dan perangkat lunak kesiapan penjualan.





Gambar 1.69 *Google Classroom*, Perangkat Lunak Pendidikan.
Sumber Gambar: https://i.ytimg.com/vi/_I7eh0mfFVE/maxresdefault.jpg

- 6) Perangkat lunak pengembangan media, meliputi perangkat lunak seni grafis dan penyuntingan media.



Gambar 1.70 *Corel Draw Suite 2021*
Sumber Gambar: <https://www.coreldraw.com/en/product/coreldraw/>

- 7) Perangkat lunak rekayasa produk meliputi rekayasa perangkat keras dan rekayasa perangkat lunak.

b. Jenis - Jenis Gim (*Genre Game*)

Jenis *game* biasa disebut dengan istilah dengan *genre game* juga berarti gaya atau format dari sebuah *game*. Menurut Henry dalam bukunya yang berjudul "Panduan Praktis Membuat Game 3D" mengatakan format sebuah *game* bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa



genre lain. Jenis-jenis *game* menurut Henry adalah sebagai berikut:

1) *Maze Game*

Permainan *maze* atau yang dikenal dengan istilah labirin adalah sebuah permainan yang biasanya mencari jalan keluar dari jalan yang bercabang dan berliku sebagai sebuah tantangan/permasalahan. Jenis *game* ini biasanya menggunakan *maze* sebagai *setting* atau latar *game*. Jenis *game* *Maze* ini termasuk jenis *game* yang paling awal muncul. Contoh *game* ini adalah *game* Pacman dan Digger.



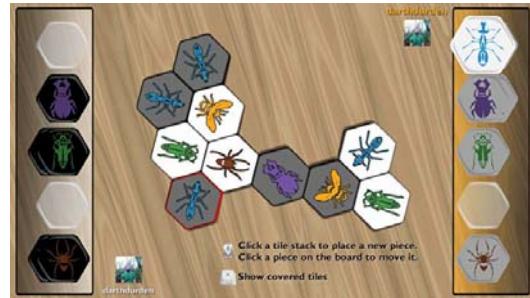
Gambar 1.71 Tampilan Permainan Pacman

(Sumber gambar: <https://pixabay.com/id/illustrations/pacman-pac-man-petualangan-lucu-4285160/>)

2) *Board Game*

Game jenis ini sama dengan *game board* tradisional pada umumnya. Hanya saja permainan ini dimainkan melalui komputer. Permainan papan atau *board game* menjadikan papan sebagai ciri khas dalam permainan tersebut.



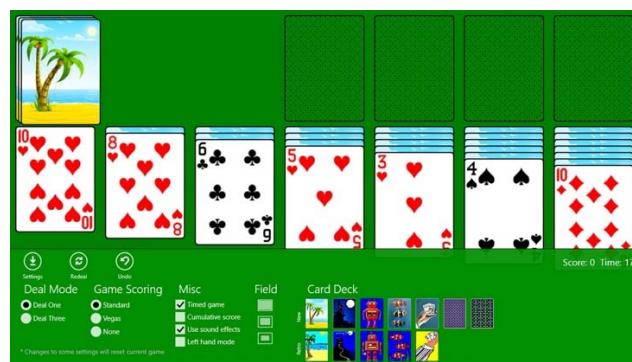


Gambar 1.72 Tampilan Permainan HIVE

(Sumber gambar: <https://www.dicebreaker.com/topics/digital-board-games/best-games/best-digital-board-games>)

3) *Card Game*

Jenis *game* kartu juga tidak jauh berbeda dari *game* tradisional aslinya. Namun, tampilannya lebih bervariasi dari versi tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah Solitaire dan Hearts.



Gambar 1.73 Tampilan Permainan Solitaire

(Sumber gambar: <https://www.microsoft.com/en-us/p/classic-solitaire-free/9wzdncrfj2sm?activetab=pivot:overviewtab#>)

4) *Battle Card Game*

Contoh *game* ini yang populer yaitu Battle Card Pokemon. *Game* ini jarang ditemukan di Indonesia. Film kartun yang bercerita tentang permainan *battle card* ini pernah ditayangkan di stasiun televisi Indonesia.



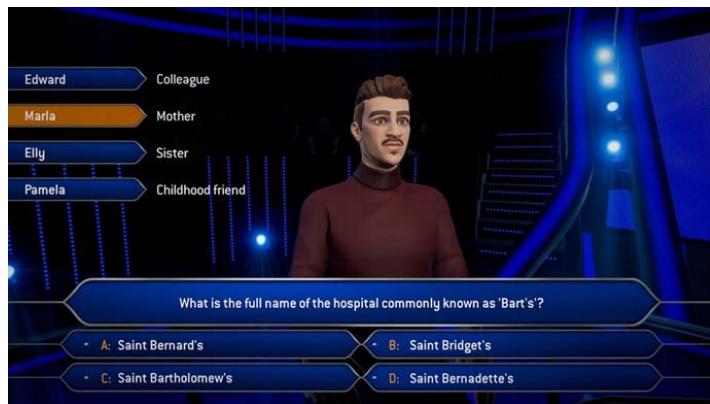


Gambar 1.74 Battle Card Game Pokemon

Sumber gambar: <https://www.pokemon.com/us/pokemon-tcg/play-online/>

5) Quiz Game

Game jenis ini merupakan *game* dengan bentuk kuis. Contoh *Quiz Game* yang pernah beredar yaitu *game* kuis Who Wants to Be Millionaire.



Gambar 1.75 Quiz Game Who Wants To Be A Millionre

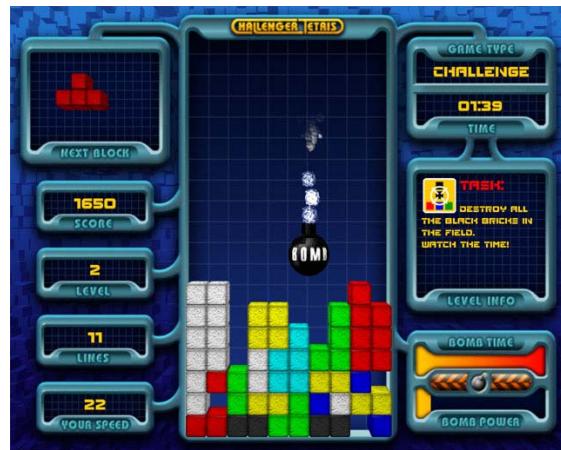
Sumber gambar:

https://store.steampowered.com/app/1356240/Who_Wants_To_Be_A_Millionaire/

6) Puzzle Game

Jenis *game* ini memberi tantangan dengan cara menjatuhkan atau melenyapkan sesuatu dari sisi atas ke bawah atau dari kiri ke kanan. Contoh *game* ini adalah Tetris.





Gambar 1.76 *Puzzle Game Tetris*

Sumber gambar:

<http://www.terminalstudio.com/screens/challtetris/big1.jpg>

7) *Shoot Them Up*



Gambar 1.77 *Shoot Them Up Space Impact*

Sumber gambar:

<http://www.3dfiggins.com/Archive/pages/spaceImpact.html>

Game jenis ini biasanya musuh berbentuk pesawat atau bentuk lain yang datang dari arah kanan, kiri, atau atas yang harus kita tembak sebanyak dan secepat mungkin. Dulu *game* ini berbentuk dua dimensi (2D), tetapi sekarang sudah berkembang dan menggunakan efek tiga dimensi (3D).

8) *Side Scroller Game*

Saat pertama kali muncul *game* ini berbentuk 2D. Sekarang sudah banyak yang dibuat dengan efek 3D. Pada *game* jenis



ini pemain diharuskan bergerak searah di alur yang disediakan. Dia diharuskan untuk berjalan, meloncat, merunduk serta menghindari rintangan-rintangan. Contoh *game* ini yang populer yaitu Mario Bros dan Prince of Persia.



Gambar 1.78 *Side Scroller Game Mario Bros 3D*

Sumber gambar:

<http://www.3dfiggins.com/Archive/pages/spaceImpact.html>

9) *Fighting Game*



Gambar 1.79 *Street Fighter*

Sumber gambar: <https://www.hienzo.com/2016/03/street-fighter-v-free-download-pc.htm>

Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh *game* ini yaitu Street Fighter, Samurai Showdown, Virtual Fighter dan Kungfu.



10) Racing Game



Gambar 1.80 Racing Game Project Cars Go

Sumber gambar: <https://allstarsyt.com/project-cars-go-download-android-ios-beta-apkobb-2/>

Adalah *game* tentang mengendarai kendaraan dengan cepat.

Contoh *game* ini yaitu Need for Speed Underground dan Toca Race Driver.

11) Turn-Based Strategy



Gambar 1.81 Turn Based Strategy Gear Tactics

Sumber gambar: <https://www.pcgamesn.com/best-turn-based-strategy-games>

Pemain dalam *game* ini melakukan gerakan setelah pemain lain melakukan gerakan jadi saling bergantian. Contoh *game* yang terkenal adalah Empire dan Civilization.



12) *Real-Time Strategy Game*



Gambar 1.82 *Real Time Strategy Warcraft III*

Sumber gambar: <https://www.filehorse.com/download-warcraft-3-reforged/>

Game ini seperti *game Turn-Based Strategy* (RTS), namun pada *game* ini pemain tidak perlu menunggu pemain lain. Pemain tercepatlah yang akan menang. Contoh *game* ini yaitu Warcraft.

13) *Simulation Video Game*



Gambar 1.83 *Euro Truck Simulator 2*

Sumber gambar: <https://www.filehorse.com/download-euro-truck-simulator-2/>

Genre ini merupakan bentuk permainan simulasi. Di sini pemain membangun sebuah area, kota, negara atau koloni.



Contoh *game* ini yaitu Ship Simulator, Train Simulator, dan Crane Simulator.

14) *First Person Shooter*



Gambar 1.84 *First Person Shooter Counterstrike*

Sumber gambar: <https://www.youtube.com/watch?v=ZMNNZzOzbh4>

Disebut *first person shooter* karena pandangan pemain adalah pandangan orang pertama. Banyak baku tembak dan mengutamakan kecepatan gerakan. Contoh *game* ini yaitu *game* Counterstrike dan Doom.

15) *First Person Shooter 3D*



Gambar 1.85 *First Person Shooter 3D Firestorm*

Sumber gambar: <https://www.rockpapershotgun.com/love-and-loathing-in-battlefield-firestorm>



Vehicle Based Game ini sama dengan FPS hanya saja pandangan pemain bukan dari orang pertama, tetapi dari kendaraan atau mesin yang digunakan. Kendaraan itu bisa berupa tank atau kapal.

16) *Third Person 3D Game*



Gambar 1.86 *Third Person 3D Game*

Sumber gambar: <https://www.slant.co/topics/1708/~best-open-world-third-person-action-rpgs-for-pc>

Game ini juga hampir sama dengan FPS hanya sudut pandang pemain merupakan sudut pandang orang ketiga.

17) *Role Playing Game*



Gambar 1.87 *Role Playing Game Beyond Divinity*

Sumber gambar: <https://www.pcgamebenchmark.com/beyond-divinity-system-requirements>

Jenis *game* ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan.



Contoh *game* ini adalah Legacy of Kain, Blade of Sword, dan Beyond Divinity

18) *Adventure Game*



Gambar 1.88 *Adventure Beyond Good and Evil 2*

Sumber gambar: <https://www.https://otakukart.com/440512/beyond-good-and-evil-2-ubisoft-announcement-new-deadline/>

Adventure game merupakan *genre game* petualangan. Di sepanjang perjalanan pemain akan menemukan peralatan yang akan disimpan dan berguna sebagai petunjuk perjalanan. Contoh : *Game* Sam and Max atau Beyond and Evil.

19) *Educational and Edutainment*



Gambar 1.89 *Educational and Edutainment*

Sumber gambar:

https://twitter.com/Taka_Radjiman/status/1263919875346411520/photo/1



Game ini lebih mengacu pada isi dan tujuan dari *game*. *Game* ini bertujuan memancing minat belajar anak sambil bermain. Contoh *game* ini adalah *Game* Boby Bola.

20) *Game Sport*



Gambar 1.90 *Game Sport Super Mega Baseball 3*

Sumber gambar: <https://sea.pc当地/pc-games/41917/the-best-pc-sports-games-for-2021>

Jenis *game* ini memiliki tema olahraga. *Game* yang mengetengahkan *genre* olahraga disebut *Sport Game*.

5. Karakteristik Perangkat Lunak dan Gim

Perangkat lunak (software) dan gim merupakan kumpulan instruksi dalam bentuk kode program yang ditulis menurut aturan bahasa pemrograman tertentu, disimpan, dan dijalankan pada perangkat keras untuk mengerjakan fungsi tertentu. Kita sehari-hari melihat perangkat lunak dalam bentuk aplikasi atau gim seperti pengolah kata atau pengolah grafis namun pada dasarnya semua aplikasi ini terbentuk dari sekumpulan instruksi dalam bentuk kode program. Jika perangkat lunak tersebut bersifat *open source* maka kita dapat melihat kode program tersebut.



Melalui definisi tersebut kita juga mengetahui bahwa perangkat lunak dan gim disimpan dan dijalankan pada perangkat keras. Dengan demikian saling menopang. Sebuah perangkat lunak atau gim tidak dapat bekerja jika tidak ada perangkat keras dan demikian pula perangkat keras tidak akan berfungsi. Kenyataan bahwa perangkat lunak atau gim disimpan secara digital pada perangkat keras memberikan ciri khas bahwa karakteristik perangkat lunak bersifat *invisible* (tidak terlihat). Karena bersifat digital atau non-fisik maka kualitas perangkat lunak atau gim tidaklah mudah untuk dinilai tidak seperti perangkat keras yang dapat dilihat dan diraba. Karena hal tersebut juga berarti setiap kerusakan atau cacat pada perangkat lunak atau gim membutuhkan penelusuran mendalam.

Konsekuensi lainnya dari bersifat non fisik adalah tidak mengalami kehausan karena pemakaian. Tidak terpengaruh oleh kondisi lingkungan seperti suhu, debu, dan sebagainya melainkan yang terpengaruhi adalah perangkat kerasnya. Lebih lanjut lagi perangkat lunak atau gim tidak memiliki suku cadang yang diproduksi masal. Bagaimanapun ini bukan berarti perangkat lunak tidak mengalami *error* atau cacat. Cacat ini terutama datang dari pengembangan perangkat lunak atau gim bukan dari pengguna. Berbeda dengan perangkat keras, katakanlah mobil yang baru diproduksi masih sempurna tanpa cacat, lama kelamaan aus dan menjadi cacat setelah digunakan.

Untuk memperbaiki cacat pada perangkat lunak atau gim, pengembang tentunya tidak pernah menyediakan suku cadang melainkan menggunakan *patch* (tambalan). *Patch* ini sifatnya bervariasi berdasarkan cacat yang ditemukan oleh pengguna.



Sebagian perangkat lunak membuat *patch* bukannya malah menambal celah kekurangan namun terkadang malah membuka celah baru. Keberadaan tambalan ini juga mengindikasikan bahwa karakteristik perangkat lunak atau gim adalah dapat digunakan kembali (*reusable*). Setiap cacat diberikan tambalan dengan mengubah atau menambah sebagian kode sedangkan kode lainnya tentu masih dapat digunakan kembali. Hal ini berlaku juga ketika ada versi baru dari perangkat lunak. Perangkat lunak terbaru tersebut sedikit banyak dapat mengambil potongan kode program versi lawas untuk digunakan kembali.

6. Proses Bisnis Pengembangan Perangkat Lunak dan Gim

Untuk mengetahui serangkaian aktivitas yang dilakukan dalam membuat produk perangkat lunak atau gim dapat kita ketahui melalui proses bisnis. Adapaun pengertian lain dari proses bisnis (Sparx System, 2004) adalah sekumpulan kegiatan atau aktifitas yang dirancang untuk menghasilkan suatu keluaran tertentu bagi pelanggan tertentu. Menurut Hammer dan Champy dalam Weske (2007) proses bisnis adalah sekumpulan kegiatan yang mengambil salah satu atau banyak masukan dan menciptakan sebuah keluaran yang berguna bagi pelanggan.

Bisnis Proses dapat digambarkan melalui *Business Process Modelling Notation*, yaitu suatu metodologi yang dikembangkan *Business Process Modelling Initiative* (BPMI) dalam memodelkan proses bisnis (Object Management Group, 2008). Tujuan dari BPMN adalah menyediakan notasi yang mudah dipahami oleh semua pengguna bisnis dan yang tidak kalah pentingnya adalah untuk



memastikan bahwa bahasa XML yang dirancang untuk pelaksanaan proses bisnis dapat dinyatakan secara visual dengan notasi yang umum (Owen and Raj, 2003). BPMN memiliki notasi khusus untuk menggambarkan peristiwa berbasis pesan dan pesan lewat diantara organisasi.

Berikut adalah elemen dan objek yang terdapat pada BPMN:

Tabel 1. 1 Elemen dan Objek BPMN

Shape	Element/Object
	Event
	Task/Activity
	Gateway
	Sequence Flow

a. *Event*



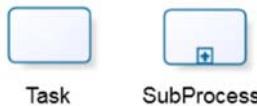
Gambar 1.91 *Event BPMN*

Event adalah sesuatu yang "terjadi" selama jalannya proses atau koreografi yang mempengaruhi aliran dari model dan biasanya memiliki penyebab (pemicu) atau dampak (hasil). *Event* digambarkan dalam lingkaran terbuka untuk membedakan



fungsinya. Ada tiga jenis *event*, berdasarkan pengaruh aliran proses yaitu Awal, Menengah, dan Akhir.

b. *Activity*



Gambar 1.92 *Activity BPMN*

Aktivitas adalah sebuah istilah umum untuk suatu kegiatan yang memperlihatkan perusahaan melakukan proses. Jenis kegiatan yang merupakan bagian dari proses sebuah model digambarkan bulat persegi panjang.

c. *Gateway*



Gambar 1.93 *Gateway BPMN*

Gateway digunakan untuk mengontrol perbedaan dan konvergensi dari urutan Arus dalam Proses. Dengan demikian, akan menentukan percabangan, forking, penggabungan, dan bergabung dengan jalur.

d. *Flow*

1) *Sequence Flow*



Gambar 1.94 *Sequence Flow*

Sebuah arus urutan digunakan untuk menunjukkan urutan kegiatan yang akan dilakukan dalam proses



2) *Message Flow*



Gambar 1.95 *Message Flow*

Digunakan untuk menunjukkan aliran pesan antara dua pelaku yang telah dipersiapkan untuk mengirim dan menerima mereka. Dalam BPMN, dua *Pools* terpisah dalam diagram kolaborasi akan mewakili dua peserta (misal: *partner* entitas atau *partner roles*).

3) *Association*



Gambar 1.96 *Association*

Digunakan untuk menghubungkan informasi dan artefak dengan elemen BPMN grafis. Teks penjelasan dan artefak lain dapat terkait dengan grafis elemen. Semua mata panah pada asosiasi menunjukkan arah aliran (misalnya: data)

4) *Pool*



Gambar 1.97 *Pool*

Adalah representasi grafis dari pelaku/peserta kolaborasi. Hal ini juga bertindak sebagai "*swimlane*" dan wadah grafis untuk partisi satu set kegiatan jadi *pools* lain, biasanya dalam konteks



situasi B2B. *Pool A* mungkin memiliki internal yang rinci, dalam bentuk proses yang akan dieksekusi.

5) *Lane*



Gambar 1.98 *Lane*

Lane adalah partisi sub-dalam proses, terkadang dalam *pool*, akan memperpanjang seluruh proses baik secara vertikal ataupun horisontal. Jalur yang digunakan untuk mengatur dan mengkategorikan kegiatan.

6) *Data Object*

Data Object



Data Objec (Collection)



Data Input Data Output



Gambar 1.99 *Data Object*

Data Object memberikan informasi tentang kegiatan apa yang perlu diadakan dan atau apa yang mereka hasilkan. *Data Object* dapat mewakili benda tunggal atau koleksi benda-benda. *Data Input* dan *Data Output* memberikan informasi yang sama untuk proses.



7) *Group*



Gambar 1.100 *Group*

Adalah pengelompokan unsur-unsur grafis yang berada dalam kategori yang sama. Jenis pengelompokan tidak mempengaruhi *Sequence Flow* dalam *Group*. Nama kategori muncul pada diagram sebagai label kelompok.

Kategori dapat digunakan untuk dokumentasi atau analisis tujuan. *Group* adalah salah satu cara dimana kategori benda dapat secara visual ditampilkan pada diagram.

7. Model Proses Pengembangan Perangkat Lunak dan Gim

Pada materi sebelumnya kita sudah mempelajari bagaimana sebuah aktivitas proses bisnis digambarkan melalui *Business Process Modelling Notation* (BPMN). Selanjutnya perusahaan pengembang perangkat lunak dan gim dapat mengembangkan atau memelihara produk/jasa yang akan dibuat berdasarkan proses bisnis yang telah ditetapkan.

Rekayasa perangkat lunak adalah ilmu yang melibatkan semua aspek pengembangan dan pemeliharaan produk perangkat lunak atau gim. Dalam banyak hal, rekayasa perangkat lunak sangat mirip dengan jenis rekayasa lainnya. Baik dalam membangun gedung, jembatan, rumah, pesawat terbang, dan lainnya. Dalam menyelesaikan tugas-tugas tersebut dibutuhkan sebuah rencana,



membuat draf perancangan, membangun hingga menguji apa yang telah Anda buat.

Menurut IEEE, rekayasa perangkat lunak adalah "*The Application of systematic, disciplined, quantifiable approach to the development, operation and maintenance of software*". Dengan demikian rekayasa perangkat lunak merupakan disiplin ilmu yang melibatkan semua aspek pengembangan, pembuatan hingga pemeliharaan produk perangkat lunak. Berdasarkan definisi tersebut maka bukan hanya seperti apa perangkat lunak yang dihasilkan itu penting, tetapi juga bagaimana proses pengembangannya.

Tujuannya rekayasa perangkat lunak adalah penciptaan perangkat lunak yang memenuhi kebutuhan pelanggan dan dapat diandalkan, efisien, serta mudah dipelihara. Selain itu harus dapat diproduksi secara ekonomis, memenuhi jadwal proyek, dan anggaran. Dalam mengembangkan perangkat lunak dan gim secara garis besar dibagi menjadi 3 tahapan yaitu :

a. Perancangan

Pemodelan perangkat lunak adalah abstraksi struktur perangkat lunak yang akan di implementasikan. Hal ini termasuk model data, struktur kode program, antarmuka, *input*, *output*, struktur navigasi atau alur kerja perangkat lunak. Perancangan ini diperlukan agar lebih memahami persyaratan perangkat lunak yang akan dibuat dan bentuk implementasi yang akan memenuhi persyaratan. Bentuk perancangan perangkat lunak itu sendiri dapat dipecah menjadi beberapa kategori yaitu:



1) Perancangan arsitektur

Merupakan perancangan yang menggambarkan keseluruhan perangkat lunak yang akan dibuat. Desain arsitektur untuk perangkat lunak serupa dengan denah ketika kita akan membangun sebuah rumah (*blueprint*). Gambaran secara menyeluruh yang dapat menggambarkan cara kerja, proses alir data, input hingga output.

2) Perancangan komponen

Perancangan komponen dapat memberikan gambaran yang rinci dan spesifik. Menggambarkan detail internal masing-masing komponen perangkat lunak seperti struktur data dan algoritma kode perangkat.

3) Perancangan basis data

Perancangan basis data adalah pengorganisasian data sesuai dengan model basis data. Perancangan basis data menentukan data apa yang harus disimpan dan bagaimana elemen-elemen data saling berhubungan.

4) Perancangan antarmuka

Perancangan antarmuka atau *interface* adalah proses menentukan bagaimana sistem akan berinteraksi dengan entitas eksternal yaitu pengguna aplikasi. Rancangan ini bertugas untuk menentukan cara bagaimana *user* akan berinteraksi dengan sistem serta *input* dan *output* yang diterima dan dihasilkan oleh sebuah perangkat lunak.

b. Implementasi

Fase ini merupakan fase inti dalam proses rekayasa perangkat lunak karena tanpa adanya implementasi tidak akan



pernah ada perangkat lunak atau gim yang akan dibuat. Bentuk implementasinya adalah kegiatan pemrograman. Pemrograman untuk mengimplementasikan rancangan yang telah dibuat pada tahap sebelumnya. Proses perancangan, implementasi dan pengujian dilakukan secara berurutan.

c. Pengujian

Pengujian dilakukan tiga level, pertama modul individual yang diuji oleh pengembang. Kedua modul terintegrasi dan diuji untuk memastikan bahwa mereka berinteraksi dengan benar. Ketiga, setelah semua modul terintegrasi, seluruh sistem diuji untuk memastikan bahwa itu memenuhi persyaratan pengguna. Pengujian sistem biasanya dilakukan oleh tim pengujian atau penjaminan kualitas independen. Proses pengujian ini dikenal dengan proses verifikasi. Tujuannya untuk menjawab pertanyaan apakah kita sudah membuat perangkat lunak atau gim dengan benar dan sesuai dengan kebutuhan?



REFLEKSI I

Setelah mempelajari materi ini, banyak manfaat yang akan kamu rasakan dalam memahami konsep dasar dan model pengembangan perangkat lunak gim yang tentunya dapat membekali pengetahuan anda dalam mendeskripsikan konsep dasar dan model pengembangan perangkat lunak yang kemudian dapat membantu anda dalam proses pengembangan perangkat lunak dan gim.





ASESMEN I

I. Tugas Mandiri

Coba Lakukan salah satu perencanaan bisnis proses dalam bidang pengembangan perangkat lunak atau gim yang kamu ketahui, dan cobalah tuangkan Business Process Modelling Notation!

II. Latihan Soal

a. Pilihan Ganda

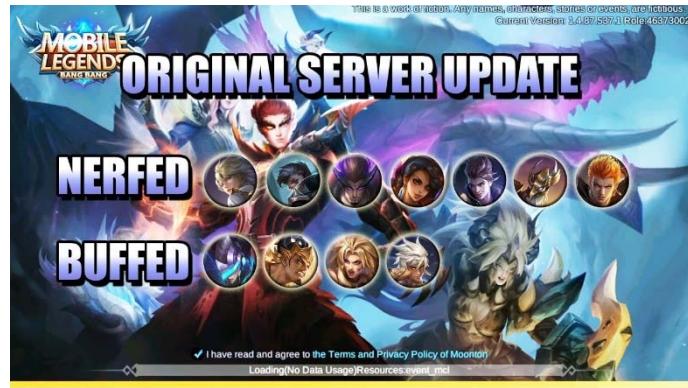
1. Manakah di bawah ini yang merupakan definisi dari perangkat lunak yang paling tepat...
 - a. Suatu kegiatan dengan beberapa rules dan tujuan, dimana sering kali (tetapi tidak selalu) melibatkan konflik, baik dengan User lain atau dengan sistem.
 - b. Aktivitas yang berlangsung dalam beberapa Batasan ruang dan waktu, dalam urutan yang terlihat, berdasarkan kepada urutan yang secara bebas diterima dan diluar lingkup kebutuhan.
 - c. Kumpulan Instruksi atau program komputer (syntaks/ code yang ditulis dalam bahasa pemrograman) dan ketika dijalankan menyediakan fungsi dan performa yang diinginkan.
 - d. Kegiatan yang menghadirkan konsep imajinatif, pengalaman koheren, sehingga dalam pengembangannya seorang desainer harus memiliki visi.



- e. Aktivitas yang menawarkan tantangan kecerdasan dan pengalaman yang mulus, sehingga dalam proses pembuatnya developer harus mengerti teknologi.
- 2. Manakah di bawah ini yang merupakan definisi dari gim yang paling tepat...
 - a. Kumpulan Instruksi atau program komputer (syntaks/code yang ditulis dalam bahasa pemrograman) dan ketika dijalankan menyediakan fungsi dan performa yang diinginkan.
 - b. Kumpulan program komputer (sintak/kode), prosedur, fungsi, modul (kumpulan program yang memiliki tujuan dan fungsi tertentu), dokumentasi dan kumpulan data yang diperlukan untuk mengoperasikan suatu sistem.
 - c. Suatu kegiatan dengan beberapa rules dan tujuan, dimana sering kali (tetapi tidak selalu) melibatkan konflik, baik dengan User lain atau dengan sistem.
 - d. Suatu aktivitas yang bersifat sistematik, dalam pengembangannya menerapkan prinsip - prinsip rekayasa, metode pengembangan, operasi, dan maintenance.
 - e. Kumpulan program komputer yang dalam pembuatannya menerapkan serangkaian studi pendekatan rekayasa.



3. Perhatikan gambar di bawah ini!



Berdasarkan gambar diatas, gambar tersebut adalah gambar *loading patch* pada game *mobile legends*. Dari halaman tersebut bisa kita ketahui bahwa sebuah game atau perangkat lunak mempunyai karakteristik...

- a. Efisien
- b. Mudah dirawat (maintainable)
- c. Dapat diterima
- d. Dapat diandalkan
- e. Mudah dijumpai

b. Essay

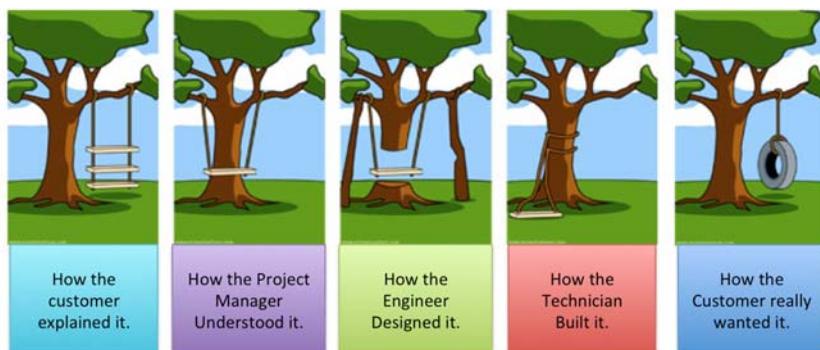
- 1. Apakah yang dimaksud dengan perangkat lunak dan gim?
- 2. Jelaskan secara singkat sejarah pengembangan perangkat lunak dan gim!
- 3. Apa saja aspek dan elemen yang membedakan perangkat lunak dan gim?
- 4. Sebutkan jenis – jenis perangkat lunak dan gim!
- 5. Apa yang dimaksud dengan Business Process dan Business Process Modelling Notation?



B. Budaya Mutu dan Manajemen Proyek Pengembangan Perangkat Lunak dan Gim

1. Manajemen Proyek

Perhatikan gambar dibawah ini!



Gambar 1.261 *Manajemen Projek Tree*

Sumber gambar: <http://www.cs.virginia.edu/~up3f/cs1111/inclass/inclass-ambiguity-wrapup.html>

Manajemen proyek merupakan bagian penting dalam perangkat lunak dan gim. Hal yang dibahas dalam melaksanakan manajemen proyek adalah perencanaan, manajemen risiko, spektrum manajemen 5P (*project, process, people, price* dan *place*), dokumentasi dan pelaporan serta manajemen kualitas.

Dalam membuat perangkat lunak atau gim perlu dikelola dengan baik karena setiap produk selalu dibatasi dengan anggaran dan jadwal yang ada. Tugas seorang manajer proyek adalah memastikan bahwa proyek yang dibuat memenuhi batasan serta memberikan produk yang berkualitas. Manajemen proyek yang baik memang tidak dapat menjamin *output* yang baik, namun manajemen yang buruk sangat berpotensi menghasilkan kegagalan proyek seperti tidak selesainya produk perangkat lunak atau gim, sehingga menghabiskan biaya lebih dari perkiraan anggaran atau gagal dalam memenuhi kebutuhan pelanggan.



Kriteria keberhasilan dalam manajemen proyek jelas bervariasi dari satu proyek ke proyek lainnya, namun secara garis besar tujuan terpenting adalah untuk menyerahkan produk yang sesuai ekspektasi pengguna akhir dengan anggaran dan waktu yang disepakati. Tujuan ini tidak untuk rekayasa perangkat lunak namun merupakan tujuan dari semua manajemen proyek termasuk dalam pembuatan *game*. Beberapa komponen yang perlu diperhatikan dalam manajemen perangkat lunak atau gim :

- a. Perangkat lunak dan gim tidak berwujud, adalah produk *intangible*, tidak berwujud fisik. Hal inilah yang paling membedakan manajemen rekayasa perangkat lunak dengan rekayasa teknik lainnya seperti teknik sipil bangunan, transportasi atau jalanan. Karena tidak berwujud fisik estimasi biaya serta progres pengembangan perangkat lunak dan gim sering kali sulit untuk dinilai. Berbeda dengan teknik pembangunan jembatan misalnya progres dapat dilihat langsung pada bentuk fisik, struktur yang ada.
- b. Perubahan sangat cepat, salah satu hal yang paling menantang pada manajemen perangkat lunak dan gim adalah perubahan teknologi yang begitu cepat. Berbeda dengan bangunan yang sekali dibuat mungkin bisa bertahan lama, perangkat lunak dan gim dapat berubah-ubah dengan cepat sesuai dengan teknologi dan kebutuhan yang ada.
- c. Proses perangkat lunak dan gim bersifat variabel dan spesifik, organisasi proses rekayasa untuk beberapa jenis sistem seperti jembatan dan bangunan memiliki standar. Namun, proses



perangkat lunak dan gim sangat bervariasi dari satu organisasi ke organisasi lainnya.

Manajemen proyek diperlukan untuk memastikan bahwa tujuan telah ditetapkan, dapat dipantau, dan akhirnya dapat terpenuhi. Hal ini diperlukan untuk menjaga anggota tim tetap berjalan dan fokus pada masalah yang dihadapi, tanpa terganggu oleh isu-isu sampingan yang tidak penting. Pada proyek yang lebih kecil, seseorang mungkin memainkan peran manajemen ganda. Pada proyek yang sangat kecil, satu orang dapat melakukan semua peran termasuk manajer, pengembang, pengujji, dan pengguna akhir. Salah satu komponen utama dari manajemen perangkat lunak dan gim adalah perencanaan proyek. Ini merupakan tahap pertama pada sebuah proyek perangkat lunak dan gim.

2. Perencanaan Proyek

Proses pengembangan perangkat lunak dan gim dimulai dengan kegiatan yang dikenal dengan perencanaan. Sebelum proyek dapat dimulai, tim harus dapat memperkirakan pekerjaan yang akan dilakukan, sumber daya yang akan dibutuhkan, estimasi waktu yang dijadwalkan dari awal sampai akhir. Setelah kegiatan ini selesai, tim harus menetapkan jadwal proyek yang mendefinisikan tugas dan mengidentifikasi siapa yang bertanggung jawab untuk melakukan setiap tugas, dan menentukan keterkaitan antar tugas.

Tujuan perencanaan proyek adalah untuk menyediakan kerangka kerja untuk membuat perkiraan sumber daya, biaya, dan jadwal yang dibutuhkan. Nantinya hasil analisis ini akan menentukan kelayakan dari proyek yang akan dibuat. Jika dirasa proyek tidak layak maka langkah tahapan berikutnya tidak akan diteruskan. Meskipun



ada saja ketidak pastian selama perkiraan, tim harus bisa memulai menjalankan rencana yang telah ditetapkan. Karena itu, rencananya harus bisa disesuaikan dan diperbaharui saat proyek berjalan. Pada fase perencanaan proyek ada beberapa langkah yang perlu dilalui di antaranya:

a. Definisi lingkup proyek

Mendefinisikan cakupan proyek adalah langkah pertama untuk perencanaan proyek. Pada tahap ini Anda menetapkan tujuan dan batasan proyek yang nantikan akan digunakan untuk mengalokasikan sumber daya.

b. Analisis sumber daya

Analisis sumber daya manusia adalah proses analisis dan identifikasi yang dilakukan kelompok terhadap kebutuhan akan sumber daya manusia, sehingga kelompok tersebut dapat menentukan langkah yang harus diambil guna mencapai tujuannya. Kebutuhan sumber daya manusia (SDM) di sini mulai dari tingkat manajer hingga teknik dengan spesialisasi tertentu seperti perancang basis data, sistem analisis. Untuk proyek yang relatif kecil seseorang mungkin dapat merangkap beberapa peran sekaligus, bagaimana pun pada proyek yang lebih rumit peran yang dibutuhkan mungkin akan lebih spesifik.

c. Analisis alat bantu

Sejumlah *vendor* menjual alat bantu untuk membantu perekayasa dalam mengembangkan aplikasi perangkat lunak ataupun gim. Peralatan ini dikenal dengan istilah *Computer Aided Software Engineering* (CASE). Ada sejumlah CASE yang tersedia untuk menyederhanakan berbagai tahap siklus hidup



pengembangan seperti alat analisis, alat desain, alat manajemen proyek, alat manajemen basis data, dan alat dokumentasi. Secara keseluruhan penggunaan alat CASE mempercepat pengembangan proyek untuk menghasilkan hasil yang diinginkan dan membantu mengungkap kekurangan sebelum bergerak maju dengan tahap berikutnya dalam pengembangan perangkat lunak.

d. Analisis jadwal

Penjadwalan proyek adalah proses menentukan bagaimana pekerjaan dalam suatu proyek akan didekomposisi sebagai tugas-tugas terpisah, dan kapan tugas-tugas ini akan dilaksanakan. Anda memperkirakan waktu kalender yang dibutuhkan untuk menyelesaikan setiap tugas, dan siapa yang akan mengerjakan tugas. Ingat jadwal ini hanya rencana dan dapat disempurnakan atau dimodifikasi selama tahap pembangunan.

e. Analisis biaya

Saat Anda mengajukan penawaran untuk kontrak proyek pembuatan perangkat lunak atau gim, Anda harus menentukan harga proyek. Sebagai titik awal untuk menghitung harga ini, Anda perlu membuat estimasi biaya untuk menyelesaikan pekerjaan proyek. Estimasi ini menghitung berapa banyak usaha yang diperlukan untuk menyelesaikan setiap aktivitas dan menghitung total biaya. Anda harus selalu menghitung biaya secara obyektif agar dapat memprediksi secara akurat biaya pengembangan perangkat lunak ataupun gim. Banyak faktor yang mempengaruhi penetapan harga proyek perangkat lunak bukan sekedar biaya ditambah keuntungan harga. Ada beberapa



komponen yang perlu diperhitungkan ketika menentukan harga perangkat lunak di antaranya :

- 1) Biaya usaha (upah tim pengembang perangkat lunak)
- 2) Biaya perangkat keras/lunak
- 3) Biaya pemeliharaan
- 4) Biaya lainnya (biaya perjalanan, pelatihan, dokumentasi dsb.)

f. Analisis risiko

Semua proyek memiliki risiko dengan bermacam-macam tingkatan. Sebagian risiko berpotensi menyebabkan masalah seperti keterlambatan jadwal atau peningkatan biaya proyek sebagian lainnya bahkan mengakibatkan proyek gagal. Mungkin bagian tersulit dari manajemen risiko adalah identifikasi, karena itu memerlukan membayangkan bagian dari proses yang mungkin pada awalnya tidak tampak mengandung risiko.

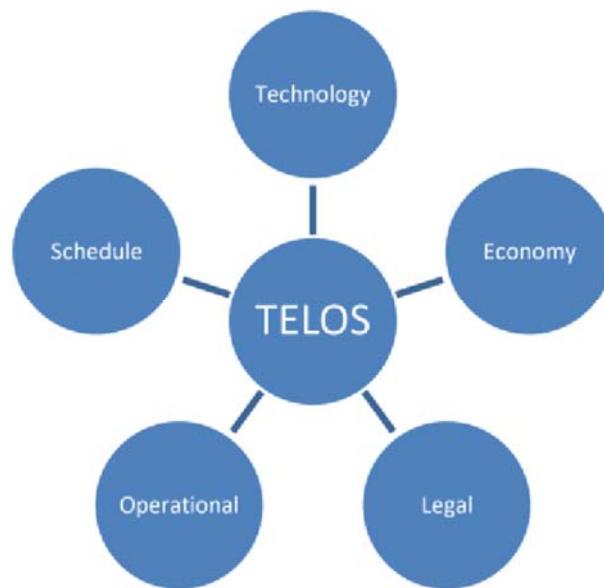
g. Analisis kelayakan

Salah satu tujuan utama dari perencanaan proyek perangkat lunak atupun gim adalah analisis kelayakan. Perencanaan merupakan pondasi bagi seluruh proses perangkat lunak ataupun gim karena proses ini menentukan apakah sebuah proyek dapat dilanjutkan atau tidak. Bayangkan Anda baru saja mendapatkan ide untuk proyek baru. Anda yakin bahwa proyek ini akan menghemat waktu dan uang organisasi Anda, kemudian dengan penuh antusias, Anda menyampaikannya kepada manajer Anda. Namun, dia punya beberapa pertanyaan. Dia ingin tahu apakah proyek Anda layak secara komersial, apakah itu kompatibel dengan teknologi yang digunakan organisasi Anda, dan bagaimana hal itu dapat memengaruhi pekerjaan yang sedang



berlangsung. Untuk menjawab pertanyaan seperti ini, Anda dapat menggunakan kerangka kerja TELOS.

Dengan menggunakan TELOS, Anda dapat meningkatkan desain proyek agar lebih berhasil, dan Anda dapat melihat proyek yang pada dasarnya sudah cacat, sebelum Anda menginvestasikan banyak waktu dan tenaga di dalamnya. TELOS adalah akronim dari lima bidang utama yang perlu Anda jelajahi sebagai bagian dari studi kelayakan proyek Anda yakni Teknologi, Ekonomis, Legal, Operasional dan Schedule.



Gambar 1.102 Elemen TELOS

3. Teknik Pemasaran

Dalam bisnis, pemasaran adalah proses penting agar bisnis bisa berhasil. Sebagus apa pun produknya, termasuk produk aplikasi, jika produk itu tidak diterima oleh pasar, produk itu tidak ada gunanya.

a. Teknik Pemasaran Perangkat Lunak



Ada beberapa cara yang dapat dilakukan untuk penetrasi/pemasaran pasar pada pengembangan perangkat lunak:

1) Penetrasi langsung ke pasar

- Biasanya dilakukan oleh pengusaha yang *track record*-nya belum dikenal
- Menawarkan langsung produk aplikasi ke calon pengguna. Bisa melalui website atau dengan pertemuan bisnis.
- Yang penting mudah dalam memperkenalkan produk pada orang/pihak yang sudah dikenal terlebih dahulu.

2) Sub kontrak

- Menjadi mitra sub kontrak pada perusahaan – perusahaan aplikasi yang sudah mapan dan terkenal.
- Jika hasil perusahaan dinilai berkualitas, maka ada peluang untuk mendapatkan order pekerjaan secara terus-menerus.
- Konsekuensi, tingkat keuntungan lebih rendah dibandingkan jika mendapatkan pekerjaan langsung dan konsumen.

3) Pinjam nama

- Banyak dilakukan oleh perusahaan yang belum terkenal
- Jika ingin menjalin hubungan bisnis dengan lembaga pemerintah karena adanya peraturan presiden

b. Teknik Pemasaran Gim

Berikut adalah teknik publishing / pemasaran dalam gim:

1) Melakukan Publikasi Mandiri

Langsung memasukkan game yang telah dibuat ke Google playstore, Windows game, app store, atau yang lainnya. Tentu saja agar game tersebut menarik calon pengguna harus dilakukan promosi yang tepat.



2) Melakukan Publikasi Publishing

Contoh bekerja sama dengan lyto atau megaxus (dalam negeri) atau dengan NoodleCake, Crescent Moon Games, Chillingo, Bullkypix (luar negeri).

Kelebihan:

- Program promosi menjadi tanggung jawab publishing, termasuk biayanya.
- Publishing yang terkenal akan menjamin produk tepat sasaran.

Kekurangan:

- Keuntungan dibagi antara developer dan publishing.
- Kadang – kadang publishing memiliki “moment bisnis” yang berbeda dengan developer. Jika produk tidak laku, maka bisa terjadi saling menyalahkan. Untuk itu harus dipastikan adanya surat perjanjian yang jelas.

Setelah melakukan publishing, kita dapat menentukan bagaimana cara melakukan promosi agar game laku di pasaran:

- a) Memasang iklan pada game lain
- b) Tempat yang pas di Google Play Store dan Apple Store
- c) Mengadakan game berhadiah
- d) Membuat merchandise
- e) Mengikuti game exhibition (pameran game)

4. Budaya Mutu dan K3LH

a. Budaya Mutu Upaya Peningkatan Kualitas Perangkat Lunak dan Gim

Budaya Mutu merupakan sistem nilai dari sebuah organisasi yang menghasilkan keadaan lingkungan yang kondusif dalam



pembentukan perbaikan yang berkelanjutan dalam segi mutu. Budaya Mutu terdiri dari nilai-nilai, tradisi, prosedur, dan harapan yang mengedepankan mutu.

Kita di sini semua akan setuju bahwa perangkat lunak dan gim berkualitas tinggi adalah tujuan utama. Tetapi bagaimana kita mendefinisikan berkualitas? Dalam arti yang paling umum, kualitas dapat didefinisikan sebagai proses yang efektif diterapkan dengan cara yang menciptakan produk yang bermanfaat yang memberikan nilai terukur bagi mereka yang memproduksinya dan mereka yang menggunakannya. Berdasarkan definisi tersebut setidaknya ada dua poin penting: Produk yang bermanfaat memberikan konten, fungsi, dan fitur yang diinginkan pengguna, tetapi yang lebih penting, diberikan dengan cara yang andal, bebas kesalahan. Kedua, memberikan manfaat bagi organisasi atau pengguna akhir.

Pengembang perangkat lunak dan gim memperoleh nilai tambah karena produk berkualitas tinggi memerlukan lebih sedikit upaya pemeliharaan, lebih sedikit bug yang diperbaiki, dan berkurangnya biaya dukungan pelanggan. Ini memungkinkan para pengembang menghabiskan lebih banyak waktu untuk membuat aplikasi baru dan lebih sedikit untuk mengerjakan ulang.

Berikut ini adalah faktor – faktor yang dapat menentukan sebuah kualitas perangkat lunak menurut McCall:

Tabel 1.2 Faktor Penentu Kualitas Perangkat Lunak

No	Faktor	Keterangan
1	<i>Correctness</i>	Kebenaran dari <i>output</i> perangkat lunak. Sejauh mana suatu program memenuhi



		spesifikasinya dan memenuhi tujuan misi pelanggan
2	<i>Reliability</i>	Tingkat kegagalan maksimum yang diizinkan dari sistem perangkat lunak, dan dapat merujuk ke seluruh sistem atau satu atau lebih dari fungsinya yang terpisah
3	<i>Efficiency</i>	Jumlah sumber daya yang dibutuhkan oleh suatu program untuk menjalankan fungsinya. Mencakup kemampuan pemrosesan (MHz), kapasitas penyimpanannya (MB/GB) dan kemampuan komunikasi data (MBPS).
4	<i>Integrity</i>	Berkaitan dengan keamanan sistem perangkat lunak. Sejauh mana akses ke perangkat lunak atau data oleh orang yang tidak berwenang dapat dikendalikan
5	<i>Usability</i>	Upaya yang diperlukan bagi seorang karyawan baru untuk belajar dan mengoperasikan perangkat lunak
6	<i>Maintainability</i>	Upaya untuk menemukan dan memperbaiki kesalahan dalam suatu program
7	<i>Flexibility</i>	Usaha yang diperlukan untuk memodifikasi program untuk mendukung kegiatan pemeliharaan perangkat lunak adaptif
8	<i>Testability</i>	Upaya diperlukan untuk menguji suatu program untuk memastikan bahwa ia menjalankan fungsi yang dimaksudkan
9	<i>Portability</i>	Kemampuan adaptasi dari sistem perangkat lunak ke lingkungan lain yang terdiri dari



		perangkat keras yang berbeda, sistem operasi yang berbeda, dan sebagainya
10	<i>Reusability</i>	Sejauh mana suatu program atau bagian dari suatu program dapat digunakan kembali dalam aplikasi lain
11	<i>Interoperability</i>	Upaya yang diperlukan dari suatu perangkat lunak untuk dihubungkan ke sistem lain.

b. Kesehatan, Keselamatan, Kerja dan Lingkungan Hidup (K3LH)

Selain budaya mutu, penting bagi perusahaan pengembang perangkat lunak dan gim untuk selalu menjamin kesehatan, keselamatan, kerja dan lingkungan hidup para karyawan.

Kesehatan, Keselamatan Kerja dan Lingkungan Hidup adalah suatu menciptaan kenyamanan dalam pekerjaan agar terhindar atau tercegah dari bahaya akibat kecelakaan kerja, serta lingkungan yang nyaman.

Menurut Sama'mur tahun 1992, Keselamatan kerja merupakan sarana utama untuk mencegah kecelakaan seperti cacat dan kematian akibat kecelakaan kerja. Kecelakaan kerja dalam hubungannya dengan perlindungan tenaga kerja adalah salah satu segi penting dari perlindungan tenaga kerja. Keselamatan kerja yang dilaksanakan sebaik-baiknya akan membawa iklim yang aman dan tenang dalam pekerja sehingga sangat membantu hubungan kerja dan menejemen.

Keselamatan kerja memang menjadi poin penting yang wajib diperhatikan oleh setiap perusahaan. Semakin minim kecelakaan, akan semakin besar produksi yang dihasilkan sehingga produktivitas bisa semakin ditingkatkan tanpa adanya kendala yang



bisa merugikan perusahaan. Namun pada kenyataannya masih banyak perusahaan yang tidak memperhatikan hal ini dan cenderung memandangnya dengan sebelah mata saja. Sangat disayangkan jika keselamatan kerja dikesampingkan.

Untuk mengetahui efektivitas dari suatu pelatihan K3 dapat diukur dengan memperhatikan indikator keberhasilan pelatihan (Widuri, 1992), yaitu:

1. Prestasi kerja karyawan
2. Kedisiplinan karyawan
3. Absensi karyawan
4. Tingkat kerusakan produksi, alat-alat, dan mesin
5. Tingkat kecelakaan karyawan
6. Tingkat pemborosan bahan baku, tenaga, dan waktu
7. Tingkat kerja sama karyawan
8. Tingkat upah karyawan
9. Prakarsa karyawan
10. Kepemimpinan dan kepuasaan manajerial

Pada umumnya Kesehatan dan Keselamatan Kerja (K3) dalam menggunakan perangkat teknologi komputer yang sering digunakan dalam bidang pengembangan perangkat lunak dan gim, meliputi:

- 1) Persyaratan pencahayaan ruang komputer

Pencahayaan ruang komputer harus memungkinkan orang bekerja dengan enak dan mata tidak mudah lelah. Pekerjaan dengan tingkat ketelitian dan kecermatan seperti di ruang komputer dibutuhkan intensitas nilai pencahayaan 40 sampai dengan 80 *foot candles*, pada bidang 30 inci dari lantai.



Untuk daerah penempatan *visual display units*, intensitas nilai pencahayaan sebaiknya tidak lebih dari 50 *foot candles*, sedangkan penempatan konsol dan panel kontrol harus dihindarkan dari sinar matahari langsung.

2) Tata letak ruang komputer

Berdasarkan pertimbangan-pertimbangan yang telah ditentukan, ruang komputer sebaiknya memenuhi:

- a) Pemilihan material untuk pembuatan ruang dianjurkan memenuhi ketentuan ketahanan terhadap api, berdasarkan standar NEPA.
- b) Dinding dan plafon mampu menyerap suara yang ditimbulkan dalam ruang, dan menahan suara serta panas dari luar ruangan.
- c) Jauh dari daerah atau vents pelepas panas dan asap.
- d) Jauh dari daerah kegiatan mesin pres atau sejenisnya yang menimbulkan sumber getar.
- e) Tidak langsung dibawah lantai yang banyak tandon air atau kegiatan yang menggunakan banyak air.
- f) Tidak terletak di bawah lantai yang airnya tidak bisa dikontrol dengan baik.
- g) Jauh dari pusat pembangkit medan listrik dan medan magnet, seperti gardu induk, gardu transformator, saklar pemutus beban listrik besar, saluran listrik berdaya besar dengan kawat telanjang, motormotor listrik.
- h) Jauh dari daerah untuk kegiatan proses kimia, seperti pembuatan printed dengan proses etching.
- i) Jauh proses material yang menimbulkan debu atau asap.



3) Tata letak ruang komputer

Syarat teknis ruang komputer tidak lepas dari sifat amannya ruang terhadap gangguangangguan, yaitu :

- a) Terjaminnya nilai temperatur ruang
- b) Terjaminnya nilai kelembaban ruang
- c) Bebas debu
- d) Bebas pengaruh medan magnet dan listrik
- e) Bebas getaran
- f) Bebas asap
- g) Bebas dari gas-gas tertentu
- h) Bebas zat kimia
- i) Terjaminnya nilai pencahayaan
- j) Akustik ruang

4) Mengatur posisi duduk



Gambar 1.103 Posisi duduk

Sumber Gambar: <https://katigaku.top/2020/12/11/penerapan-k3-di-bidang-teknologi-informasi-yang-diperlukan-di-masa-modern-saat-ini/>



Penempatan kursi, meja mouse, keyboard, dan layar komputer yang benar akan membantu membuat perubahan dalam mencegah resiko gangguan kesehatan.

a) Area komputer

Biarkan area komputer di ruangan teratur. Alat-alat yang sering digunakan sebaiknya dekat sehingga mudah menjangkaunya. Hindari menyimpan sesuatu dibawah meja yang dapat mengganggu posisi kaki.

b) Duduk dengan posisi yang baik

Ketika duduk, tempatkan pantat tepat dikursi. Duduk tegap dan cobalah untuk menjaga pinggul, bahu dan telinga dalam posisi lurus.

c) Kaki sebaiknya menyentuh lantai

Jika kaki tidak menyentuh lantai, rendahkan kursi atau gunakan sandaran kaki.

d) Posisi monitor

Monitor harus ditempatkan di mana bagian atas monitor berada tepat di mata dan langsung berhadapan. Jarak antara operator dengan monitor kira - kira 15 – 30 inch.

e) Istirahat dan ganti posisi

Jalan – jalan sebentar dapat mengurangi stress dan ketegangan pada otot dengan melentangkan badan membuat perbedaan yang besar

5) Memperhatikan jarak pandang dengan komputer

Dalam menggunakan komputer yang baik dan demi keselamatan kerja dari pengguna komputer (*brainware*) harus



terlebih dahulu mengetahui prosedur-prosedur yang aman dalam bekerja. Sikap posisi duduk yang baik, jarak pandang mata terhadap komputer haruslah ditaati demi kesehatan dan keselamatan kerja.

Pengaturan posisi duduk dalam menggunakan komputer, jarak pandang dan juga prosedur perawatan komputer perlu diketahui oleh para *brainware*.

Usaha dalam mengurangi kelelahan mata, punggung dan leher dapat dilakukan sebagai berikut:

- a) Garis pandang dari mata harus tegak lurus pada monitor berjarak 50 cm
- b) Bagian belakang punggung belakang sandaran kursi harus keras, tapi berbantal empuk dan tegak posisi 90°
- c) Lakukan gerakan untuk melemaskan otot
- d) Istirahatlah sebentar-sebentar tapi sering
- e) Tinggalkan komputer sejenak dan lakukan refresing
- f) Usahakan penerangnya tidak menyilaukan mata
- g) Tinggi atau letak monitor sesuai dengan arah pandang mata, agar mudah melihatnya
- h) Perbanyaklah makan makanan yang mengandung vitamin A, seperti : wortel, pisang, dan sebagainya

6) Menghidupkan komputer sesuai dengan prosedur

Dalam mengaktifkan komputer yang baik agar tidak terjadi kerusakan pada komputer haruslah mengikuti prosedur yang ada, langkah-langkah yang dilakukan dapat mengikuti petunjuk sebagai berikut :

- a) Sambungkan kabel listrik



- b) Hidupkan stavol (sebagai penyeimbang arus)
- c) Hidupkan CPU dengan menekan tombol *power (on/off)*
- d) Hidupkan monitor dengan menekan tombol *power (on/off)*
- e) Tunggu hingga muncul tampilan yang disebut dengan *deskstop*

Dalam mematikan komputer juga perlu diperhatikan prosedur yang benar. Langkah-langkah dalam mematikan komputer yang baik sebagai berikut :

- a) *Close* program aplikasi yang telah selesai digunakan dengan cara *close windows* atau dengan klik *file* lalu *exit*
- b) Klik *start/begin/mulai* yang biasanya berada disebelah kiri bawah
- c) Pilih *shut down* (mematikan komputer)
- d) Muncul *dialog box shut down*, pilih *shut down*
- e) Klik *yes*, tunggu hingga monitor sudah tidak aktif.
- f) Matikan komputer atau layar pada tombol *power* monitor.
- g) Matikan tombol *power* pada CPU (jika sistem *shut down* tidak otomatis, jika CPU otomatis mati maka tombol *power* tidak perlu ditekan).



REFLEKSI II

Setelah mempelajari materi ini, banyak manfaat yang akan kamu rasakan dalam meningkatkan budaya mutu dan manajemen proyek pada suatu perangkat lunak atau gim, maka banyak sekali peluang yang akan kamu rasakan setelah memahami manajemen proyek, perencanaan proyek, teknik pemasaran dan budaya mutu pada perangkat lunak dan gim yang tentunya dapat membekali



pengetahuan dan keterampilan peserta didik dalam menjalankan proyek yang berkualitas pada perusahaan atau *startup* dalam perangkat lunak dan gim.



ASESMEN II

I. Tugas Mandiri

Buatlah sebuah proposal perencanaan manajemen proyek mulai dari perencanaan proyek hingga teknik pemasaran dalam bidang pengembangan perangkat lunak atau gim!

II. Latihan Soal

1. Apa yang dimaksud dengan manajemen proyek?
2. Jelaskan apa yang di maksud perencanaan proyek?
3. Bagaimana teknik pemasaran pada bidang perangkat lunak dan gim?
4. Apa yang dimaksud dengan Budaya Mutu?
5. Sebut dan jelaskan apa saja K3 bidang perangkat lunak dan gim?



C. Perkembangan teknologi industri 4.0

1. Industri 4.0

Era Revolusi Industri keempat ini diwarnai oleh kecerdasan buatan (*artificial intelligence*), super komputer, rekayasa genetika, teknologi nano, mobil otomatis, dan inovasi. Perubahan tersebut terjadi dalam kecepatan eksponensial yang akan berdampak terhadap ekonomi, industri, pemerintahan, dan politik. Pada era ini semakin terlihat wujud dunia yang telah menjadi kampung global.

Industri 4.0 adalah sebuah istilah yang diciptakan pertama kali di Jerman pada tahun 2011 yang ditandai dengan revolusi digital. Industri ini merupakan suatu proses industri yang terhubung secara digital yang mencakup berbagai jenis teknologi, mulai dari 3D *printing* hingga robotik yang diyakini mampu meningkatkan produktivitas. Sebelum ini telah terjadi tiga revolusi industri yang ditandai dengan:

- a. Ditemukannya mesin uap dan kereta api tahun 1750-1930;
- b. Penemuan listrik, alat komunikasi, kimia, dan minyak tahun 1870-1900;
- c. Penemuan komputer, internet, dan telepon genggam tahun 1960-sekarang.

2. Isu Pengembangan Perangkat Lunak dan Gim

Sebagai pengembang perangkat lunak dan gim, penting bagi kita untuk mengetahui berbagai isu yang berkembang seperti dampak positif negatif pengembangan perangkat lunak, *internet of thing*, *cloud computing*, *information security*, *personal branding* dan hak kekayaan intelektual.



a. Dampak Positif dan Negatif Perangkat Lunak dan Gim

1) Pengaruh Positif dan Negatif pada perangkat lunak

Pengembangan perangkat lunak diarahkan pada realisasi sistem aplikasi yang mampu menunjang proses transaksi ekonomi yang cepat dan aman, serta pengambilan keputusan yang benar dan cepat. Harga yang terjangkau dan daya saing pada tingkat internasional merupakan salah satu kriteria yang dipersyaratkan, khususnya mendukung kebijakan substitusi impor. Perangkat lunak sistem operasi dengan kehandalan tinggi dan kebutuhan sumber daya memori maupun prosesor yang minimal serta fleksibel terhadap perangkat keras maupun program aplikasi yang baru, merupakan prioritas yang harus dikembangkan. Program aplikasi juga perlu dikembangkan, terutama yang terkait dengan sektor perekonomian, industri, pendidikan, maupun pemerintahan.

Dalam mempercepat pengembangan dan pendayagunaan perangkat lunak, perlu pula ditinjau implementasi konsep open source. Penerapan konsep open source ini diharapkan mampu menggalakkan industri perangkat lunak dengan partisipasi seluruh lapisan masyarakat tanpa melakukan pelanggaran hak cipta.

2) Pengaruh Positif dan Negatif pada Gim

Menurut Anhar (2010:27) Beberapa dampak positif bermain gim adalah sebagai berikut:

- Membantu perkembangan koordinasi tangan, mata, motorik, dan kemampuan spasial



- Meningkatkan kemampuan membuat analisa, keputusan yang cepat dan berpikir secara mendalam

Sedangkan Menurut Anhar (2010: 33), dampak negatif dari gim timbul karena 89% dari gim mengandung beberapa konten kekerasan. Menurut pendapat Darma (2011:67) dampak negatif *game online* pada siswa atau anak - anak adalah sebagai berikut:

- Anak lebih banyak menghabiskan waktu bermain gim pada jam di luar sekolah
- Konsentrasi belajar terganggu karena pikiran siswa cenderung mengarah pada permainan yang ada di dalam gim
- Tertidur di sekolah
- Sering melalaikan tugas dan tanggung jawab sebagai siswa
- Nilai di sekolah menurun

b. *Internet of Thing* (IoT)

IoT merupakan segala aktifitas yang pelakunya saling berinteraksi dan dilakukan dengan memanfaatkan internet. Dalam penggunaannya, *Internet of Thing* banyak ditemui dalam berbagai aktifitas, contohnya : banyaknya transportasi *online*, *e-commerce*, pemesanan tiket secara *online*, *live streaming*, *e-learning*, dan lain-lain bahkan sampai alat-alat untuk membantu dibidang tertentu seperti *remote temperature sensor*, *GPS tracking*, dan sebagainya yang menggunakan internet atau jaringan sebagai media untuk melakukannya.



c. *Cloud Computing*

Cloud Computing merupakan teknologi yang memberikan pelayanan secara luas dengan akses internet dimanapun berada, media penyimpanan *cloud computing* berada di internet.

Cloud computing menyimpan semua data di *server* yang tidak tau dimana letak server tersebut. Ada 3 layanan *cloud computing* yang dapat digunakan yaitu:

1) *Software as a Service* (SaaS)

Layanan *cloud computing* dimana pengguna dapat menggunakan aplikasi atau perangkat lunak (*software*) yang disediakan oleh *cloud provider* (penyedia jasa *cloud computing*). Contoh dari layanan SaaS adalah layanan produktivitas seperti *Office365*, *GoogleDocs*, dan *Adobe Creative Cloud*. Layanan email seperti *Gmail*, *YahooMail*, dan *LiveMail*.

2) *Platform as a Service* (PaaS)

Layanan yang difasilitasi oleh *cloud provider* untuk menyediakan *platform* bagi pengembangan aplikasi-aplikasi. Pengguna dapat berfokus pada pengembangan aplikasi tanpa perlu mengkhawatirkan *platform* aplikasi tersebut. Contoh dari layanan PaaS adalah *Amazon Web Service*, *Windows Azure*, dan *GoogleApp Engine*.

3) *Infrastructure as a Service* (IaaS)

Pada layanan ini pengguna dapat menyewa infrastruktur yang disediakan oleh *cloud provider* (unit komputasi, *storage*, *memory*, *network*, dan sebagainya). Pada layanan ini seluruhnya pengguna yang menentukan



perangkat perangkat untuk *cloud computing* yang akan digunakan, jika sistem *virtual* di *cloud* tersebut menggunakan *source* yang besar, pengguna dapat menambahkan RAM sesuai kebutuhan. Contoh dari layanan IAAS adalah *Amazon EC2*, *Rackspace Cloud*, dan *Windows Azure*.

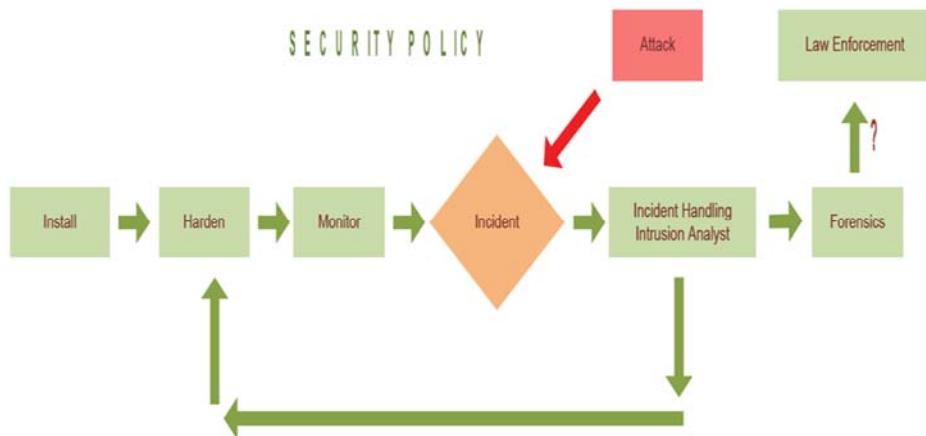
d. *Information Security*

Saat ini Indonesia tengah dalam keadaan mendesak *cyber-security* atau keamanan dunia maya karena melihat kenyataan bahwa tingkat kejahatan di dunia maya atau *cyber crime* di Indonesia sudah mencapai tahap memprihatinkan. Namun berbeda dengan penanganan kejahatan lainnya, *cyber-security* membutuhkan pemikiran yang komprehensif untuk menanggannya.

Cyber-security adalah kumpulan alat, kebijakan, konsep keamanan, perlindungan keamanan, pedoman, pendekatan manajemen risiko, tindakan, pelatihan, praktik terbaik, jaminan dan teknologi yang dapat digunakan untuk melindungi lingkungan *cyber* dan organisasi dan aset pengguna. Organisasi dan aset pengguna dalam *cyber-security* termasuk perangkat yang terhubung komputasi, personil, infrastruktur, aplikasi, layanan, sistem telekomunikasi dan totalitas informasi yang dikirimkan dan/atau disimpan dalam lingkungan maya.



Berikut adalah diagram ruang lingkup *cyber security*:



Gambar 1.104 *Cyber Security*

e. *Personal Branding*

Branding atau pelabelan lebih akrab dilakukan di dunia pemasaran. Pelabelan dilakukan karena begitu banyak produk sejenis yang beredar, sehingga diperlukan pembeda bagi produk yang satu dengan produk yang lain. Hal itu bertujuan untuk membuatnya lebih menonjol dibanding produk sejenis lainnya. Prinsip dasar itulah yang kemudian diterapkan dalam konteks *personal branding*.

Personal branding dapat mengatur persepsi seseorang terhadap orang lain, dengan menceritakan pengalaman kepada orang lain secara natural sehingga orang lain berpikir bahwa persepsi tersebut dibangun dengan sendirinya (Montoya, 2002).

Personal branding juga dapat dikatakan sebagai proses di mana seseorang dipandang sebagai sebuah *brand* (merek) oleh *target market* (Lair, Sullivan, dan Cheney, 2005:35). Dapat disimpulkan bahwa *personal branding* adalah suatu proses pembentukan persepsi masyarakat terhadap aspek-aspek yang dimiliki



seseorang, diantaranya kepribadian, kemampuan, atau nilai-nilai, dan bagaimana stimulus-stimulus ini menimbulkan persepsi positif dari masyarakat yang pada akhirnya dapat digunakan sebagai alat pemasaran (McNally & Speak, 2002).

f. Hak Kekayaan Intelektual (HKI)

Hak Kekayaan Intelektual (HKI) adalah hak kekayaan yang timbul atau lahir karena kemampuan intelektual manusia. Karya-karya intelektual bisa dari berbagai bidang, seperti ilmu pengetahuan, seni, sastra, atau teknologi. Karya-karya tersebut biasanya dilahirkan dengan pengorbanan tenaga, waktu dan biaya, sehingga menghasilkan nilai-nilai tertentu.

Ada beberapa macam HKI, namun yang penting bagi para pengusaha pembuat gim adalah:

1) Hak Cipta

Hak Cipta diperlukan untuk menghindari sengketa pemilik hak cipta. Sengketa bisa saja terjadi apabila suatu aplikasi dinilai telah sukses, yaitu telah menghasilkan aliran keuntungan. Mungkin ada karyawan perusahaan yang keluar dan membuat perusahaan sendiri, dengan menjiplak sebuah karya aplikasi. Hal semacam inilah yang bisa dihindari dengan pendaftaran dan pencatatan Hak Cipta.

Pencatatan hak cipta dilakukan di Direktorat Jenderal Kekayaan Intelektual Kementerian Hukum Dan Hak Asasi Manusia, sesuai ketentuan UU No. 28/2014 Tentang Hak Cipta. Yang diakui sebagai pemilik hak cipta adalah



pendeklarator pertama dari sebuah karya, termasuk karya aplikasi.

Pengurusan pencatatan hak cipta ini bisa dilakukan secara Online dan bisa diakses melalui : <https://e-hakcipta.dgip.go.id>. Setelah semua persyaratan dikirimkan, pengirim akan mendapatkan *email* balasan resmi yang mencantumkan *username* dan *password* yang bisa digunakan untuk mengakses e-hak cipta.

2) Hak Merek

Pengertian merek menurut UU No. 20 tahun 2016 adalah tanda yang dapat ditampilkan secara garis berupa gambar, logo, nama, kata, huruf, angka, susunan warna, dalam bentuk 2 (dua) dimensi dan/atau 3 (tiga) dimensi, suara, hologram, atau kombinasi dari 2 (dua) atau lebih unsur tersebut untuk membedakan barang dan/atau jasa yang diproduksi oleh orang atau badan hukum dalam kegiatan perdagangan barang dan/atau jasa. Oleh karena itu merek perlu didaftarkan agar tidak terjadi sengketa antar merek dan melindungi pemiliknya.

Merek yang sudah terdaftar mendapat perlindungan hukum untuk jangka waktu 10 tahun dan dapat diperpanjang untuk jangka waktu yang sama 10 tahun. Perlindungan merek terdaftar selama 10 tahun tersebut berlaku surut sejak tanggal penerimaan permohonan merek yang bersangkutan.

Pengurusan pendaftaran merek bisa diawali dengan pengecekan kemiripan merek yang didaftarkan dengan



merek-merek yang sudah terdaftar dengan cara penelusuran merek di fasilitas *online* data merek Indonesia yang disediakan oleh Kementerian Hukum dan Hak Asasi Manusia Direktorat Jenderal Hak Kekayaan Intelektual pada <http://penelusuran-merek.dgip.go.id/>.

Diharapkan dengan sudah daftarkannya merek produk, maka dikemudian hari tidak ada masalah sengketa terkait dengan merek.

3) Rahasia Dagang

Rahasia dagang di Indonesia diatur dalam Undang - Undang Nomor 30 Tahun 2000 tentang Rahasia Dagang. Beberapa hal pokok yang terdapat dalam Undang-Undang Rahasia Dagang yang perlu diketahui adalah sebagai berikut:

- a) Hak Rahasia Dagang adalah informasi yang tidak diketahui oleh umum di bidang teknologi dan/atau usaha, mempunyai nilai ekonomi karena berguna dalam kegiatan usaha, dan dijaga kerahasiaannya oleh pemilik Rahasia Dagang.
- b) Objek pengaturan Rahasia Dagang meliputi metode produksi, metode pengolahan, metode penjualan atau informasi lain di bidang teknologi dan/atau usaha yang memiliki nilai ekonomi dan tidak diketahui oleh masyarakat umum. Dengan demikian, lingkup perlindungan rahasia dagang meliputi:
 - (1) Metode produksi
 - (2) Metode pengolahan



- (3) Metode penjualan
- (4) Informasi lain di bidang teknologi dan/atau usaha yang memiliki nilai ekonomis dan tidak diketahui masyarakat secara umum
- c) Beberapa faktor yang dapat digunakan untuk menilai informasi yang dimiliki dilindungi sebagai rahasia dagang, antara lain adalah:
 - (1) Sejauh mana informasi tersebut diketahui oleh kalangan di luar perusahaannya;
 - (2) Sejauh mana informasi tersebut diketahui oleh para karyawan di dalam perusahaannya;
 - (3) Sejauh mana upaya-upaya yang dilakukan untuk melindungi kerahasiaan informasinya;
 - (4) Nilai dari informasi tersebut bagi dirinya dan bagi pesaingnya;
 - (5) Derajat kesulitan atau kemudahan untuk mendapatkan atau menduplikasikan informasi yang sama oleh pihak lain.
- d) Perlindungan rahasia dagang timbul berdasarkan Undang-Undang Rahasia Dagang.
- e) Rahasia dagang dapat beralih atau dialihkan dengan:
 - (1) Pewarisan;
 - (2) Hibah;
 - (3) Wasiat;
 - (4) Perjanjian tertulis; atau
 - (5) Sebab-sebab lain yang dibenarkan oleh peraturan perundang-undangan.



f) Jangka Waktu Perlindungan

Dalam perlindungan rahasia dagang, tidak ada ketentuan yang membatasi tentang jangka waktu berlakunya perlindungan rahasia dagang, yaitu selama pemiliknya tetap merahasiakan dan melakukan usaha-usaha untuk melindungi kerahasiaannya maka selama itu pula berlaku perlindungan hukum.

g) Pelanggaran dan Sanksi

Seseorang dianggap melanggar rahasia dagang orang lain apabila ia memperoleh atau menguasai rahasia dagang tersebut dengan cara-cara yang bertentangan dengan peraturan perundang-undangan yang berlaku, seperti:

- Pencurian;
- Penyadapan;
- Spionase industri;
- Membujuk untuk mengungkapkan atau membocorkan rahasia dagang melalui penyuapan, paksaan dll;
- Dengan sengaja mengungkapkan atau mengingkari kesepakatan atau kewajiban yang tertulis untuk menjaga rahasia dagang yang bersangkutan.

Selanjutnya ketentuan pidananya ialah "Barangsiapa dengan sengaja dan tanpa hak menggunakan Rahasia Dagang pihak lain, atau melakukan perbuatan sebagaimana dimaksud dalam Pasal 13, atau Pasal 14



dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau denda paling banyak Rp. 300.000.000,- (tiga ratus juta rupiah). Tindak pidana dimaksud termasuk delik aduan.



REFLEKSI III

Setelah mempelajari materi ini, banyak manfaat yang akan kamu rasakan dalam meningkatkan pengetahuan terhadap isu yang berkembang pada industry 4.0. Isu pengembangan perangkat lunak dan gim bidang IoT, Cloud Computing, Information Security, Personal Branding dan Hak Kekayaan Intelektual dapat membekali pengetahuan kalian dalam mempersiapkan produk yang berkualitas yang sesuai dengan perkembangan jaman.



ASESMEN III

I. Tugas Mandiri

Buatlah sebuah perencanaan produk bidang pengembangan perangkat lunak atau gim yang memperhatikan berbagai isu seperti IoT, Cloud Computing, Information Security, Personal Branding dan Hak Kekayaan Intelektual!

II. Latihan Soal

1. Jelaskan apa yang dimaksud dengan industry 4.0!
2. Sebut dan jelaskan dampak positif dan negative pada pengembangan perangkat lunak dan gim!
3. Jelaskan apa yang dimaksud dengan internet of thing, cloud computing dan cyber security!



4. Jelaskan apa yang dimaksud dengan Personal branding! Dan berikan contoh personal branding bidang perangkat lunak dan gim!
5. Jelaskan apa yang dimaksud dengan HKI!

D. Profil dan profesi di bidang pengembangan perangkat lunak dan gim

1. Profil profesi pengembang perangkat lunak dan gim

Pada dasarnya profesi yang berkembang dalam bidang pengembangan perangkat lunak dan gim mempunyai kesamaan yaitu ada beberapa SDM yang mengatur bidang *management*, perencanaan, desain, pemrograman dan testing, berikut adalah profil profesi pengembang perangkat lunak dan gim:

- a) Managemen Perusahaan *Startup*

Perusahaan rintisan (*startup*) adalah istilah yang merujuk pada semua perusahaan yang belum lama beroperasi (Wikipedia, 2019). Perusahaan-perusahaan ini sebagian besar merupakan perusahaan yang baru didirikan dan berada dalam fase pengembangan dan penelitian untuk menemukan pasar yang tepat. Namun, bisnis *startup* ini lebih identik dengan perusahaan yang berbau teknologi, web, internet dan berhubungan dengan ranah tersebut (Techinasia, 2015). Berikut adalah profil profesi pada perusahaan *startup*:

- 1) *Chief Executive Officer* (CEO)

CEO merupakan posisi dengan jabatan tertinggi dalam sebuah *startup*. CEO adalah pemimpin *startup* dan bertindak sebagai perwakilan perusahaan. Tanggung jawab CEO meliputi:



- a) Membuat dan mengkomunikasikan visi, misi, tujuan jangka panjang, dan tujuan jangka pendek perusahaan
- b) Menyusun kebijakan-kebijakan strategis untuk mencapai visi dan misi perusahaan
- c) Melakukan pengawasan terhadap seluruh aktivitas yang ada di dalam perusahaan
- d) Bertanggung jawab dalam menjalankan perusahaan termasuk kerugian yang dihadapi perusahaan maupun keuntungan perusahaan
- e) Mengambil keputusan yang berkaitan dengan keseluruhan bisnis
- f) Menjalin hubungan yang baik dengan investor dan seluruh pemangku kepentingan
- g) Memotivasi anggota tim.

2) *Chief Technical Officer (CTO)*

CTO merupakan orang yang bertanggung jawab pada kualitas akhir, karena ia yang mengelola tim *engineer* selama prosesnya. CTO berperan penting dalam sebuah perusahaan teknologi karena bertindak sebagai ahli teknologi atau *programmer*. Tanggung jawab CTO meliputi:

- a) Membuat aplikasi CusTrip
- b) Mengelola pengembangan produk dan pembaruan (*update*) terhadap sistem secara berkala
- c) Memahami perkembangan teknologi saat ini dan pengaplikasiannya
- d) Melakukan pengawasan terhadap segala hal yang berhubungan dengan teknologi informasi



e) Memberikan jawaban terhadap *review* pengguna di *PlayStore* dan *AppStore*

3) *Chief Finance Officer* (CFO)

CFO merupakan orang yang bertanggung jawab atas pembukuan keuangan, penggalangan dana, dan penganggaran. Pada tahap awal sebuah *startup*, ia juga bertanggung jawab pada penggajian dan sumber daya manusia. Tanggung jawab CFO meliputi:

- a) Membuat laporan keuangan, termasuk di dalamnya penjualan, pengeluaran, dan keuntungan perusahaan dalam satu hari
- b) Membuat pembukuan laporan keuangan selama satu bulan
- c) Mengatur pembiayaan dan anggaran
- d) Menyampaikan pembukuan kepada CEO setiap bulan
- e) Membentuk strategi penggalangan dana
- f) Membuat keputusan yang berhubungan dengan sumber daya manusia termasuk penggajian karyawan

4) *Chief Marketing Officer* (CMO)

CMO adalah orang yang bertanggung jawab mengembangkan strategi pemasaran perusahaan. CMO meliputi bagian riset pasar, hubungan masyarakat, dan pencitraan. Tanggung jawab CMO meliputi:

- a) Melakukan pemasaran aplikasi kepada *locals* dan *travelers*
- b) Meramalkan tren pasar.
- c) Mengembangkan strategi pemasaran.



- d) Melakukan pengawasan terhadap kegiatan pemasaran yang telah dilakukan.
- e) Memahami kebutuhan konsumen dan mengembangkan nilai tambah yang menarik.
- f) Mengelola semua media sosial (Facebook, Instagram, Twitter, dan lain-lain).

5) *Chief Operating Officer (COO)*

COO adalah orang yang bertanggung jawab atas urusan organisasi perusahaan seperti operasional kantor, karyawan, sampai bisnis. Tanggung jawab COO meliputi:

- 1) Mengatur kegiatan operasional perusahaan
- 2) Menjadi penghubung antara karyawan lain dengan CEO
- 3) Menganalisis kebutuhan untuk membangun suatu sistem
- 4) Melakukan perekrutan dan seleksi karyawan
- 5) Membantu menjadi *customer service*

b. Profesi Tahap Perencanaan pada Perangkat Lunak dan Gim

1) Sistem Analis

Sistem Analis adalah personil tenaga ahli yang memiliki kemampuan untuk melakukan analisa terhadap sebuah sistem, memilih alternative pemecahan masalah dan menyelesaikan masalah tersebut dengan menggunakan komputer.

Tenaga Ahli Sistem Analis memiliki peran penting dalam pembangunan Sistem Informasi, memberikan ide dan saran untuk mendukung dan meningkatkan proses bisnis, membantu merancang proses bisnis baru yang didukung oleh Teknologi Informasi, merancang sistem informasi baru dan



menjamin bahwa semua sistem informasi dapat dilakukan *maintenance/Perawatan*.

Berikut ini adalah tugas dan tanggung jawab Sistem Analis dalam projek pengembangan perangkat lunak:

- a) Menterjemahkan isi kerangka acuan kerja ke dalam konsep metodologi kerja.
- b) Bertanggung jawab dalam hal-hal teknis yang berkaitan dengan pekerjaan secara keseluruhan seperti pelaporan, pelaksanaan *survey*, pengembangan aplikasi, instalasi dan pelatihan.
- c) Melakukan perancangan sistem dan desain aplikasi hasil dari tahapan *survey* kebutuhan (*requirement analysis*)

2) *Data analyst*

Data analyst merupakan suatu pekerjaan yang berhubungan dengan berbagai jenis data. Secara umum, *data analyst* akan bertanggung jawab untuk mengelola data, mengembangkan laporan, dan mengatasi masalah terkait data. Selain itu, *data analyst* juga berperan untuk menciptakan visualisasi data agar dapat dimengerti oleh orang awam.

3) *Game Designer*

Game Designer termasuk profesi dalam *pre-production stage* (perencanaan), yang berarti *Game Designer* bekerja sebelum *Artist* dan *Programmer*. *Game Designer* adalah seseorang yang bertugas untuk membuat alur cerita atau *blueprint* dari sebuah gim yang akan dibuat. Hal ini bertujuan



untuk memperkuat daya analisis dan logika berpikir para pemainnya untuk pengambilan keputusan sebaik mungkin. Sehingga pekerjaan *artist* dan *programmer* akan lebih terarah dalam mengembangkan sebuah gim. Berikut ini beberapa tugas yang perlu dikerjakan oleh *Game Designer*:

- a) Membuat *Mock-up game* yang merupakan gambaran kasar/sketsa yang menggambarkan gim seperti apa yang ingin dibuat
- b) Membuat *Storyboard*, yaitu gambar berurutan yang menunjukkan level dari tiap gim atau menggambarkan adegan-adegan tiap kejadian yang berbeda-beda
- c) Menentukan *genre*, menjelaskan jenis apakah yang akan digunakan pada gim tersebut
- d) *Gameplay*, mendeskripsikan apa yang akan *player* lakukan di dalam gim
- e) *Feature*, bagian ini menampilkan kumpulan nilai jual utama dari gim tersebut, yang dapat membantu mengidentifikasi kemampuan dari gim tersebut
- f) *Setting*, mendeskripsikan dunia gimnya. Termasuk konsep *art* yang akan digunakan. Jika gim ini kuat di *story*-nya, maka tambahkan fitur yang mempengaruhi jalan cerita
- g) *Target audience*, di bagian ini akan dijelaskan mengenai target utama pengguna gim
- h) *Hardware Platform*, di bagian ini akan dijelaskan untuk *device* apa gim ini akan dibuat. Apakah untuk PC, *Console*, *Mobile*, *Handheld*, dan lain-lain.



c. Profesi Tahap Implementasi pada Perangkat Lunak dan Gim

1) *UI Designer*

UI Designer merupakan sebutan untuk orang yang mendesain *interface* untuk perangkat lunak komputer, ponsel pintar, dan lainnya. *UI Designer* bertugas untuk mendesain tampilan secara menarik baik dari sisi bentuk, warna, juga tulisan

2) *UX Designer*

UX Designer merupakan jenis pekerjaan yang berhubungan dengan bagaimana meningkatkan kepuasan pengguna aplikasi maupun pengunjung situs yang dilihat dari nilai guna, manfaat, juga kesenangan yang didapatkan oleh pengguna. *UX Designer* bertanggung jawab untuk kepuasaan pengguna (*user*) dan pengalaman yang lebih baik.

3) *Game Artist*

Menurut Dwiperdana, dkk (2012, h. 10), *game artist* berperan membuat semua aspek visual dari gim, mulai dari karakter, *background*, sampai menu dan antarmuka/UI. *Game artist* bertanggung jawab agar gim terlihat menarik. Terdapat dua macam bentuk visualisasi yang sering di pakai dalam pengembangan sebuah gim yaitu visualisasi bentuk 2D dan 3D.

4) *Front End Developer*

Memiliki tugas untuk merancang antarmuka/tampilan sebuah *website*. Tampilan *sidebar*, *header*, menu navigasi, dan *footer* adalah hasil pekerjaan dari *front end*.



5) *Back End Developer*

Back End Developer bertugas memberikan aksi untuk melakukan sesuatu dari sebuah web. Mengisi formulir dan juga menekan tombol kirim merupakan hasil pekerjaan dari *back end*.

6) *Full Stack Developer*

Full Stack Developer memiliki kemampuan *front end* dan juga *back end*

7) *Software Engineer*

Seseorang yang berperan dalam penerapan prinsip - prinsip teknik pengembangan perangkat lunak. Melakukan analisa, membuat rekayasa, menyusun spesifikasi, mengimplementasikan dan memvalidasi suatu rancangan sistem perangkat lunak untuk menjawab suatu permasalahan.

d. Profesi Tahap Pengujian pada Perangkat Lunak dan Gim

1) *Quality Assurance dan Tester*

Quality assurance atau *tester* adalah orang yang bertugas sebagai penjamin mutu, melakukan test terhadap perangkat lunak atau gim yang dibuat, Melakukan uji coba terhadap produk yang dibuat oleh programmer, mencari bug dan memberikan penilaian terhadap produk perangkat lunak dan gim.

2. Peluang Usaha

Saat ini industri perangkat lunak dan gim di Indonesia memang sedang berkembang. Akan tetapi jika dibandingkan dengan jumlah pengguna produk dan pengembang perangkat lunak dan gim



(*Software and game developer*) lokal ternyata cenderung mengalami pertumbuhan yang lambat. Seharusnya jumlah pengguna yang banyak bisa disertai dengan jumlah pengembang sehingga nantinya akan terjadi keseimbangan antara jumlah produk dengan kebutuhan pasar.

a. Pengertian Peluang Usaha

Peluang usaha adalah kesempatan yang datang pada waktu tertentu yang harus diambil oleh seorang wirausahawan untuk menciptakan suatu usaha dengan keberanian mengambil risiko, sehingga mendapatkan hasil yang diinginkan. Banyak sekali peluang usaha yang berada di sekitar kita, peluang usaha itu bisa datang dari orang terdekat maupun dari apa yang kita baca atau dengar. Bahkan peluang usaha bisa kita ciptakan sendiri dengan kreativitas dan inovasi yang kita lakukan dari usaha-usaha sebelumnya yang sudah ada. Selain itu, peluang usaha juga akan mudah didapatkan apabila kita memiliki keterampilan. Misalnya, bagi peserta didik jurusan pengembangan perangkat lunak dan gim dapat membuka usaha pembuatan perangkat lunak dan gim (*Software and Game Development*).

Namun yang lebih penting sebagai seorang yang akan memulai sebuah bisnis, kemampuan menangkap peluang usaha yang ada harus disertai keberanian, kreativitas, kemampuan berinovasi dan kesungguhan dalam menjalankan peluang usaha serta mau mengambil risiko. Dengan demikian, peluang usaha tersebut dapat menjadi sebuah usaha yang benar-benar matang dan kuat dalam bersaing dengan usaha-usaha lainnya. Dengan



begitu keuntungan atau hasil yang didapatkan akan sesuai dengan yang diharapkan.

Jadi peluang usaha merupakan suatu kesempatan yang harus benar-benar dimanfaatkan oleh seseorang yang ingin memulai suatu usaha dengan melihat faktor internal dan eksternal. Faktor internal atau diri sendiri berupa, keterampilan, minat, dan bakat yang dimiliki, sedangkan faktor eksternal, yaitu kondisi lingkungan sekitar, kondisi, kebutuhan masyarakat, dan sesuatu yang sedang tren saat ini. Sehingga usaha yang dilakukan berdasarkan peluang usaha yang ada dapat memperbaiki kehidupan orang yang menjalankannya.

b. Identifikasi Peluang Usaha

Perkembangan di bidang perdagangan sangat pesat saat ini, karena didukung oleh perkembangan ilmu pengetahuan dan teknologi khususnya internet, memberikan banyak peluang usaha yang dapat dikembangkan. Namun ada beberapa unsur yang harus diperhatikan dalam menangkap peluang usaha yang ada di sekitar kita, di antaranya sebagai berikut.

- 1) Lihat keterampilan yang dimiliki dan cocokkan dengan karakter usaha yang akan diambil. Dengan begitu, akan lebih mudah bagi kita untuk mengembangkan usaha tersebut sesuai dengan yang kita inginkan, selain itu kreativitas dan inovasi sangat diperlukan dalam mengembangkan sebuah usaha.
- 2) Usaha yang kita lakukan adalah usaha yang kita sukai. Rasa suka kita terhadap usaha yang kita geluti adalah syarat mutlak. Karena pada dasarnya, jika kita menyukai apa yang



kita kerjakan, maka kita akan lebih giat, tekun, dan pantang menyerah dalam menjalankannya. Memulai usaha dari hobi dapat berpotensi lebih besar usaha tersebut akan berhasil, namun diperlukan dukungan analisa dari aspek yang lainnya.

- 3) Mengukur kemampuan diri dalam menjalankan usaha. Hal ini diperlukan agar usaha yang kita jalankan lebih matang, karena kita akan melakukan analisa dan riset yang pada akhirnya dibandingkan dengan kemampuan yang dimiliki.
- 4) Kemampuan melihat dan memanfaatkan peluang usaha yang ada di sekitarnya, seperti, membuat inovasi baru dari usaha-usaha yang telah ada maupun menemukan ide usaha baru yang belum ada sebelumnya, disesuaikan dengan kebutuhan sekitar, memanfaatkan koneksi dan relasi untuk promosi maupun pengembangan produk, mengevaluasi produk barang dan jasa yang telah dibuat, dan mengikuti tren.

Mengidentifikasi peluang usaha dapat juga dilakukan dengan cara-cara di antaranya, yaitu, dengan berpikir positif terhadap ide-ide baru yang muncul di dunia usaha, menerima saran-saran dari berbagai pelaku usaha, banyak membaca untuk mencari referensi perkembangan usaha saat ini, dan memberlakukan konsep ATM (amati, tiru, dan modifikasi) dengan memanfaatkan kreativitas yang dimiliki.

Selain itu beberapa hal yang harus diamati dalam membantu identifikasi peluang usaha, di antaranya sebagai berikut:

- a) Segmentasi pasar
- b) Sumber daya manusia
- c) Posisi produk



- d) Keuangan
- e) Tanggung jawab sosial
- f) Pengembangan usaha

c. Peluang Usaha Bidang Pengembangan Perangkat Lunak dan Gim

1) Peluang Usaha Bidang Perangkat Lunak

Adapun peluang usaha bidang perangkat lunak dapat dikategorikan menjadi:

- Produk Jadi
- Penawaran pesanan aplikasi sesuai kebutuhan calon konsumen dengan cara lelang atau tender

a) Jasa *maintance* dan *service*

Setelah menentukan peluang usaha yang ada pada bidang perangkat lunak, kita dapat menentukan model bisnis yang cocok pada bidang perangkat lunak, yaitu:

(1) *Business to Business* (B2B)

Transaksi bisnis antara pelaku bisnis dengan pelaku bisnis lainnya. Dapat berupa kesepakatan spesifik yang mendukung kelancaran bisnis. Contoh: aplikasi *software* kasir, aplikasi sistem informasi sekolah, aplikasi sistem informasi perpajakan, aplikasi sistem manajemen parkir, dan lain sebagainya.

Model bisnis ini mendapatkan uang (*monetize*) dengan cara menjual aplikasi, menjual jasa dan *maintance* hingga valuasi perusahaan (perusahaan dengan portfolio bagus, akan dapat menjual sahamnya kepada investor).

(a) Kelebihan



- Market jelas dan spesifik
- Segmentasi lebih sempit
- Mendapat penghasilan dari *maintance* dan *service*
- Lebih pasti dalam menghasilkan uang

(b) Kekurangan

- Kreativitas mengikuti keinginan konsumen
- Modal lebih besar
- Harus memiliki portfolio yang jelas untuk mendapatkan pengguna
- *Deadline* ketat

(2) *Business to Consumer* (B2C)

Aktivitas yang dilakukan produsen kepada konsumen secara langsung. Contoh: tiket.com, MatahariMall.com yang memfasilitasi konsumen untuk lebih mudah mendapatkan barang yang mereka butuhkan. Model bisnis ini mendapatkan uang (*monetize*) dengan cara menjual aplikasi yang dibeli oleh pengguna, iklan, *in app-purchase*(fitur tertentu berbayar), *subscription* dan promosi.

(a) Kelebihan

- Market lebih spesifik, dapat menasarkan pada unit usaha yang sudah ada.
- Alternatif monetize beragam seperti iklan, in apps dll.

(b) Kekurangan

- Kompetitor banyak



- Tidak ada kepastian monetize dapat menghasilkan uang dalam jumlah besar.

(3) *Consumer to Consumer* (C2C)

Aktivitas bisnis (penjualan) yang dilakukan oleh individu (konsumen) kepada individu (konsumen) lainnya.

Contoh: OLX, Tokopedia, dan Bukalapak. Model bisnis ini mendapatkan uang (*monetize*) dengan cara menjual aplikasi yang dibeli oleh pengguna, iklan, *in app purchase*(fitur tertentu berbayar), *subscription* dan promosi.

(a) Kelebihan

- Market luas
- Modal yang diperlukan terukur, sesuai dengan kebutuhan
- Lebih fleksibel dalam kreativitas
- Alternatif monetize banyak, seperti iklan in app, dll

(b) Kekurangan

- Segmentasi terlalu luas
- Kompetitor lebih banyak
- Tidak ada kepastian monetize yang menghasilkan uang dalam jumlah besar.

2) Peluang Usaha Bidang Gim

Indonesia memiliki pasar gim yang menjanjikan. Dari 66 juta orang yang memiliki akses internet, 42,8 juta di antaranya mengakses gim.



Industri gim sudah muncul sejak lama. Mulai era *console game* sederhana, *handheld game*, *PC console game*, *game Online*, *web browser game*, hingga *Mobile game*. Seiring dengan perkembangan ponsel dan internet, gim juga semakin berkembang dan makin beragam juga *genre*-nya, seperti *role playing game*, *strategy game*, *shooter game*, atau *game simulation*.

Setiap *genre* memiliki pangsa pasarnya masing-masing. Semua gim yang ada saat ini tidak muncul begitu saja, namun melalui proses yang cukup panjang.

Berdasarkan analisis pasar dan peluang usaha yang ada bidang gim dapat dipasarkan melalui:

a) Menjual Gim

Konsumen harus membeli gim tersebut agar dapat bisa dimainkan. Segmentasi gim berbayar ini harus berbeda dengan gim yang bisa diunduh gratis.

b) Memasang Iklan

Uang bisa dihasilkan dari iklan yang ditampilkan oleh rekanan gim. Jika ada pemain yang ingin menghilangkan iklan pada gim, maka dia harus membayar sejumlah uang.

c) Menjual Konten dan Karakter

Beberapa gim dibuat dengan level kesulitan berbeda. Jika ada pemain yang ingin naik tingkat, maka dia harus membayar produsen gim juga bisa memperoleh uang dari *ringtone*, *wallpaper*, gim versi premium, atau meningkatkan kemampuan karakter dalam gim tersebut.



Lalu produsen bisa menjual set paket *upgrade* karakter dari tokoh yang dimainkan.

d) Menjual Souvenir

Jika gim sudah dikenal, maka bisa dibuat turunan produknya untuk dijual contohnya kaos, boneka karakter, topi atau bahkan menjual ide gim tersebut kepada produsen film.



E. REFLEKSI IV

Setelah mempelajari materi ini, banyak manfaat yang akan kamu rasakan, kamu lebih dapat mendalami profesi dan peluang usaha apa saja yang ada dalam pengembangan perangkat lunak dan gim, sehingga kamu dapat lebih jelas menggapai cita–cita yang kamu inginkan.



ASESMEN IV

A. Tugas Mandiri

Buatlah sebuah slide perentasi yang berisikan gambaran cita – cita kalian atau ide peluang usaha dalam bidang pengembangan perangkat lunak dan gim!

B. Latihan Soal

1. Sebutkan berbagai macam jenis profesi pada pengembang perangkat lunak!
2. Sebutkan berbagai macam jenis profesi pada pengembang gim
3. Apa yang dimaksud dengan peluang usaha?



4. Sebut dan jelaskan berbagai macam jenis peluang usaha dalam bidang perangkat lunak!
5. Sebut dan jelaskan berbagai macam jenis peluang usaha dalam bidang gim!

RANGKUMAN

1. Perangkat lunak (*software*) adalah kumpulan Instruksi atau program komputer (*syntaks/code* yang ditulis dalam bahasa pemrograman) yang ketika dijalankan menyediakan fungsi dan performa yang diinginkan.
2. Gim adalah suatu kegiatan dengan beberapa *rules* dan tujuan.
3. Perangkat lunak berevolusi selama empat era, yaitu: Era Pioneer, Era Stabil, Era Mikro dan Era Stabil.
4. Era Pre-History pada sejarah perkembangan gim, Fusajiro Yamauchi mendirikan Koppai Nitendo sebagai sebuah perusahaan kartu permainan di Jepang. Kemudian mereka menghilangkan kata "Koppai" dari nama dan kini hanya di kenal sebagai "Nitendo".
5. Perangkat lunak aplikasi dapat digolongkan menjadi beberapa kelas, yaitu: perangkat lunak enterprise, infrastruktur perusahaan, informasi kerja, media dan hiburan, pendidikan, pengembangan media dan rekayasa produk.
6. Jenis – jenis gim menurut Henry dibagi menjadi 20, yaitu: *Maze Game, Board Game, Card Game, Battle Card Game, Quiz Game, Puzzle Game, Shoot Them Up, Side Scroller Game, Fighting Game, Racing Game, Turn-Based Strategy, Real-Time Strategy Game, Simulation Video Game, First Person Shooter, Third Person 3D Game, Role Playing Game, Adventure Game, Educational and Edutainment* dan *Game Sport*.



7. Krakteristik perangkat lunak adalah *invisible* (tidak terlihat) dan *reusable* (dapat digunakan kembali).
8. Proses Bisnis adalah sekumpulan kegiatan atau aktifitas yang dirancang untuk menghasilkan suatu keluaran tertentu bagi pelanggan tertentu.
9. Bisnis Proses dapat digambarkan melalui *Business Process Modelling Notation*.
10. Model Proses Pengembangan Perangkat Lunak dan Gim secara garis besar dibagi menjadi 3 yaitu perencanaan, implementasi dan pengujian.
11. Hal-hal yang dibahas dalam manajemen proyek adalah perencanaan, manajemen risiko, spektrum manajemen 5P (*Project, process, people, price* dan *place*), dokumentasi dan pelaporan serta manajemen kualitas.
12. Fase perencanaan proyek meliputi: Definisi lingkup proyek, Analisis sumber daya, Analisis alat bantu, Analisis jadwal, Analisis biaya, Analisis risiko dan Analisis kelayakan.
13. Pemasaran adalah proses penting agar bisnis bisa berhasil.
14. Faktor-faktor yang dapat menentukan sebuah kualitas perangkat lunak menurut McCall adalah *Correctness, Reliability, Efficiency, Integrity, Usability, Maintainability, Flexibility, Testability, Portability, Reusability* dan *Interoperability*.
15. Keselamatan kerja merupakan sarana utama untuk mencegah kecelakaan seperti cacat dan kematian akibat kecelakaan kerja.
16. Industri 4.0 adalah sebuah istilah yang diciptakan pertama kali di Jerman pada tahun 2011 yang ditandai dengan revolusi digital.



17. IoT merupakan segala aktifitas yang pelakunya saling berinteraksi dan dilakukan dengan memanfaatkan internet.

18. *Cloud Computing* merupakan teknologi yang memberikan pelayanan secara luas dengan akses internet dimanapun berada, media penyimpanan cloud computing berada di internet.

19. *Cyber-security* adalah kumpulan alat, kebijakan, konsep keamanan, perlindungan keamanan, pedoman, pendekatan manajemen risiko, tindakan, pelatihan, praktik terbaik, jaminan dan teknologi yang dapat digunakan untuk melindungi lingkungan cyber dan organisasi dan aset pengguna.

20. *Branding* atau pelabelan lebih akrab dilakukan di dunia pemasaran. Pelabelan dilakukan karena begitu banyak produk sejenis yang beredar, sehingga diperlukan pembeda bagi produk yang satu dengan produk yang lain.

21. Hak Kekayaan Intelektual (HKI) adalah hak kekayaan yang timbul atau lahir karena kemampuan intelektual manusia.

22. CEO merupakan posisi dengan jabatan tertinggi dalam sebuah startup. CEO adalah pemimpin startup dan bertindak sebagai perwakilan perusahaan

23. Sistem Analis adalah personil tenaga ahli yang memiliki kemampuan untuk melakukan analisa terhadap sebuah sistem, memilih alternative pemecahan masalah dan menyelesaikan masalah tersebut dengan menggunakan komputer.

24. *Game artist* berperan membuat semua aspek visual dari game, mulai dari karakter, background, sampai menu dan antarmuka/UI.

25. Peluang usaha adalah kesempatan yang datang pada waktu tertentu yang harus diambil oleh seorang wirausahawan untuk



menciptakan suatu usaha dengan keberanian mengambil risiko, sehingga mendapatkan hasil yang diinginkan.

26. Peluang usaha bidang perangkat lunak dapat dikategorikan menjadi: Produk Jadi, Penawaran pesanan aplikasi sesuai kebutuhan calon konsumen dengan cara lelang atau tender dan Jasa *maintance* dan *service*.

27. Peluang usaha yang ada bidang gim adalah menjual gim, memasang iklan, menjual konten dan karakter dan menjual souvenir



F. PENGAYAAN

Untuk lebih menambah luas wawasanmu mengenai BAB ini, berikut link yang bisa kamu kunjungi dan memotivasi kamu agar menjadi lebih giat dan meningkatkan kualitas kemampuan kamu dibidang pengembang perangkat lunak dan gim:



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
DIREKTORAT JENDERAL PENDIDIKAN VOKASI
DIREKTORAT SEKOLAH MENENGAH KEJURUAN
Dasar - Dasar Pengembangan Perangkat Lunak dan Gim
SMK Kelas X
Penulis : Yustiana Amita Utama dan Esa Hergatama
ISBN : xxx-xxx-xxx-xxx-x

BAB II

ASET DESAIN PADA PERANGKAT LUNAK DAN GIM



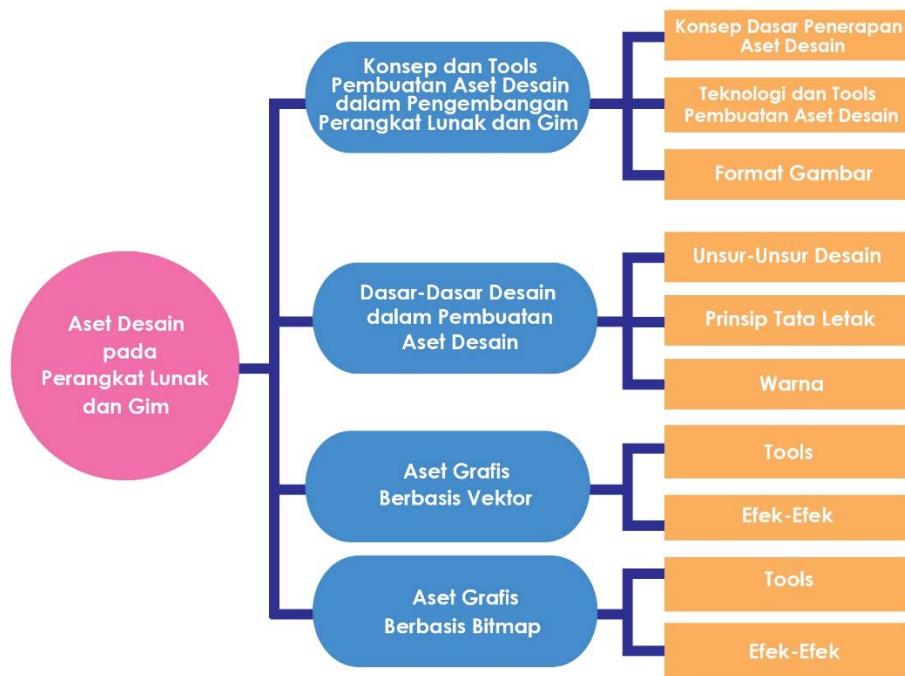
TUJUAN PEMBELAJARAN

Setelah mempelajari BAB II, Anda diharapkan untuk mampu:

- a. Mengidentifikasi tools di bidang pengembangan perangkat lunak dan gim.
- b. Memahami konsep dasar penerapan aset desain
- c. Memahami berbagai format gambar dalam pengembangan perangkat lunak dan gim
- d. Menerapkan unsur – unsur desain sebagai dasar pengelolaan asset grafis dan user interface dalam pengembangan perangkat lunak dan gim
- e. Menempatkan berbagai fungsi dan unsur warna CMYK dan RGB sebagai dasar pengolalan asset grafis dan user interface dalam pengembangan perangkat lunak dan gim
- f. Menerapkan prinsip – prinsip tata letak sebagai dasar pengolalan asset grafis dan user interface dalam pengembangan perangkat lunak dan gim
- g. Menggunakan perangkat lunak pengolah gambar vector
- h. Menerapkan manipulasi gambar vektor sebagai dasar pengolalan asset grafis dan user interface
- i. Menerapkan efek vektor sebagai dasar pengolalan asset grafis dan user interface
- j. Membuat asset grafis berbasis gambar vektor untuk kebutuhan asset dan user interface
- k. Menerapkan perangkat lunak pengolah gambar bitmap (raster) sebagai dasar pengolalan asset grafis dan user interface
- l. Menerapkan manipulasi gambar raster dengan menggunakan fitur efek sebagai dasar pengolalan asset grafis dan user interface
- m. Menerapkan efek raster dengan menggunakan fitur efek sebagai dasar pengolalan asset grafis dan user interface
- n. Membuat asset dan desain user interface vektor dan bitmap (raster)



PETA KONSEP



BAB II

ASET DESAIN PADA PERANGKAT LUNAK DAN GIM



APERSEPSI

Pernahkah kamu pergi ke sebuah taman, dengan harapan melihat pemandangan yang indah dengan tujuan menghirup udara yang segar dan menikmati suasana. Tapi bagaimana jika suasana taman tersebut tanpa bunga? Masihkah disebut taman?



Gambar 2. 1 Gambar Taman Bunga

Sumber Gambar : <https://www.merdeka.com/gaya/ditutup-begini-pemandangan-taman-bunga-terindah-di-dunia-keukenhof-tanpa-pengunjung.html>

Secara fungsi taman tersebut masih disebut taman karena masih bisa dinikmati udara dan lainnya. Tapi dengan adanya Bungan maka kondisi taman itu semakin indah. Begitupun juga asset desain pada sebuah perangkat lunak ataupun gim.

Apakah asset desain pada perangkat lunak dan gim, dapat diibaratkan seperti bunga tersebut. Sebuah perangkat lunak dan gim masih



bisa di nikmati fungsi nya walau tampilan dari perangkat lunak dan gim tersebut tidak begitu indah.



KATA KUNCI

Konsep Dasar Desain, Dasar – Dasar Desain, Aset Berbasis Vektor, dan Aset Berbasis Bitmap

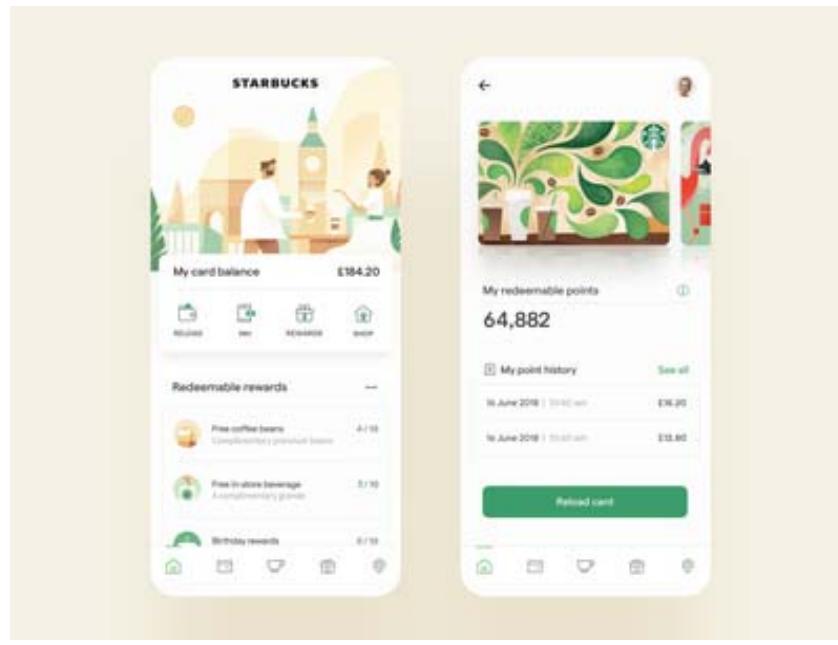
A. Konsep dasar dan tools pembuatan Aset Desain

Setelah memahami materi pada BAB I yaitu proses bisnis, perkembangan dunia kerja dan profesi pengembangan perangkat lunak dan gim. Pada BAB II ini kalian akan mempelajari aset desain pada perangkat lunak dan gim.

Pada saat ini multimedia sangat dibutuhkan pada perangkat lunak dan gim. Multimedia merupakan media komunikasi dan informasi dengan mengombinasikan antara teks, grafik, video dan animasi yang digunakan untuk menyampaikan suatu pesan tertentu melalui media komputer dan peralatan elektronik lainnya.

Dalam perangkat lunak, multimedia digunakan sebagai sarana menyampaikan informasi agar lebih efektif dan mempercantik tampilan perangkat lunak. Sedangkan multimedia pada gim menjadi *branchmark* / standar perkembangan pada setiap generasi *console*.





Gambar 2.2 Penerapan multimedia grafik, teks dan animasi dalam aplikasi starbucks.
Sumber: <https://dribbble.com/shots/7297745-Starbucks-UI-UX-Dashboard>



Gambar 2.3 Peningkatan Kualitas gambar

Pengembangan sebuah perangkat lunak dan *game* tidak terlepas dari aset di dalamnya. Bahkan, faktanya adalah apapun yang kita lihat, dengar dan gunakan dalam sebuah perangkat lunak adalah *asset* dan data. Apakah *asset* dalam *game* itu?

Asset adalah semua kebutuhan atau bahan - bahan yang digunakan dalam perangkat lunak atau gim. *Asset* dalam sebuah

SMK KELAS X | 130



perangkat lunak atau gim bisa berupa 2D, 3D, animasi, teks, gambar statis, ikon, efek suara, musik atau elemen multimedia lainnya.

1. Objek - Objek Multimedia

Untuk membuat sebuah asset pada perangkat lunak atau gim, ada beberapa objek - objek pada multimedia yang perlu kita ketahui, yaitu:

a. Teks

Teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia. Teks bertujuan untuk menjelaskan informasi agar dapat dengan mudah dan cepat dipahami oleh pembacanya.

b. Gambar

Secara umum image atau grafik dapat berupa foto dan gambar. Manusia sangat berorientasi pada visual (*visual-oriented*), dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi.

c. Animasi

Animasi merupakan serangkaian gambar bergerak berurutan untuk menyajikan proses tertentu yang biasa dilengkapi teks penjelasan dan audio. Elemen ini adalah salah satu elemen multimedia yang disukai dan menarik. Penggunaan animasi berperan penting dalam mempermudah dalam memahami informasi yang kompleks dan abstrak.

d. Audio

Suara merupakan gelombang yang dihasilkan dari benda bergetar di udara. Benda bergetar tersebut mengakibatkan molekul yang terdapat di udara merapat merenggang dan menyebar, jika sampai ditelinga manusia akan terdengar suara. Suara dalam



sajian multimedia dapat berupa narasi suara manusia, latar musik, efek suara dan lain-lain. Suara dapat bermanfaat untuk penyampaian informasi teks ataupun gambar.

e. Video

Video adalah hasil rekaman proses kejadian yang berisikan gambar berurutan disertai suara. Jika dibandingkan dengan animasi maka video lebih realistik.

f. Link Interaktif

Link interaktif digunakan untuk menggabung berbagai elemen multimedia agar menjadi informasi yang padu.

2. Penggunaan Multimedia

Multimedia dapat digunakan untuk bermacam – macam bidang pekerjaan tergantung dari kreativitas untuk mengembangkannya.

Pemanfaatan penggunaan multimedia, yaitu:

a. Multimedia pada Perangkat Lunak

Perkembangan teknologi menambah penggunaan multimedia dalam memenuhi kebutuhan komunikasi. Berbagai persoalan dalam multimedia pada perangkat lunak lainnya seperti *chatting*, *e-learning*, *video conference*, dan lainnya dapat disebut MoIP (*Multimedia Over Internet Protocol*).

b. Multimedia pada Gim

Beberapa gim dilengkapi dengan animasi, animasi yang mengandung gambar, video, teks, dan audio yang didalamnya mengandung berbagai informasi memperkuat *genre* dari gim tersebut. Contohnya *genre* gim yang berupa kuis, teks pada



screen hanya berisi pertanyaan, dapat dikomunikasikan menggunakan grafik.

Tidak hanya aspek multimedia tetapi aspek teknik juga diperlukan pada waktu merancang gim. Penerapan multimedia dan gim memerlukan pemrograman yang baik.

3. Desain Grafis

Setelah memahami objek-objek multimedia dalam membuat aset penting bagi kita untuk memahami apa itu desain grafis? Desain grafis sebagai sarana visual yang tercipta dapat menjadi sarana penyampaian informasi atau pesan secara jelas dan efektif, bahkan mampu membentuk persepsi manusia akan sebuah hal.

Blanchard (1986) mendefinisikan desain grafis sebagai "suatu seni komunikatif yang berhubungan dengan industri, seni dan proses dalam menghasilkan gambaran visual pada segala permukaan". Dari definisi itu dapat disimpulkan bahwa produk dari desain grafis merupakan gambaran visual yang merupakan suatu karya seni yang komunikatif yang dapat digunakan pada segala permukaan.

Jadi, desain grafis adalah ilmu yang mempelajari dan mengembangkan bahasa visual (salah satunya) untuk keperluan informasi dan komunikasi dari pelontar pesan. Pesan visual yang ingin disampaikan dapat berupa informasi produk, jasa atau gagasan kepada sasaran (audiensi) secara komunikatif dan persuasif (dengan tujuan publikasi, promosi, dan pemasaran) menggunakan media informasi (misalnya: televisi, *website*, majalah, surat kabar, brosur).



4. Tools Desain Grafis

Adapun peralatan yang digunakan dalam desain grafis konvensional adalah sebagai berikut:

- Drawing Board (Meja Gambar)
- Box Light (meja *layout*)
- Berbagai pisau potong (*Art knife*)
- Berbagai peralatan menempel
- Berbagai penggaris
- Berbagai macam karton dan kertas
- Pulpen (*art pen*) dengan segala macam ujung penanya
- Pensil dan crayon berwarna

Pada masa digital, semua tahapan desain grafis secara konvensional dapat dilengkapi atau digantikan dengan komponen sebagai berikut :

- Sistem *hardware* (perangkat keras) meliputi : PC dan monitor
- *Input device* yang meliputi: kamera, *scanner*, dan *pen tablet*
- *Output device*: *Laser printer*, *plotter* dan lain – lain.
- Sistem *software* (perangkat lunak) terdiri dari *software* pengolah kata, *software page make-up/layout software* spreadsheet, *software graphic*, *software clip art* dan *software* penunjang.





Refleksi I

Setelah mempelajari materi ini, manfaat yang akan kamu rasakan adalah anda dapat memahami konsep – konsep dasar dan *tools* yang digunakan dalam membuat aset desain. (Elemen – elemen, penggunaan multimedia, desain grafis dan *tools* desain grafis) yang tentunya dapat membekali pengetahuan anda dalam memahami konsep multimedia dan *asset* dalam perangkat lunak dan gim.



Assesmen I

I. Tugas Mandiri

Coba lakukan identifikasi objek – objek multimedia yang terdapat pada sebuah perangkat lunak atau gim yang kamu ketahui!

II. Latihan Soal

1. Apakah yang dimaksud dengan aset pada perangkat lunak dan gim?
2. Jelaskan fungsi multimedia pada perangkat lunak dan gim!
3. Sebut dan jelaskan objek – objek multimedia?
4. Apa yang dimaksud dengan desain grafis?
5. Sebut dan jelaskan tools – tools dalam desain grafis!

B. Dasar – Dasar Desain dalam pembuatan Aset Desain

Untuk membuat sebuah karya desain grafis yang bagus ada beberapa kriteria dan komponen penting yang tidak boleh dilewatkan. Dalam menciptakan sebuah karya desain grafis harus memperhatikan unsur-unsur desain yang minimal harus sesuai dengan ketentuan yang ditetapkan. Jika salah dalam menentukan satu komponen dapat

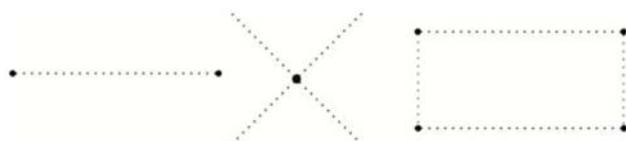


berakibat pada kualitas pada desain itu sendiri. Hal yang penting dalam desain adalah unsur tata letak yang harus sesuai dan lengkap karena unsur-unsur tata letak tidak boleh terpisahkan dan mengandung arti yang sangat dalam

1. Komponen Desain Grafis

Komponen desain grafis merupakan unsur pembentuk desain grafis itu sendiri. Dalam membuat sebuah desain grafis, diperlukan komponen – komponen yang menjadi dasar visual dalam memberikan peranan komunikasi secara visual. Untuk itu, komponen desain grafis terdiri atas:

a. Titik



Gambar 2.4 Contoh Komponen Desain Titik

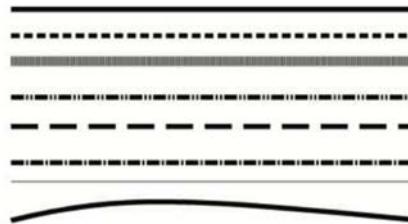
Titik sebagai salah satu komponen desain grafis yang paling mendasar, baik berdiri sebagai individual maupun sebagai kelompok, komponen titik memberikan nuansa komunikasi yang berarti dalam sebuah desain. Potensi titik, antara lain sebagai berikut:

- 1) Membentuk sebuah makna: titik tunggal bermakna selesai, titik berkelompok bermakna kesatuan atau kebersamaan;
- 2) mengekspresikan nilai statis dan emosi yaitu titik tunggal yang berdiri sebagai kemantapan atau bernilai absolut serta berdiri sebagai penekanan selesai;



- 3) dalam desain, bisa berperan sebagai pemberian aksen (sebagai elemen dekoratif, elemen emosi, dan elemen bahasa gambar);

b. Garis



Gambar 2.5 Contoh komponen desain garis

Garis adalah sebuah komponen desain grafis yang merupakan perapatan dari sekelompok titik yang berurut dan berunut, baik berdiri sebagai elemen utama maupun bagian dari (aksen) komunikasi. Komponen garis memberikan juga nuansa komunikasi yang berarti dalam sebuah desain. Potensi garis, antara lain sebagai berikut:

- 1) Membentuk garis tepi/kontur sebuah benda;
- 2) Mengekspresikan gerak dan emosi
- 3) dalam desain, bisa berperan sebagai pemberian aksen (sebagai pembatas, kolom, dan dekoratif desain);

Menurut Adi Kusrianto (2007) goresan garis dapat memiliki beberapa arti yaitu:

- Garis tegak : kuat, kokoh, tegas, dan hidup.
- Garis datar : lemah, tidur, dan mati.
- Garis lengkung : lemah, lembut, mengarah.
- Garis patah : tegas, tajam, hati-hati, naik-turun.
- Garis miring : sedang, menyudutkan.
- Garis berombak : halus, lunak, berirama.



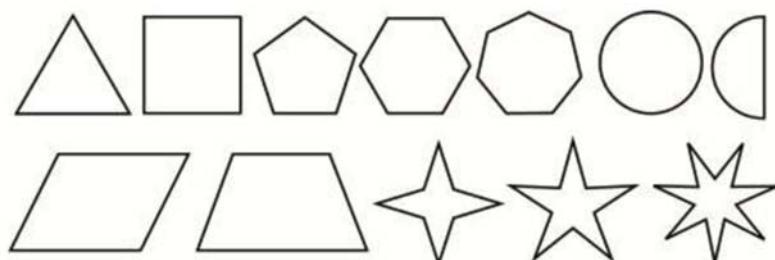
c. Bentuk

Bentuk/raut/kurva adalah sebuah komponen desain grafis yang merupakan bertemuannya titik awal garis dengan titik akhir garis. Tentunya, hasil dari garis yang bertemu ini membentuk sebuah bidang. Bidang tersebut, baik berdiri sebagai elemen utama maupun bagian dari (aksen) komunikasi, sangat berperan dalam memberikan nuansa komunikasi yang berarti dalam sebuah desain. Potensi Bentuk, antara lain sebagai berikut :

- 1) Istilah bentuk digunakan untuk menyatakan suatu bangun/shape yang tampak dari suatu benda.
- 2) Bentuk merupakan rupa keliling dari sebuah rancang.
- 3) Bentuk mempunya garis luar (outline) atau pembatas disekitarnya.
- 4) Bentuk memaksimalkan karakter/kesan yang ingin dicapai.

Ada 2 bentuk dasar dalam desain grafis, yaitu:

- 1) Geometris dan Organis

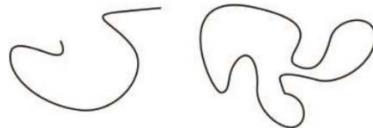


Gambar 2.6 Bentuk – Bentuk Geometris

Bentuk Geometris (segitiga, segiempat, segilima, lingkaran, trapesium, dan seterusnya). Bentuk geometris ini, merupakan sebagian dari bentuk dasar yang dipergunakan sebagai acuan dasar desainer dalam merancang.



2) Bentuk Organis, Lengkungan Bebas dan Fleksibel



Gambar 2.7 Bentuk – bentuk Organis

d. Ruang

Ruang adalah sebuah komponen desain grafis yang merupakan kelanjutan dari bentuk yang dikembangkan dapat membentuk ruang imajiner yang terkait persepsi pengamatnya. Hasil dari pengembang bidang yang menjadi ruang ini lebih banyak berperan sebagai elemen utama (berdasarkan tingkat kerumitannya), tetapi tidak menutup kemungkinan pula dapat berdiri sebagai bagian dari komunikasi. Ruang sangat berperan dalam memberikan nuansa 3 dimensi dalam unsur komunikasi visual. Potensi ruang, antara lain sebagai berikut.

- 1) Ruang menjadi pengembangan bidang yang menjadi bidang lain, tetapi tidak hanya sekadar bidang baru. Ruang harus memiliki dimensi sehingga melahirkan berat/*massa* pada bidang baru yang disebut ruang.
- 2) Sebuah bentuk 3 dimensi dapat digambarkan pada permukaan papar yang terdiri atas beberapa bentuk 2 dimensi.
- 3) Selain membentuk nuansa 3 dimensi, ruang juga dapat memperlihatkan penggabungan bentuk dari sisi/cara pandang yang berbeda (*gestalt*).
- 4) Ruang kosong dimanfaatkan agar rancangan tidak terlalu penuh.
- 5) Ruang kosong dimanfaatkan sebagai pemisah.



e. Warna

Warna dapat membantu menciptakan mood dan membuat teks lebih berbicara. Sebagai contoh, desain publikasi yang menggunakan warna-warna soft dapat menyampaikan kesan lembut, tenang dan romantis. Warna-warna kuat dan kontras dapat memberikan kesan dinamis, cenderung meriah (Supriyono, 2010:70).

Supriyono (2010:72) mengatakan bahwa berdasarkan *hue* (spektrum warna) warna dibagi menjadi tiga golongan, yaitu :

- 1) Warna primer (*primary colors*) terdiri dari warna merah, kuning dan biru
- 2) Warna sekunder (*secondary color*), merupakan campuran dua warna primer dengan perbandingan seimbang, menghasilkan warna orange (merah+kuning), hijau (kuning+biru), dan ungu (biru+merah)
- 3) Warna tersier (*tertiary colors*) merupakan gabungan antara warna primer dan sekunder, yaitu: kuning-orange, merah-orange, merah-ungu, biru-ungu, biru-hijau, dan kuning-hijau.

f. Teskstur

Tekstur adalah sebuah komponen desain grafis yang merupakan tampilan/karakteristik/gambaran/representasi sifat dari suatu permukaan. Tekstur dapat dibiarkan sebagaimana adanya atau diolah secara khusus menurut kehendak perancangnya. Permukaan dapat polos, bersisik, licin, kasar, pudar, kusam, kilap, lembut, halus, berlendir, terasa gatal, berbulu, dan lain-lain. Tekstur merupakan salah satu unsur seni



yang unik karena ia dapat mengaktifkan 2 proses penginderaan sekaligus (visual dan raba). Tekstur dapat berukuran kecil, menekankan pada kedua dimensian permukaan sebagai hiasan atau berukuran besar yang menekankan pada kesan raba pada 3D. Potensi tekstur, antara lain sebagai berikut:

- 1) memperkaya kenikmatan visual;
- 2) dipakai dalam memvisualkan objek;
- 3) membangkitkan perasaan untuk meraba;
- 4) memperjelas kesan adanya ruang;
- 5) tekstur dan kesan ruang: tekstur detail/halus (jarak dekat) dan tekstur blur/kurang jelas (jarak jauh).

g. Terang – Bayang / Gradasi

Terang-Bayang adalah sebuah komponen desain grafis yang merupakan permainan baru dalam bentuk ranah/lingkup arsir. Dari penambahan elemen arsir tersebut (biasanya adalah efek gradasi), bentuk yang dikembangkan akan memiliki nuansa realistik yang diambil dari prinsip jatuhnya unsur gelap-terang pada sebuah benda bila terkena sumber cahaya. Hasil dari komponen terang-bayang ini berperan sebagai ilustrasi utama (berdasarkan nilai realistiknya), tetapi tidak menutup kemungkinan pula dapat berdiri sebagai bagian dari komunikasi yang kreatif. Potensi terang-bayang, antara lain sebagai berikut.

- 1) Terang-bayang dari arsiran adalah upaya menambah nuansa jatuhnya *highlight* dan *shadow* pada sebuah benda.
- 2) Pendekatan terang-bayang ini menghasilkan nuansa realistik dan atau fotografik pada sebuah komponen desain.



- 3) Peranannya bisa kepada ilustrasi yang utama juga bisa sebagai sebuah elemen aksen dekoratif.

2. Prinsip-Prinsip Tata Letak

a. Kesatuan

Kesatuan atau *unity* merupakan salah satu prinsip yang menekankan pada keselarasan dari unsur-unsur yang disusun, baik dalam wujudnya maupun kaitannya dengan ide yang mendasarinya. Kesatuan diperlukan dalam suatu karya grafis yang mungkin terdiri dari beberapa elemen di dalamnya. Dengan adanya kesatuan itulah, elemen-elemen yang ada saling mendukung sehingga diperoleh fokus yang dituju. Secara elementer, ada beberapa cara untuk mencapai kesatuan, yaitu:

- 1) Menentukan dominasi agar diperoleh pengaruh yang tepat
- 2) Dominasi pada ukuran
- 3) Dominasi pada warna
- 4) Dominasi pada letak atau penempatan
- 5) Ukuran sebagai daya tarik
- 6) Menyatukan arah
- 7) Menyatukan bentuk

b. Keseimbangan

Keseimbangan atau *balance* merupakan prinsip dalam komposisi yang menghindari kesan berat sebelah atas suatu bidang atau ruang yang diisi dengan unsur-unsur rupa. Keseimbangan dapat dibagi menjadi 3 yaitu Balans simetri dan asimetris, Balans memusat dan Balans menyebar (Kusrianto, 2007:38).



Faktor yang mendukung keseimbangan yaitu posisi atau penempatan, proporsi, kualitas, dan arah dari unsur-unsur pendukungnya. Komposisi yang bagus, artinya yang indah dipandang mata merupakan keharusan dalam mendesain. Tata letak sangatlah penting karena itu akan membuat orang yang melihatnya terkagum-kagum. Jadi tata letak yang benar menentukan bagaimana nilai dari karya desain tersebut.

c. Irama

Irama atau ritme adalah penyusunan unsur-unsur dengan mengikuti suatu pola penataan tertentu secara teratur agar didapatkan kesan yang menarik. Penataannya dapat dilaksanakan dengan mengadakan pengulangan maupun pergantian secara teratur.

d. Kontras

Kontras di dalam suatu komposisi diperlukan sebagai vitalitas agar tidak terkesan monoton. Tentu saja, kontras ditampilkan secukupnya saja karena bila terlalu berlebihan, maka muncul ketidakteraturan dan kontradiksi yang jauh dari kesan harmonis.

e. Fokus

Fokus atau pusat perhatian selalu diperlukan dalam suatu komposisi untuk menunjukkan bagian yang dianggap penting dan diharapkan menjadi perhatian utama. Penggunaan keharmonisan dalam membuat suatu fokus dilakukan dengan menjadi segala sesuatu yang berada di sekitar fokus mendukung fokus yang telah ditentukan.



f. Proporsi

Proporsi adalah perbandingan ukuran antara bagian dengan bagian dan antara bagian dengan keseluruhan. Prinsip komposisi tersebut menekankan pada ukuran dari suatu unsur yang akan disusun dan sejauh mana ukuran itu menujukan keharmonisan tampilan sesuatu desain.

3. Tipografi

Tidak dapat dipungkiri bahwa teks adalah bagian dari desain grafis yang sangat penting. Oleh karena itu, untuk menguasai desain grafis, harus dipelajari pula tipografi, yaitu ilmu yang mempelajari segala sesuatu tentang huruf cetak.

Dalam desain grafis, tipografi didefinisikan sebagai suatu proses seni untuk menyusun bahan publikasi menggunakan huruf cetak. Oleh karena itu, "menyusun" meliputi merancang bentuk huruf cetak hingga merangkainya dalam sebuah komposisi yang tepat untuk memperoleh suatu efek tampilan yang dikehendaki. Desain komunikasi visual tidak bisa lepas dari tipografi sebagai unsur pendukungnya. Perkembangan tipografi banyak dipengaruhi oleh faktor budaya serta teknik pembuatan. Karakter tipografi yang ditimbulkan dari bentuk hurufnya bisa dipersepsikan berbeda.

Pemilihan huruf tidak semudah yang dibayangkan, ribuan bahkan jutaan jumlah huruf menyebabkan desainer harus cermat dalam memilih tipografi yang tepat untuk karyanya. Rangkaian huruf dalam sebuah kata atau kalimat bukan saja berarti suatu makna yang mengacu kepada sebuah objek ataupun gagasan, tetapi juga memiliki kemampuan untuk menyuarakan suatu citra ataupun kesan secara visual. Hal itu dikarenakan terdapat nilai fungsional dan nilai estetika



dalam suatu huruf. Pemilihan huruf disesuaikan dengan citra yang ingin diungkapkan.

Lazlo Moholy berpendapat bahwa sebuah kata atau kalimat bukan saja bisa berarti suatu makna yang mengacu kepada sebuah obyek ataupun gagasan, tetapi juga memiliki kemampuan untuk menyuarakan suatu citra ataupun kesan secara visual. Hal itu dikarenakan terdapat nilai fungsional dan nilai estetika dalam suatu huruf. Pemilihan jenis huruf disesuaikan dengan citra yang ingin diungkapkan (Kusrianto, 2007 : 190-191).

a. Jenis – Jenis Huruf

- 1) *Oldstyle*
- 2) *Modern*
- 3) *Slab Serif*
- 4) *Sans Serif*

b. *Legibility* dan keterbacaan

Legibility adalah tingkat kemudahan mata mengenali suatu tulisan dengan mudah. Hal ini bisa ditentukan oleh:

- 1) Kerumitan desain huruf, seperti penggunaan serif, kontras stroke, dan lain-lain
- 2) Penggunaan warna
- 3) Frekuensi pengamat menemui huruf tersebut dalam kehidupan sehari-hari.

Keterbacaan adalah tingkat kenyamanan sesuatu susunan huruf saat dibaca, yang dipengaruhi oleh:

- 1) Jenis huruf
- 2) Ukuran
- 3) Pengaturan, termasuk di dalamnya alur, spasi,



- 4) perataan dan lain-lain
- 5) Kontras warna terhadap latar belakang.

Refleksi II



Setelah mempelajari materi ini, manfaat yang akan kamu rasakan adalah Anda dapat memahami dasar – dasar desain grafis: komponen desain dan prinsip – prinsip tata letak yang tentunya dapat membekali pengetahuan anda dalam menerapkan pembuatan aset pada perangkat lunak dan gim.



Asesmen II

Penugasan ini melatih Anda dalam melakukan proses pengamatan aplikasi web atau gim yang akan mengamati tampilan aplikasi dari elemen desain grafis yang digunakan untuk menyusun tampilan tersebut.

1. Buatlah kelompok dengan anggota 3 – 4 orang
2. Amatilah 2 aplikasi web dan 2 aplikasi gim
3. Tunjukkanlah komponen desain dan prinsip tata letak yang digunakan untuk menyusun tampilan tersebut.
4. Buatlah tabel seperti gambar berikut perbedaan dan persamaan yang anda dapatkan diskusikan dengan teman sekelompok
5. Buat laporan dan diskusikan dengan teman sekelompok

No	Nama Aplikasi	Komponen Desain	Prinsip Tata Letak



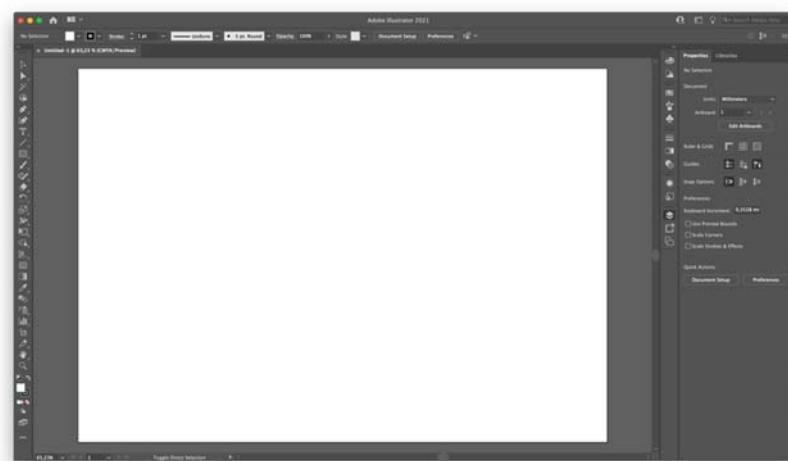
C. Aset Grafis Berbasis Vector

1. Mengenal *Adobe Illustrator*

a. Membuat Dokumen Baru

Setelah membuka aplikasi *Adobe Illustrator CC*, langkah pertama yang dilakukan adalah membuat dokumen baru atau membuka dokumen yang ada. Dikarenakan ini adalah pertama kali menggunakan *Adobe Illustrator* maka perintah pertama yang dilakukan adalah membuat dokumen baru. Proses pembuatan dokumen baru yaitu :

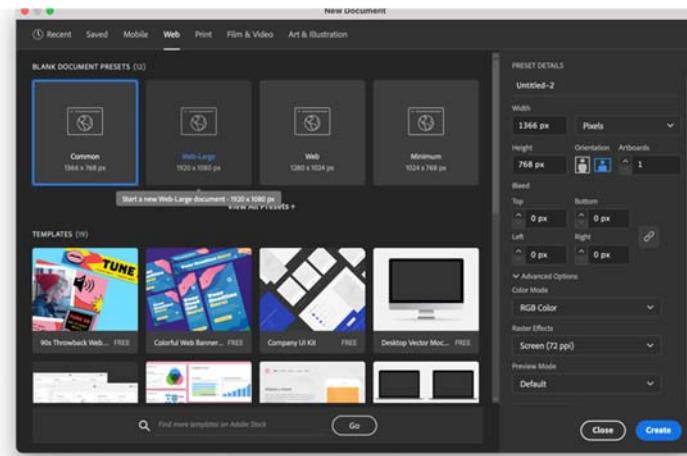
- 1) Pilih menu **File > New**



Gambar 2. 8 Tampilan *Adobe Illustrator* 2021

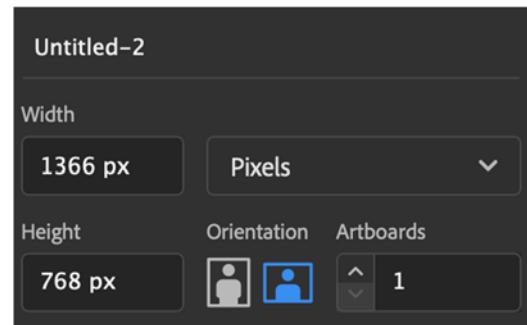
- 2) Jika ingin membuat desain untuk aset pada perangkat lunak atau gim dapat memilih *template* pada kelompok **Mobile/Web/Art & Illustration**





Gambar 2. 9 Memilih *Template* dari Kelompok *Web*

- 3) Pilih ukuran yang diinginkan, jika ingin mengubah ukuran masukan ukuran dan satuan ukuran yang digunakan pada kotak isian **width** atau **height**



Gambar 2. 10 *Width* dan *Height* menentukan ukuran *artboards*

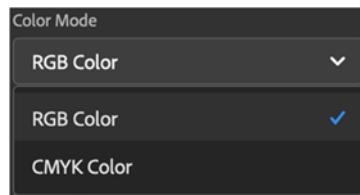
- 4) Opsi **Orientation** digunakan untuk mengatur posisi *portrait* (berdiri) atau *landscape* (mendatar)
- 5) **Bleed** digunakan untuk membuat area tambahan diluar ukuran yang telah ditetapkan, jika ingin mengatur *bleed* anda dapat mengisi *top*, *bottom*, *left* dan *right* pada kotak kolom isian.





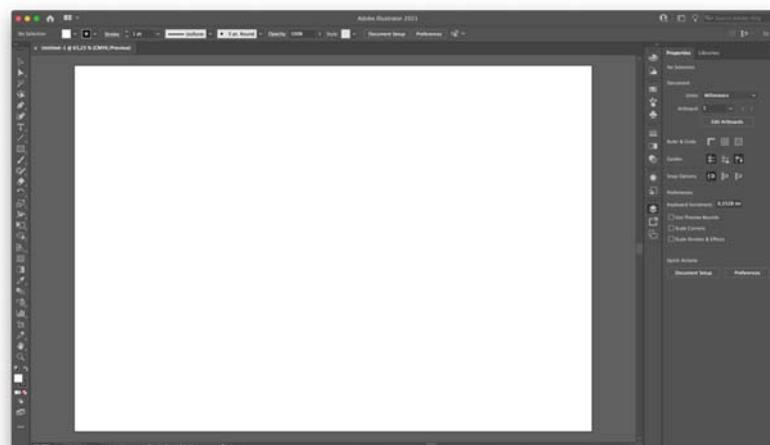
Gambar 2. 11 Pengaturan *Bleed*

- 6) Terakhir **Color Mode**. Untuk mode warna yang digunakan dengan tujuan pencetakan dapat memilih CMYK. Sedangkan untuk hasil objek desain yang hanya digunakan pada layar monitor saja dapat menggunakan mode **RGB**.



Gambar 2. 22 Pengaturan *Color Mode*

- 7) Kemudian tekan **Create**.
Tunggu hingga beberapa saat, hingga kemudian *Adobe Illustrator* membuat dokumen baru.



Gambar 2. 33 Tampilan dokumen baru pada *Adobe Illustrator*



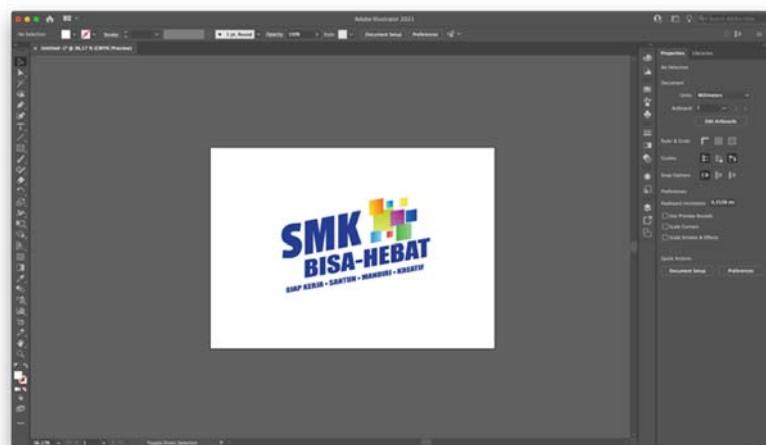
b. Meletakan Gambar (*Place*)

Perintah *Open* digunakan untuk membuka *file* dan akan dianggap sebagai dokumen. Sedangkan *Place* digunakan untuk meletakkan *file* ke dalam dokumen yang saat ini sedang aktif.

Jadi, jika Anda membuat poster dan membutuhkan *file* foto, *file* itu harus dimasukkan ke dalam dokumen menggunakan perintah **Place**.

Berikut langkah-langkah menggunakan perintah *Place* :

- 1) Pastikan sudah ada dokumen yang sedang aktif
- 2) Pilih menu **File > Place**
- 3) Cari *file* yang akan diletakkan di dalam dokumen
- 4) Tekan tombol **Place**
- 5) Arahkan kursor *mouse* untuk menempatkan *file* tersebut ke dalam dokumen
- 6) Klik dan Anda bisa mengatur ukuran *file* itu sehingga sesuai dengan kebutuhan.



Gambar 2. 14 Memasukan Gambar pada *Artboard*



7) Jika *Selection Tool* sedang aktif, kliklah di sembarang tempat untuk menonaktifkan *bounding box* yang mengelilingi *file* tersebut. Sekarang, *file* itu akan menjadi objek baru di dalam dokumen

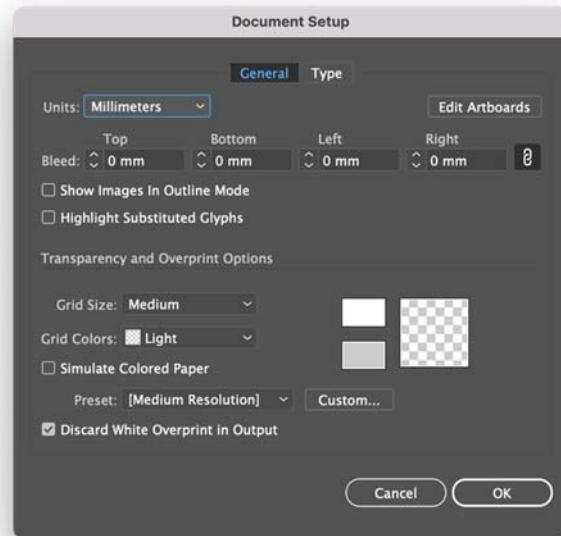
c. *Document Setup*

Dokumen yang sudah dibuat dapat diubah kapan pun menggunakan *Document Setup*. Tombol ini dapat ditemukan di dalam toolbar dan bisa diklik sewaktu-waktu.



Gambar 2. 15 *Toolbar Document Setup*

Ada beberapa parameter yang dapat digunakan sewaktu membuat dokumen baru. Namun, ada juga parameter-parameter yang belum diatur secara khusus.



Gambar 2. 16 *Document Setup*



- 1) **Show Images in Outline Mode** : dalam kondisi aktif, gambar bitmap akan tetap ditampilkan meskipun mode yang aktif adalah *Outline Mode*
- 2) **Highlight Substituted Glyphs** : dalam kondisi aktif, maka apabila terjadi *glyphs (special character/karakter khusus)* yang diganti dengan lainnya, akan ditandai dengan *highlight*
- 3) **Transparency and Overprint Options** : bagian ini digunakan untuk mengatur area kerja ketika bekerja dengan objek transparan atau *overprint*
- 4) **Grid Size** : mengatur ukuran *grid* untuk menunjuk area transparan pada objek
- 5) **Grid Colors** : mengatur warna *grid* untuk menunjuk area transparan pada objek

Transparency Grid dapat diaktifkan dengan memilih menu **View > Show Transparency Grid**.



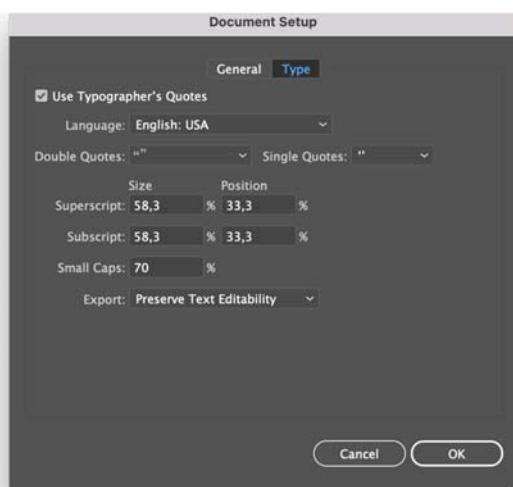
Gambar 2. 17 *Transparency Grid* dalam posisi aktif



6) **Simulate Colored Paper** : dalam kondisi aktif, Anda bisa membuat simulasi jika kertas yang digunakan tidak berwarna putih tapi warna lain.

Selain digunakan untuk mengatur dokumen, *Document Setup* juga bisa mengatur *Type*. Tekan tab *Type* dan ada beberapa parameter yang dapat digunakan:

- 1) **Use Typographer's Quotes** : dalam kondisi aktif, penggunaan tanda kutip ganda akan berbentuk (6 dan » dan bukan hanya sekadar berbentuk
- 2) **Language** : digunakan untuk mengatur penggunaan bahasa selama Anda bekerja dengan dokumen. Ini akan memengaruhi spelling dan pemotongan kata secara baik (*hyphenation*). Sayangnya, belum ada opsi Bahasa Indonesia.
- 3) **Double Quotes** : saat Anda menulis tanda kutip ganda, pilih bentuk yang diinginkan.
- 4) **Single Quotes** : apabila Anda menulis tanda kutip tunggal, pilih bentuk yang diinginkan



Gambar 2. 18 *Document Setup tab (Type)*

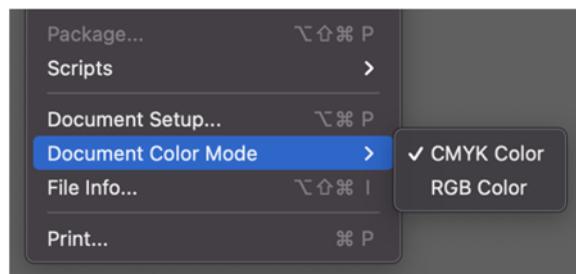


- 5) **Superscript** : atur seberapa tinggi teks yang akan diformat sebagai *superscript* terlihat di dalam dokumen
- 6) **Subscript** : atur seberapa rendah teks yang akan diformat sebagai *subscript* dalam dokumen
- 7) **Small Caps** : mengatur ukuran teks jika diformat sebagai *small caps*
- 8) **Export** : mengatur bagaimana teks ketika diekspor, apakah ingin memprioritaskan untuk diedit atau dibaca

d. *Document Color Mode*

Adobe Illustrator menyediakan dua mode warna, yaitu CMYK dan RGB. Salah satu mode warna akan aktif. Anda bisa memilih mode warna itu saat pembuatan dokumen dari awal.

Namun, jika Anda ingin mengganti mode warna pada saat dokumen sudah dibuat. kliklah menu **File > Document Color Mode**. Pilih mode warna yang diinginkan.



Gambar 2. 19 Memilih *Document Color Mode*

e. Menyimpan Dokumen

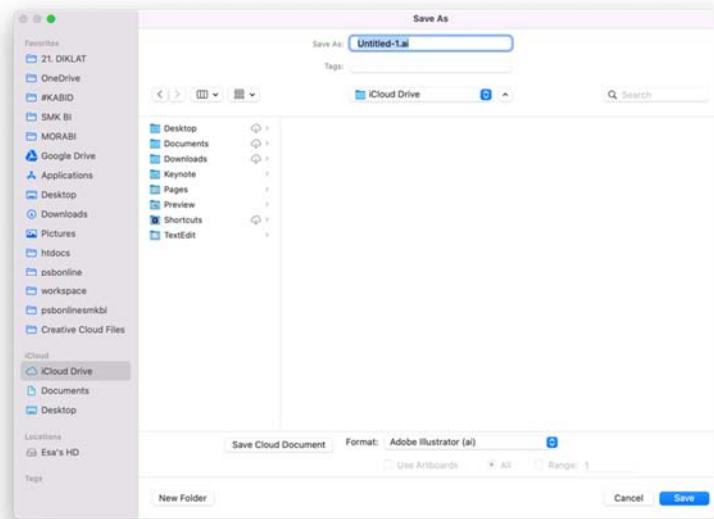
Simpanlah desain Anda sesering mungkin. Ini ditempuh untuk mencegah agar desain itu tidak hilang karena hal-hal yang tak terduga seperti system crash, mati lampu, atau faktor lainnya. Penyimpanan dokumen dibedakan menjadi dua, yaitu penyimpanan pertama kali untuk pembuatan nama *file* dan



penyimpanan berkala untuk meng-*update file* dengan desain yang baru.

Untuk penyimpanan pertama kali, lakukan langkah-langkah berikut:

- 1) Pilih menu File > Save As.
- 2) Buat nama *file* di dalam kotak isian *File Name*
- 3) Apabila ingin disimpan dalam format *Adobe Illustrator* dengan ekstensi .AI, pilihlah opsi tersebut pada menu kombo *Save as Type*.
- 4) Jika dokumen mengandung beberapa halaman, pilih *All* untuk menyimpan seluruh halaman atau *Range* jika hanya ingin menyimpan satu atau beberapa halaman saja.



Gambar 2. 20 Simpan *file* ke dalam format .AI (Mac Os)

Sekarang, dokumen Anda sudah disimpan dalam sebuah *file* dengan nama spesifik. Kemudian, lanjutkan desain Anda dan simpan secara berkala dengan hanya menekan tombol **Ctrl+S** atau pilih menu **File > Save**.



1) Menyimpan dokumen (*Save a copy*)

Perintah **Save a Copy** digunakan apabila Anda ingin mengarsipkan dokumen ini ke dalam *file* dengan nama lain. Secara *default*, nama *file* akan diberi akhiran -copy. Misalnya, desainku-copy.ai.

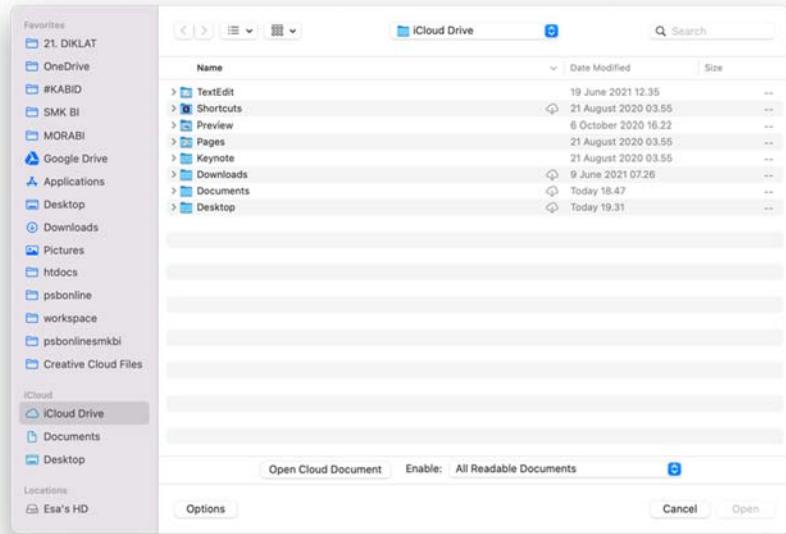
2) Menyimpan dokumen (*Save as template*)

Perintah *Save as Template* dipakai apabila Anda mendesain dokumen yang akan dijadikan template untuk desain-desain lainnya. Sebagai contoh, Anda membuat *stationery* yang bagian atasnya selalu ada logo, nama, dan alamat perusahaan. Alih-alih memasang logo dan menulis nama dan alamat perusahaan berkali-kali, lebih baik buat satu desain yang nanti dijadikan template. Desain itu sudah dilengkapi dengan logo, nama, dan alamat perusahaan.

f. Membuka Dokumen

Perintah untuk membuka *file* yang sudah disimpan sebelumnya adalah dengan menggunakan *Open*. Anda bisa mengklik menu **File > Open** atau menekan **tombol Ctrl+O**. Jika *file* itu termasuk sering Anda buka dan simpan, silakan cek dengan mengklik menu **File > Open Recent** dan pilih nama file yang dimaksud.





Gambar 2. 21 Tampilan pada saat membuka dokumen (*Mac Os*)

Secara *default*, Anda bisa membuka *file Adobe Illustrator* berakhiran .AI. Namun, *Adobe Illustrator* juga bisa membuka *file* dengan format lain. Berikut format-format populer yang dapat dibuka oleh *Adobe Illustrator* :

1) *CorelDRAW (.CDR)*

Pengguna *CorelDRAW* dapat membuka *file* desain mereka menggunakan *Adobe Illustrator*. Tidak hanya sekadar dibuka, desain yang sudah dibuat dapat di-*edit* menggunakan *tool-tool Adobe Illustrator*

2) *File bitmap (.JPEG, PNG, GIF, dan lain-lain)*

file-file ini bisa dibuka tanpa masalah *menggunakan Adobe Illustrator*

3) *File MS Word (.DOC, .DOCX, .RTF)*

apabila ingin melibatkan tulisan yang dibuat menggunakan MS Word sebagai bagian dari desain grafis, bukalah *file* itu langsung dengan memanfaatkan *Adobe Illustrator*



4) *Photoshop* (.PSD)

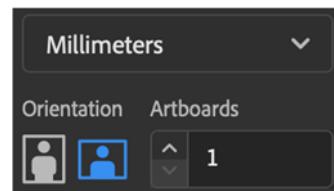
Anda bisa mengintegrasikan desain dari *Photoshop* ke dalam *Illustrator* dengan membuka *file* dengan format ini

5) *Adobe PDF* (.PDF)

File berformat .PDF juga dapat dibuka menggunakan *Adobe Illustrator*

g. Menambah Halaman

Dalam satu dokumen bisa terdapat beberapa halaman. Di lingkungan *Adobe Illustrator*, halaman-halaman dalam sebuah dokumen sering disebut dengan istilah *Artboard*. Anda bisa membuatnya sejak awal dengan menentukan jumlah *Artboard* yang ingin ditambahkan ketika membuat dokumen baru. Lihat gambar di bawah ini.

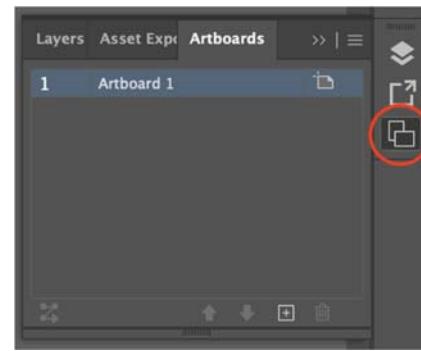


Gambar 2. 22 Pengaturan jumlah *artboard* saat pertama membuat dokumen

Kalau Anda sudah telanjur membuat dokumen atau membuka dokumen yang pernah disimpan sebelumnya, maka penambahan artboard dilakukan dengan langkah-langkah berikut:

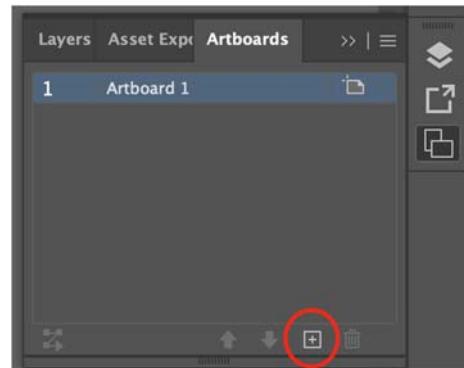
- 1) Pilih menu *Window* > *Artboards* atau klik ikon *Artboards*



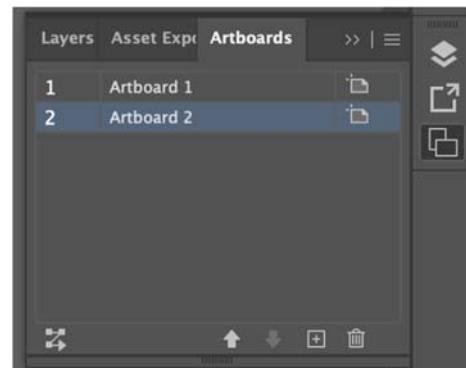


Gambar 2. 43 Icon Menu Artboards

2) Untuk menambah artboard baru, klik ikon New Artboard



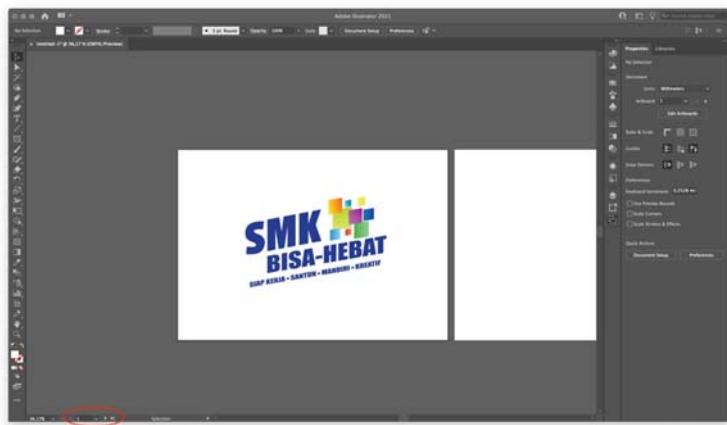
Gambar 2. 24 Icon Untuk Menambah Artboards



Gambar 2. 25 Artboards baru telah ditambah

3) Anda bisa berpindah dari satu *artboard* ke *artboard* lainnya





Gambar 2. 265 Tampilan *Adobe Illustrator* saat memiliki 2 *Artboards*

- 4) melalui menu *Artboard Navigation* yang ada di *Status bar*.

h. *Revert*

Perintah *Revert* dapat dijalankan dengan mengklik menu **File > Revert**. Fungsinya adalah untuk mengembalikan kondisi dokumen seperti terakhir kali disimpan. Perintah ini bisa Anda gunakan apabila desain yang Anda buat terlihat jelek dan Anda ingin mengembalikan kondisi seperti semula.

Yang dimaksud seperti semula adalah kondisi ketika *file* tersebut disimpan terakhir kali.

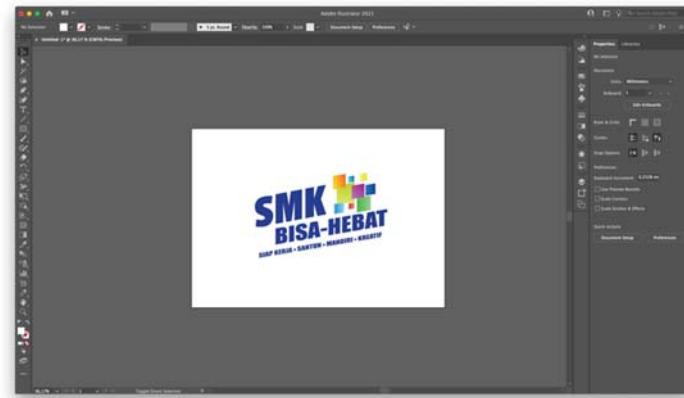
i. Memilih *Workspace*

Adobe Illustrator digunakan untuk beragam keperluan. Oleh karena itu, Anda bisa memilih *workspace* sesuai kebutuhan kerja. Jika *workspace* sudah dipilih, peletakan panel akan disesuaikan berdasarkan kebutuhan.

Cara memilih *workspace* adalah dengan mengklik menu **Window > Workspace**.



- 1) **Essential** : *workspace default* untuk berbagai keperluan. Jika ingin belajar fitur-fitur umum tentang *Adobe Illustrator*, pilihlah *workspace Essential*.



Gambar 2. 276 Tampilan *Workspace Essential Classic*

- 2) **Layout** : *workspace* yang digunakan untuk membantu Anda me-*layout* halaman. Cocok digunakan apabila Anda membuat desain, seperti majalah, brosur, *flyer*, dan sebagainya yang membutuhkan penataan objek secara presisi.
- 3) **Painting** : *workspace* yang lebih menitikberatkan pada proses melukis objek di dalam dokumen. Di sini, *Brush* mudah diakses menggunakan panel-panel yang aktif secara otomatis.
- 4) **Printing & Proofing** : *workspace* untuk membantu Anda mencetak atau mengukur kualitas cetakan di berbagai macam media.
- 5) **Tracing** : *workspace* untuk membantu Anda melakukan *tracing* pada objek. *Tracing* sendiri adalah salah satu proses mengubah objek bitmap menjadi vektor.



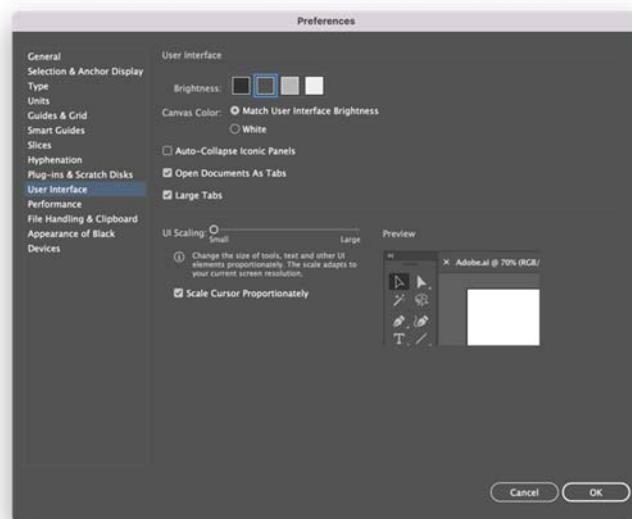
- 6) **Typography** : workspace untuk membantu Anda menata tulisan.
- 7) **Web** : *workspace* untuk membantu Anda membuat desain untuk keperluan desain *website*.



Gambar 2. 287 Tampilan *Workspace Web*

j. Mengatur *Brightness*

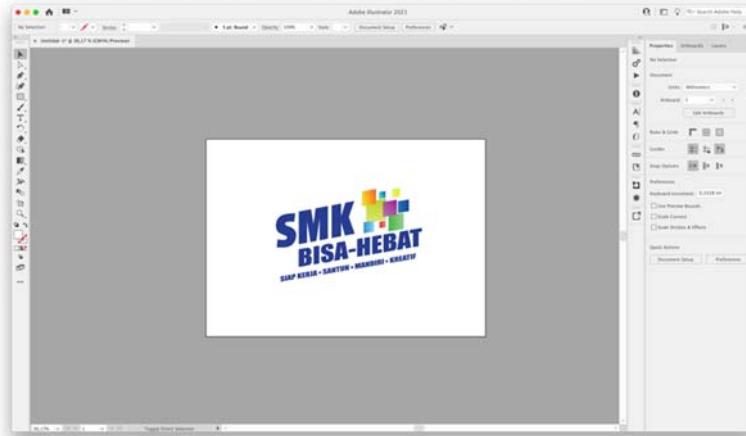
Mengatur *Brightness* *Workspace* bisa diatur terang-gelapnya. Caranya adalah dengan memilih menu **Edit > Preferences**. Lalu, pilih menu **User Interface**.



Gambar 2.29 Memilih Menu *Interface*



Lalu, pilih *Brightness* yang diinginkan mulai dari warna keabuabuan hingga gelap menjelang hitam.



Gambar 2. 30 *Adobe Illustrator Light Mode*

k. *Undo*

Undo digunakan untuk membatalkan perintah. Sebagai contoh, kalau Anda membuat lingkaran di dalam dokumen dan menjalankan perintah *Undo*, maka lingkaran itu tidak jadi dibuat.

Secara default, perintah *Undo* dapat dijalankan dengan menekan tombol **Ctrl+Z** atau Anda juga dapat menekan menu **Edit > Undo**.

l. *Redo*

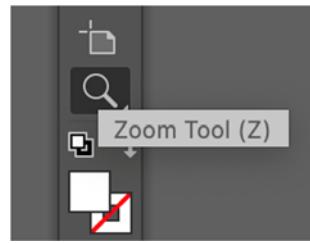
Undo sendiri adalah perintah untuk pembatalan. Nah, *Undo* dapat dibatalkan dengan menggunakan perintah lain yang diberi nama *Redo*. Jadi, ketika Anda telanjur menggunakan *Undo* dan ingin membatalkannya, kliklah menu **Edit > Redo** atau tekan tombol **Shift+Ctrl+Z**.

m. *Zoom*

Zoom pada dasarnya adalah untuk memperbesar atau mengecilkan tampilan di dalam dokumen. Secara *default*, *Zoom*



dapat digunakan setelah Anda mengaktifkan *Zoom Tool*. Berikut beberapa cara menggunakan *Zoom Tool*:

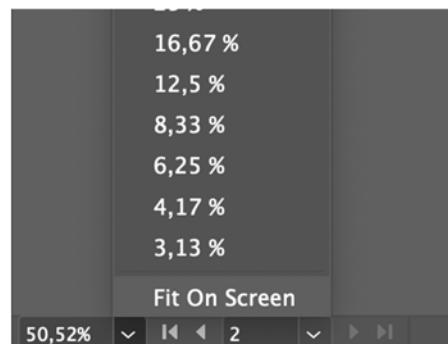


Gambar 2. 81 *Toolbar Zoom Tool*

- 1) Klik di dalam dokumen untuk memperbesar tampilan atau shortcut **Ctrl dan +**
- 2) Klik-ganda untuk membuat tampilan dokumen sebesar 100%.
- 3) Tekan *Alt+klik* untuk *zoom out* (mengecilkan tampilan dokumen) atau shortcut **Ctrl dan -**

n. Penggunaan *Zoom Level*

Zoom Level digunakan untuk dua tujuan yaitu pertama, menginformasikan level *zoom* (pembesaran) yang sedang digunakan dan kedua, melakukan *zoom* terhadap dokumen.



Gambar 2. 32 Menentukan *Zoom Level* melalui *Status Bar*

Zoom Level diletakkan di *Status bar*, tepatnya di sisi kiri bawah pada jendela *Adobe Illustrator*. Di sebelah kanan *Zoom*



Level/terdapat anak panah kecil. Klik untuk mengganti *zoom level* dengan memilih nilai baru yang dikehendaki.

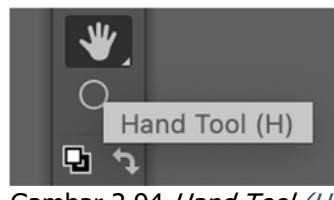
Pilihan *Fit On Screen* digunakan untuk menampilkan ukuran dokumen secara utuh di dalam jendela *Adobe Illustrator* sehingga seluruh objek di dalam dokumen menjadi terlihat.



Gambar 2. 33 *Fit On Screen* membantu melihat *artboards* secara utuh

o. *Hand Tool*

Hand Tool tak bisa dilepaskan dari *Zoom*. Jika ukuran objek melebihi kapasitas jendela *Adobe Illustrator*, maka cara untuk objek yang tertutup adalah dengan menggeser dokumen.



Gambar 2.94 *Hand Tool (H)*

Ada beragam cara untuk menggeser dokumen, salah satunya menggunakan *scrollbar* horizontal dan vertikal. Namun, untuk lebih praktisnya adalah menggunakan saja *Hand Tool*. Cara menggunakan *tool* ini cukup gampang, yaitu:



- 1) Klik *Hand Tool*
- 2) Lalu, *klik-drag* di dokumen untuk menggeser tampilan yang ada di dalamnya
- 3) Lepas tombol kiri *mouse* jika sudah

p. Beralih dari *Zoom Tool* ke *Hand Tool*

Zoom Tool dan *Hand Tool* sering digunakan bergantian. Oleh karena hal itu, ada tombol pintas untuk beralih dari *Zoom Tool* menuju *Hand Tool* secara cepat. Caranya adalah sebagai berikut :

- 1) Gunakan *Zoom Tool* untuk memperbesar atau mengecilkan tampilan di dalam dokumen
- 2) Tekan tombol *Spacebar* untuk mengubah *Zoom Tool* menjadi *Hand Tool*
- 3) *Klik-drag* untuk menggeser dokumen
- 4) Lepas tombol *Spacebar* untuk mengembalikan *Hand Tool* menjadi *Zoom Tool* kembali

q. *Actual Size*

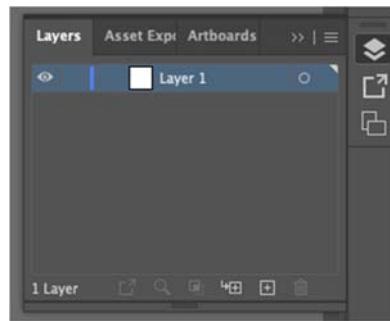
Actual Size adalah pembesaran terhadap objek mengikuti ukuran asli objek tersebut (100%). *Actual Size* dapat diaktifkan dengan menggunakan salah satu dari alternatif berikut :

- 1) Pilih 100% pada kotak isian *Zoom Level* pada *Status Bar*
- 2) Klik-ganda *Zoom Tool*
- 3) Pilih menu **View > Actual Size**
- 4) Tekan tombol shortcut **Ctrl+1**



r. Bekerja dengan *Layers*

Adobe Illustrator memiliki *layer*. Untuk mengaktifkan panel *Layers*, klik ikon *> Layers*. Berikut adalah langkah menggunakan *layer*.

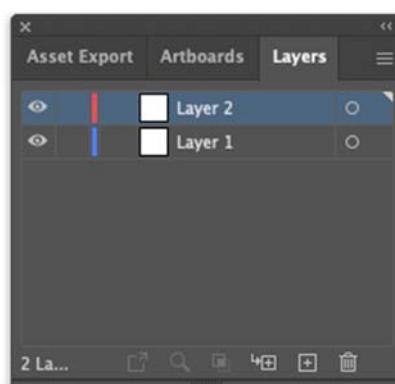


Gambar 2. 35 *Layers*

- 1) Buat objek di dalam dokumen. Secara *default*, objek ini akan diletakkan di *Layer 1*
- 2) Klik ikon *Create New Layer* untuk membuat layer baru



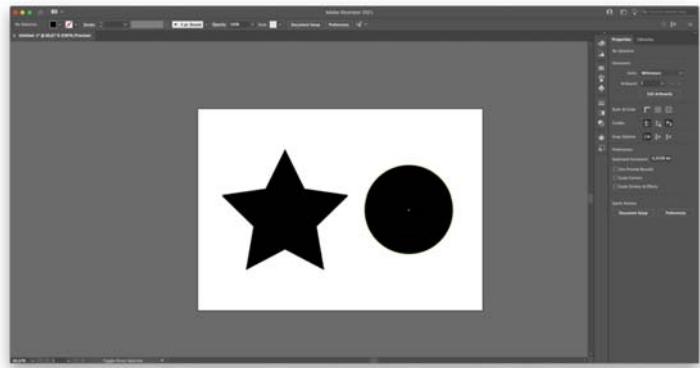
Gambar 2. 36 *Create New Layer*



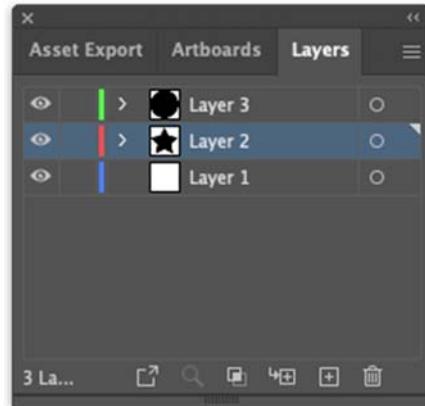
Gambar 2. 37 Tampilan saat *Layer* baru telah dibuat

- 3) Buat objek di dalam dokumen.





Gambar 2. 38 Objek - objek di dalam dokumen



Gambar 2. 39 Tampilan Objek-objek pada masing-masing *layers*

4) *Layer* dapat memudahkan kalian dalam mengelompokan objek dengan cepat. Tanpa membuat *layer* baru secara manual, objek-objek itu akan ditata di dalam satu *layer* yang sama.

s. Memilih Bentuk *Strokes*

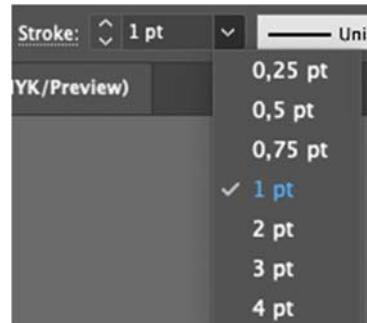
Stroke pada dasarnya adalah sebuah garis yang membentuk objek. Nah, karena merupakan sebuah garis maka *stroke* sendiri memiliki bentuk-bentuk.



Gambar 2. 40 *Toolbar Adobe Illustrator*



Stroke Weight adalah menu yang digunakan untuk mengatur ketebalan garis *stroke*. Setelah membuat atau mengklik objek, Anda bisa klik menu *Stroke Weight* di dalam *toolbar* dan pilih ukuran yang dikehendaki.



Gambar 2. 41 Memilih ukuran stroke yang diinginkan

Anda bisa memasukkan angka secara langsung ke dalam menu tersebut dan akhiri dengan tombol **Enter**.

Variable Width Profile secara umum digunakan untuk memilih bentuk ketebalan *stroke*. Untuk *stroke* yang dibuat menggunakan *Line*, Anda akan bisa langsung melihat perbedaan antara satu *profile* dengan *profile* lainnya:



Gambar 2. 42 Pilihan *width profile* untuk *stroke*

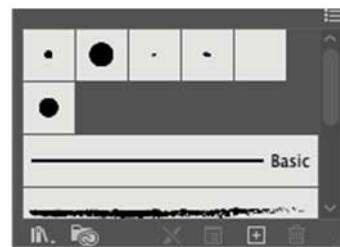




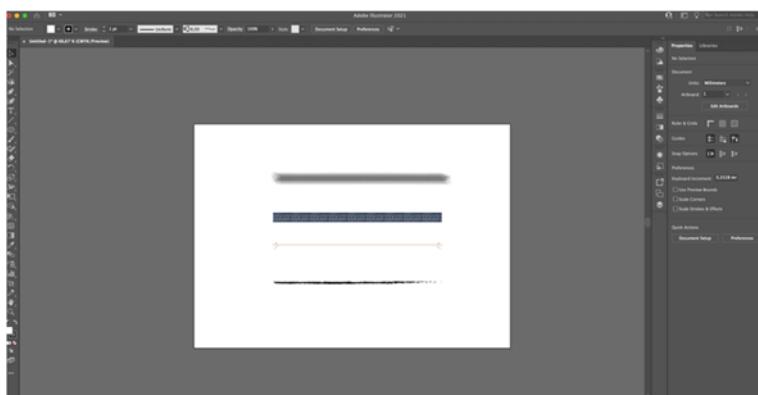
Gambar 2. 43 Objek dengan berbagai *variable width profile* di dalam dokumen

t. *Brush Definition*

Menu *Brush Definition* digunakan untuk memilih bentuk *stroke*. Anda bisa memilih *stroke* yang dibentuk dari pola lingkaran hingga bentuk-bentuk lainnya.



Gambar 2. 44 *Brush Definition Properties*



Gambar 2. 45 Berbagai objek dengan *Brush Definition* di dalam dokumen



2. Dasar – dasar membuat objek

a. *Rectangle Tools*

Rectangle Tool digunakan untuk membuat persegi empat.

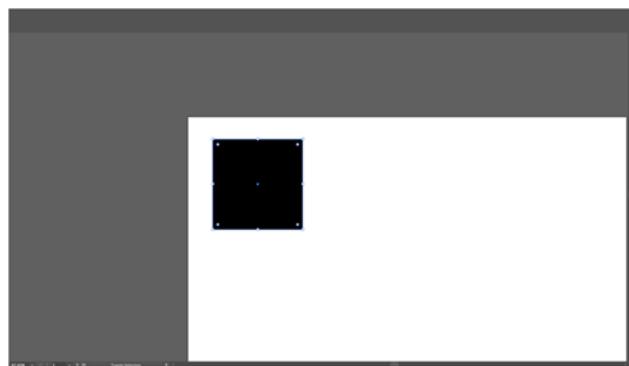
Berikut langkah-langkah menggunakannya:

1) Pilih **Rectangle Tool**



Gambar 2. 46 *Rectangle Tool (M)*

- 2) Klik-drag di dalam dokumen untuk membuat persegi empat
- 3) Pilih *Selection Tool* dan klik objek tersebut

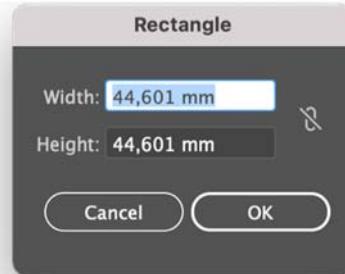


Gambar 2. 47 Objek persegi empat di dalam dokumen

- 4) Untuk membuat segi empat sama sisi, tekanlah tombol **Shift+klik+ drag**

Apabila ingin membuat persegi empat dengan ukuran presisi, kliklah di dalam dokumen. Kotak dialog akan muncul untuk membantu Anda mengisi *Width* dan *Height* persegi empat.



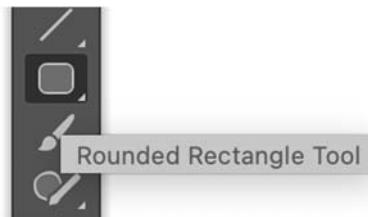


Gambar 2. 48 Membuat persegi empat secara presisi

b. *Rounded Rectangle Tools*

Rounded Rectangle Tool digunakan untuk membuat persegi empat dengan sudut tumpul. Seberapa besar tingkat ketumpulan yang diinginkan bisa diatur menggunakan berbagai cara. Berikut langkah-langkah pembuatannya:

1) Pilih **Rounded Rectangle Tool**



Gambar 2. 49 *Rounded Rectangle Tool*

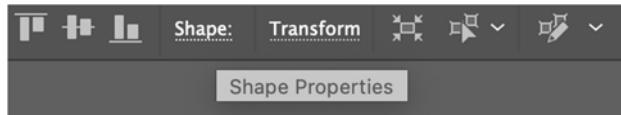
- 2) Selanjutnya, klik-drag di dalam dokumen untuk membuat *Rounded Rectangle*
- 3) Anda bisa pindahkan objek itu menggunakan *Selection Tool*



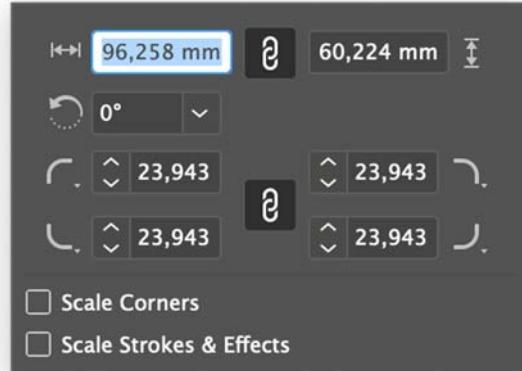
Gambar 2. 50 Objek *Rounded Rectangle* di dalam dokumen



Mengatur tumpul persegi empat baik *rectangle* maupun *rounded rectangle* dilengkapi atribut untuk melengkungkan setiap sudutnya. Caranya adalah dengan terlebih dulu mengklik *Shape* yang dapat ditemukan di dalam *toolbar*.

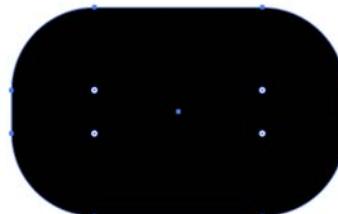


Gambar 2. 51 *Toolbar Shape*



Gambar 2. 52 *Shape Properties*

Cukup naikkan atau turunkan nilai di dalam kotak isian *Corner Radius* masing-masing sudut untuk mengatur tumpul-tajamnya persegi empat.



Gambar 2. 53 Persegi Empat dengan nilai *corner* yang tinggi

Selain itu, Anda bisa langsung *klik-drag* menggunakan *Selection Tool* pada titik *corner radius* 6 yang ada di setiap sudut persegi empat. Klik-drag ke arah dalam untuk menampilkan persegi empat dan ke arah luar untuk menjamkannya kembali.



c. *Ellipse Tools*

Ellipse Tool digunakan untuk membuat objek lingkaran. Cara penggunaannya adalah sebagai berikut:

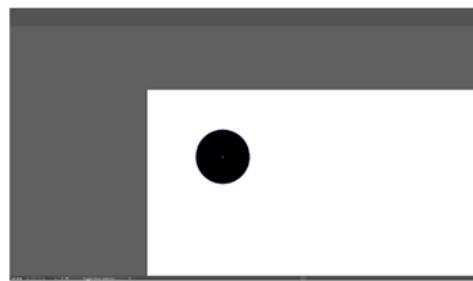
1) Pilih ***Ellipse Tool***



Gambar 2. 54 *Ellipse Tool (L)*

2) Klik-drag di dalam dokumen untuk membuatnya.

Apabila muncul tanda silang di dalam lingkaran, ini menunjukkan ukuran Width maupun Height lingkaran tersebut sama.



Gambar 2. 55 *Objek Ellipse* di dalam dokumen

3) Pilih *Selection Tool* untuk memindahkan objek itu.

Selain metode *klik-drag*, ada beberapa variasi lain untuk menggambar objek lingkaran, yaitu:

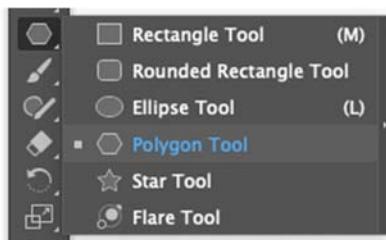
- 1) Tekan **Shift+klik-drag** untuk membuat objek lingkaran sempurna.
- 2) Tekan **Alt+klik-drag** untuk membuat objek lingkaran dari tengah lingkaran.



d. *Polygon Tool*

Polygon Tool digunakan untuk menggambar objek bersegi banyak. Langkah-langkah penggunaannya adalah sebagai berikut:

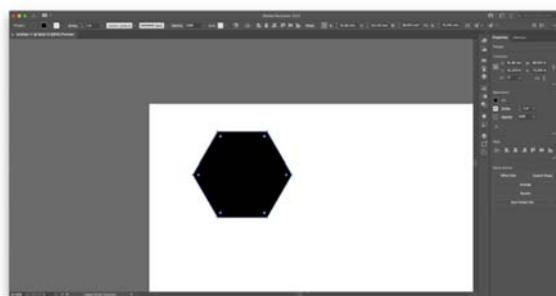
1) Pilih **Polygon Tool**



Gambar 2. 56 *Polygon Tool*

2) Klik-drag di dalam dokumen

3) Posisikan dengan menggunakan *Selection Tool*



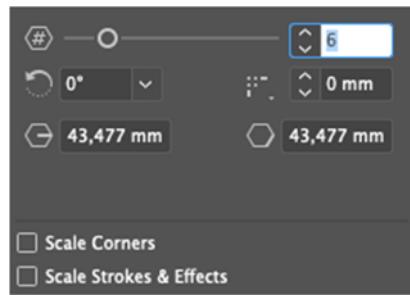
Gambar 2. 57 Objek *Polygon* di dalam dokumen

Bentuk poligon di dalam dokumen dapat diatur ulang, seperti jumlah sudut pada objek, kemiringan, tumpulnya sudut, dan sebagainya. Caranya adalah dengan mengklik *Shape*.



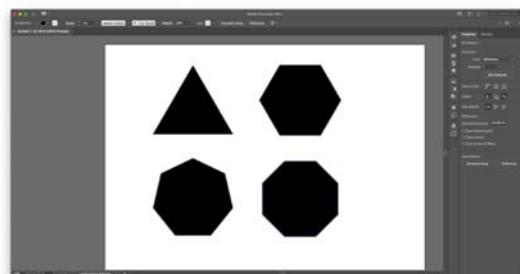
Gambar 2. 58 *Toolbar Shape*





Gambar 2. 59 *Polygon Shape Properties*

- 1) **Polygon Side Count** : mengatur jumlah sudut yang ingin dibuat
- 2) **Polygon Angle** : mengatur derajat perputaran objek poligon
- 3) **Polygon Radius** : mengatur ukuran poligon
- 4) **Corner Radius** : menumpulkan sudut poligon
- 5) **Polygon Side Length** : mengatur panjang sisi poligon



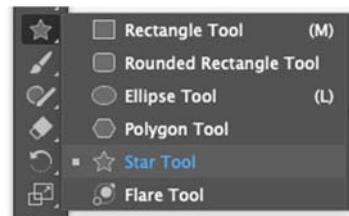
Gambar 2. 60 Objek *Polygon* dengan berbagai sisi di dalam dokumen

e. *Star Tool*

Star Tool digunakan untuk membuat objek bintang. Langkah-langkah penggunaan *Star Tool* adalah sebagai berikut:

- 1) Pilih **Star Tool**





Gambar 2. 61 *Star Tool*

- 2) Buat objek bintang dengan cara *klik-drag* di dalam dokumen
- 3) Posisikan dengan *Move Tool* jika sudah

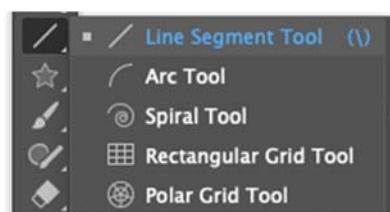


Gambar 2. 62 Objek Bintang di dalam dokumen

f. *Line Segment Tool*

Line Segment Tool digunakan untuk membuat garis di dalam dokumen. Langkah-langkah pembuatannya adalah sebagai berikut :

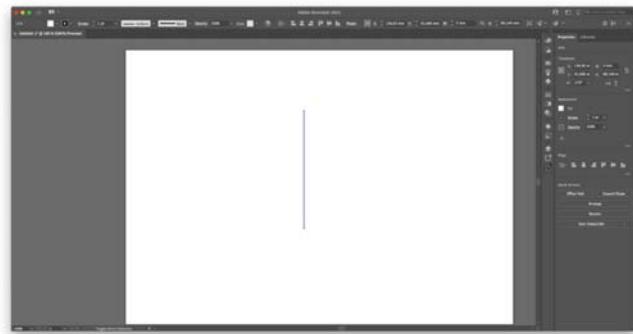
- 1) Pilih **Line Segment Tool**



Gambar 2. 63 *Line Segement Tool (//)*

- 2) *Klik-drag* di dalam dokumen untuk membuat garis
- 3) Pilih *Selection Tool* untuk memindah garis itu



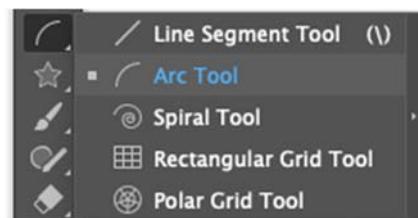


Gambar 2. 64 Objek *Line* di dalam dokumen

g. *Arc Tool*

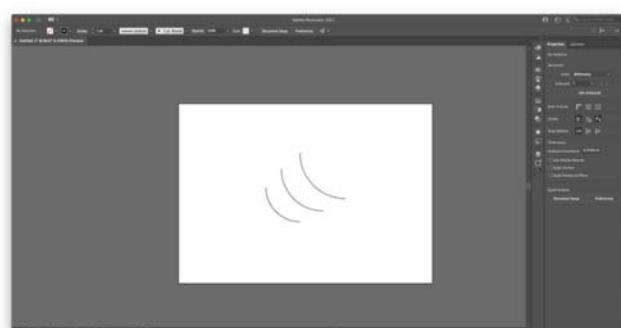
Arc Tool digunakan untuk membuat objek berbentuk garis lengkung. Langkah-langkah membuat *Arc Tool* adalah sebagai berikut :

1) Pilih **Arc Tool**



Gambar 2. 65 *Arc Tool*

- 2) Buat garis lengkung dengan cara *klik-drag* di dalam dokumen
- 3) Lepas tombol kiri *mouse*



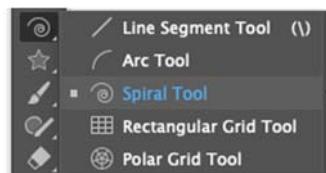
Gambar 2. 66 Objek *Arc Tool* di dalam dokumen



h. *Spiral Tool*

Spiral Tool digunakan untuk membuat objek berbentuk spiral. Langkah-langkah menggunakan *Spiral Tool* adalah sebagai berikut:

1) Pilih **Spiral Tool**



Gambar 2. 67 *Spiral Tool*

- 2) *Klik-drag* di dalam dokumen untuk membuat objek spiral
- 3) Pilih *Selection Tool* dan *klik-drag* objek ke tempat yang Anda inginkan

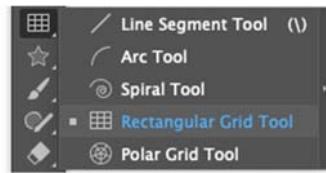


Gambar 2. 68 Objek Spiral di dalam dokumen

i. *Rectangular Grid Tool*

Rectangular Grid Tool adalah *tool* yang digunakan untuk membuat kotak-kotak bersusun (*grid*). Anda bisa memanfaatkannya untuk tabel atau kotak isian *form* pendaftaran. Berikut langkah-langkah membuat *Rectangular Grid* :

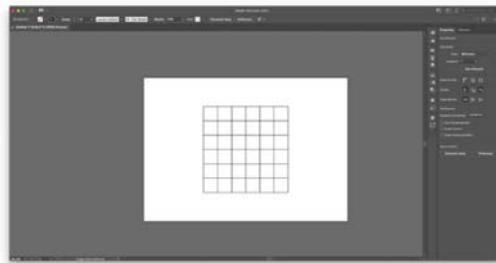
1) Pilih **Rectangular Grid Tool**



Gambar 2. 69 *Rectangular Grid Tool*



- 2) Klik-drag di dalam dokumen untuk membuat objek tersebut
- 3) Posisikan menggunakan *Selection Tool* kalau sudah

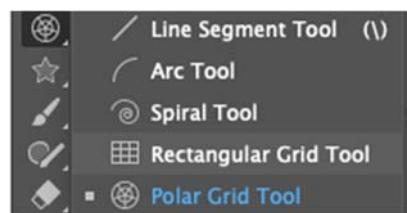


Gambar 2. 7010 Objek *Rectangular Grid Tool* di dalam dokumen

j. *Polar Grid Tool*

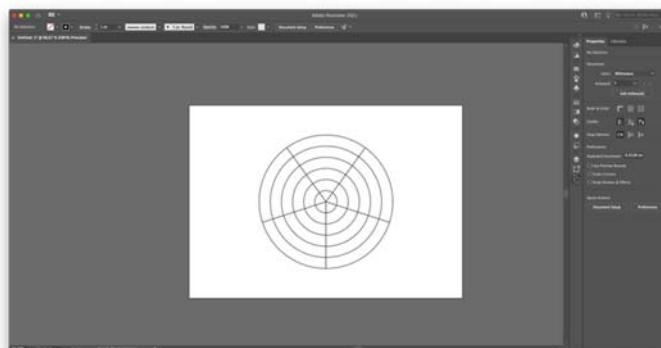
Polar Grid Tool menyerupai *Rectangular Grid Tool*. Hanya saja, bentuk yang dihasilkan adalah lingkaran. Berikut langkah-langkah menggunakannya :

1) Pilih **Polar Grid Tool**



Gambar 2. 71 *Polar Grid Tool*

2) Buat objek di dalam dokumen

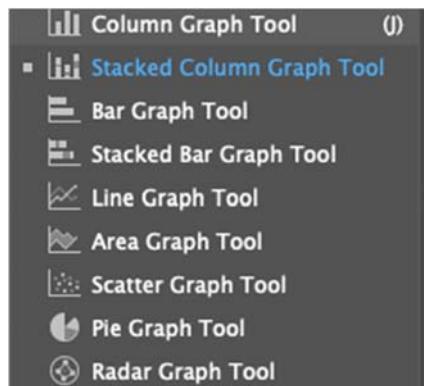


Gambar 2. 72 Buat objek



k. *Chart*

Anda bisa membuat *chart* (grafik) di dalam dokumen dengan menggunakan beragam *tool* yang telah disediakan. Berikut *tool-tool* untuk keperluan pembuatan chart:



Gambar 2. 73 Berbagai Objek *Chart* di Adobe Illustrator

- 1) **Column Graph Tool** digunakan untuk membuat *chart* berbentuk batang (*column*)
- 2) **Stacked Column Graph Tool** ni digunakan untuk membuat *chart* batang (*column*) yang bertumpuk
- 3) **Bar Graph Tool** bentuk *bar* digunakan untuk membuat *chart* berbentuk *bar*
- 4) **Stacked Bar Graph Tool** digunakan untuk membuat *chart* berbentuk *bar* yang bertumpuk
- 5) **Line Graph Tool** digunakan untuk membuat *chart* berbentuk garis (*line*)
- 6) **Area Graph Tool** digunakan untuk membuat *chart* berbentuk area (garis namun memiliki warna isian)
- 7) **Scatter Graph Tool** digunakan untuk membuat *chart* berbentuk titik
- 8) **Pie Graph Tool** digunakan untuk membuat *chart* berbentuk *pie*



9) **Radar Graph Tool** digunakan untuk membuat *chart* berbentuk radar lingkaran

Ketika Anda menggambar *chart* menggunakan *tool-tool* di atas, maka panel baru akan terbuka di mana data-data *chart* wajib Anda isi. Berdasarkan data-data tersebut, *chart* akan dibuat.

Pada umumnya, pembuatan *chart* satu dengan lainnya adalah sama. Maka di bagian ini hanya dibuat satu sampel saja untuk membantu Anda memahami cara kerja masing-masing *tool*.

Berikut langkah-langkah membuat *chart* batang menggunakan *Column Graph Tool*:

1) Pilih **Column Graph Tool**

2) *Klik-drag* di dalam dokumen untuk membuat *chart* baru

3) Isi data-data untuk membentuk *chart*. Pengisiannya menyerupai pembuatan *spreadsheet* menggunakan MS Excel. Contohnya sebagai berikut:

4) Klik Ikon

Ketika ikon tersebut diklik, maka data-data akan diolah menjadi sebuah *chart* berbentuk batang (*column*).

Anda bisa menutup panel data tersebut dengan mengklik ikon *Close*. Namun, setelah ikon *Close* ditekan dan panel menutup maka data-data yang membentuk *chart* tak bisa di-*edit* lagi. Contohnya seperti berikut :

I. Mengubah Warna *Column*

Warna *Column* bisa diubah sesuai kebutuhan atau selera.

Berikut langkah-langkah mengubah warna *column* :

1) Pilih **Direct Selection Tool**

2) Klik pada *column* yang akan diwarnai



- 3) Lalu klik menu *Fill* di dalam *toolbar* dan pilih warna yang dikehendaki

Jangan lupa untuk mewarnai kotak *legend* dengan warna yang sesuai. Tampilan *chart* akan lebih berwarna dengan kehadiran warna-warna yang variatif.

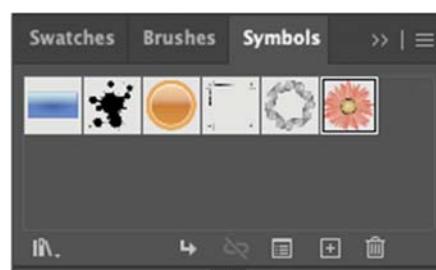
m. *Symbol Sprayer Tools*

Symbol Sprayer Tool digunakan untuk menambahkan objek-objek berbentuk simbol ke dalam dokumen. Selain *Symbol Sprayer Tool*, Anda juga bisa memanfaatkan *tool-tool* lain yang memang disediakan untuk bekerja dengan simbol. Berikut cara penggunaannya :



Gambar 2. 74 Berbagai *tool* untuk menambahkan *symbol*

1) Pilih **Symbol Sprayer Tool**

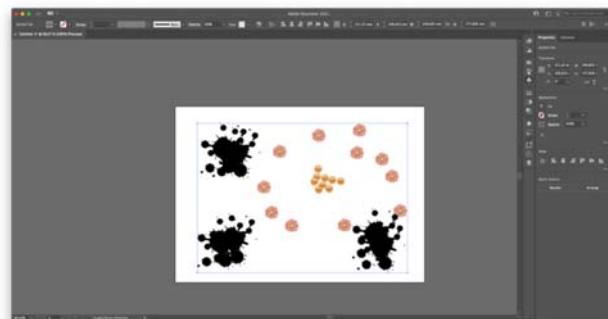


Gambar 2. 75 Panel *Symbols*

- 2) Klik ikon *Symbols* untuk membuka panel *Symbols* agar anda bisa memilih simbol yang akan ditambahkan
- 3) Lalu pilih simbol yang diinginkan

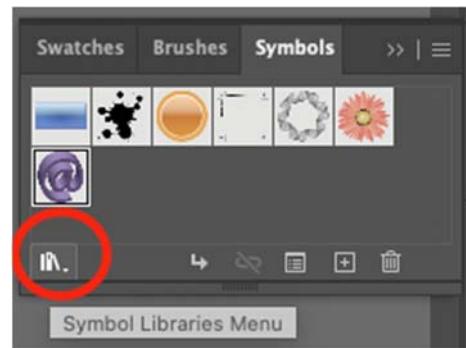


4) Arahkan kursor *mouse* ke atas dokumen dan *klik-drag* untuk membuat simbol itu



Gambar 2. 76 Objek *Symbols* di dalam dokumen

Untuk menambah koleksi simbol yang telah disediakan oleh *Adobe Illustrator*, klik ikon *Symbol Libraries Menu*.



Gambar 2. 77 *Symbol Libraries Menu*

Simbol yang sudah dibuat di dalam dokumen bisa di-*edit* dengan menggunakan *tool-tool* yang berbeda. Berikut fungsi masing-masing *tool* :

- 1) **Symbol Shifter Tool** berfungsi untuk menggeser posisi simbol.
- 2) **Symbol Scruncher Tool** berfungsi untuk menciutkan jarak simbol-simbol.
- 3) **Symbol Sizer Tool** berfungsi mengatur ukuran simbol yang sudah dibuat.



- 4) **Symbol Spinner Tool** berfungsi memutar simbol yang sudah dibuat
- 5) **Symbol Stainer Tool** berfungsi untuk menurunkan saturasi simbol
- 6) **Symbol Screener Tool** berfungsi untuk memudarkan (menurunkan nilai *opacity*) simbol.
- 7) **Symbol Styler Tool** berfungsi untuk menambahkan *style* pada simbol. Pertama Anda perlu memilih *style* pada panel *Graphic Styles* (klik ikon aktifkan panel tersebut).

3. Menggambar Objek

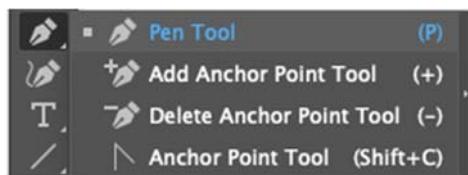
Pada materi sebelumnya, kita telah mempelajari membuat objek–objek dasar pada *Adobe Illustrator*, sekarang kita akan melanjutkan menggambar sebuah objek. Hal ini bertujuan untuk membuat objek yang sesuai dengan keinginan kita. Berikut adalah *tools-tools* yang biasa digunakan untuk menggambar objek:

a. *Pen Tool*

Pen yang digunakan untuk menggambar di lingkungan *Adobe Illustrator* adalah *Pen Tool*. Jika Anda menggunakan *CorelDRAW*, maka fungsi *tool* ini sama dengan *Bezier Tool*.

Cara menggambar menggunakan *Pen Tool* pada dasarnya seperti ini:

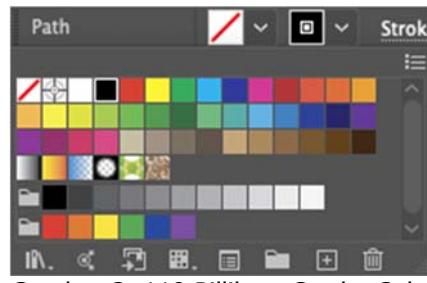
1) Klik **Pen Tool**



Gambar 2. 78 *Pen Tool (P)*

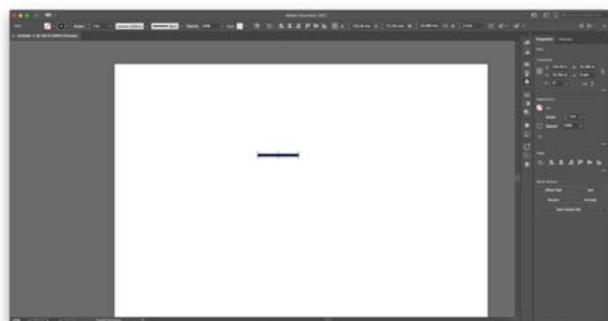
2) Pilih warna *Stroke* yang akan digunakan melalui *toolbar*





Gambar 2. 119 Pilihan *Stroke Color*

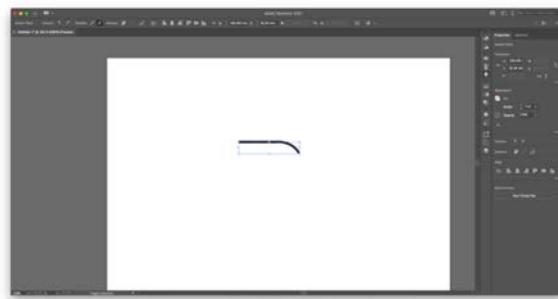
- 3) Anda juga bisa mengatur warna *Fill* apabila ingin membuat objek tertutup yang diisi dengan warna. Pilih warna *Fill* melalui *toolbar*.
- 4) Arahkan kursor *mouse* di atas dokumen
- 5) Klik untuk membuat titik baru. Lepas tombol kiri *mouse*
- 6) Klik di tempat berbeda untuk membuat titik kedua. Kedua titik akan tersambung dengan sebuah garis



Gambar 2. 80 Membuat garis lurus menggunakan *Pen Tool*

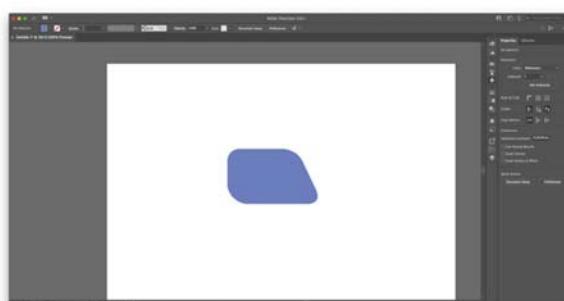
- 7) Arahkan kursor *mouse* ke tempat lain untuk membuat titik ketiga
- 8) *Klik-drag* gerakan *mouse* akan mengakibatkan bentuk garis melengkung





Gambar 2. 81 Garis lengkung dibuat dengan metode *klik-drag*

- 9) Lanjutkan dengan membuat titik-titik baru
- 10) Sambungkan titik akhir dan titik awal sehingga membentuk objek yang tertutup



Gambar 2. 82 *Asset Vector* yang dibuat dari *Pen Tool*

Objek di atas bisa dilanjutkan untuk membuat desain yang lebih bermakna. Namun sebelum itu, ada baiknya bagian-bagian yang membentuk objek di atas kita kenali istilah-istilahnya:

- 1) **Segment** : garis yang Anda buat menggunakan *Pen Tool* atau garis yang membentuk objek pada dasarnya disebut dengan istilah segment.
- 2) **Anchor** : titik-titik yang Anda buat untuk membuat segment disebut *anchor*. Lihat gambar di bawah ini. Ada empat buah *anchor*, yaitu titik pertama, titik kedua, dan titik-titik yang ada di *direction lines*.



3) **Direction lines** : garis yang digunakan untuk mengatur bentuk lengkung (kurva). *Direction lines* ini dikontrol menggunakan *anchor* yang ada di ujung-ujungnya.

b. *Direct Selection Tools*

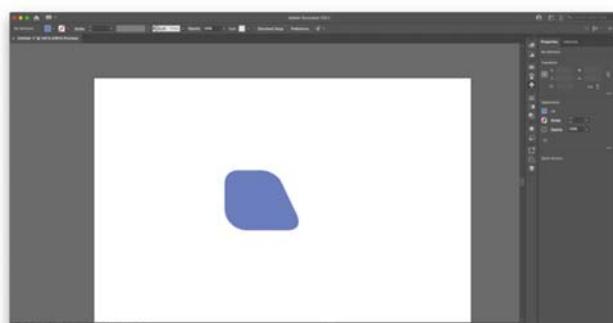
Direct Selection Tool digunakan untuk mengubah bentuk *direction lines* dengan cara mengontrol posisi *anchor*. Cara menggunakan tool ini cukup mudah.

1) Pilih **Direct Selection Tool**



Gambar 2. 83 *Direct Selection Tool (A)*

- 2) Arahkan cursor pada *anchor* yang akan digunakan untuk menggeser *direction lines*
- 3) *Klik-drag* untuk mengubah arah *direction lines* itu



Gambar 2. 84 Mengatur titik menggunakan *Direct Selection Tool*

Fungsi tombol *Alt* secara *default*, *direction lines* selalu lurus. Namun Anda bisa mematahkannya menggunakan kombinasi tombol **Alt+klik+drag** saat *Direct Selection Tool* aktif. Berikut contoh bentuk-bentuk *direction lines* yang patah.



c. *Anchor Point Tool*

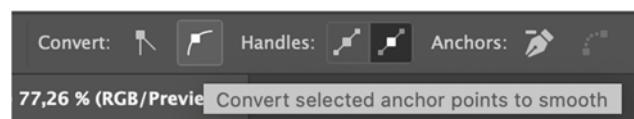
Setiap sudut lancip dibentuk oleh *anchor point*. Dengan demikian, persegi empat di bawah ini terbentuk dari empat *anchor point*.

Anda bisa mengubah sudut lancip itu menjadi kurva, dan sebaliknya, dengan memanfaatkan ikon-ikon yang ada di dalam *toolbar*.

Ikon digunakan untuk mengubah sudut tajam menjadi tumpul (kurva). Sebagai contoh, lihat perbedaan sudut-sudut persegi empat di bawah ini :

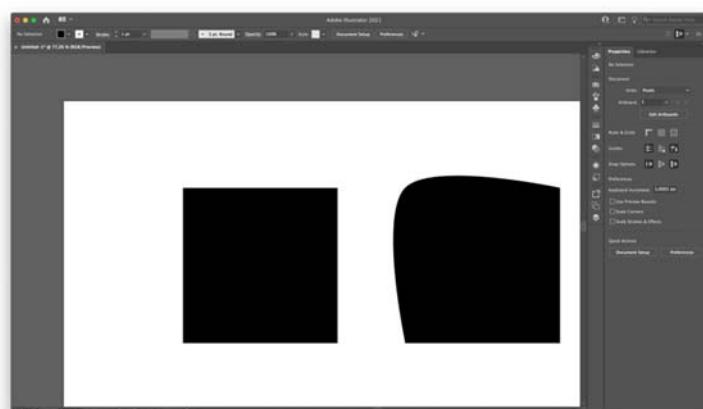
1) Pilih **Direct Selection Tool**

2) **Convert selected anchor points to smooth** yang akan diubah menjadi sudut lengkung pada menu *toolbar* di bagian atas



Gambar 2. 85 *Convert selected anchor points to smooth*

3) Tekan ikon

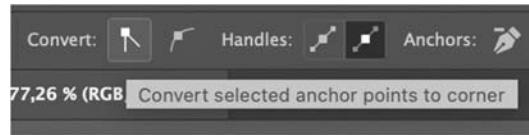


Gambar 2. 86 Kanan adalah objek setelah di *convert selected anchor points to smooth*



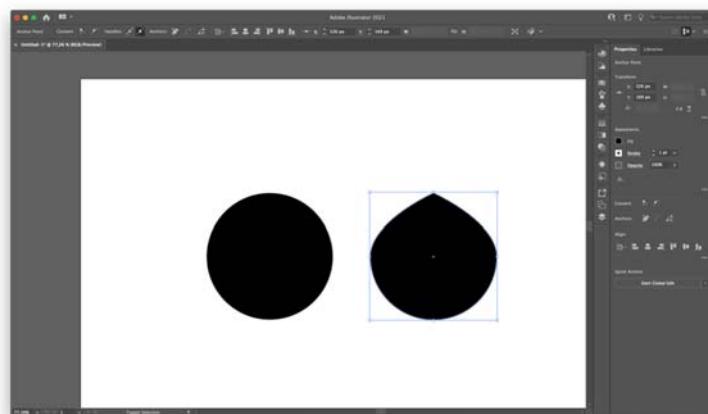
Ikon digunakan untuk mengubah sudut lengkung menjadi tajam. Lihat perbedaan bentuk lingkaran di bawah ini.

- 1) Pilih **Direct Selection Tool** atau dengan *shortcut a*
- 2) Klik *anchor* yang akan dilancipkan



Gambar 2. 87 *Convert selected anchor points to corner*

- 3) Tekan ikon



Gambar 2. 88 Kanan adalah objek setelah di *convert anchor point to corner*

Ikon **Direct Selection Tool** digunakan untuk memotong sudut yang dihasilkan *anchor* sehingga menjadi segment yang terpisah. Lihat bentuk persegi empat di bawah ini yang salah satu segment (garis) terlepas. Cara menggunakannya sebagai berikut:

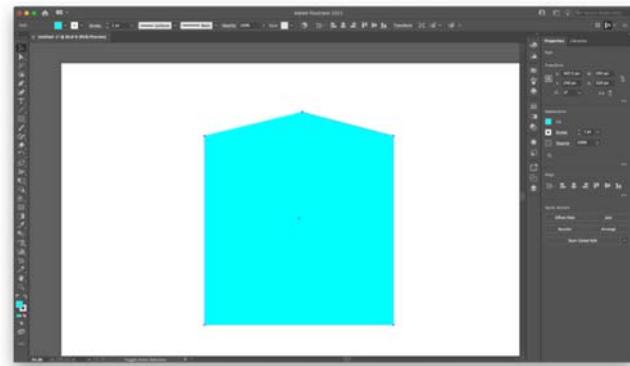
- 1) Klik **Direct Selection Tool**
- 2) Klik pada *anchor* yang akan dipotong



Gambar 2. 89 *Cut path at selected anchor points*

- 3) Lalu klik ikon segmen dan *klik-drag* untuk memisahkan





Gambar 2. 90 Hasil *object cut path at selected anchor points* di dalam dokumen

Add Anchor Point

Add Anchor Point Tool digunakan untuk menambah titik *anchor* baru. Sebagai contoh, satu segmen terbentuk dari dua buah *anchor*, yaitu titik awal dan titik akhir. Dengan menggunakan *Add Anchor Point Tool*, Anda bisa menambah titik *anchor* baru di sepanjang segmen tersebut.

- 1) Buat segmen dengan memanfaatkan *Pen Tool*



Gambar 2. 121 *Pen Tool (P)*

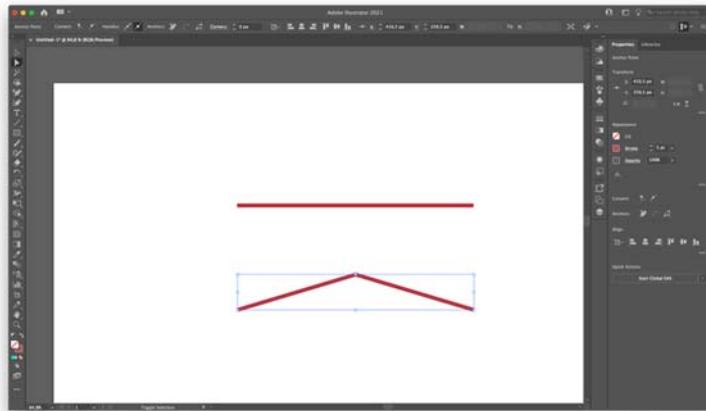
- 2) Lalu pilih *Add Anchor Point Tool*





Gambar 2. 132 Add Anchor Point Tool (+)

- 3) Arahkan kursor *mouse* di salah satu segmen
- 4) Klik sehingga muncul titik baru di segment tersebut

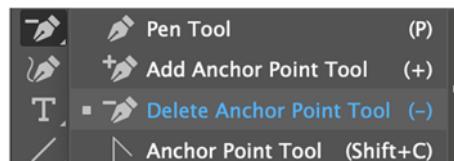


Gambar 2. 93 Titik baru berhasil ditambahkan kepada garis pada objek bagian bawah

Delete Anchor Point

Delete Anchor Point Tool digunakan untuk menghapus *anchor*. Karena *anchor* membentuk sudut, maka penghapusan *anchor* akan secara otomatis mengubah bentuk objek.

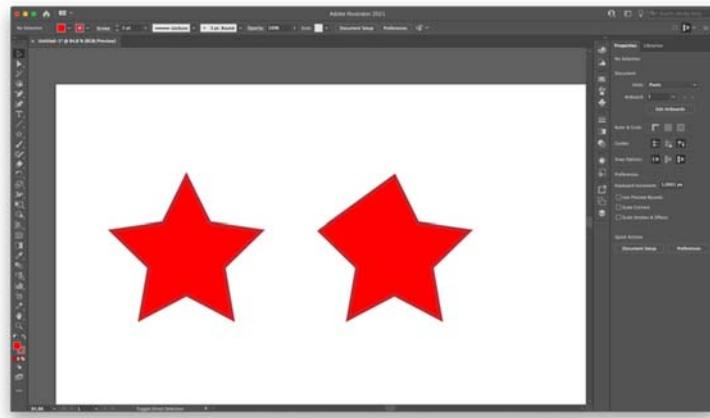
- 1) Pilih **Delete Anchor Point Tool**



Gambar 2. 144 Delete Anchor Point Tool (-)

- 2) Arahkan kursor *mouse* pada *anchor* yang akan dihapus





Gambar 2. 155 Contoh objek yang telah dihapus titiknya ada di sebelah kanan

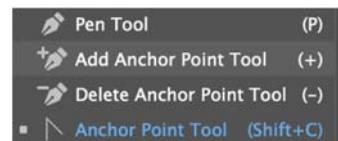
- 3) Klik dan *anchor* itu akan dihapus diikuti dengan bentuk objek yang berubah.

Jika anchor keburu terhapus dan ingin membatalkannya, cukup tekan **Ctrl+Z**.

Anchor Point

Anchor Point Tool digunakan untuk mengubah sudut tajam menjadi tumpul dengan kontrol yang lebih luwes terhadap bentuk sudut tersebut. Berikut langkah-langkahnya :

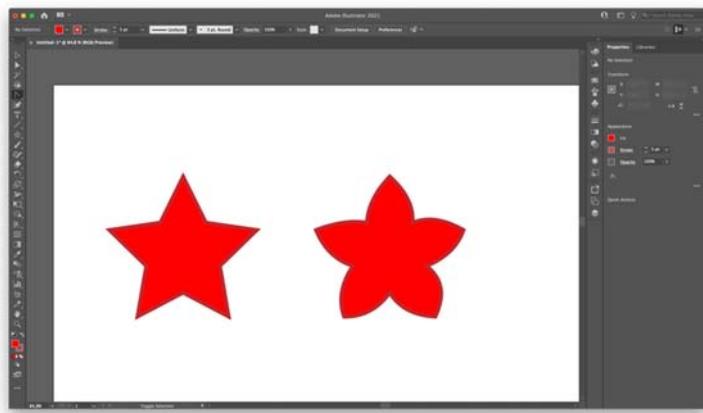
- 1) Pilih **Anchor Point Tool**



Gambar 2. 166 *Anchor Point Tool* (**Shift+C**)

- 2) Arahkan pada *anchor* yang akan diubah menjadi sudut tumpul
- 3) *Klik-drag* dan atur gerak *mouse* untuk membentuk sudut tersebut





Gambar 2. 177 Sebelah kanan adalah objek yang sudah ditambahkan *Anchor Point*

d. *Curvature Tool*

Curvature Tool digunakan khusus untuk membuat garis lengkung. Pembuatannya diawali dengan segmen (garis) lurus dan kemudian Anda dapat membelokkan garis itu menjadi lengkung. Berikut langkah-langkah menggunakan *Curvature Tool*:

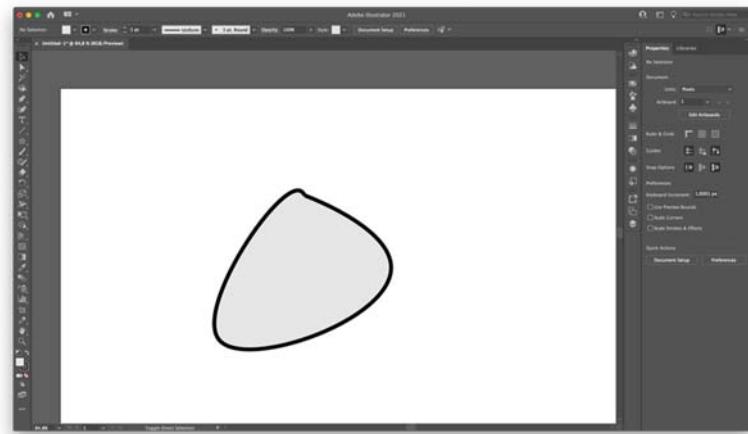
1) Pilih **Curvature Tool**



Gambar 2. 188 *Curvature Tool (Shift + ')*

- 2) Klik untuk membuat *anchor* pertama
- 3) Arahkan cursor *mouse* ke tempat lain untuk membuat *anchor* kedua
- 4) Klik dan lepas tombol kiri *mouse*
- 5) *Drag* sehingga membentuk lengkungan





Gambar 2. 199 *Object Curvature* di dalam dokumen

e. *Eraser Tool*

Eraser Tool digunakan untuk menghapus objek. Segment yang dilalui oleh *Eraser Tool* saat Anda lakukan penghapusan, akan ditandai dengan *anchor* baru. Contohnya adalah sebagai berikut:

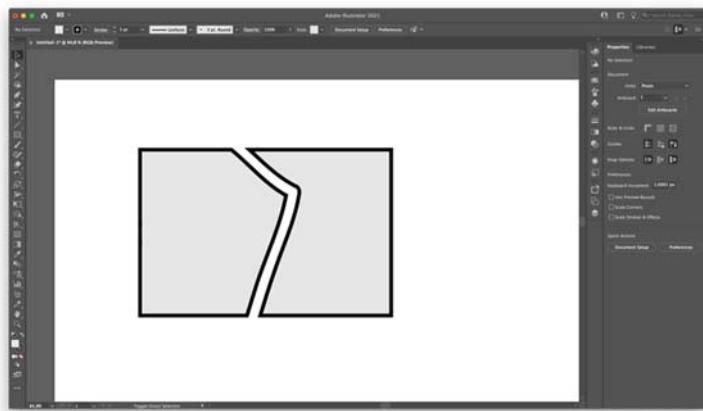
- 1) Siapkan objek yang akan dihapus. Misalnya objek berbentuk lingkaran seperti gambar
- 2) Pilih *Eraser Tool*



Gambar 2. 100 *Eraser Tool (Shift + E)*

- 3) *Klik-drag* pada bagian objek yang akan dihapus. Lihat gambar di bawah ini





Gambar 2. 201 Objek yang telah dihapus bagian tengahnya oleh *eraser tool*

Bagian objek yang dilewati *Eraser Tool* akan dibentuk oleh *anchor*. Dengan menggunakan *Direct Selection Tool*, posisi *anchor* itu bisa digeser atau dibentuk ulang.

f. *Scissors Tool*

Scissors Tool digunakan untuk memotong *segment* (garis) atau *anchor*. Kita pernah mempelajari cara memotong *anchor* ketika menggunakan *Direct Selection Tool*. *Anchor* yang terpotong bisa diubah menjadi segmen yang terpisah dengan *anchor* pada segmen lain.

Sedangkan pada *segment* (garis), *Scissors Tool* bisa dipakai untuk memotong garis itu sehingga menjadi dua garis terpisah. Berikut langkah-langkah penggunaannya:

- 1) Pastikan sudah ada segmen atau *anchor* yang akan dipotong.
- 2) Pilih *Scissors Tool*





Gambar 2. 212 Scissors Tool (C)

- 3) Klik pada segmen atau *anchor*
- 4) Anda bisa memindah segmen atau *anchor* yang terpotong itu menggunakan *Direct Selection Tool*



Gambar 2. 223 Perbandingan objek setelah dilakukan *scissors tool*

g. *Knife Tool*

Knife Tool digunakan untuk memotong objek vektor agar potongan itu bisa berdiri sebagai objek baru. Berikut langkah-langkah menggunakan *Knife Tool*:

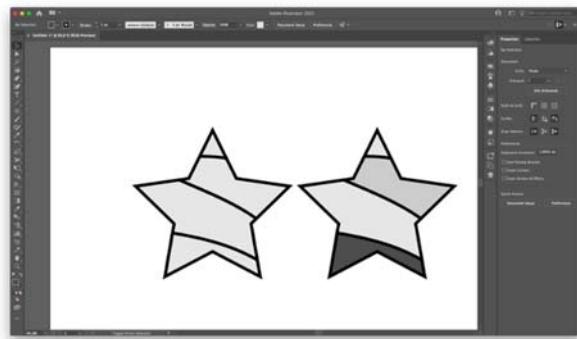
1. Siapkan objek yang akan dipotong.
- 2) Pilih *Knife Tool*



Gambar 2. 234 Knife Tool

- 3) *Klik-drag* membentuk potongan objek yang Anda inginkan.
- 4) Pilih *Selection Tool* dan klik-ganda objek hasil potongan tersebut
- 5) *Klik-drag* untuk melepas objek hasil potongan itu



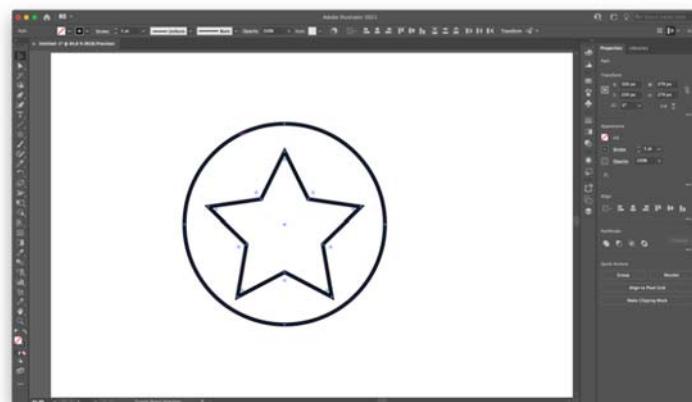


Gambar 2. 245 Hasil Objek yang dilakukan *Knife Tool*

h. *Devide Object Below*

Divide Objects Below adalah perintah yang digunakan untuk memotong objek menggunakan objek lain yang ada di atasnya. Cara menggunakannya adalah sebagai berikut:

1. Buat objek, misalnya berbentuk lingkaran
- 2) Buat objek bintang dan letakkan di atas lingkaran



Gambar 2. 256 Objek Lingkaran dan Bintang

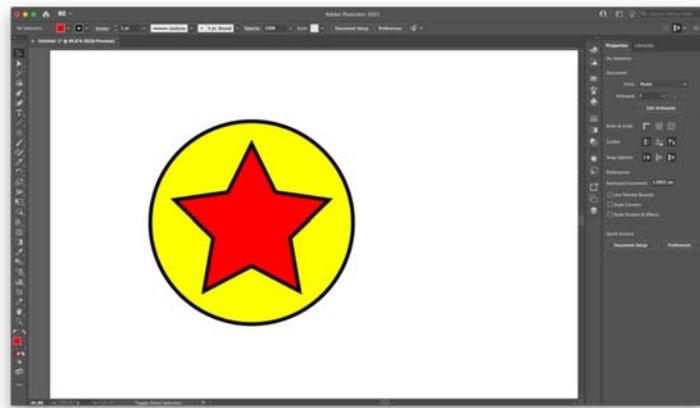
- 3) Pilih *Selection Tool* dan klik objek bintang
- 4) Lanjutkan dengan memilih menu **Object > Path > Divide**





Gambar 2. 267 *Devide Objects Below*

- 5) Masih dengan menggunakan *Shape Tool*, geser objek bintang ke luar lingkaran
- 6) Klik lagi objek lingkaran dan berilah warna atau *Graphic Style*. Hasil yang diperoleh kira-kira akan menjadi seperti ini



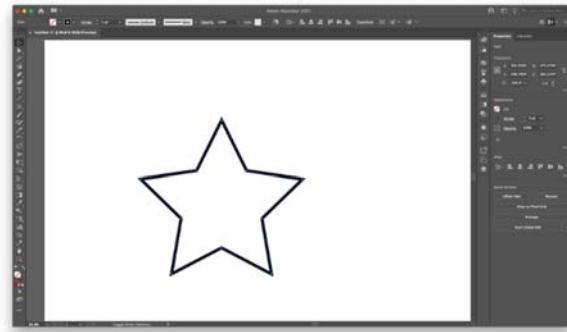
Gambar 2. 278 Hasil *Devide Objects Below*

i. *Reshape Tool*

Reshape Tool digunakan untuk mengubah bentuk objek dengan cara menarik satu *anchor* ke arah yang diinginkan. Bentuk objek akan berubah seperti karet mengikuti arah *mouse* yang Anda gerakkan. Berikut langkah-langkahnya:

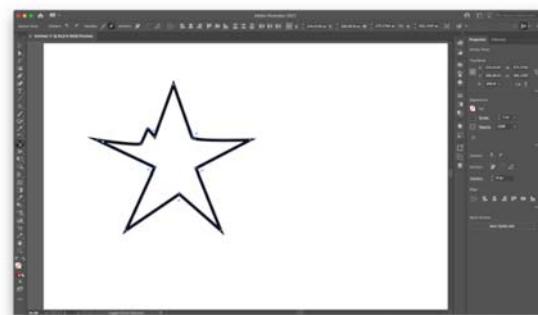
- 1) Siapkan objek yang akan dibentuk ulang.





Gambar 2. 289 Objek yang akan di *reshape*

- 2) Pilih *Direct Selection Tool* dan klik pada objek agar muncul seluruh *anchor* yang membentuk objek tersebut
- 3) Klik pada *anchor* yang akan di-*edit*
- 4) Lalu pilih *Reshape Tool*
- 5) *Klik-drag* titik *anchor* yang akan diubah
- 6) Lakukan terhadap titik-titik *anchor* lainnya



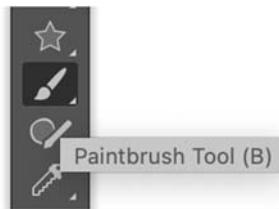
Gambar 2. 2910 Hasil Objek yang telah di *reshape*

j. *Paintbrush Tool*

Paintbrush Tool digunakan untuk melukis menggunakan prinsip kerja kuas. Jadi, Anda harus mahir menggerakkan *mouse* atau alat gambar agar objek tampak menarik. Berikut langkah-langkah menggunakan *Paintbrush Tool*:

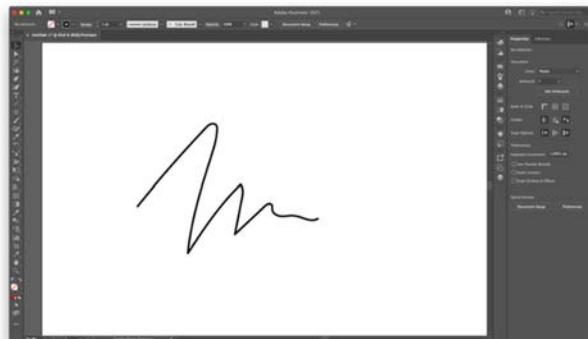
- 1) Pilih *Paintbrush Tool*





Gambar 2. 301 *Paintbrush Tool*

- 2) Atur warna *Stroke* yang dikehendaki
- 3) Mulailah menggambar menggunakan *klik-drag*

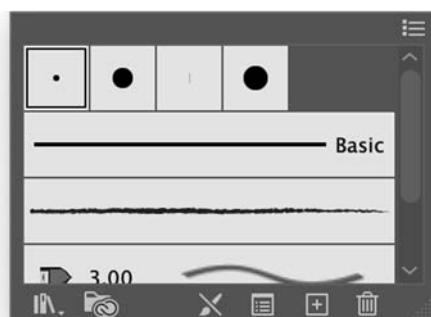


Gambar 2. 312 Hasil Objek menggunakan paintbrush tool

Untuk membuat gambar, Anda bisa mengatur ketebalan garis dengan menggunakan menu *Stroke* di dalam *toolbar*. Atau, cukup gunakan tombol [dan] untuk pengaturan melalui tombol *keyboard*.

Brush Definition

Brush Definition mengatur bagaimana bentuk sapuan kuas yang akan digunakan. Menu ini dapat Anda gunakan di *toolbar*.

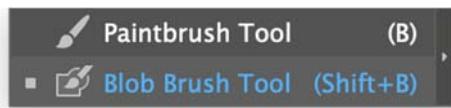


Gambar 2. 323 *Brush Definition*



Namun, pengaturan *Brush Definition* yang lebih lengkap dapat ditemukan di dalam panel *Brush Definition* dengan mengklik ikon. Anda juga bisa membukanya melalui menu **Window** **Brushes > Brushes**.

k. *Blob Brush Tool*



Gambar 2. 334 *Blob Brush Tool (Shift+B)*

Blob Brush Tool juga dimanfaatkan untuk menggambar menggunakan prinsip kerja kuas. Namun, yang membedakan dengan *Paintbrush Tool* adalah bagaimana objek sapuan kuas itu dibuat. Meskipun Anda menggambar menggunakan sapuan kuas, *Illustrator* tetap menganggapnya sebagai sebuah garis vektor. Jika Anda menggunakan *Paintbrush Tool*, garis vektor itu ada di dalam sapuan kuas. Sedangkan jika menggunakan *Blob Brush Tool*, garis vektor ada di tepi luar sapuan kuas.



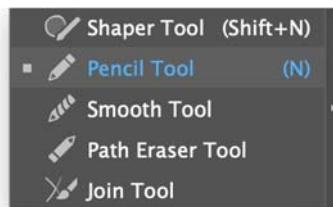
Gambar 2. 345 Objek Hasil *Blob Brush Tool*



I. *Pencil Tool*

Pencil Tool digunakan untuk menggambar objek dengan meniru prinsip sebuah pensil. Berikut langkah-langkah menggunakan *Pencil Tool*:

1) Pilih *Pencil Tool*



Gambar 2. 356 *Pencil Tool (N)*

- 2) Atur warna *Stroke* yang akan digunakan.
- 3) Buat gambar dengan menggunakan *klik-drag*

Ukuran *Pencil Tool* juga diatur melalui panel *Stroke*



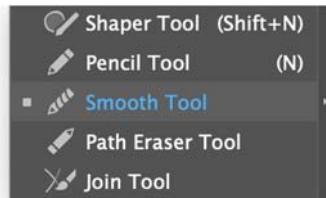
Gambar 2. 367 Objek Hasil *Pencil Tool*

m. *Smooth Tool*

Smooth Tool ideal jika dimanfaatkan bersama-sama dengan *Pencil Tool*. Setiap Anda menggambar menggunakan *tool* tersebut dan hasilnya kurang memuaskan, Anda bisa mengoreksi posisi anchor pada garis vektor objek. Caranya adalah dengan membuat garis yang menyambung satu titik dengan titik lainnya. Contohnya sebagai berikut:



- 1) Pastikan objek yang akan dihaluskan sudah ada.
- 2) Pilih *Smooth Tools*



Gambar 2. 378 *Smooth Tool*

- 3) *Klik-drag* membuat garis untuk menyambung satu titik dengan lainnya.

Lakukan berulang-ulang sampai bentuk objek berubah sesuai keinginan.



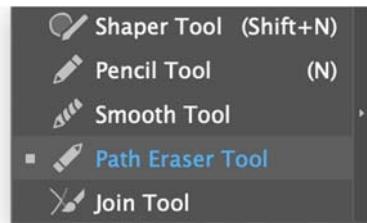
Gambar 2. 389 Perbedaan Hasil Objek dengan *smooth tool*

n. *Path Eraser Tool*

Path Eraser Tool digunakan untuk menghapus *path* untuk objek yang dibuat menggunakan *Pencil Tool*. Cukup buat garis yang menghubungkan beberapa *anchor* untuk menghapus *path* yang dibentuk oleh *anchor* tersebut. Lakukan langkah-langkah sebagai berikut:

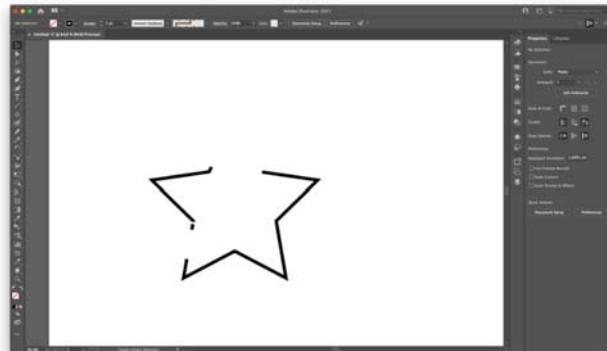
- 1) Klik *Path Eraser Tool*





Gambar 2. 3920 *Path Eraser Tool*

- 2) Buat garis yang menyambungkan satu *anchor* dengan *anchor* lainnya.



Gambar 2. 401 Hasil Objek yang dihapus menggunakan *Path Eraser Tool*

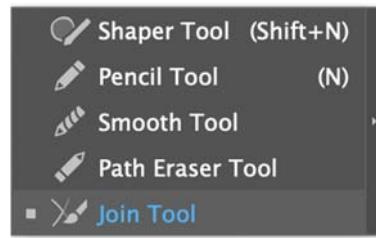
o. *Join Tool*

Join Tool digunakan untuk menyambung *anchor* yang tidak presisi (tidak pas) saat Anda menggambarnya menggunakan *Pencil Tool*. Lihat objek di bawah ini:

Untuk menyempurnakan bentuk anak panah dengan cara mengoreksi garis-garis yang berpotongan, maka Anda bisa memanfaatkan *Join Tool*. Berikut langkah-langkahnya:

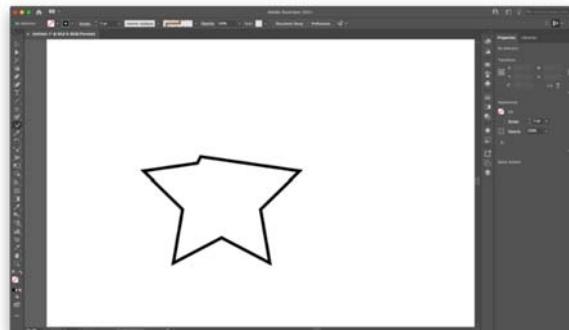
- 1) Objek yang akan dikoreksi tidak perlu diseleksi terlebih dulu.
- 2) Pilih *Join Tool*





Gambar 2. 412 *Join Tool*

- 3) *Klik-drag* pada garis yang bersinggungan. Lihat hasilnya seperti di bawah ini

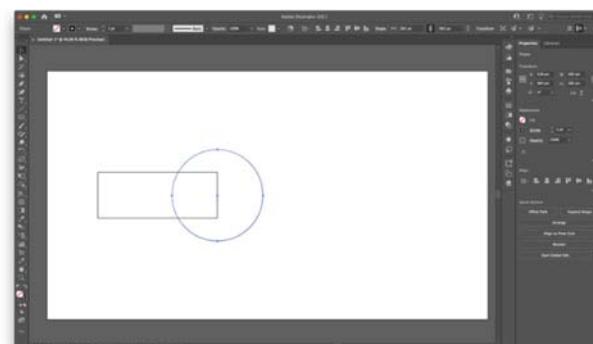


Gambar 2. 423 Hasil Objek setelah di *join tool*

p. *Shape Builder Tool*

Shape Builder Tool digunakan untuk membentuk objek baru dari beberapa objek yang bersinggungan. Berikut contoh penggunaannya:

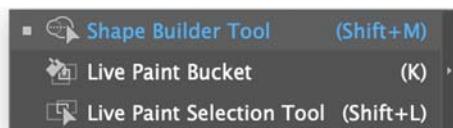
- 1) Buat dua objek yang bersinggungan seperti contoh di bawah ini



Gambar 2. 434 Contoh dua objek yang bersinggungan

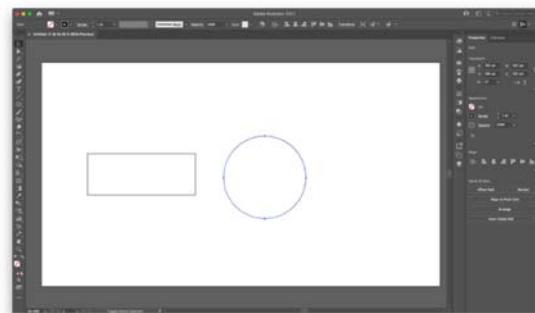


- h. Pilih *Direct Selection Tool* dan seleksi seluruh objek
- i. Pilih *Shape Builder Tool*

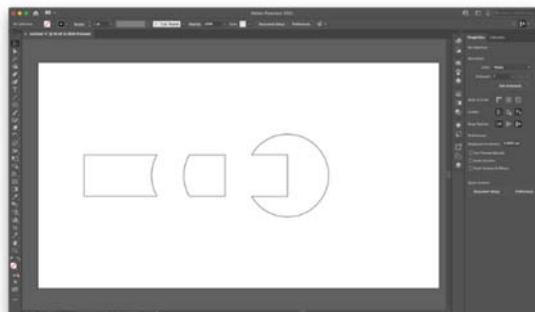


Gambar 2. 445 *Shape Builder Tool (Shift + M)*

- j. *Klik-drag* area yang berada di dalam salah satu objek ke arah area di objek lain sehingga membentuk area terarsir



Gambar 2. 456 Objek dipisah sebelum menggunakan *Shape Builder*



Gambar 2. 467 Objek dipisah setelah menggunakan *Shape Builder*

4. *Editing Objek*

Pada materi kali ini kalian akan mempelajari *tools-tools* dasar yang digunakan dalam meng-*edit* objek sehingga objek yang telah dibuat sebelumnya dapat diubah kembali.

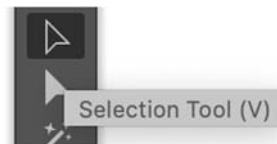


a. Menyeleksi Objek

Sebelum meng-*edit*, objek yang akan diedit harus diseleksi terlebih dulu. Ini penting agar *tool* maupun perintah tahu objek mana yang akan di-*edit*. Lalu objek bisa diseleksi dengan menggunakan beberapa cara yaitu :

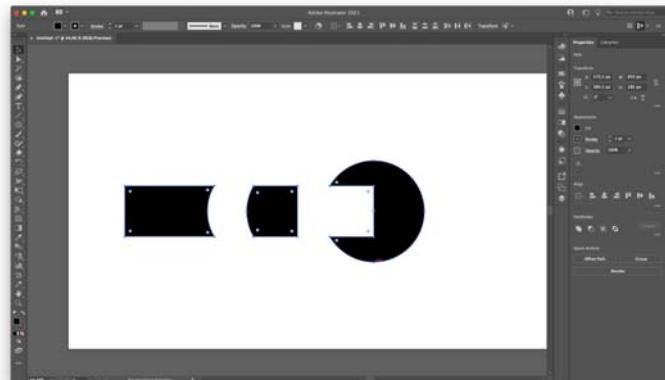
1) *Selection Tool* lakukan salah satu dari berbagai metode seleksi berikut :

(a) Klik *Selection Tool* untuk menyeleksi satu objek



Gambar 2. 478 *Selection Tool*

(b) Tekan *Shift+klik* untuk menyeleksi lebih dari satu objek



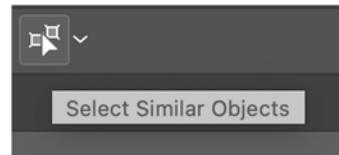
Gambar 2. 489 Tampilan saat berbagai objek di seleksi

(c) Buat area seleksi dengan cara *klik-drag* untuk menyeleksi banyak objek secara praktis

Objek terseleksi akan ditandai dengan *bounding box*, yaitu garis berbentuk segi empat yang mengelilingi objek terseleksi tersebut.



2) *Select Similar Objects*



Gambar 2. 130 *Select Similar Objects* pada *Toolbar*

Select Similar Objects adalah perintah yang dapat Anda temukan di dalam *toolbar* pada saat *Selection Tool* aktif.

Pada intinya, perintah ini digunakan untuk menyeleksi objek berdasarkan kesamaan mirip. Jadi pilih satu objek dan ketika perintah tersebut digunakan maka objek yang memiliki kemiripan akan terseleksi.

Kemiripan itu bisa diatur dengan mengklik anak panah yang ada di samping kanan ikon *Select Similar Objects*, yaitu:

- (a) **All** : menyeleksi objek yang memiliki seluruh kemiripan dengan objek yang terseleksi
- (b) **Fill Color** : hanya menyeleksi objek yang memiliki kemiripan warna isian
- (c) **Stroke Color** : hanya menyeleksi objek yang memiliki kemiripan warna *stroke*
- (d) **Fill & Color Stroke** : hanya menyeleksi objek yang memiliki kemiripan warna isian dan *stroke*
- (e) **Stroke Weight** : hanya menyeleksi objek yang memiliki kemiripan ketebalan garis
- (f) **Opacity** : hanya menyeleksi objek yang memiliki kemiripan *opacity*



(g) **Graphic Style** : hanya menyeleksi objek yang memiliki kemiripan penggunaan *graphic style*

(h) **Appearance** : hanya menyeleksi objek yang sama-sama terlihat di dalam dokumen

(i) **Appearance Attribute** : hanya menyeleksi objek yang memiliki kemiripan atribut

(j) **Shape** : hanya menyeleksi objek yang memiliki kemiripan bentuk (*shape*)

3) *Lasso Tools*

Lasso Tool digunakan untuk menyeleksi beberapa objek sekaligus dengan prinsip kerja menyerupai tali *lasso*. Jadi jika posisi objek-objek itu tidak teratur, *Lasso Tool* bisa dimanfaatkan untuk menyeleksinya. Berikut langkah-langkah menggunakan *Lasso Tool* :

(a) Pilih **Lasso Tools**

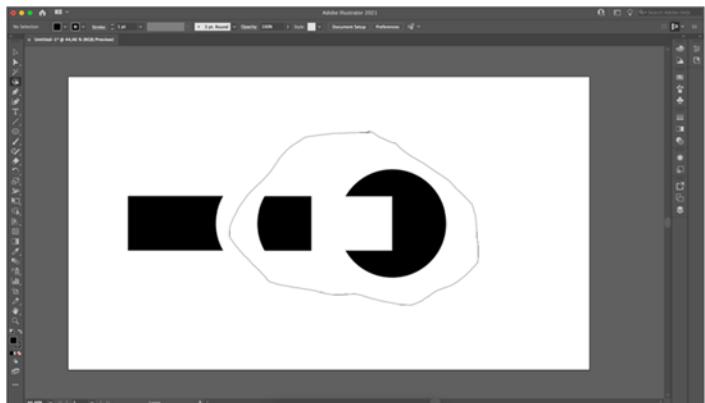


Gambar 2. 491 *Lasso Tool (Q)*

(b) *Klik-drag* membentuk area seperti tali *lasso* mengitari objek objek yang akan diseleksi.

(c) Pertemukan titik awal dan titik akhir untuk mengakhiri penggunaan *Lasso Tool*. Objek-objek pun akan terseleksi.

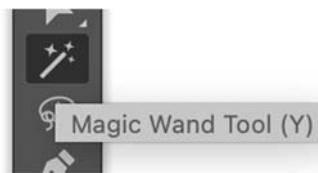




Gambar 2. 502 Tampilan saat menyeleksi menggunakan *lasso tools*

4) *Magic Wand Tool*

Lasso Tool digunakan untuk menyeleksi objek berdasarkan kesamaan warna. Jadi jika ada lima buah objek dengan warna merah, klik salah satu objek menggunakan *Magic Wand Tool*. Dalam sekejap, seluruh objek dengan warna sama akan terseleksi.



Gambar 2. 513 *Magic Wand Tool (Y)*

5) Menu *Select*

Seleksi dengan cara lain dapat dilakukan dengan memanfaatkan menu *Select*. Di dalam menu ini terdapat banyak perintah untuk menyeleksi objek.





Gambar 2. 524 Pilihan Menu *Select*

Berikut perintah-perintah di dalam menu *Select* yang dapat digunakan:

- (a) **All** : digunakan untuk menyeleksi seluruh objek
- (b) **All on Active Artboard** : untuk menyeleksi seluruh objek yang ada di dalam semua *artboard*
- (c) **Deselect** : untuk menonaktifkan seluruh seleksi
- (d) **Reselect** : untuk menyeleksi objek yang terakhir kali diseleksi
- (e) **Inverse** : untuk membalik seleksi sehingga objek terseleksi menjadi tak terseleksi, dan sebaliknya
- (f) **Next Object Above** : menyeleksi objek urutan berikut yang ada di atas objek terseleksi
- (g) **Next Object Below** : menyeleksi objek urutan berikut yang ada di bawah objek terseleksi



(h) **Same** : memilih objek yang memiliki kesamaan karakteristik. Sama seperti penggunaan perintah *Select Similar Objects*.

(i) **Object** : menyeleksi objek yang memiliki kondisi tertentu

(j) **Save Selection** : menyimpan seleksi

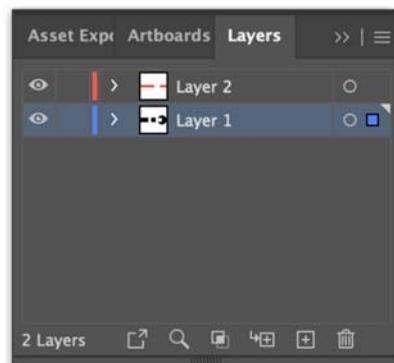
(k) **Edit Selection** : mengedit seleksi yang aktif

6) Seleksi melalui *layer*

Cara lain menyeleksi objek adalah dengan melalui *layer*. Langkah ini bisa ditempuh dengan catatan apabila setiap objek diletakkan di *layer-layer* terpisah, seperti cara kerja ketika bekerja menggunakan *Photoshop*.

Untuk menyeleksi objek melalui *layer*, lakukan langkah-langkah berikut:

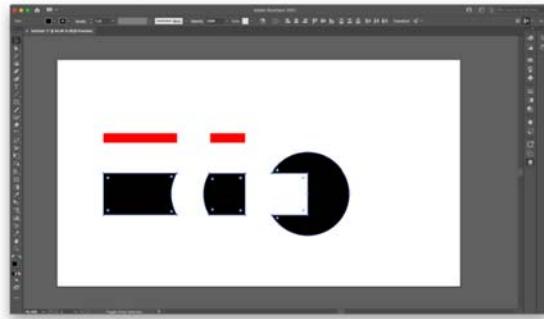
- Aktifkan panel *Layers* dengan mengklik ikon
- Tekan *Alt+klik* *layer* yang objeknya akan diseleksi



Gambar 2. 535 Tampilan Menu *Layer*

Objek yang terseleksi akan dikelilingi *bounding box*. Warna *bounding box* sama dengan warna *layer* yang terlihat di dalam panel *Layers* itu.





Gambar 2. 546 Objek berwarna hitam terseleksi

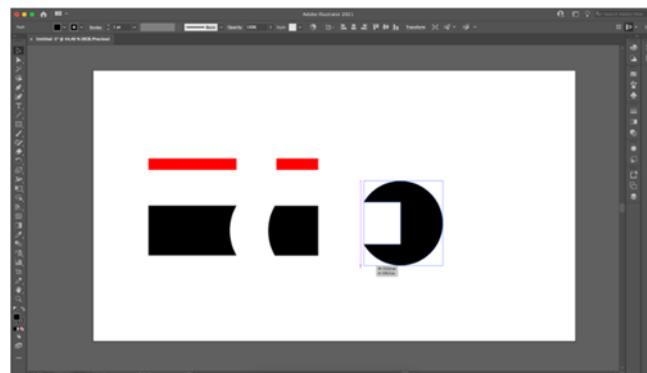
b. Mengatur Ukuran Objek

Objek yang sudah dibuat dapat diatur ulang ukurannya. Karena berbentuk objek vektor, maka pembesaran pada skala berapa pun tidak akan menurunkan kualitas ketajaman objek.

Ada beberapa cara mengatur ukuran objek. Di dalam sub-sub bab berikut akan dijelaskan berbagai macam cara mengatur ukuran objek yang sudah dibuat di dalam dokumen.

1) *Selection Tool*

Setelah objek dibuat, pilih *Selection Tool* dan klik objek yang ingin diperbesar atau diperkecil ukurannya. Caranya, *klik-drag handle* (kotak-kotak kecil) yang ada di setiap sisi dan sudut persegi empat objek.



Gambar 2. 557 Mengatur ukuran objek menggunakan *Selection Tools*



Ukuran riil objek akan terlihat pada panel *pop-up* yang muncul selama proses menskala objek. Pada contoh di atas, ukuran objek sebesar 1,15 inchi x 1,25 inchi.

2) *Scale Tool*

Scale Tool digunakan untuk memperbesar atau mengecilkan ukuran objek. Penggunaan *Scale Tool* diawali dengan menyeleksi tool yang akan diskalakan.

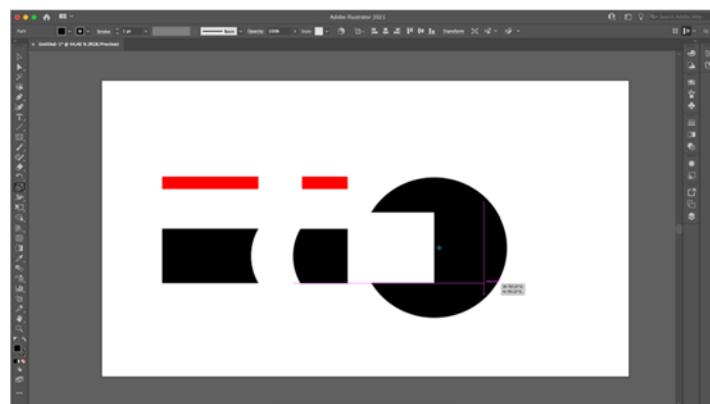
Berikut langkah-langkah penggunaannya:

- (a) Seleksi objek yang akan diskala menggunakan *Selection Tool*
- (b) Pilih *Scale Tool*



Gambar 2. 568 *Scale Tool (S)*

- (c) Arahkan kursor *mouse* pada salah satu *handle* untuk digunakan sebagai pengatur ukuran objek.
- (d) *Klik-drag* ke arah luar untuk memperbesar objek atau ke arah dalam untuk mengecilkannya.

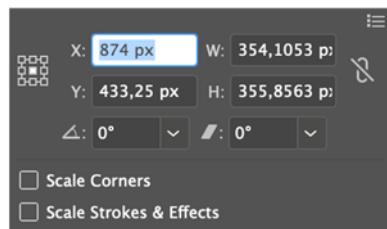


Gambar 2. 579 Mengatur ukuran objek menggunakan *scale tool*



3) Menggunakan *Width* dan *Height*

Untuk mengubah ukuran pada objek anda juga dapat mengatur nya menggunakan kolom isian width dan height pada toolbar bagian atas saat kalian menyeleksi objek tersebut kemudian tekan **transform** pada toolbar bagian atas.



Gambar 2. 140 Kotak isian *width* dan *height* pengatur ukuran

Panel *Transform* akan muncul saat Anda mengklik tulisan yang sama di dalam *toolbar*.

Di dalam panel tersebut, Anda bisa mengelola berbagai macam atribut objek, yaitu:

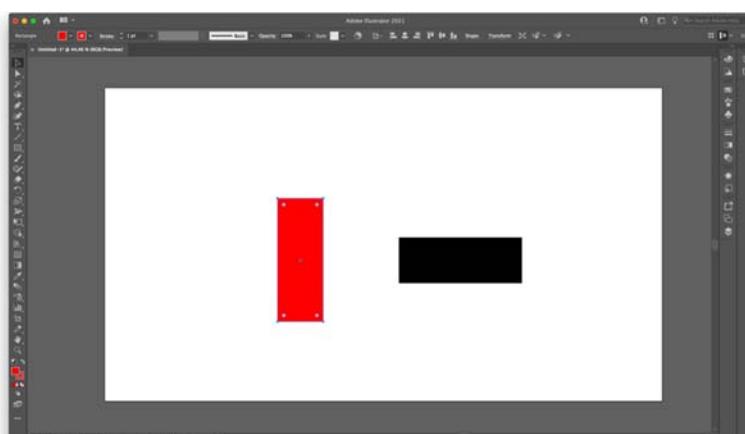
- (a) **X dan Y** : mengatur posisi objek menurut sumbu X dan Y di dalam dokumen
- (b) **W dan H** : mengatur ukuran lebar dan tinggi objek
- (c) **Rotate** : untuk memutar objek berdasarkan derajat tertentu
- (d) **Shear** : untuk mencondongkan objek berdasarkan derajat tertentu
- (e) **Scale Corners** : dalam kondisi aktif (tercentang), sudut objek akan ikut diskala (diperbesar atau diperkecil)



(f) **Scale Strokes & Effects** : dalam kondisi aktif, *stroke* (garis tepi) dan efek pada objek akan ikut diperbesar atau diperkecil

c. *Reference Point*

Di dalam panel *Transform* terdapat *Reference Point*. Fungsinya untuk menentukan titik sumbu pada objek. Anda bisa mengklik titik titik yang ada di dalam *Reference Point* itu dan warna hitam menunjukkan lokasi titik sumbu.



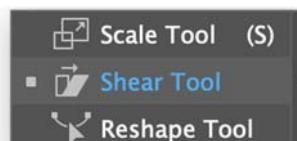
Gambar 2. 581 Contoh Objek Merah yang diputar dengan *Reference Point* di tengah

Reference Point sangat penting digunakan saat Anda ingin memutar objek.

d. *Shear Tool*

Shear Tool digunakan untuk mencondongkan objek. Berikut langkah-langkah menggunakan *tool* ini:

- 1) Siapkan objek yang akan dicondongkan
- 2) Pilih *Shear Tool*



Gambar 2. 592 *Shear Tool*



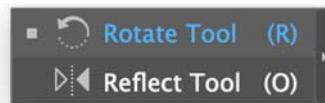
3) *Klik-drag* untuk mencondongkan objek



Gambar 2. 603 Hasil objek menggunakan *shear tool*

e. *Rotate Tool*

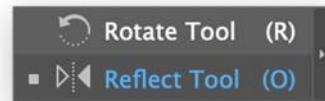
Rotate Tool digunakan untuk memutar objek. Setelah *tool* ini aktif, *klik-drag* untuk memutar objek.



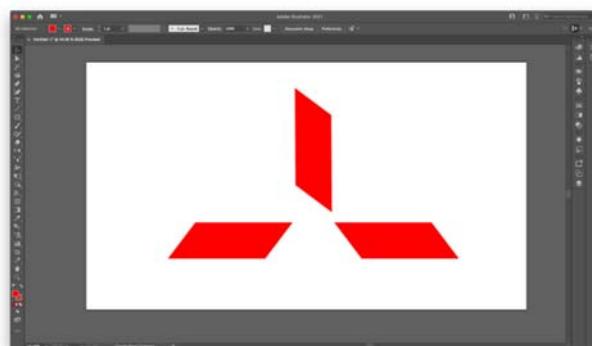
Gambar 2. 614 *Rotate Tool (R)*

f. *Reflect Tool*

Reflect Tool digunakan untuk *flip* objek secara horizontal maupun vertikal. *Flip* sendiri artinya memutar sebesar 180°.



Gambar 2. 625 *Reflect Tool (O)*



Gambar 2. 636 Objek dengan *Reflect Tool*



g. *Free Transform Tool*

Free Transform Tool adalah *tool* yang dapat digunakan untuk mengubah ukuran dan memutar objek sekaligus. Selain itu, *tool* ini juga dilengkapi kemampuan untuk melakukan distorsi terhadap objek.



Gambar 2. 647 *Free Transform (E)*

Ketika *tool* ini diklik, maka akan muncul *tool flyout* yang berisi *tool-tool* lain dengan fungsi berbeda-beda. Berikut fungsinya:

- 1) **Constrain** : ketika dalam kondisi aktif, maka ukuran *Width* akan mengikuti *Height* secara proporsional, dan begitu pula sebaliknya. Apabila objek diputar, perputarannya sebesar kelipatan 45
- 2) **Free Transform** : *tool* untuk memutar atau mengatur ukuran objek
- 3) **Perspective Distort** : *tool* ini digunakan untuk mendistorsi objek secara perspektif
- 4) **Free Distort** : *tool* ini digunakan untuk mendistorsi objek secara bebas

h. *Group*

Group digunakan untuk mengelompokkan beberapa objek menjadi satu. Ketika beberapa objek sudah di-*group*, maka transformasi yang terjadi terhadap salah satu objek, misalnya dengan mengubah ukuran, memindah, dan lain-lain, akan



memengaruhi objek-objek lainnya. Berikut ini langkah-langkah untuk meng-*group* objek:

- 1) Seleksi objek-objek yang akan di-*group*
- 2) Pilih menu *Object > Group*
- 3) Coba pindahkan atau ubah ukuran objek itu dan Anda akan melihat seluruh obiek akan berubah seluruhnya

i. *Ungroup*

Ungroup digunakan untuk melepas *group* yang sudah dibuat. Langkah-langkahnya adalah sebagai berikut:

- 1) Pilih *Selection Tool* dan klik objek yang akan di-*ungroup*
- 2) Lalu pilih menu *Object > Ungroup*
- 3) Coba pindahkan atau ubah ukuran objek itu. Masing-masing objek akan berubah tanpa memengaruhi lainnya.

j. *Distribute*

Distribute digunakan untuk menyamakan jarak beberapa objek. Jarak objek bisa diatur secara horizontal maupun vertikal, tergantung dari susunan objek yang Anda buat. Berikut salah satu contoh menggunakan *Distribute* :

- 1) Seleksi objek-objek yang akan di-*distribute*



Gambar 2. 658 Tampilan Objek dengan jarak beragam



2) Karena objek-objek di atas disusun vertikal, pilihlah *Distribute vertical*



Gambar 2. 669 Berbagai macam jenis *distribute*

3) Pilih salah satu *distribute* untuk objek vertikal, misalnya



Gambar 2. 6750 Hasil Objek dengan *Distribute Vertical*

Objek akan tertata dengan rapi setelah ikon tersebut diklik.

Jarak antar objek akan sama satu dengan lainnya.

k. *Align*

Align digunakan untuk merapikan posisi beberapa objek. Objek yang pertama kali diseleksi secara *default* akan menjadi acuan ketika objek-objek lainnya diposisikan. Berikut langkah-langkah penggunaan *Align* :

- 1) Masih menggunakan objek-objek yang sama seperti gambar sebelumnya (lihat penggunaan *Distribute* di atas)
- 2) Klik *Align Center*



Gambar 2. 681 Berbagai jenis *align*

- 3) Pilih model perataan yang diinginkan. Sebagai contoh, perataan kiri bisa dilakukan dengan menggunakan ikon





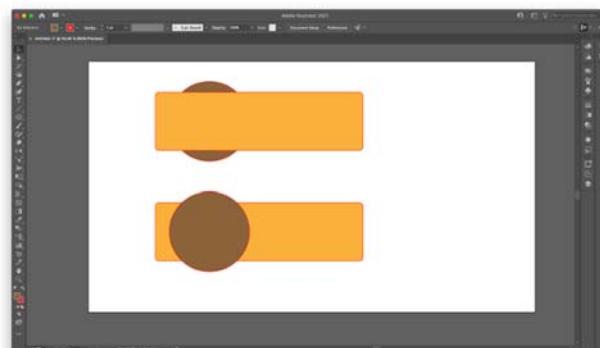
Gambar 2. 692 Hasil Penerapan *Align Center*

I. *Arrange*

Ketika Anda membuat beberapa objek di dalam dokumen, maka berlaku aturan objek terbaru akan menumpuk objek yang lebih dulu dibuat.

Arrange digunakan untuk mengubah posisi tersebut. Pilih menu **Object > Arrange** dan pilih salah satu perintah yang diinginkan, yaitu:

- 1) **Bring to Front** : meletakkan objek di urutan teratas
- 2) **Bring Forward** : meletakkan objek satu tingkat di atas
- 3) **Send Backward** : meletakkan objek satu tingkat di bawah
- 4) **Send to Back** : meletakkan objek di urutan terbawah



Gambar 2. 703 Perbandingan objek lingkaran ketika menggunakan *Bring to Front*



5. Efek – Efek

a. *Opacity*

Perintah *Opacity* digunakan untuk mengatur tingkat transparansi objek. Semakin kecil dan jauh angkanya dari 100%, maka objek itu akan menjadi semakin transparan.

Opacity dapat ditemukan di dalam toolbar dan akan aktif ketika ada objek yang terseleksi.

b. *Blending Mode*

Blending Mode digunakan untuk mencampur *pixel* dari dua buah objek yang saling menumpuk. *Blending Mode* dapat diaktifkan atau pilih menu dengan mengklik ikon **Transparency Window > Transparency**.

Untuk menggunakan *Blending Mode*, lakukan contoh langkah-langkah di bawah ini:

- 1) Buat dua buah objek dan usahakan posisinya bertumpuk.
Berikut contohnya:
- 2) Aktifkan panel *Transparency*
- 3) Pilih *blending mode* dengan mengklik menu kombo *Blending Mode*

Berikut beberapa variasi hasil dari penggunaan *Blending Mode*. Pilih satu demi satu untuk mendapat *blending mode* lainnya.

c. *Brush Definition*

Menu *Brush Definition* digunakan untuk memilih bentuk *stroke*. Anda bisa memilih *stroke* yang dibentuk dari pola lingkaran hingga bentuk-bentuk lainnya.





Refleksi III

Setelah mempelajari materi ini, manfaat yang akan kamu rasakan adalah anda dapat membuat membuat aset sederhana menggunakan aplikasi pengolah grafis berbasis vector yang dapat digunakan pada perangkat lunak dan gim.



Asesmen III

Penugasan ini melatih anda dalam melakukan praktik pembuatan aset sederhana untuk perangkat lunak atau gim menggunakan aplikasi pengolah vektor.

D. Aset Grafis Berbasis Bitmap

1. Mengenal *Adobe Photoshop*

a. Membuat *Canvas* Baru

Kanvas adalah dokumen tempat membuat desain baru atau mengedit foto-foto digital. Jika Anda ingin membuat kanvas baru, berikut ini langkah-langkahnya:

- 1) Klik *Create New*
- 2) Ada pilihan *preset* yang bisa digunakan secara langsung, yaitu:
 - (a) **Recent** : kanvas dengan ukuran yang akhir-akhir ini Anda gunakan
 - (b) **Saved** : kanvas dengan ukuran yang Anda simpan
 - (c) **Photo** : kanvas dengan ukuran-ukuran standar foto cetak (*portrait*, *landscape*, dan sebagainya)
 - (d) **Print** : kanvas dengan ukuran-ukuran kertas cetak (*Letter*, *A3*, *A4*, dan seterusnya)



(e) **Art & Illustration** : kanvas dengan ukuran mengikuti kanvas untuk lukisan, ilustrasi, dan media artistik lainnya

(f) **Web** : kanvas dengan ukuran mengikuti standar desain *website*

(g) **Mobile** : kanvas *smartphone/tablet* dengan ukuran mengikuti layar (*iPhone*, *Android*, *iPad*, dan sebagainya)

(h) **Film & Video** : kanvas dengan ukuran mengikuti standar untuk film dan video (HDTV, HDV, dan seterusnya).

3) Jika tidak ada ukuran yang cocok, maka aturlah sendiri ukuran kanvas menggunakan *Preset Details*. Pengaturan ukuran kanvas menggunakan *Preset Details* :

- (a) **Width** : ukuran lebar kanvas ditentukan di bagian ini
- (b) **Height** : ukuran tinggi kanvas ditentukan di bagian ini
- (c) **Orientation** : mengatur apakah kanvas berorientasi *portrait* atau *landscape*
- (d) **Artboards** : jika ingin dilengkapi dengan *Artboards*, centanglah bagian ini
- (e) **Resolution** : mengatur ketajaman resolusi dalam satuan *pixels/inch* (banyaknya *pixel per inch*) atau *pixel/centimeters*
- (f) **Color Mode** : mengatur mode warna yang digunakan
- (g) **RGB Color** : untuk desain *desktop*, seperti *website*, ikon, ilustrasi digital, *interface* aplikasi *smartphone*, atau desain untuk keperluan *movie/video*



Pengeditan foto digital juga lebih cenderung menggunakan RGB *Color* jika tidak diarahkan untuk pencetakan.

- (h) **CMYK Color** : untuk desain *printing*, seperti sampul buku, poster, brosur, dan sebagainya.
- (i) **Grayscale** : untuk desain dengan warna hitam putih dan skala abu-abu.
- (j) **Bitmap** : untuk desain *desktop* yang lebih mengarah pada lingkungan *MS Windows*.
- (k) **LAB Color** : untuk pengeditan foto digital. Namun, penggunaan *LAB Color* cukup jarang.

- 4) Di sebelah kanan menu kombo *Color Mode*, Anda bisa pilih kedalaman warna dalam satuan bit. Secara *default*, pilih 8 bit.
- 5) Pilih warna dasar kanvas. Secara *default* berwarna putih (*White*). Namun, Anda juga bisa memilih warna hitam atau warna latar belakang yang saat itu sedang aktif.
- 6) Jika Anda sering bekerja dengan ukuran kanvas yang sama seperti yang Anda buat pada langkah-langkah di atas, maka Anda bisa menyimpannya agar bisa digunakan terus-menerus tanpa harus mengurnya dari awal. Caranya, beri nama yang spesifik dan klik ikon seperti ditunjukkan pada gambar berikut.
- 7) Tekan tombol *Create*



b. Menyimpan *File*

Desain yang Anda buat dapat disimpan dengan menekan tombol **Shift+Ctrl+S**. Jika Anda ingin mempertahankan layer (tentang layer akan dibahas dalam bab khusus), maka:

- 1) Simpan dalam format *Photoshop .PSD*.
- 2) Pastikan kotak cek *Layers* dalam kondisi tercentang.

c. Membuka *File*

Apabila Anda sudah membuka atau menyimpan beberapa desain, *file-file* itu akan diletakkan di jendela *Photoshop* sebagai *shortcut*. Anda bisa meng-klik *thumbnail file* tersebut untuk membukanya. Namun, jika *file* yang ditemukan tidak ada, maka pilih tombol *Open* dan cari *file* yang dimaksud.

d. *Canvas Size*

Canvas Size digunakan untuk mengatur ulang ukuran kanvas. Ketika Anda membuat kanvas baru atau membuka foto digital, mungkin ukuran kanvas yang ada saat ini terlihat kurang besar/ kecil. Anda bisa mengoreksinya menggunakan langkah-langkah berikut:

- 1) Pastikan Anda sudah memiliki kanvas yang ingin diatur ulang ukurannya
- 2) Pilih menu **Image > Canvas Size**
- 3) Anda bisa mengatur ukuran kanvas dengan memasukkan nilai *Width* dan *Height* yang baru

2. Menggambar Objek

a. *Rectangle Tools*

Rectangle Tool digunakan untuk membuat persegi empat. Berikut langkah-langkah menggunakannya:



- 1) Pilih **Rectangle Tool**
- 2) *Klik-drag* di dalam dokumen untuk membuat persegi empat
- 3) Pilih *Selection Tool* dan klik objek tersebut
Untuk membuat segi empat sama sisi, tekanlah tombol **Shift+klik+ drag**.

Apabila ingin membuat persegi empat dengan ukuran presisi, kliklah di dalam dokumen. Kotak dialog akan muncul untuk membantu Anda mengisi Width dan Height persegi empat.

b. *Rounded Rectangle Tools*

Rounded Rectangle Tool digunakan untuk membuat persegi empat dengan sudut tumpul. Seberapa besar tingkat ketumpulan yang diinginkan bisa diatur menggunakan berbagai cara.

Berikut langkah-langkah pembuatannya:

- 1) Pilih **Rounded Rectangle Tool**
- 2) Selanjutnya, *klik-drag* di dalam dokumen untuk membuat *Rounded Rectangle*
- 3) Anda bisa pindahkan objek itu menggunakan *Selection Tool*

Mengatur tumpul persegi empat baik *Rectangle* maupun *Rounded Rectangle* dilengkapi atribut untuk melengkungkan setiap sudutnya. Caranya adalah dengan terlebih dulu mengklik *Shape* yang dapat ditemukan di dalam *toolbar*.

Cukup naikkan atau turunkan nilai di dalam kotak isian *Corner Radius* masing-masing sudut untuk mengatur tumpul-tajamnya persegi empat

Selain itu, Anda bisa langsung *klik-drag* menggunakan *Selection Tool* pada titik *corner radius* 6 yang ada di setiap sudut



persegi empat. *Klik-drag* ke arah dalam untuk menampilkan persegi empat dan ke arah luar untuk menajamkannya kembali.

c. *Ellipse Tools*

Ellipse Tool digunakan untuk membuat objek lingkaran. Cara penggunaannya adalah sebagai berikut:

- 1) Pilih **Ellipse Tool**
- 2) *Klik-drag* di dalam dokumen untuk membuatnya.
- 3) Apabila muncul tanda silang di dalam lingkaran, ini menunjukkan ukuran *Width* maupun *Height* lingkaran tersebut sama.
- 4) Pilih *Selection Tool* untuk memindahkan objek itu.

Selain metode *klik-drag*, ada beberapa variasi lain untuk menggambar objek lingkaran, yaitu:

- 1) Tekan **Shift+klik-drag** untuk membuat objek lingkaran sempurna.
- 2) Tekan **Alt+klik-drag** untuk membuat objek lingkaran dari tengah lingkaran.

d. *Polygon Tool*

Polygon Tool digunakan untuk menggambar objek bersegi banyak. Langkah-langkah penggunaannya adalah sebagai berikut :

- 1) Pilih **Polygon Tool**
- 2) *Klik-drag* di dalam dokumen
- 3) Posisikan dengan menggunakan *Selection Tool*

Bentuk poligon di dalam dokumen dapat diatur ulang, seperti jumlah sudut pada objek, kemiringan, tumpulnya sudut, dan sebagainya. Caranya adalah dengan mengklik *Shape*.



- 1) **Polygon Side Count** : mengatur jumlah sudut yang ingin dibuat.
- 2) **Polygon Angle** : untuk mengatur derajat perputaran objek poligon.
- 3) **Polygon Radius** : untuk mengatur ukuran poligon.
- 4) **Corner Radius** : untuk menampilkan sudut poligon.
- 5) **Polygon Side Length** : untuk mengatur panjang sisi poligon.

e. *Star Tool*

Star Tool digunakan untuk membuat objek bintang. Langkah-langkah penggunaan *Star Tool* adalah sebagai berikut:

- 1) Pilih **Star Tool**
- 2) Buat objek bintang dengan cara *klik-drag* di dalam dokumen.
- 3) Posisikan dengan *Move Tool* jika sudah.

f. *Line Segment Tool*

Line Segment Tool digunakan untuk membuat garis di dalam dokumen. Langkah-langkah pembuatannya adalah sebagai berikut:

- 1) Pilih **Line Segment Tool**
- 2) *Klik-drag* di dalam dokumen untuk membuat garis
- 3) Pilih *Selection Tool* untuk memindah garis itu

g. *Arc Tool*

Arc Tool digunakan untuk membuat objek berbentuk garis lengkung. Langkah-langkah membuat *Arc Tool* adalah sebagai berikut:

- 1) Pilih **Arc Tool**



- 2) Buat garis lengkung dengan cara *klik-drag* di dalam dokumen
- 3) Lepas tombol kiri *mouse*

h. Spiral Tool

Spiral Tool digunakan untuk membuat objek berbentuk spiral. Langkah-langkah menggunakan *Spiral Tool* adalah sebagai berikut:

- 1) Pilih **Spiral Tool**
- 2) *Klik-drag* di dalam dokumen untuk membuat objek spiral.
- 3) Pilih *Selection Tool* dan *klik-drag* objek ke tempat yang Anda inginkan.

i. Rectangular Grid Tool

Rectangular Grid Tool adalah tool yang digunakan untuk membuat kotak-kotak bersusun (*grid*). Anda bisa memanfaatkannya untuk tabel atau kotak isian form pendaftaran. Berikut langkah-langkah membuat *Rectangular Grid*:

- 1) Pilih **Rectangular Grid Tool**
- 2) *Klik-drag* di dalam dokumen untuk membuat objek tersebut
- 3) Posisikan menggunakan *Selection Tool* kalau sudah

3. Editing Objek

a. Group

Group digunakan untuk mengelompokkan beberapa objek menjadi satu. Ketika beberapa objek sudah *di-group*, maka transformasi yang terjadi terhadap salah satu objek, misalnya dengan mengubah ukuran, memindah, dan lain-lain, akan memengaruhi objek-objek lainnya. Berikut ini langkah-langkah untuk meng-*group* objek:



- 1) Seleksi objek-objek yang akan di-*group*
- 2) Pilih menu **Object > Group**
- 3) Coba pindahkan atau ubah ukuran objek itu dan Anda akan melihat seluruh obiek akan berubah seluruhnya

b. *Ungroup*

Ungroup digunakan untuk melepas *group* yang sudah dibuat. Langkah-langkahnya adalah sebagai berikut:

- 1) Pilih *Selection Tool* dan klik objek yang akan di-*ungroup*
- 2) Lalu, pilih menu **Object > Ungroup**
- 3) Coba pindahkan atau ubah ukuran objek itu. Masing-masing objek akan berubah tanpa memengaruhi lainnya

4. Efek - Efek

a. *Opacity*

Perintah *Opacity* digunakan untuk mengatur tingkat transparansi objek. Semakin kecil dan jauh angkanya dari 100%, maka objek itu akan menjadi semakin transparan.

Opacity dapat ditemukan di dalam toolbar dan akan aktif ketika ada objek yang terseleksi.

b. *Blending Mode*

Blending Mode digunakan untuk mencampur *pixel* dari dua buah objek yang saling menumpuk. *Blending Mode* dapat diaktifkan atau pilih menu dengan mengklik ikon **Transparency Window > Transparency**.

Untuk menggunakan *Blending Mode*, lakukan contoh langkah-langkah di bawah ini:

- 1) Buat dua buah objek dan usahakan posisinya bertumpuk
- 2) Aktifkan panel *Transparency*



3) Pilih *blending mode* dengan mengklik menu kombo *Blending Mode*

Berikut beberapa variasi hasil dari penggunaan *Blending Mode*. Pilih satu demi satu untuk mendapat *blending mode* lainnya.



E. Refleksi IV

Setelah mempelajari materi ini, manfaat yang akan kamu rasakan adalah anda dapat membuat membuat aset sederhana menggunakan aplikasi pengolah grafis berbasis vector yang dapat digunakan pada perangkat lunak dan gim.



Asesmen IV

Penugasan ini melatih anda dalam melakukan praktik pembuatan aset sederhana untuk perangkat lunak atau gim menggunakan aplikasi pengolah vektor.

RANGKUMAN

Aset adalah semua kebutuhan atau bahan - bahan yang digunakan dalam perangkat lunak atau gim.



F. PENGAYAAN

Untuk lebih menambah luas keterampilan mengenai BAB ini, berikut link yang bisa kamu kunjungi dan memotivasi kamu agar menjadi lebih giat dan meningkatkan kualitas kemampuan kamu dibidang pembuatan aset desain pada perangkat lunak dan gim:



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
DIREKTORAT JENDERAL PENDIDIKAN VOKASI
DIREKTORAT SEKOLAH MENENGAH KEJURUAN
Dasar - Dasar Pengembangan Perangkat Lunak dan Gim
SMK Kelas X
Penulis : Yustiana Amita Utama dan Esa Hergatama
ISBN : xxx-xxx-xxx-xxx-x

BAB III PROGRAMASI DASAR

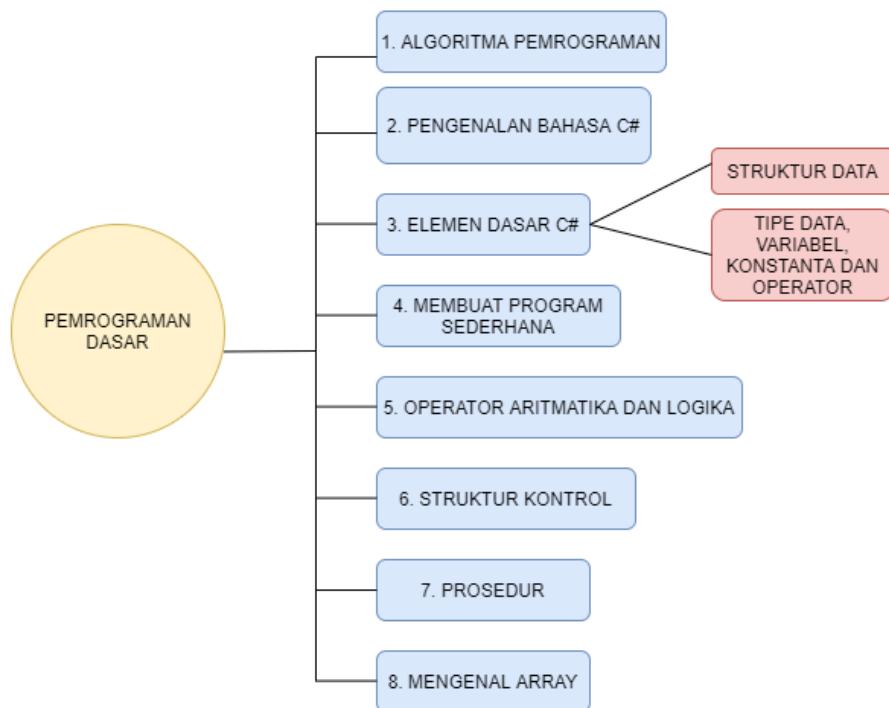


TUJUAN PEMBELAJARAN

Setelah mempelajari Bab III ini, peserta didik diharapkan mampu :

1. Mengimplementasikan algoritma pemrograman
2. Melakukan instalasi dan konfigurasi *software* tool pemrograman
3. Menggunakan struktur data
4. Menggunakan tipe data, variabel, konstanta dan operator
5. Membuat program sederhana
6. Menerapkan operasi aritmatika dan logika
7. Menggunakan struktur kontrol perulangan dan percabangan
8. Membuat program menggunakan prosedur
9. Membuat program menggunakan *array*

PETA KONSEP



BAB III

PEMROGRAMAN DASAR

Apersepsi



Saat ini komputer sudah menjadi barang yang dibutuhkan untuk membantu memecahkan permasalahan dalam kehidupan. Seperti kita ketahui di bank, komputer digunakan dalam berbagai macam aktivitas seperti simpan pinjam dan transaksi keuangan lainnya. Pada bidang kesehatan, komputer dipakai untuk mendiagnosa penyakit, analisa organ tubuh bagian dalam. Di bidang jasa pengiriman dokumen, komputer dan internet dapat digunakan untuk mengirim *e-mail* untuk mengirim berbagai macam dokumen, dll. Komputer sebenarnya merupakan seperangkat rangkaian elektronika yang tidak dapat mengerjakan apa-apa. Tetapi komputer dapat menyelesaikan berbagai permasalahan dan membantu kerja manusia pada saat komputer tersebut diberi instruksi dan perintah yang benar. Langkah-langkah untuk memberikan instruksi kerja sebuah komputer ini disebut dengan pemrograman komputer. Untuk dapat memberikan instruksi kepada komputer, maka diperlukan sebuah Bahasa pemrograman untuk menjembatani antara keinginan si pemberi perintah (manusia) dengan yang diperintah (komputer).

Kata Kunci



Algoritma Pemrograman, instalasi, struktur data, tipe data, variabel, konstanta, operator, struktur kontrol, prosedur, *array*



A. Algoritma Pemrograman

Saat mempelajari bahasa pemrograman untuk pertama kali, kalian akan mendapatkan kesan bahwa bagian yang sulit untuk memecahkan masalah di komputer adalah menerjemahkan ide-ide dalam bahasa tertentu yang akan dimasukkan ke dalam komputer. Untuk itu diperlukan langkah-langkah solusi dan menulisnya dalam urutan langkah sederhana, seolah-olah instruksinya diberikan kepada manusia. Urutan instruksi yang dinyatakan dengan cara ini disebut sebagai algoritma. Sehingga bisa disimpulkan bahwa algoritma merupakan urutan instruksi yang tepat untuk memberikan solusi (Savitch, Walter J., 2014).

Sedangkan dalam Kamus Besar Bahasa Indonesia (KBBI) algoritma diartikan sebagai prosedur sistematis untuk memecahkan masalah matematis dalam langkah-langkah terbatas. Penulisan algoritma yang tidak baku memudahkan untuk memahami alur penyelesaian masalah. Ada beberapa cara dalam penulisan algoritma, yaitu :

1. Teks Algoritma untuk menentukan besar kecilnya bilangan dengan masukan / inputan 2 bilangan.

Contoh :

Teks algoritma untuk menentukan

- a. Masukkan bilangan pertama (i), bilangan kedua (j)
- b. Jika i lebih besar daripada j maka cetak i.
- c. Selain itu cetak j.

2. *Pseudocode*

Pseudocode merupakan deskripsi tingkat tinggi yang ringkas dan informal dari suatu program menggunakan konversi Bahasa



pemrograman. *Pseudocode* bukanlah bahasa pemrograman yang sebenarnya. Jadi tidak dapat dikompilasi menjadi program yang dieksekusi. *Pseudocode* ini menggunakan sintaks bahasa Inggris sederhana untuk menulis kode sebelum diubah menjadi bahasa pemrograman.

Contoh beberapa keyword yang sering digunakan:

- a) *INPUT*: digunakan untuk memasukkan sesuatu dari keyboard
- b) *OUTPUT*: keluaran yang tampil di layar.
- c) *WHILE*: *loop* (iterasi yang memiliki kondisi di awal).
- d) *FOR*: *loop* penghitungan (iterasi).
- e) *IF-THEN-ELSE*: seleksi (keputusan) dimana pilihan dibuat.

Contoh kasus : Perkalian dua bilangan

```
Pseudocode:  
  
BEGIN  
NUMBER a, b, result  
OUTPUT ("Input number 1 : ")  
INPUT a  
OUTPUT ("Input number 2 : ")  
INPUT b  
result = a*b  
OUTPUT result  
END
```

3. Flowchart

Flowchart adalah sarana grafis untuk mendokumentasikan urutan operasi. *Flowchart* berfungsi sebagai sarana bergambar untuk komunikasi tentang urutan waktu peristiwa atau tindakan. Sebagai format bergambar, *flowchart* telah menjadi subjek



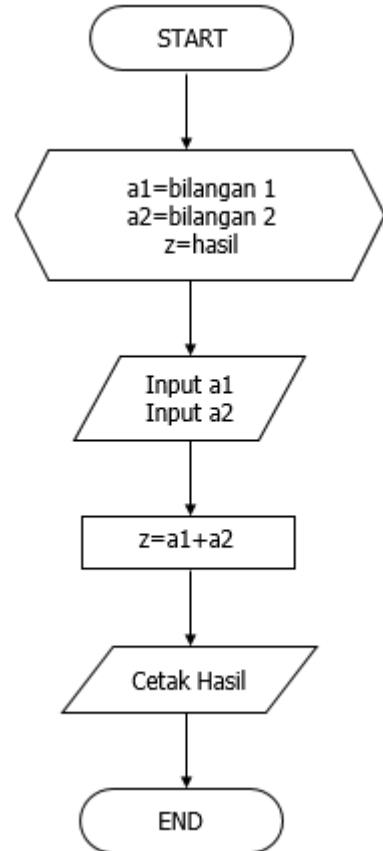
Standar Nasional Internasional dan Amerika (ANSI, 1970; chapin, 1979). Di bawah ini merupakan *symbol flowchart* :

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 3.1 *Symbol flowchart* (sumber : <https://informatikalogi.com/>)



Contoh :



Gambar 3.2 Contoh Flowchart penjumlahan

Struktur Algoritma

Dalam membuat algoritma, ada berbagai macam urutan pemrosesan instruksi. Struktur dasar algoritma dibagi menjadi 3 bagian :

1. Runtutan (*Sequence*)

Dalam tahap ini instruksi dalam algoritma diproses secara runtut.

Contoh :

Algoritma Luas Segitiga

Diketahui alas dan tinggi sebuah segitiga, akan dihitung luasnya dengan menggunakan algoritma sebagai berikut :

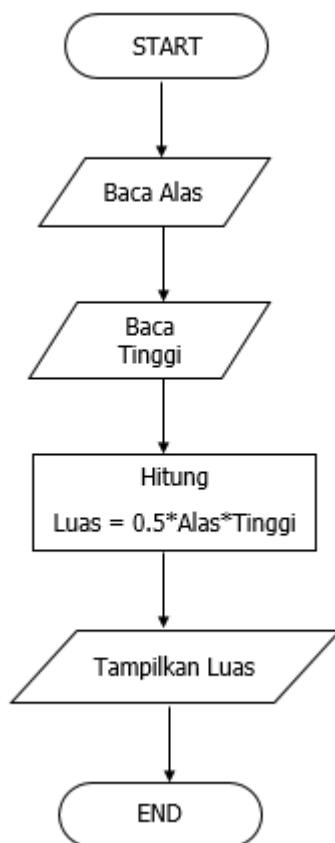


Diketahui sebuah segitiga yang memiliki alas dan tinggi.

Deskripsi Struktur :

- a. Mulai
- b. Baca alas
- c. Baca tinggi
- d. Hitung luas = $0.5 * \text{alas} * \text{tinggi}$
- e. Tampilkan luas
- f. Selesai

Flowchart Luas_Segitiga



Gambar 3.3 Contoh Flowchart Runtutan



2. Pemilihan (*Selection*)

Dalam tahap ini instruksi dalam algoritma digunakan untuk kasus-kasus yang mengandung pemilihan dalam pengambilan keputusan.

Contoh :

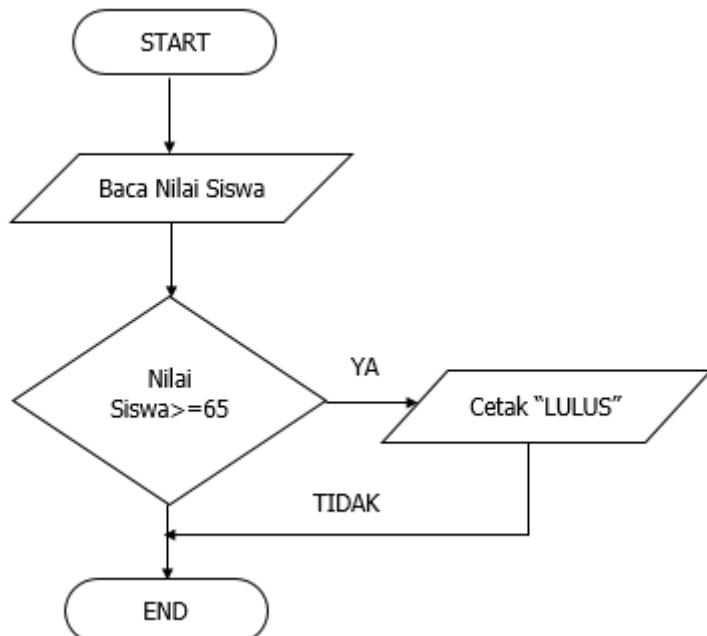
Algoritma Kelulusan Siswa

Diketahui seorang siswa bisa dikatakan lulus jika nilainya ≥ 65 .

Deskripsi Struktur :

- a. Mulai
- b. Baca nilai_siswa
- c. Jika nilai_siswa ≥ 65 maka kerjakan langkah d
- d. Cetak "LULUS"
- e. Selesai

Flowchartnya sebagai berikut:



Gambar 3.4 Contoh Flowchart Pemilihan



3. Pengulangan (*Repetition*)

Dalam tahap ini instruksi dalam algoritma digunakan untuk kasus-kasus yang membutuhkan pengulangan

Algoritma Cetak Angka

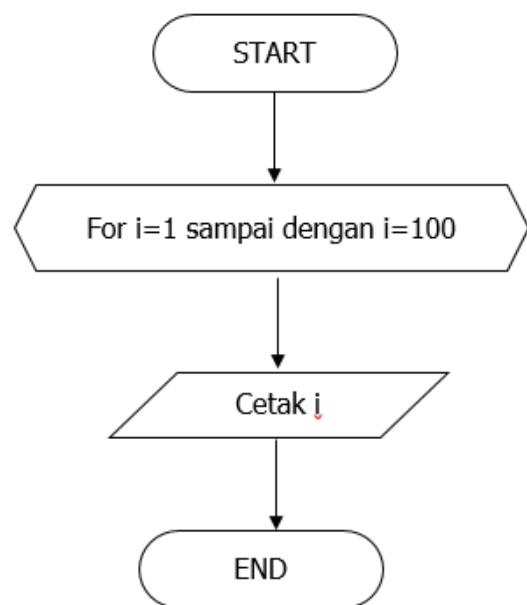
Akan dicetak angka 1 sampai 100 dengan menggunakan perulangan for.

Dicetak angka 1 sampai 100 dengan perulangan for.

Deskripsi Struktur :

- a. Mulai
- b. Kerjakan langkah c mulai $i=1$ sampai $i=100$
- c. Cetak i
- d. Selesai

Flowchart Cetak Angka Dengan For



Gambar 3.5 Contoh Flowchart Pengulangan



Tugas Mandiri

Buatlah algoritma untuk menghitung konversi suhu dari Reamur menjadi Celcius dan Fahrenheit, kemudian tampilkan hasilnya.

Diskusi Kelompok

Buatlah kelompok kecil, analisis mengenai algoritma dua bilangan untuk menentukan proses penjumlahan, pengurangan, perkalian dan pembagian.

Uji Kompetensi

1. Jelaskan dengan bahasa kalian tentang pengertian algoritma!
2. Jelaskan dengan bahasa berbagai macam teknik untuk penulisan algoritma!
3. Berikan gambar 5 contoh diagram alir (flowchart) beserta fungsinya!
4. Buatlah flowchart untuk penghitungan luas segitiga!
5. Susunlah algoritma untuk menghitung total dan rata-rata nilai yang dimiliki oleh seorang siswa sebanyak 4 nilai. Nama dan nilai dibaca dari keyboard. Hasil akhir yang diharapkan adalah tampilan nama siswa, nilai, jumlah dan rata-rata.



B. Pengenalan Bahasa C#

1. Mengapa C#?

C# (baca : C sharp) adalah bahasa pemrograman sederhana, modern, berorientasi objek dan bekerja dengan berbagai compiler, serta platform khusus seperti Microsoft .NET Framework yang dapat dikembangkan di berbagai aplikasi seperti *web, mobile, desktop* dan *game*.

Syntax bahasa pemrograman C# mirip dengan bahasa pemrograman lainnya, seperti Java atau C++. Sebagai contoh, kode program terdiri dari pernyataan yang diakhiri dengan titik koma (:), dan kurung kurawal ({dan}) digunakan untuk mengelompokkan pernyataan, seperti dalam perulangan *foreach*.

2. Instalasi dan konfigurasi *software tool* pemrograman

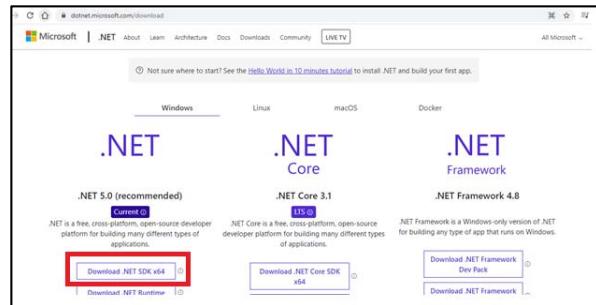
Ada berbagai macam C# IDE (*Integrated Development Environment*) *software* untuk pemrograman C# seperti *microsoft visual studio, notepad++, monodevelop, Visual Studio Code, slickedit, glot.io, replit, online GDB, online Csharp*, dll.

Visual Studio Code merupakan salah satu *text editor* yang dikembangkan *Microsoft*. Aplikasi ini dapat berjalan pada *Linux, Mac OS, Windows*. *VS Code* ini memiliki banyak keunggulan diantaranya adalah dapat mendukung berbagai macam bahasa pemrograman, ringan, memiliki banyak *plugin*, tersedia gratis, memiliki terintegrasi dengan *Git*, dll.

Untuk instalasi C# dengan VS Code. Langkah pertama dimulai dengan *install* .NET Framework SDK dengan *download file installer* di <https://dotnet.microsoft.com/download>.



Pilih file yang diberi tanda kotak merah, kemudian lakukan instalasi.

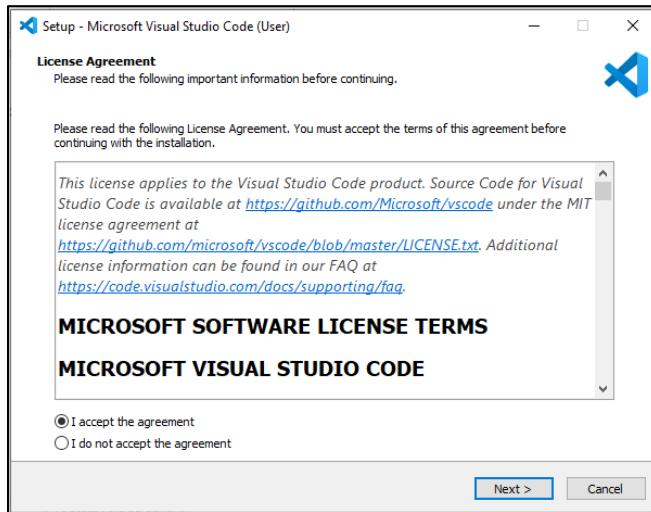


Gambar 3.6 Download .NET SDK x64
(sumber : <https://code.visualstudio.com/>)

Untuk mendownload VS Code bisa melalui halaman resmi di <https://code.visualstudio.com/> dan silahkan kalian pilih sistem operasi yang kalian gunakan. Setelah berhasil *download*, maka bisa dilanjutkan ke proses instalasi :

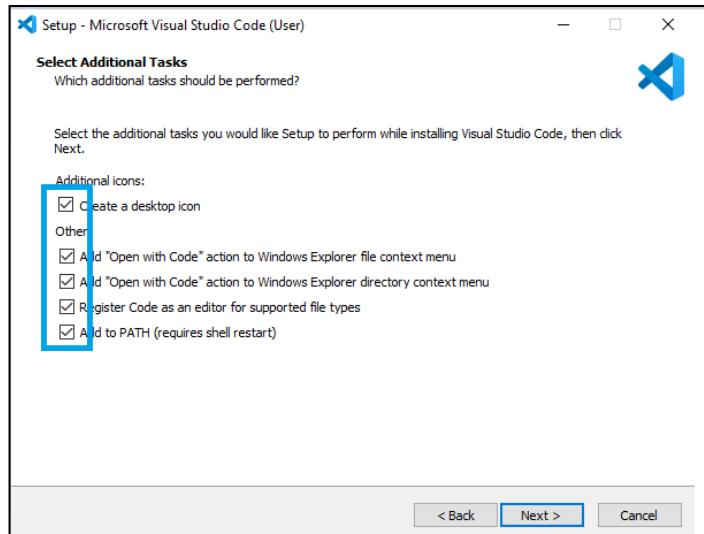
- a. Double klik file installernya atau klik kanan kemudian pilih Run as Administrator.
- b. Klik "Yes", jika muncul peringatan "Run As Administrator".
- c. Pilih "I accept the agreement" untuk menyetujui "License Agreement". Lalu klik "Next"





Gambar 3.7 License Agreement VS Code
(sumber : Microsoft Visual Studio Code)

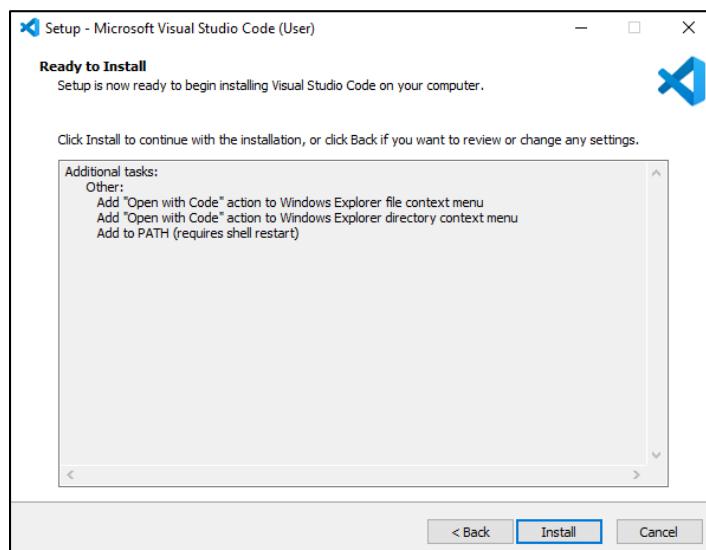
- d. Pada Select Destination Location bisa dibiarkan saja jika lokasi instalasi tidak akan dirubah. Lalu klik Next
- e. Jika tidak ingin merubah Start Menu Folder, klik Next.
- f. Pada bagian Select Additional Tasks centang semua. Klik Next.



Gambar 3.8 Select Additional Tasks
(sumber : Microsoft Visual Studio Code)

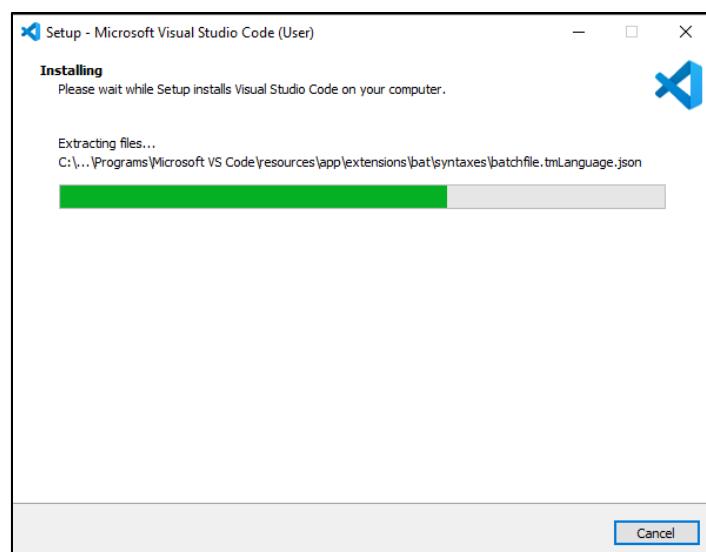


g. Untuk memulai instalasi silahkan pilih klik Install



Gambar 3.9 Ready to Install
(sumber : Microsoft Visual Studio Code)

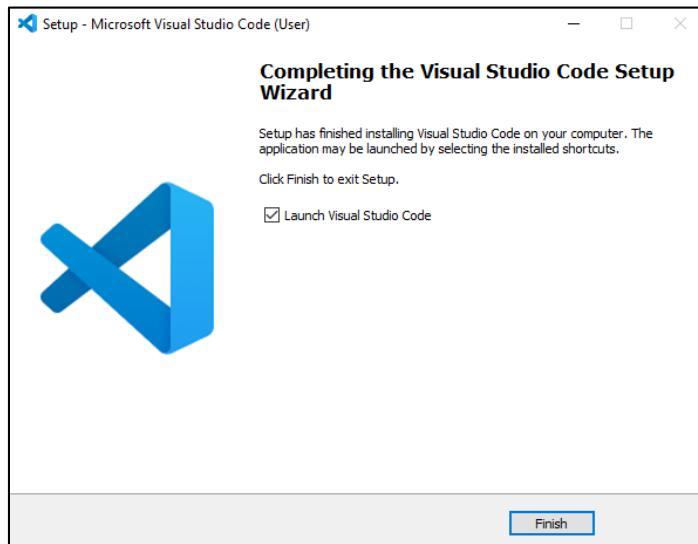
h. Tunggu sampai proses instalasi selesai



Gambar 3.10 Proses Install (sumber : Microsoft Visual Studio Code)



i. Instalasi selesai. Klik Finish.

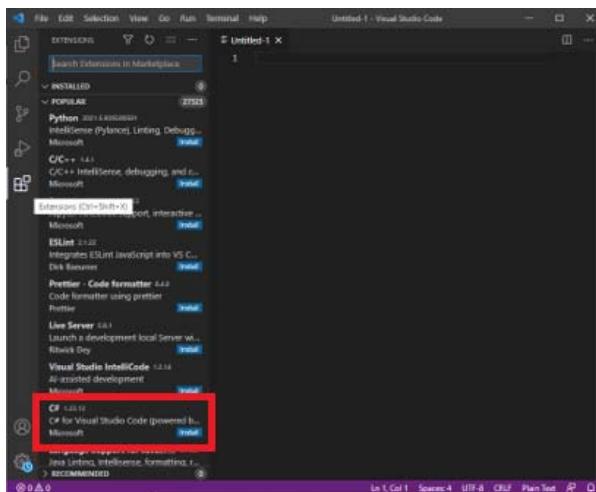


Gambar 3.11 Instalasi selesai (sumber : Microsoft Visual Studio Code)

Setelah instalasi VS Code selesai. Maka bisa dilanjutkan dengan install Extensions di VS Code. Berikut ini langkah-langkah install Extensions C# :

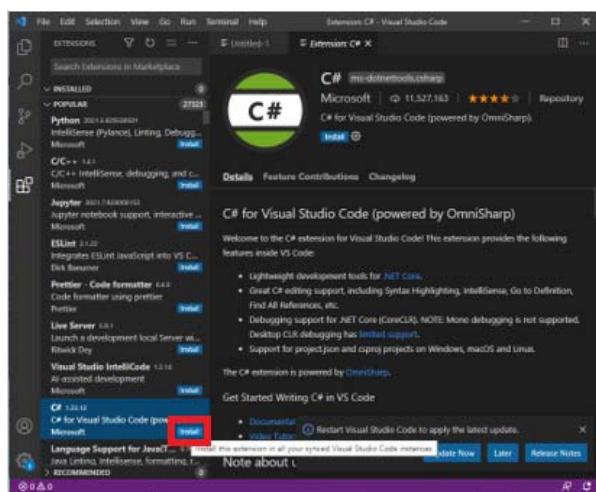
- a. Buka Visual Studio Code (VS Code)
- b. Klik tab Extensions yang ada di sebelah kiri. Kemudian ketik "C#" di pencarian
- c. Pilih C#





Gambar 3.12 Pilih Extensions C# (sumber : Microsoft Visual Studio Code)

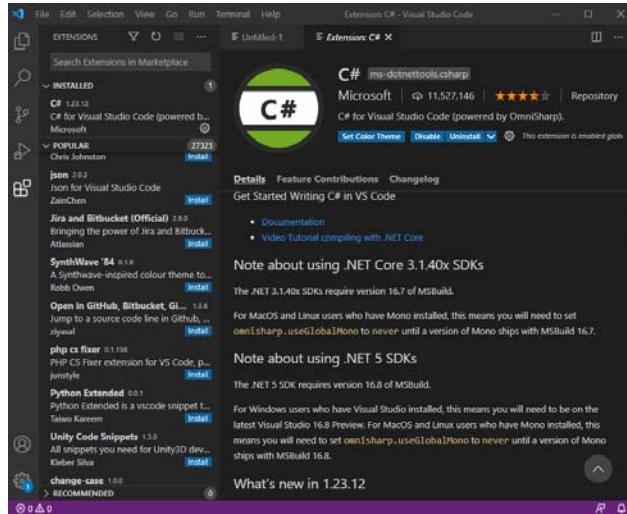
d. Kemudian klik install pada extensions C#



Gambar 3.13 Install Extensions C# (sumber : Microsoft Visual Studio Code)



e. Proses instalasi extensions C# selesai.



Gambar 3.14 Hasil instalasi C# (sumber : Microsoft Visual Studio Code)

f. Lalu close aplikasi VS Code.

3. Menjalankan Aplikasi VS Code

a. Buat sebuah folder di lokasi manapun di dalam direktori komputer anda. Jika kalian menggunakan sistem operasi *Windows*, buka folder tersebut, lalu ketik *cmd* di *directory bar* pada *Windows Explorer* kemudian tekan enter untuk membuka *command prompt* pada direktori tersebut.



Gambar 3.15 Pembuatan folder



```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

E:\Belajar C#>
```

Gambar 3.16. Tampilan command prompt di *Windows*

b. Ketik **dotnet new console** pada *command prompt* untuk membuat aplikasi *console* di .NET

```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

E:\Belajar C#>dotnet new console
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on E:\Belajar C#\Belajar C#.csproj...
  Determining projects to restore...
    Restored E:\Belajar C#\Belajar C#.csproj (in 180 ms).
Restore succeeded.
```

Gambar 3.17 Tampilan perintah dotnet new console

c. **Setelah** selesai ketik **code .** pada jendela *command prompt* untuk membuka Visual Studio Code

```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

E:\Belajar C#>dotnet new console
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on E:\Belajar C#\Belajar C#.csproj...
  Determining projects to restore...
    Restored E:\Belajar C#\Belajar C#.csproj (in 180 ms).
Restore succeeded.

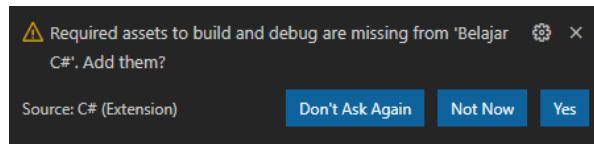
E:\Belajar C#>code .

E:\Belajar C#>
```

Gambar 3.18 Tampilan perintah code .

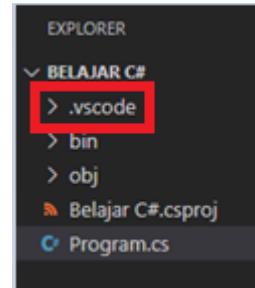
d. Setelah VS Code terbuka, maka muncul pesan agar kalian menambahkan aset yang dibutuhkan untuk debugging seperti gambar di bawah ini. Tekan tombol “Yes”.





Gambar 3.19 Tampilan Required Assets

e. Akan muncul folder .vscode di panel *Explorer* di sisi kiri editor.



Gambar 3.20 Tampilan folder .vscode

f. Menulis kode program

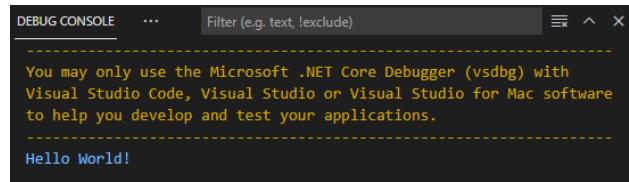
Pilih file Program.cs di panel *Explorer*. File tersebut berisi kode dasar untuk mencetak kalimat "*Hello World*".

```
Program.cs
using System;
namespace Belajar_C_ {
    class Program {
        static void Main(string[] args) {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Gambar 3.21 Tampilan file Program.cs

g. Jalankan program tersebut dengan menekan tombol **Ctrl+F5** pada VS Code untuk menjalankan program tanpa debugging. Dan akan menampilkan output seperti gambar di bawah ini :





```
DEBUG CONSOLE ... Filter (e.g. text, | exclude) ⌂ ⌄ X
You may only use the Microsoft .NET Core Debugger (vsdbg) with
Visual Studio Code, Visual Studio or Visual Studio for Mac software
to help you develop and test your applications.
-----
Hello World!
```

Gambar 3.22 Output Hello World!

C. Elemen Dasar C#

1. Struktur Penulisan Kode C#

Kode program pada C# diawali dengan mendeklarasikan nama *class*, seperti pada contoh di bawah ini :

```
class Program
```

Kemudian ditambahkan tanda "{" sebagai tanda awal kode dan pada akhir kode ditutup dengan tanda "}".

```
class Program
{
}
```

C# dibangun oleh satu atau beberapa fungsi yang diletakkan dalam *class*. Nama fungsi pada C# harus diawali dengan huruf, garis bawah "_" dan diikuti oleh huruf, angka atau garis bawah. Pada akhir nama fungsi ditambahkan tanda kurung buka dan kurung tutup "()". Untuk memberi nama fungsi tidak boleh diberi spasi. Tanda "{ }" merupakan tanda awal dan akhir sebuah fungsi. Berikut ini merupakan contoh penamaan fungsi yang diletakkan di dalam *class*.

```
class HelloWorld {
    nama fungsi(){}
}
```



Fungsi utama yang dipakai dalam C# adalah Main(). Contoh penulisan fungsi Main() dalam kode program :

```
public static void Main (string[] args)
```

Sebuah fungsi berisi kumpulan-kumpulan perintah yang diakhiri dengan tanda ";".

```
class HelloWorld {  
    // Bagian utama program C#  
    public static void Main() {  
        System.Console.WriteLine("Hello, World");  
    }  
}
```

2. Penulisan Komentar

Ada dua acara untuk menuliskan komentar pada C#, yaitu :

- Komentar satu baris menggunakan tanda "//". Seperti contoh di bawah ini :

```
//using System;
```

- Komentar multi baris, biasa digunakan untuk penulisan komentar yang panjang. Penulisan komentar tersebut diawali dengan menggunakan tanda "/*" dan diakhiri dengan tanda "*/". Seperti contoh di bawah ini :

```
/*class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine("Hello World!");  
    }  
}*/
```



3. Karakter khusus dalam C#

Berikut ini tabel karakter khusus pada C#:

Tabel 3.1 Daftar Karakter Khusus C#

Escape Sequence	Keterangan
\a	Menghasilkan suara (beep)
\b	Mengembalikan posisi kursor ke sebelumnya
\t	Meletakkan posisi kursor di pemberhentian tab berikutnya
\n	Meletakkan posisi kursor pada baris baru
\v	Melakukan tab secara vertikal
\r	Memberikan nilai enter
\"	Menampilkan double quote ("")
\'	Menampilkan apostrophe (')
\?	Menampilkan tanda tanya
\\\	Menampilkan blackslash (\)
\0	Menampilkan karakter null

4. Tipe Data

Sekarang silahkan lihat kondisi rumah atau bangunan sekitar kalian. Saat proses pembangunan rumah ataupun gedung perhatikan di sekitar lokasi pasti akan banyak dijumpai bahan atau material bangunan yang bermacam-macam yang ditempatkan berdasarkan jenis dan tidak tercampur dalam satu tempat penyimpanan yang sama. Hal ini dimaksudkan agar mudah dalam pengambilan sesuai kebutuhan bahan atau material bangunan. Selain itu tempat penyimpanan dibedakan agar tidak mudah terkontaminasi bahan lain sehingga mempengaruhi kualitas bahan atau material bangunan yang berimbang juga pada kualitas bangunan itu sendiri.





Gambar 3.23 Lokasi Pembangunan
(sumber <http://pasarbatang.desa.id/>)

Dari penjelasan diatas dalam permograman komputer mempunyai kasus yang sama. Dalam pemrograman komputer bahan atau material bangunan bernama tipe data. Yang mana tipe data merupakan material dasar dalam membentuk sebuah rumah yang bernama program. Tipe data juga memiliki karakteristik yang berbeda setiap karakternya sehingga hasil yang dikeluarkan juga berbeda. Tipe data pada pemrograman C# memiliki dua jenis yaitu tipe dasar dan tipe *reference*. Tipe dasar sendiri terbagi atas tipe numerik dan tipe tekstual.

a. Tipe dasar

Dalam karakteristiknya tipe data numerik memiliki 4 tipe *primitive* yaitu *Byte*, *short*, *int*, *long* dan 2 tipe *primitive* untuk bilangan pecahan yaitu *float* dan *double*. Adapun perbedaan dari setiap tipe tersebut adalah dalam pengelolaannya di dalam memori komputer. Dalam memori komputer data disimpan dalam bentuk bit. Bit terdiri dari bilangan 0 dan 1. Dikatakan 1 bit jika terdiri dari 0 dan 1. Kondisi 2 bit jika memungkinkan untuk 4 kombinasi yang berbeda seperti tabel dibawah ini.



Tabel 3.2 Empat kombinasi untuk kondisi 2 bit

A	B
0	0
0	1
1	0
1	1

Dan kondisi 2 bit dapat dihitung dalam bentuk 2^2 . Begitu juga dengan 3 bit dan seterusnya. Khusus untuk kondisi 8 bit bisa disebut dengan 1 Byte karena memiliki 256 kemungkinan kombinasi dan dapat dihitung dengan 2^8 . Adapun dalam byte nilai yang dapat disimpan dari -128 sampai 127. -128 nilai sebelum 0 dan 127 nilai setelah 0. Untuk lebih lanjutnya silahkan dilihat tabel dibawah ini.

Tabel 3.3 Tabel jenis tipe data

Tipe	Panjang	Nilai yang dapat disimpan	Nilai minimal	Nilai tertinggi
Byte	8 bit	2^8 , atau... 256	- 2^7 , atau... -128	2^7-1 , atau... 127
short	16 bit	2^{16} , atau... 65,535	- 2^{15} , atau... -32,768	$2^{15}-1$, atau... 32,767
int	32 bit	2^{32} , atau... 4,294,967,296	- 2^{31} , atau... - 2,147,483,648	$2^{31}-1$, atau... 2,147,483,647
long	64 bit	2^{64} , atau ... 18,446,744,073,709,551,616	- 2^{63} , atau... - 9,223,372,036, 854,775,808L	2^{63} , atau... 9,223,372,036, 854,775,808L

Tipe data *primitive* diatas merupakan tipe data numerik *primitive* untuk bilangan bulat. Dalam penggunaanya tipe data numerik *primitive* yang paling tinggi akurasinya dalam bilangan bulat adalah tipe data int. Selain itu dalam tipe data numerik



untuk bilangan pecahan menggunakan *float* dan *double* yang mana memiliki perbedaan seperti tabel dibawah ini.

Tabel 3.4 Tabel float dan double

Tipe	Panjang	Nilai minimal	Nilai Maksimal
Float	32 bit	$-3,4 \times 10^{-38}$	$3,4 \times 10^{38}$
Double	64 bit	$-1,7 \times 10^{-308}$	$1,7 \times 10^{308}$

Dari tipe data *float* dan *double* untuk akurasi *range* yang terbaik adalah tipe data *double*. Dalam memudahkan pemrograman dan perhitungan nilai pecahan sering dijumpai menggunakan tipe data *double*.

Tipe tekstual berbeda dengan tipe *numeric* yaitu berfungsi menuliskan karakter seperti huruf. Tipe data tekstual yang digunakan yaitu *char*. Dalam penggunaan satu karakter *char* membutuhkan memori 16 bit. Namun tipe data ini terbatas hanya menampilkan satu karakter saja. Jika membutuhkan lebih dari satu karakter maka harus menggunakan objek *string*. *String* merupakan objek yang spesial karena dapat digunakan ke berbagai objek yang lain. Nama lain dari *string* adalah penggabung, karena dapat menggabungkan berbagai objek yang berbeda.



Pada pengaplikasian tipe data dasar pemrograman C# pembacaan tipe data dapat dilihat dibawah ini :

Tabel 3.5 Pembacaan tipe data (Ali, 2014:7)

Type	Deskripsi	Deklarasi	Keterangan
bool	Boolean	<i>True, false</i>	Memberikan nilai true/false
char	16 bit <i>Unicode character</i>	'A', '\x0041', '\u0041	Karakter
sbyte	8 bit <i>signed integer</i>		
byte	8 bit <i>unsigned integer</i>		
short	16 bit <i>signed integer</i>		Bilangan bulat pendek
ushort	16 bit <i>unsigned integer</i>		Bilangan bulat pendek yang tidak bertanda
int	32 bit <i>signed integer</i>	Suffix:U	Bilangan bulat
uint	32 bit <i>unsigned integer</i>	Suffix:L/I	Bilangan bulat tak bertanda
long	64 bit <i>signed integer</i>	Suffix:U/u eller L/I	Bilangan bulat panjang
ulong	64 bit <i>unsigned integer</i>	Suffix:F/f	Bilangan panjang bulat tak bertanda
float	32 bit <i>floating-point number</i>	Suffix:D/d	Float bernilai presisi tunggal
double	64 bit <i>floating-point number</i>	Suffix:M/m	Float bernilai presisi ganda
Decimal	96 bit <i>decimal number</i>	"C:\\test.txt;@ "C:\\test.txt"	Bilangan desimal
string	<i>Character string (text)</i>		Gabungan dari beberapa karakter

Tipe data string berbeda dengan yang lainnya, sehingga dapat disebut dengan tipe data referensi. Nilai string dimulai dengan karakter @, sehingga karakter *escape* tidak akan diterjemahkan. Karakter *escape* merupakan karakter string khusus yang selalu dimulai dengan simbol \ dan diikuti karakter string dibelakangnya. Contoh karakter *escape* \f.



b. Tipe data referensi

Tipe data referensi tidak berisi data namun menyimpan data sebagai referensi. Dalam tipe data referensi terdapat tiga tipe bawaan yaitu tipe data string, objek, dan dinamis. Selain itu dapat mendeklarasikan *class* delegasi dan *interface*.

Tipe data string sangat special karena selain masuk tipe data kontekstual dalam tipe data bahasa pemrograman komputer lainnya, namun *string* juga dapat masuk tipe data referensi dalam bahasa pemrograman C#. Hal ini dikarenakan dalam pemrograman C# sering ditemui kebutuhan penyimpanan data teks dan disistem namespace menggunakan string sebagai kata kunci. Tipe *string* dapat bernilai nol atau diatur ke nol.

Kelas objek di deklarasikan pada *system namespace* pada bahasa C#. Hal ini dikarenakan merupakan classpaling dasar untuk kelas lainnya. Maksudnya tipe nilai saat digunakan dan tipe referensi yang ditentukan pengguna diturunkan dari kelas objek yang juga dapat menggunakan nama sebutan objek. Hal ini dikarenakan objek merupakan tipe dasar untuk semua tipe nilai. Sehingga dapat dikonversikan menjadi variabel dari tipe data lain ke tipe objek begitu sebaliknya. Hal ini yang dinamakan *boxing*(yang pertama) dan *unboxing*(selanjutnya), contoh penggunaan :

```
int a = 3;  
object aBoxing = a;  
int aUnboxing = (int)aBoxing;
```



Bahasa pemrograman C# mendukung deklarasi *class* bersamaan dengan konstruktor, *finalizer*, konstanta, *fields*, *properties*, *index*, *events*, *methods*, dan operator sebagai delegasi. Pada class juga mendukung pewarisan dan interface. Pada class dapat menerapkan satu interface atau lebih. Yang artinya *class* harus menggunakan semua *methods*, *event*, dan *index* yang ditentukan dalam interface. Sehingga dalam menggunakan *interface* C# cukup menggunakan *keyword interface*. Dan referensi delegasi memungkinkan pengesahan spesifikasi dari sebuah method.

5. Variabel

Variabel merupakan tempat dimana menampung data di memori, data tersebut mempunyai nilai dan dapat dirubah selama program masih dalam proses eksekusi. Contoh dalam menggunakan variabel :

```
int id_pegawai;  
char nama_depan;
```

Pengunaan simbol (_) sebagai penghubung dalam variabel karena spasi tidak bisa digunakan. Dan dalam variabel tetap menggunakan tipe data terlebih dahulu sebagai penentu tipe dari variabel tersebut.

6. Konstanta

Berbeda dengan variabel diatas, konstanta bernilai tetap yang berarti tidak dapat dirubah. Konstanta sendiri memiliki lima macam yaitu sebagai berikut:



a. Konstanta Oktal

Penulisan konstanta oktal selalu diawali dengan nilai 0. Karena dalam oktal yang digunakan 0 sampai 7. Contoh penulisan 010 yang berarti dalam desimal adalah 8.

b. Konstanta Heksadesimal

Pada heksadesimal menggunakan simbol angka dan huruf yang dalam angka dimulai dari 0-9 dan huruf A-F. Yang mana A-F dalam desimal seperti tabel dibawah ini.

Tabel 3.6 Konstanta Heksadesimal

Heksadesimal	Desimal
A	10
B	11
C	12
D	13
E	14
F	15

Penulisan konstanta heksa diawali dengan nol dan x, contoh : 0xA1 dalam desimal adalah 161.

c. Konstanta String

Konstanta yang dalam penulisannya menggunakan awalan dan akhiran ("") petik ganda sebagai pembeda dengan konstanta karakter. Dalam konstanta string dapat diisi menggunakan karakter *escape*. Contoh : "Hello World"

d. Konstanta Bernama

Memungkinkan dalam mendefinisikan konstanta dengan menggunakan kata kunci const. Contoh : const float phi = 3,14. Jadi phi konstanta bernama bernilai 3,14.



7. Operator

Dalam pengoperasian dibutuhkan simbol yang memiliki fungsi sebagai pemanipulasi agar variabel dan nilai berjalan dengan signifikan. Operator memiliki tiga sifat yaitu unary, binary, dan ternary. Operator unary hanya melibatkan satu operator saja contoh -3 . Binary melibatkan dua operator contoh $3+6$. Sedangkan Ternary melibatkan tiga operator contoh $(1<2)?1:2$. Adapun simbol operator (Kadir, 1995:101) akan dijelaskan dibawah ini.

Tabel 3.7 Symbol Operator

Simbol	Keterangan
::	Resolusi Lingkup
++	Post-Increment (naik +1)
--	Post-Decrement (turun -1)
()	Pemanggil fungsi
[]	Elemen Array
->	Pointer ke anggota struktur
.	Anggota struktur
!	Pre-Increment
-	Pre-Decrement
~	Bitwise Komplemen
+	Unary Plus
&	Alamat
*	Indirection
new	Pengalokasian memory
delete	Deallocasi memory
sizeof	Ukuran Tipe Data
(tipe)	Type Casting
tipe()	Type Casting
*	Perkalian
/	Pembagian
%	Sisa pembagian
+	Penjumlahan
-	Pengurangan
<<	Geser kiri
>>	Geser kanan
<	Kurang dari
>	Lebih dari
<=	Kurang dari sama dengan
>=	Lebih dari sama dengan
==	Sama dengan

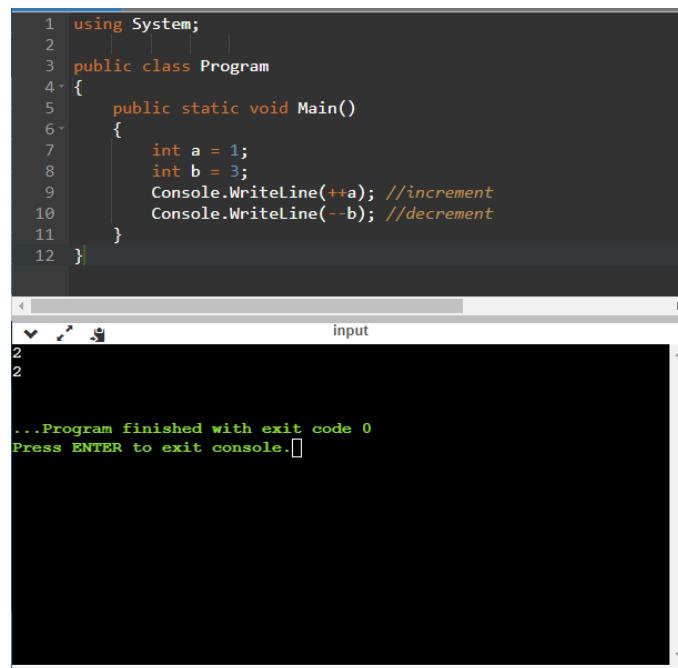


!=	Tidak sama dengan
&	Bitwise dan (AND)
^	Bitwise exclusive OR (XOR)
	Bitwise atau (OR)
&&	Logika dan (AND)
	Logika atau (OR)
?:	Operator kondisi
= *= /= %= += -= <<= >>= &= ^= = ,	Operator Majemuk
,	Operator Koma

Setiap operator diatas walaupun dengan simbol yang sama namun dapat memberikan hasil yang berbeda. Hal ini mengharuskan penggunaan operator harus disesuaikan dengan keadaan dan kasus yang sesuai. Contoh simbol – pada unary aritmatika digunakan sebagai penanda minus misalnya -3. Sedangkan dalam aritmatika biasa digunakan sebagai simbol pengurangan misalnya 3-1. Dan terdapat operator yang sama namun beda nama seperti ++ dan --. Hal ini dimaksudkan adalah pembacaan. Post-increment berarti pembacaan dari kiri contoh ++a. Dan pre-increment pembacaan sebaliknya contoh a++. Fungsinya sama yaitu penaikan namun jika diaplikasikan dalam program menghasilkan eksekusi berbeda. Operator perbandingan seperti < kurang dari dan >. Berfungsi untuk membandingkan suatu nilai. Dan Operator bitwise seperti >>, <<, AND, OR,XOR merupakan untuk keperluan manipulasi data bit. Operator majemuk memiliki banyak sekali simbol, fungsinya adalah



menyederhanakan penulisan perintah. Contoh operasi majemuk $X = X+1$ jika diberi operasi majemuk $X +=1$.



```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int a = 1;
8         int b = 3;
9         Console.WriteLine(++a); //increment
10        Console.WriteLine(--b); //decrement
11    }
12 }
```

input
2
2
...Program finished with exit code 0
Press ENTER to exit console.█

Gambar 3.24 Contoh Konstanta Bernama

Tugas Mandiri

Buatlah sebuah variabel dengan minimal 5 tipe data yang berbeda lalu jelaskan dengan bahasa kalian maksud dari tipe data tersebut dalam gabungan variabel.



Diskusi Kelompok

Setiap tipe data memiliki karakter dan fungsi yang berbeda namun tidak memungkiri perintah dalam program masih dapat dijalankan walaupun variabel yang digunakan berbeda tipe data. Hal ini dikarenakan dalam tipe data terdapat perintah konversi yang memungkinkan program dapat dijalankan walaupun dengan tipe data yang berbeda.

Buatlah kelompok kecil, analisis mengenai perintah konversi tipe data beserta macamnya, bagaimana cara kerja, alur algoritma, dan macam-macam tipe data yang dapat konversi. Presentasikan hasil diskusi kelompok kalian di kelas.

Uji Kompetensi

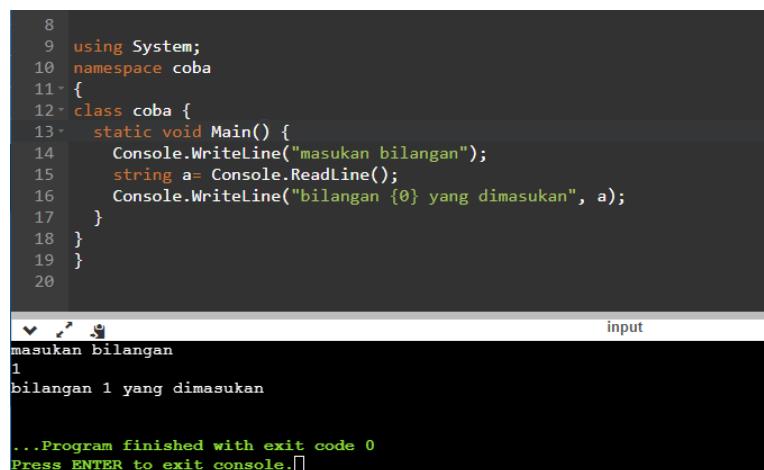
1. Dalam tipe data pecahan terdapat float dan double, jelaskan dengan bahasa kalian kondisi yang tepat menggunakan float dan double!
2. Jelaskan dengan bahasa kalian maksud dari `a++` dan `++a` pada operator increment beserta hasilnya!
3. Jelaskan dengan bahasa kalian maksud dari `a--` dan `--a` pada operator decrement beserta hasilnya!
4. Jelaskan dengan bahasa kalian cara-cara dalam konversi tipe data?
5. Jelaskan pendapatmu jika terdapat perintah yang dilakukan menggunakan konstanta dan tipe data yang berbeda dan bagaimana langkah penyelesaiannya!



D. Membuat Program Sederhana

Program sederhana yang akan dibuat adalah menggunakan fungsi output dan input pada bahasa pemrograman C#. Fungsi output menggunakan perintah Console.WriteLine() perintah ini adalah sebagai perintah menampilkan hasil. Ada juga Console.Write() hasilnya sama untuk menampilkan hasil namun perbedaanya adalah jika Console.WriteLine() maka akan muncul garis baru jika Console.Write() tidak.

Perintah input ada Console.ReadLine(), Console.Read(), dan Console.ReadKey(). Perbedaanya terletak pada pembacaan input yang dimasukan. Console.ReadLine() akan membaca seluruh teks dalam satu baris yang dikeluarkan. Console.Read() akan membaca satu huruf saja yang dimasukan dan menghasilkan tipe data int. Console.ReadKey() sama seperti Console.Read() namun tipe data yang dihasilkan berupa char.



```
8
9 using System;
10 namespace coba
11 {
12 class coba {
13 static void Main() {
14     Console.WriteLine("masukan bilangan");
15     string a= Console.ReadLine();
16     Console.WriteLine("bilangan {0} yang dimasukan", a);
17 }
18 }
19 }
20
```

masukan bilangan
1
bilangan 1 yang dimasukan

...Program finished with exit code 0
Press ENTER to exit console. █

Gambar 3.25 Contoh Menggunakan Input dan Output



Tugas Mandiri

Buatlah sebuah pengisian biodata sederhana yang terdiri dari mengisi nama, alamat, tempat dan tanggal lahir, dan hobi.

Diskusi Kelompok

Dalam perintah input serta output terdapat perintah yang hampir mirip namun mempunyai hasil yang berbeda

Buatlah kelompok kecil, analisis perbedaan pada setiap perintah dan terapkan pada program kalian. Presentasikan hasil program dan diskusi kelompok kalian di kelas.

Uji Kompetensi

1. Jelaskan dengan bahasa kalian tentang fungsi perintah input dan output dalam bahasa C#!
2. Jelaskan dengan bahasa kalian perbedaan yang signifikan antara `console.writeline()` dengan `console.write()`!
3. Jelaskan dengan bahasa kalian keunggulan dari `console.writeline()`!
4. Jelaskan dengan bahasa kalian perbedaan yang signifikan antara `console.read()` dengan `console.readkey()`!
5. Jelaskan pendapatmu kondisi dalam penggunaan `console.readkey()`!



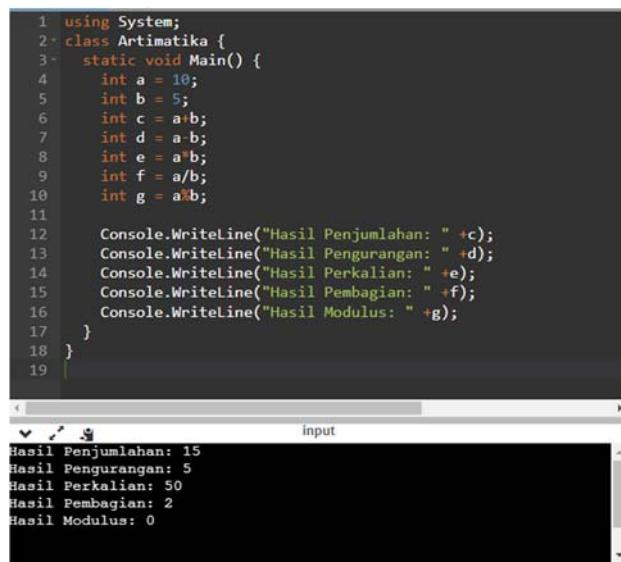
E. Operator Aritmatika dan Logika

1. Operasi Aritmatika

Operasi aritmatika terdiri dari beberapa operator yaitu penjumlahan (+), pengurangan (-), pembagian (/), perkalian (*), dan modulus (%), untuk lebih lanjut seperti tabel dibawah ini.

Tabel 3.8 Operator Aritmatika

Simbol	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa pembagian
<code>+=</code>	Penjumlahan
<code>-=</code>	Pengurangan
<code>*=</code>	Perkalian
<code>/=</code>	Pembagian
<code>%=</code>	Sisa pembagian



```
1 using System;
2 class Aritmatika {
3     static void Main() {
4         int a = 10;
5         int b = 5;
6         int c = a+b;
7         int d = a-b;
8         int e = a*b;
9         int f = a/b;
10        int g = a%b;
11
12        Console.WriteLine("Hasil Penjumlahan: " +c);
13        Console.WriteLine("Hasil Pengurangan: " +d);
14        Console.WriteLine("Hasil Perkalian: " +e);
15        Console.WriteLine("Hasil Pembagian: " +f);
16        Console.WriteLine("Hasil Modulus: " +g);
17    }
18 }
19
```

Hasil Penjumlahan: 15
Hasil Pengurangan: 5
Hasil Perkalian: 50
Hasil Pembagian: 2
Hasil Modulus: 0

Gambar 3.26 Contoh Penggunaan Operasi Aritmatika +, -, *, / dan %



Maksud dari program diatas adalah c merupakan hasil penjumlahan dari variabel a dan b ($10+5=15$), sedangkan d merupakan hasil pengurangan dari variabel a dan b ($10-5=5$), e merupakan hasil perkalian variabel a dan b ($10\times5=50$), f merupakan hasil pembagian variabel a dan b ($10:5=2$), g merupakan hasil modulus (sisa pembagian) dari variabel a dan b ($10\%5=0$).

2. Operasi Logika

Operasi logika digunakan untuk menggabungkan dan mengolah pernyataan. Untuk lebih jelas silahkan perhatikan tabel dibawah ini.

Tabel 3.9 Tabel Operasi Logika

!	Logika not
&&	Logika dan (AND)
	Logika atau (OR)

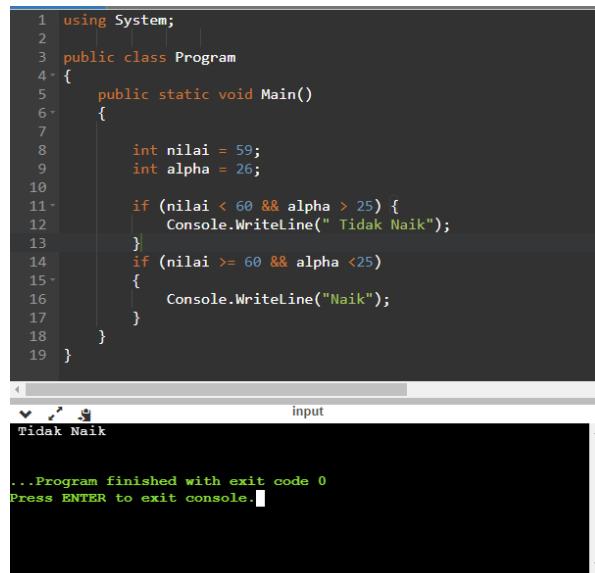
Pada operasi ini berlaku tabel kebenaran seperti dibawah ini.

a. Operator And

Tabel 3.10 Tabel Kebenaran Operator And

A	B	Hasil
False	False	False
False	True	False
True	False	False
True	True	True





```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7
8         int nilai = 59;
9         int alpha = 26;
10
11         if (nilai < 60 && alpha > 25) {
12             Console.WriteLine(" Tidak Naik");
13         }
14         if (nilai >= 60 && alpha < 25)
15         {
16             Console.WriteLine("Naik");
17         }
18     }
19 }
```

Gambar 3.27 Penggunaan And

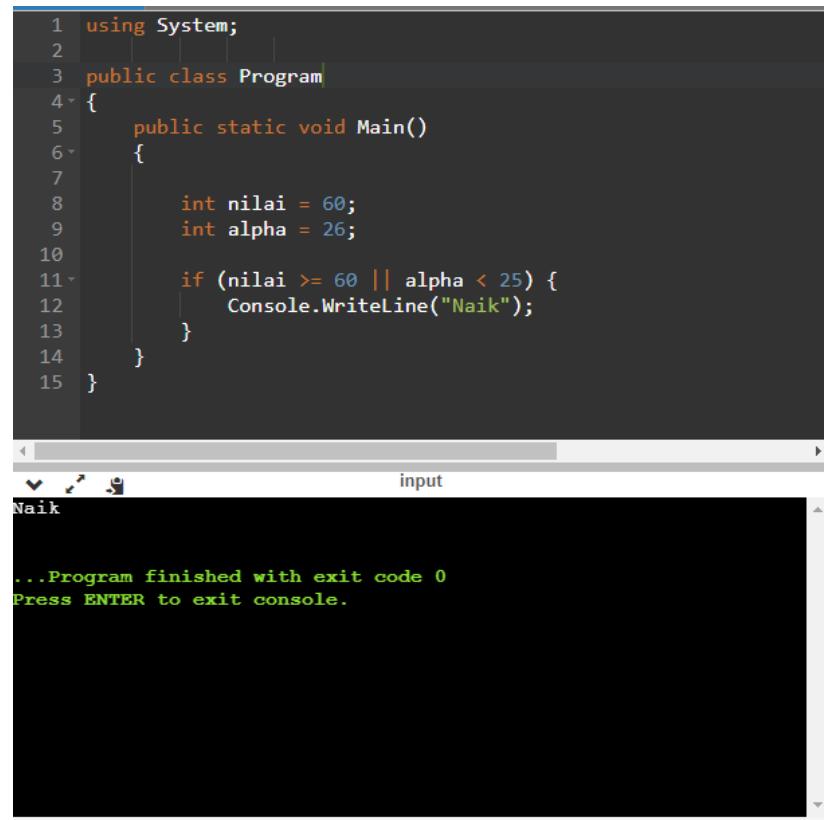
Maksud dari program diatas adalah jika kondisi a lebih dari 5 dan a kurang dari 21 maka bersifat true dan dikeluarkan kalimat Lebih dari 5 kurang dari 21.

b. Operator Or

Tabel 3.11 Tabel Kebenaran Operator OR

A	B	Hasil
False	False	False
False	True	True
True	False	True
True	True	True





```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7
8         int nilai = 60;
9         int alpha = 26;
10
11        if (nilai >= 60 || alpha < 25) {
12            Console.WriteLine("Naik");
13        }
14    }
15 }
```

input
Naik

...Program finished with exit code 0
Press ENTER to exit console.

Gambar 3.28 Pengaplikasian OR

Dari contoh diatas jika nilai lebih dari sama dengan 60 atau alpha kurang dari 25 maka bersifat true dan dikeluarkan pernyataan naik.

Khusus operasi not kebalikan dari operasi yang lain. Misalnya not and berarti kebalikan dari tabel kebenaran and dan not or kebalikan dari operasi not.



Tugas Mandiri

Buatlah perintah sederhana dengan menggunakan gerbang logika and untuk kondisi nilai syarat nilai naik kelas pada setiap mata pelajaran. Mata pelajaran yang diisikan sesuai dengan yang berlaku di semester yang sedang berlangsung dan nilai kkm yang digunakan.

Diskusi Kelompok

Setiap kenaikan kelas pasti memiliki syarat yang harus dipenuhi oleh siswa, yaitu nilai mata pelajaran dan jumlah ketentuan maksimal alpha dalam satu tahun.

Buatlah kelompok kecil, analisis syarat kkm sesuai dengan yang berlaku disekolah dengan jumlah mata pelajaran yang di masukan sesuai dengan yang digunakan di sekolah. Buat sebuah program sederhana dengan operasi logika dan presentasikan di kelas.

Uji Kompetensi

1. Jelaskan kondisi pada operasi logika and dengan bahasa kalian!
2. Jelaskan kondisi pada operasi logika or dengan bahasa kalian!
3. Jelaskan kondisi operasi logika not dengan bahasa kalian!
4. Jelaskan fungsi dari operasi aritmatika dengan bahasa kalian?
5. Jelaskan penyederhanaan penulisan operator pada operasi aritmatika!



F. Struktur Kontrol

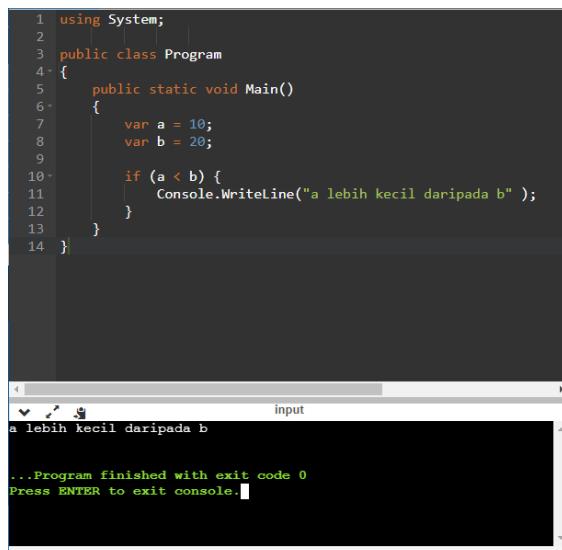
Agar program berjalan dengan lancar dan tidak saling bertabrakan maka dibutuhkan pernyataan yang berfungsi sebagai kondisi yang digunakan sebagai kontrol. Struktur kontrol dalam pemrograman C# dibagi menjadi dua yaitu struktur kontrol percabangan dan struktur kontrol perulangan

1. Struktur kontrol percabangan

Struktur kontrol percabangan digunakan untuk mengontrol sebuah kondisi yang membutuhkan sebuah aksi. Aksi tersebut muncul untuk menentukan kearahmana program yang akan dieksekusi selanjutnya. Perintah percabangan yang digunakan akan dibawah ini.

a. If

Perintah ini digunakan dalam satu kondisi benar.



The image shows a code editor window with the following C# code:

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         var a = 10;
8         var b = 20;
9
10        if (a < b) {
11            Console.WriteLine("a lebih kecil daripada b");
12        }
13    }
14 }
```

Below the code editor is a terminal window showing the output of the program. The input 'a lebih kecil daripada b' is typed into the terminal, and the output shows the message 'a lebih kecil daripada b' followed by the message '...Program finished with exit code 0' and 'Press ENTER to exit console.'

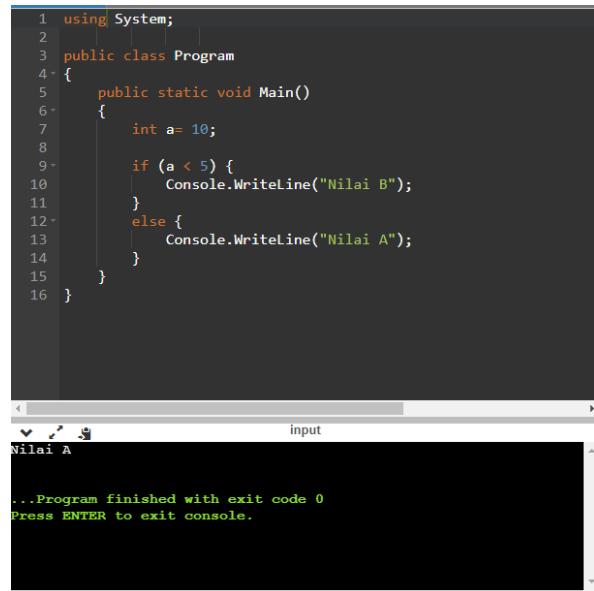
Gambar 3.29 Contoh Percabangan If

Dari contoh diatas kondisi true yang berlaku adalah satu kondisi. Jika a lebih kecil dari pada b maka keluaran berupa kalimat a lebih kecil daripada b.



b. If-Else

Perintah ini digunakan jika terdapat dua kondisi yang berbeda.



The image shows a code editor with the following C# code:

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int a= 10;
8
9         if (a < 5) {
10            Console.WriteLine("Nilai B");
11        }
12        else {
13            Console.WriteLine("Nilai A");
14        }
15    }
16 }
```

Below the code editor is a terminal window with the following output:

```
Nilai A
...Program finished with exit code 0
Press ENTER to exit console.
```

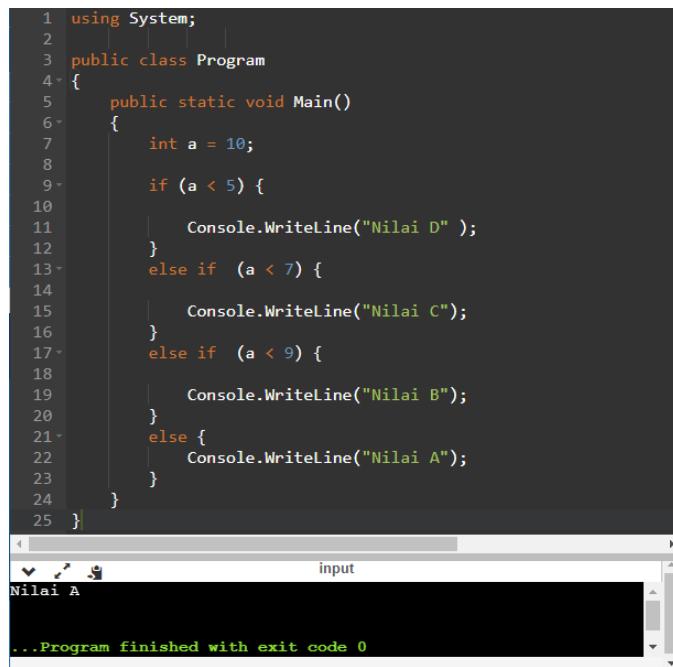
Gambar 3.30 Contoh Percabangan If-Else

Maksud perintah diatas jika kondisi nilai a kurang dari 5 maka akan mendapat nilai B, jika tidak maka nilainya A.

c. If-Else If

Berbeda dengan fungsi perintah diatas di fungsi perintah dapat digunakan untuk kondisi bercabang lebih dari dua atau bisa disebut percabangan bersarang.





```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int a = 10;
8
9         if (a < 5) {
10
11             Console.WriteLine("Nilai D" );
12         }
13         else if (a < 7) {
14
15             Console.WriteLine("Nilai C");
16         }
17         else if (a < 9) {
18
19             Console.WriteLine("Nilai B");
20         }
21         else {
22
23             Console.WriteLine("Nilai A");
24         }
25     }
26 }
```

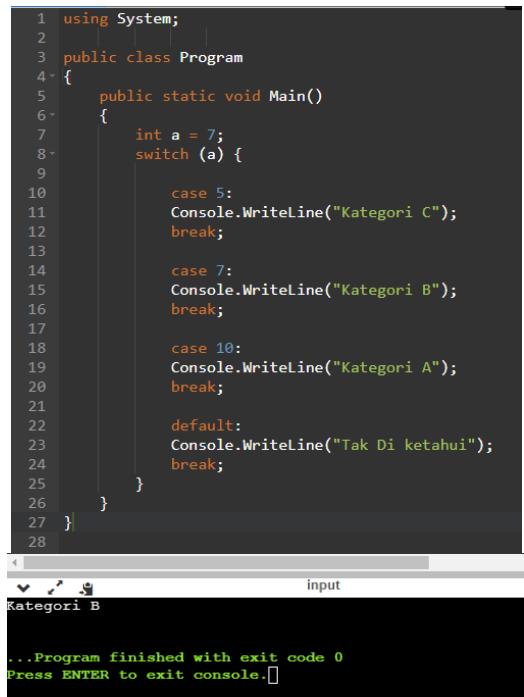
Gambar 3.31 Contoh If-Else If

Pada contoh diatas menggunakan empat kondisi yang berbeda, yang mana jika a kurang dari 5 akan mendapat nilai D, a kurang dari 7 mendapat nilai C, a kurang dari 9 akan mendapat nilai B, dan jika lebih dari A.

d. Switch-Case

Perintah ini sama dengan if-else if namun memiliki perbedaan yaitu lebih sederhana.





The screenshot shows a code editor with a C# program. The code defines a class 'Program' with a 'Main' method. Inside 'Main', an integer 'a' is set to 7, and a 'switch' statement is used to print different messages based on the value of 'a'. The 'case 7' block prints 'Kategori B'. The 'case 10' block prints 'Kategori A'. The 'default' block prints 'Tak Di ketahui'. The code editor has line numbers from 1 to 28. Below the editor is a terminal window showing the output: 'Kategori B' and '...Program finished with exit code 0 Press ENTER to exit console.'.

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int a = 7;
8         switch (a) {
9
10            case 5:
11                Console.WriteLine("Kategori C");
12                break;
13
14            case 7:
15                Console.WriteLine("Kategori B");
16                break;
17
18            case 10:
19                Console.WriteLine("Kategori A");
20                break;
21
22            default:
23                Console.WriteLine("Tak Di ketahui");
24                break;
25        }
26    }
27 }
28
```

```
input
Kategori B

...Program finished with exit code 0
Press ENTER to exit console.[]
```

Gambar3. 32 Contoh Switch-Case

Dari perintah diatas maka terlihat perbedaan setiap kondisi dimasukan kedalam case. Setiap setelah kondisi diberi break sebagai pemisah dengan kondisi yang lain.

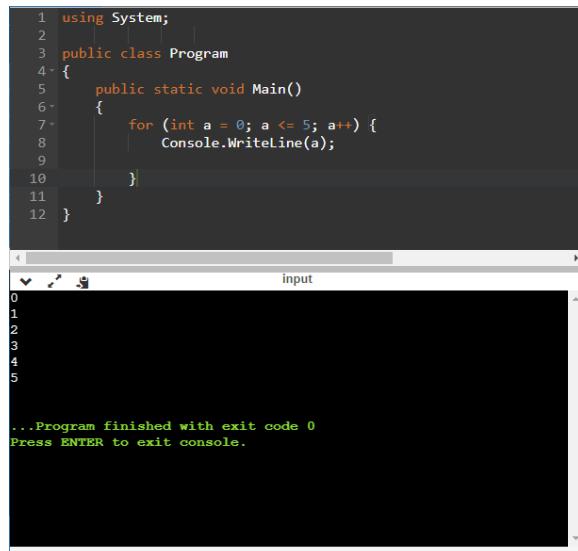
2. Struktur Kontrol Perulangan

Digunakan untuk mengontrol sebuah kondisi yang akan diulang kembali. Sehingga menghemat ruang memory dalam sebuah program dan mensederhanakan sebuah alur. Dalam kontrol perulangan menggunakan fungsi perintah dibawah ini.

a. For

Perintah for berfungsi melakukan perulangan yang didasarkan nilai dari hasil perhitungan bilangan bulat.





```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         for (int a = 0; a <= 5; a++) {
8             Console.WriteLine(a);
9         }
10    }
11 }
12 }
```

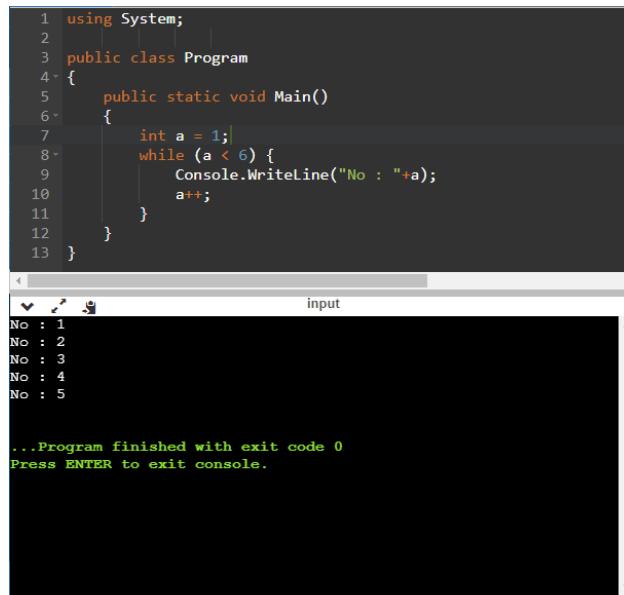
0
1
2
3
4
5

...Program finished with exit code 0
Press ENTER to exit console.

Gambar 3.33 Contoh Perintah For

b. While

Perulangan akan dilakukan pengecekan kondisi true diawal program. Dan perulangan bisa dijalankan jika kondisi true.



```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int a = 1;
8         while (a < 6) {
9             Console.WriteLine("No : "+a);
10            a++;
11        }
12    }
13 }
```

No : 1
No : 2
No : 3
No : 4
No : 5

...Program finished with exit code 0
Press ENTER to exit console.

Gambar 3.34 Contoh Perulangan While

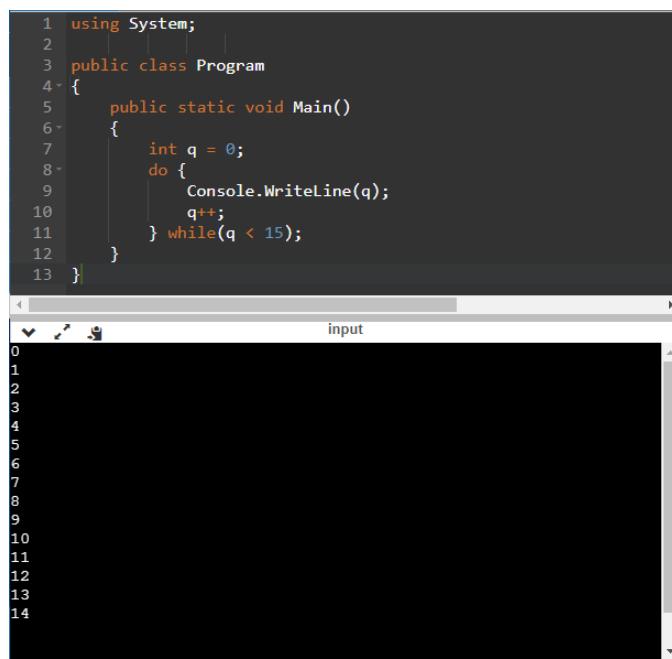
Maksud dari program diatas adalah while a kurang dari 6 maka keluarkan No dengan nilai a dan nilai a ditambah secara



increment. Sehingga nilai a akan ditambahkan secara berulang sampai kurang dari 6. Saat sudah nilai 6 maka program akan berhenti karena bernilai true dan nilai 6 tidak akan ditampilkan. Jika tidak ditambahkan operator increment maka yang akan terjadi program akan terus mengulang nilai 1 tanpa berhenti, karena kondisi masih false.

c. Do-While

Perulangan dengan do berbeda dengan while, yang mana kondisi pengecekan true dilakukan diakhir.



```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int q = 0;
8         do {
9             Console.WriteLine(q);
10            q++;
11        } while(q < 15);
12    }
13 }
```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Gambar 3.35 Contoh Perulangan Do-While

Pada do program akan dijalankan dahulu dan pengecekan diakhir proses. Jika kondisi proses masih belum true maka akan diulang kembali kondisi dari awal sampai dengan kondisi proses pada nilai 15. Jika sudah di nilai 15 program akan berhenti sesuai pernyataan while dan nilai yang ditampilkan adalah 1-14.



Tugas Mandiri

Buat sebuah daftar menu sederhana dengan menampilkan 5 daftar menu masakan nusantara. Dan apabila menu yang dipesan tidak ada maka akan menampilkan daftar menu kembali. Sertakan juga penjelasan algoritma dan alasan pemilihan struktur kontrol.

Diskusi Kelompok

Sebuah toko membutuhkan tampilan daftar barang toko, yang mana toko tersebut memiliki 7 jenis barang yang berbeda. User dapat memilih item beserta jumlah yang diinginkan. Setelah memasukan item dan jumlah akan keluar total harga yang harus dibayar. Setelah keluar total harga muncul pertanyaan belanja lagi atau tidak. Jika belanja lagi akan diarahkan kembali mulai pertanyaan pertama jika tidak maka program selesai.

Buatlah analisis dengan anggota kelompokmu berkaitan dengan alur algoritma dan struktur kontrol yang digunakan. Hasil diskusi terapkan pada program setelah itu persentasikan didepan kelas hasil diskusi kelompok kalian.



Uji Kompetensi

1. Tulislah algoritma perulangan dalam membentuk gambar dibawah ini.

```
1111111
222222
33333
4444
555
66
7

...Program finished with exit code 0
Press ENTER to exit console.
```

2. Jelaskan dengan bahasa kalian kondisi jika pada struktur perulangan while tidak menemukan kondisi true!
3. Bagaimana pendapat kalian jika dua kondisi dengan menggunakan struktur kontrol IF-IF dan IF-Else?
4. Apakah yang terjadi jika menggunakan IF-IF-Else dalam percabangan bersarang dan bandingkan dengan penggunaan IF-Else IF?
5. Jelaskan pendapatmu tentang perbedaan signifikan penggunaan IF-Else IF dan switch case dalam struktur percabangan bersarang!



G. Procedure

Procedure merupakan sub program yang tidak memiliki nilai balik. Deklarasi dari procedure :

```
private static void namaProcedure(parameter)
{
    Isi sub program
}
```

Contoh :

1. Buatlah file C# dan beri nama Procedure.cs karena memiliki nama class Procedure

```
using System;

namespace procedure_csharp
{
    class Procedure
    {
        //sub program garis
        private static void Garis()
        {
            Console.WriteLine("\n -----")
        }
        //sub program tulisan
        private static void Tulisan()
        {
            Console.WriteLine("\n Belajar Pemograman C#");
            Console.WriteLine("\n Pengembangan Perangkat Lunak dan Gim");
        }
        public void judul()
        {
            Garis();
            Tulisan();
            Garis();
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```



2. Untuk memanggil Procedure diatas maka perlu membuat sebuah objek dahulu di main class yang memiliki method main. Main class disini memiliki nama class Program yang disimpan dengan nama Program.cs. Pembuatan objek disini dengan menggunakan keyword new. Dengan format deklarasi :

```
<nama class> <nama objek>=new <konstruktor>();
```

Contoh :

Membuat objek dengan nama dasar, dapat ditulis seperti kode di bawah ini :

```
Procedure dasar = new Procedure();
```

```
using System;

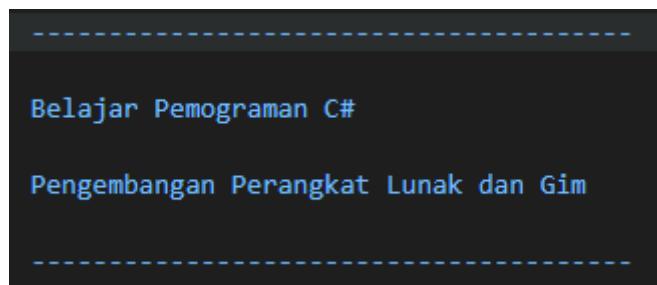
namespace procedure_csharp
{
    class Program
    {
        static void Main(string[] args)
        {
            Procedure dasar = new Procedure();
            dasar.judul();

            Console.ReadLine();
        }
    }
}
```



3. Jalankan file Program.cs dengan menekan tombol Ctrl+F5.

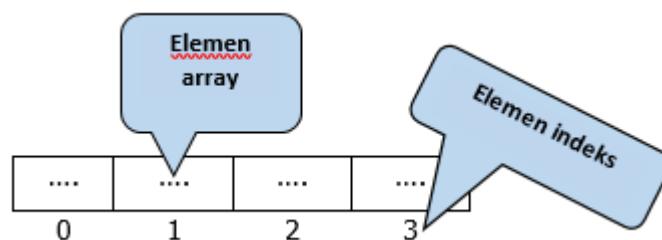
Kemudian akan muncul hasil seperti di bawah ini :



Gambar 3.36 Output procedure

H. Mengenal Array

Array merupakan kumpulan data yang memiliki tipe yang sama dan menggunakan nama yang sama. Dengan menggunakan array, sejumlah variabel dapat menggunakan nama yang sama. Antara satu variabel dengan variabel lain di dalam array dibedakan berdasarkan *subscript*. Sebuah *subscript* berupa bilangan di dalam kurung siku. Melalui *subscript* inilah masing-masing elemen array dapat diakses. Elemen-elemen dalam array bisa dijelaskan melalui gambar berikut ini :



Gambar 3.37 Elemen Array

Elemen array dalam C# diberi nomer 0,1,2,...N-1. Angka-angka tersebut dikenal dengan istilah indeks. Jumlah total elemen dalam array tertentu disebut dengan panjang array.



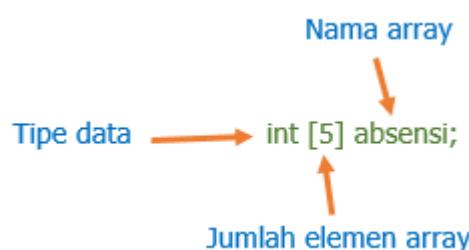
Semua elemen array memiliki tipe data yang sama. Hal ini memungkinkan untuk mewakili sekelompok elemen serupa sebagai urutan yang teratur dan mengerjakannya secara keseluruhan.

Array bisa dalam bentuk dimensi yang berbeda, tetapi yang paling banyak digunakan adalah array satu dimensi dan dua dimensi. Array satu dimensi disebut juga vektor dan array dua dimensi dikenal sebagai matriks. Aplikasi array satu dimensi bisa digunakan untuk membantu dalam pengurutan dan pencarian data (*sorting* dan *searching*).

Array dideklarasikan dengan menggunakan syntax :

```
Tipe data[] nama_array;
```

Contoh deklarasi array :



Dalam contoh ini variabel `absensi` adalah nama array yang dapat menyimpan lima buah data bertipe `int`. Ini berarti bahwa mendeklarasikan sebuah array bilangan bulat angka. Dengan `[]`, ditunjukkan bahwa yang dideklarasikan adalah sebuah array elemen, bukan elemen tunggal.



1. Array Satu Dimensi

Di C# untuk mengalokasikan memori maka digunakan kata kunci new. Seperti contoh di bawah ini kalian bisa membuat array untuk objek 10 integer :

```
int[] nomer = new int[10];
```

Di dalam brackets [] ini menyatakan angka tersebut adalah array. Jika dicontohkan posisi penempatan angka tersebut adalah sebagai berikut :

```
nomer[0]=4;  
nomer[1]=8;  
nomer[2]=16;  
nomer[3]=7;
```

Gambaran array satu dimensi ditunjukkan pada program di bawah ini. Program ini meminta pemakai untuk memasukkan 4 buah data nomer dari keyboard. Keempat data tersebut disimpan pada array bernama num. Selanjutnya data yang ada pada array tersebut ditampilkan ke layar.



Contoh :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Declare_Array
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] num = new int[4]; //Declaring Array

            //Initializing array
            num[0] = 4;
            num[1] = 8;
            num[2] = 16;
            num[3] = 7;

            //Showing value of Array
            Console.WriteLine("Nilai yang pertama : "+num[0]);
            Console.WriteLine("Nilai yang kedua : "+ num[1]);
            Console.WriteLine("Nilai yang ketiga : "+ num[2]);
            Console.WriteLine("Nilai yang keempat : "+ num[3]);

            Console.ReadLine();
        }
    }
}
```

Hasil Output program :

```
Nilai yang pertama : 4
Nilai yang kedua : 8
Nilai yang ketiga : 16
Nilai yang keempat : 7
```



2. Array Dua Dimensi

Sebagai gambaran, data kelulusan dari Bidang Keahlian Teknologi Informasi pada suatu Sekolah Menengah Kejuruan dari tahun 2019 hingga 2021 dapat dinyatakan dengan array berdimensi dua. Data kelulusan dapat digambarkan seperti tabel di bawah ini :

Tabel 3.12 Tabel Kelulusan Siswa

Program Keahlian	2019	2020	2021
1. Pengembangan Perangkat Lunak dan Gim	135	126	132
2. Teknik Komputer Jaringan dan Telekomunikasi	112	130	128

Bentuk seperti tabel diatas dituangkan ke dalam array berdimensi dua. Pendefinisianya :

```
int data_lulus[2][3];
```

Pada pendefinisian diatas :

- 2 menyatakan jumlah baris (mewakili jurusan)
- 3 menyatakan jumlah kolom (mewakili tahun kelulusan)

Masing-masing array berdimensi dua dapat diakses dengan bentuk :

```
Tipe_data nama_array[][]
```

Berikut ini merupakan contoh penomoran elemen pada array berdimensi dua :

	0	1	2
0	[0][0]	[0][1]	[0][2]
1	[1][0]	[1][1]	[1][2]



Contoh pengaksesan elemen array berdimensi dua :

a. data_lulus[0][2] = 132;

Merupakan instruksi untuk memberikan nilai 132 ke baris 0 kolom 2 (baris dan kolom dimulai dari 0)

b. Console.WriteLine(data_lulus[0][2]);

Merupakan perintah untuk menampilkan elemen data_lulus dengan subscript pertama(baris) berupa 0 dan subscript kedua (kolom) bernilai 2.

Contoh kode program array :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace program_csharp
{
    class Program
    {
        static void Main(string[] args)
        {
            //Ini adalah Contoh Array 1 Dimensi
            int[] array = new int[] { 1, 2, 3 }; //Ini adalah Contoh Array
            Console.WriteLine("\t\t Belajar Bikin Array"); //Ini akan dicetak
            Console.WriteLine("\n\n===== Array 1 Dimensi Menggunakan C# =====");
            for (int i = 0; i < array.Length; i++) //membuat perulangan
            {
                Console.WriteLine("Array Satu Dimensi" + (i + 1) + " : " + array[i]);
            }
            //Ini adalah Array 2Dimensi
```



```

Console.WriteLine("\n\n\n=====Array 2 Dimensi Menggunakan C
#=====");
int baris_array = 2;
int klm_array = 2;
int[,] matrik_array = new int[baris_array, klm_array];
int x = 1;
for (int s = 0; s < baris_array; s++)
{
    for (int j = 0; j < klm_array; j++)
    {
        matrik_array[s, j] = x;
        x++;
    }
}
for (int s = 0; s < baris_array; s++)
{
    for (int j = 0; j < klm_array; j++)
    {
        Console.Write("Array 2 Dimensi
"+matrik_array[s, j] + "\t");//array yang akan dicetak
    }
    Console.WriteLine();
}
Console.ReadLine();
}
}
}
}

```

Output :



```

Belajar Bikin Array
2
===== Array 1 Dimensi Menggunakan C#=====
Array Satu Dimensi1 : 1
Array Satu Dimensi2 : 2
Array Satu Dimensi3 : 3
3
=====Array 2 Dimensi Menggunakan C#=====
Array 2 Dimensi 1      Array 2 Dimensi 2
Array 2 Dimensi 3      Array 2 Dimensi 4

```

Gambar 3.38. Ouput Array satu dimensi dan dua dimensi



3. Array List

ArrayList digunakan untuk menyimpan data yang bersifat dinamis. Method add() digunakan untuk mengisi data pada ArrayList. Dan method sort() digunakan untuk mengurutkan data pada ArrayList.

Contoh kode program :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace Coba_List
{
    class Program{
        static ArrayList siswa = new ArrayList();
        static string nama;
        static void Main(string[] args)
        {
            Console.WriteLine ("Contoh ArrayList untuk STRING\n");
            Console.Write("Masukkan jumlah siswa : ");

            int jmlSiswa=int.Parse(Console.ReadLine());
            int i=1;
            while (i<=jmlSiswa)
            {
                Console.Write("Nama siswa ke-"+(i++)+":");
                nama=Console.ReadLine();
                siswa.Add(nama);
            }
            siswa.Sort();
            i=1;
            Console.WriteLine("\n\n Nama siswa yang diurutkan berdasarkan abjad \n");
            foreach(Object namaSiswa in siswa)
            Console.WriteLine("Siswa ke-"+(i++)+" "+namaSiswa);
            Console.ReadLine();
        }
    }
}
```



Output :

```
Contoh ArrayList untuk STRING

Masukkan jumlah siswa : 5
Nama siswa ke-1:kenzie
Nama siswa ke-2:adara
Nama siswa ke-3:nastiti
Nama siswa ke-4:mika
Nama siswa ke-5:qicha

Nama siswa yang diurutkan berdasarkan abjad

Siswa ke-1adara
Siswa ke-2kenzie
Siswa ke-3mika
Siswa ke-4nastiti
Siswa ke-5qicha
```



Tugas Mandiri

Buatlah sebuah program sederhana untuk membuat array dari 10 elemen dengan tipe data integer dan menginisialisasi setiap elemen dengan nilai yang sama dengan indeks elemen dikalikan 5. Lalu cetak elemen ke console.

Diskusi Kelompok

Buatlah kelompok kecil, untuk membuat program yang menerima inputan berupa n ($n < 10$). Kemudian sebanyak n yang diinputkan ditampung dalam array. Setelah itu jumlahkan semua bilangan yang diinputkan pada urutan genap dan tampilkan hasil penjumlahan tersebut. Analisalah bersama kelompokmu hasil penjumlahan tersebut.

Uji Kompetensi

1. Apa yang dimaksud dengan array?
2. Bagaimana cara mendeklarasikan array?
3. Bagaimanakah cara mengakses elemen sebuah array di C#?
4. Berikan contoh program deklarasi array satu dimensi!
5. Jelaskan fungsi array !





I. **Aktivitas Pembelajaran**

1. Tugas Mandiri

Untuk meningkatkan pemahamanmu mengenai materi pada Bab III, jawablah soal berikut ini :

- a. Apakah yang dimaksud dengan algoritma pemrograman?
- b. Sebutkan berbagai macam tipe data yang digunakan dalam bahasa pemrograman C#!
- c. Jelaskan penggunaan operator aritmatika dan logika!
- d. Berikan contoh program dengan menggunakan struktur control!
- e. Berikan contoh program untuk mengurutkan bilangan dengan menggunakan array!

2. Tugas Diskusi

- a. Buatlah kelompok dengan teman sebangkumu!
- b. Perhatikan penggolongan nilai di bawah ini :

Nilai 90 s.d 100 : A

Nilai 80 s.d <90 : B

Nilai 70 s.d <80 : C

Nilai <70 : D

- c. Lakukan analisis serta buatlah pseudocode dan flowchart dari deskripsi penggolongan nilai diatas!



3. Tugas Proyek

- a. Buatlah kelompok masing-masing terdiri dari 3-4 anggota.
- b. Buatlah sebuah program untuk menampilkan nilai 10 orang siswa disertai dengan rata-rata nilai.

J. Refleksi

Pada Bab III, peserta didik telah mempelajari tentang membuat algoritma pemrograman, pengenalan C#, struktur kode C#, dasar teori C#, dasar pemrograman C#, maka peserta didik diminta untuk menjawab beberapa pertanyaan di bawah ini :

1. Apakah kalian sudah memahami materi pada bab ini?
2. Tuliskan hal penting yang kalian pelajari pada materi :
 - a. Algoritma Pemrograman :
 - b. Struktur data :
 - c. Tipe data :
 - d. Variabel dan konstanta :
 - e. Operator :
 - f. Struktur control :
 - g. Prosedur :
 - h. Array :



K.  **Pengayaan**

Struktur data merupakan dasar dalam belajar menulis bahasa pemrograman. Namun materi yang disampaikan diatas hanya mengacu pada struktur bahasa pemrograman C#. Sedangkan dalam perkembangan teknologi yang sangat cepat banyak sekali muncul bahasa pemrograman baru dengan struktur yang berbeda. Oleh sebab itu untuk dapat mengenal lebih lanjut dalam struktur data dalam pemrograman komputer. Dan untuk menambah referensi dalam mengembangkan struktur data, silahkan klik tautan dibawah ini

<https://www.sololearn.com/>

<https://www.csharp-console-examples.com/>



DAFTAR RUJUKAN

Afrilia, Ascharisa Mettasatya. 2018. *Personal Branding Remaja di Era Digital*. Jurnal Komunikasi, Vol 11, 24

Albahari Joseph, Albahari Ben. 2017. *C# 7.0 in a Nutshell*. California : O'Reilly Media, Inc.

Ali, Muhamad.2014. *Kitab Belajar Pemrograman C#*. Jakarta:Graha Ilmu

ANSI. 1970. *American National Standard Flowchart Symbols and Their Usage in Information Processing*, X3.5-1970. New York: American National Standards Institute.

Asmiyatun, Siti. Dkk. 2017. *Belajar membuat game 2D dan 3D menggunakan Unity*. Yogyakarta: Deepubilsh Publisher

Davenport, T. H. 1993. *Process Innovation: Reengineering work through information technology*. Boston: Harvard Business School Press.

Enterprise, Jubilee. 2018. *Belajar Sendiri Photoshop CC 2018*. Yogyakarta. PT Elex Media Komputindo.

Enterprise, Jubilee. 2018. *Otodidak Adobe Illustrator*. Yogyakarta. PT Elex Media Komputindo.

Fajri, Choirul. 2012. *Tantangan Industri Kreatif-Game Online di Indonesia*. Jurnal Komunikasi, Volume 1, 453.

Febriyanti, Fitri. Dkk. 2019. *Produk Kreatif dan Kewirausahaan*. Jakarta. Direktorat Pembinaan SMK Kementerian Pendidikan dan Kebudayaan Republik Indonesia.

Hadiprakoso, Raden Budiarto. 2020. *Rekayasa Perangkat Lunak*. RBH.

Irianto, Heru. Dkk. 2016. *Panduan Pendirian Usaha Pengembang Aplikasi Digital*. Surakarta. Bekraf.

Dasar-Dasar Pengembangan Perangkat Lunak Dan Gim X SMK/MAK | 298

Irianto, Heru. Dkk. 2016. *Panduan Pendirian Usaha Game Digital*. Surakarta. Bekraf.

Jamro, Marcin. 2018. *C# Data Structures and Algorithms*. Birmingham: Packt Publishing Ltd.
Tersedia pada laman <https://www.packtpub.com/>

Kadir, Abdul. 1995. *Pemrograman C++*. Yogyakarta: Andi Offset

Kamus Besar Bahasa Indonesia. <https://kbbi.kemdikbud.go.id/>. Diakses tanggal 28 April 2021.

Klausen, Poul. 2012. *C#1 Introduction To Programming and The C# Language*. Online. Tersedia pada laman <https://bookboon.com/>

Kurniawan Agus, Risman Adnan, dkk. 2004. *Pengenalan Bahasa C#*. Jakarta: Team Project Otak

Mulyana, Iyan. DKK. 2019. *Buku Ajar Desain Grafis dan Multimedia Teori dan Implementasi*. Lembaga Penelitian dan Pengabdian Masyarakat Universitas Pakuan.

Munif, Abdul. 2020. *Pemodelan Perangkat Lunak (C3) Kelas XI*. Malang: PT. Kuantum buku sejahtera.

Pradana, Mahir. 2015. *Klasifikasi Jenis-Jenis Bisnis E-Commerce Di Indonesia*. Jurnal Neo-bis, Volume 9, 36.

Pressman, Roger S. 2010. *Software engineering: A Practitioner's Approach (7th. Edition)*. New York: McGraw-Hill Higher Education.

Savitch, Walter. 2015. *Problem Solving With C++ (9th edition)*. Boston : Pearson Education, Inc.

Widya, Leonardo Adi Dharma. 2016. *Pengantar Desain Grafis*. Jakarta: Direktorat Pembinaan Kursus dan Pelatihan.

DAFTAR SUMBER GAMBAR

1. Perangkat Lunak Microsoft Word.
<https://tekno.kompas.com/read/2021/02/19/18220097/microsoft-umumkan-office-2021>
2. Game Super Mario Bros.
<https://www.liputan6.com/tekno/read/4516053/fantastis-gim-super-mario-bros-lawas-dilelang-rp-45-miliar>
3. Pascalene The First Mechanical adding Machine.
<https://slideplayer.info/slide/12462754/>
4. ENIAC (Electronic Numerical Integrator And Computer).
<https://slideplayer.info/slide/12462754/>
5. IBM 701 Defense with Calculator.
https://www.ibm.com/ibm/history/exhibits/701/701_141508.html
6. Apple Lisa. https://en.wikipedia.org/wiki/Apple_Lisa
7. Perkembangan Telepon Selular.
<https://www.soloposfm.com/perkembangan-telepon-seluler-sejak-1990-an-sampai-sekarang/9616/>
8. Permainan duck shoot game oleh Ray-O-Lite Rifle, Seeburg.
<http://www.pinrepair.com/arcade/rayolit.htm>
9. Permainan Simulator Rudal Cathode Ray Tube Device Amusement.
<http://prayoga-5150.blogspot.com/2018/03/history-of-video-game.html>
10. Permainan OXO Game. <https://en.wikipedia.org/wiki/OXO>
11. Mesin Pinball SEGA. https://www.arcade-museum.com/game_detail.php?game_id=13254
12. Permainan Tennis For Two.
<https://videogamehistorydevelopment.weebly.com/tennis-for-two.html>
13. Permainan Space War. <https://en.wikipedia.org/wiki/Spacewar!>
14. Permainan The Brown Box.
https://americanhistory.si.edu/collections/search/object/nmah_1301997
15. Spacewars! Online. <https://www.masswerk.at/spacewar/>
16. PDP Space Wars!
https://www.gamasutra.com/view/feature/129861/down_the_hyperspatial_tunnel_.php?page=4
17. *Magnavox Odyssey*. <https://en.wikipedia.org/wiki/File:Magnavox-Odyssey-Console-Set.jpg>

18. Permainan Atari Pong.
<http://catatanrezapahlevi.blogspot.com/2013/07/perkembangan-teknologi-game-dari-masa.html>
19. Grand Track 10. https://www.arcade-museum.com/game_detail.php?game_id=7992
20. Permainan Pong Eksklusif. <https://www.wikiwand.com/en/Pong>
21. Console Game Fairchild dan Apple I.
https://en.wikipedia.org/wiki/Apple_I
https://en.wikipedia.org/wiki/Fairchild_Channel_F_Videocarts
22. Atari 2600 dan Nintendo Color TV Game.
[https://en.wikipedia.org/wiki/Fairchild_Channel_F_Videocarts.](https://en.wikipedia.org/wiki/Fairchild_Channel_F_Videocarts)
https://id.wikipedia.org/wiki/Color_TV_Game
23. PDP-10 Minicomputer. <https://en.wikipedia.org/wiki/PDP-10>
24. Tandy TRS80 dan Commodore PET.
<https://en.wikipedia.org/wiki/TRS-80>.
https://id.wikipedia.org/wiki/Commodore_PET
25. Apple II. https://en.wikipedia.org/wiki/Apple_II
26. Permainan Game Space Invaders dan Odyssey 2.
https://id.wikipedia.org/wiki/Space_Invaders.
https://en.wikipedia.org/wiki/Magnavox_Odyssey_2
27. Handheld Console Cosmos oleh Atari.
<http://videogamekraken.com/cosmos-by-atari>
28. Gambar Taman Bunga.
<https://www.merdeka.com/gaya/ditutup-begini-pemandangan-taman-bunga-terindah-di-dunia-keukenhof-tanpa-pengunjung.html>
29. Penerapan multimedia grafik, teks dan animasi dalam aplikasi starbucks. <https://dribbble.com/shots/7297745-Starbucks-UI-UX-Dashboard>
30. Symbol flowchart. <https://informatikalogi.com/>
31. Download .NET SDK x64. <https://code.visualstudio.com/>
32. Lokasi Pembangunan. <http://pasarbatang.desa.id/>

INDEKS

Simbol		K	
+	264,271	Konstanta	262,263
-	265,271		
=	243,271	L	
*	270,271	lane	66
%	270,271		
//	255	M	
/*	255	multi tasking	10
*/	255	multi user	10
—	262	<i>maintainable</i>	43
“	263	<i>message flow</i>	65
<	264	<i>monetize</i>	117
?	264		
++	265	N	
--	265	Not	273
>	265		
<	265	O	
>>	265	<i>open source</i>	60
<<	265		
+=	266	P	
		<i>Patch</i>	43,61
A		<i>Player</i>	45
Algoritma	237	<i>Pool</i>	65
Assembler	8	<i>Print out</i>	8
<i>Activity</i>	64	<i>Printer</i>	11
Association	65	<i>procedure</i>	283
Array	233	<i>Pseudocode</i>	237
		Punched card	8
B			
Bit	8	Q	
boolean	8	Quotes	153
brand	98		
byte	8	R	
		<i>real time</i>	10
C		<i>Reliability</i>	83
<i>Character</i>	45	<i>Reusable</i>	62
<i>cloud computing</i>	93,95		

<i>Compiler</i>	46	S	
<i>Correctness</i>	83	Sains	5
<i>cyber-security</i>	97	<i>Sequence flow</i>	64
		Software engineering	6
D		<i>Stakeholders</i>	44
<i>Debugger</i>	46	<i>Startup</i>	105,107
<i>Desktop</i>	91	<i>Storyboard</i>	110
<i>Do-while</i>	277	<i>Swimlane</i>	65
		<i>Switch case</i>	277
E			
<i>Efficiency</i>	83	T	
<i>Engineer</i>	106	<i>Testability</i>	84
		Tipe data	205
F			
<i>Feature</i>	110	U	
<i>Flowchart</i>	238	<i>Upgrade</i>	43
<i>foot candles</i>	85	<i>Usability</i>	84
for	278		
		V	
G		Vendor	77
<i>Gateway</i>	64	<i>visual display units</i>	85
<i>Genre</i>	110		
		W	
H		While	227
<i>Hybrid</i>	49		
		Z	
I		Zoom	163
<i>Integrity</i>	84		
<i>Interpreter</i>	46		
<i>Intangible</i>	65		
<i>internet of thing</i>	93,95		
<i>Invisible</i>	61		

GLOSARIUM

A

Algoritma	: prosedur sistematis untuk memecahkan masalah matematis dalam langkah-langkah terbatas
Animasi	: serangkaian gambar bergerak berurutan untuk menyajikan proses tertentu yang biasa dilengkapi teks penjelasan dan audio
<i>Asset</i>	: semua kebutuhan atau bahan - bahan yang digunakan dalam perangkat lunak atau gim

B

Bahasa Pemrograman	: bahasa yang digunakan untuk menulis program yang dapat dieksekusi oleh komputer
Bisnis	: bidang usaha
Bit	: satuan informasi terkecil dalam sistem informasi
<i>Branding</i>	: pelabelan

C

<i>Code</i>	: sandi, kode
<i>Controller</i>	: pengontrol
<i>Cloud Computing</i>	: teknologi yang memberikan pelayanan secara luas dengan akses internet dimanapun berada

D

Desainer	: perancang
----------	-------------

E

Elektronik	: alat yang dibuat berdasarkan prinsip elektronika serta hal atau benda yang menggunakan alat elektronika
------------	---

F

<i>For</i>	: loop penghitungan (iterasi)
------------	-------------------------------

G	
Garis	: komponen desain grafis yang merupakan perapatan dari sekelompok titik yang berurut dan beruntut, baik berdiri sebagai elemen utama maupun bagian dari (aksen) komunikasi
Gim/game	: permainan
I	
Imajinatif	: menggunakan imajinasi
Informasi	: data yang sudah diolah menjadi bentuk yang bernilai atau bermakna
Instalasi	: proses menyiapkan <i>computer</i> baru
Internet	: kumpulan dari jutaan <i>computer</i> yang saling terhubung
Internet of Things	: segala aktifitas yang pelakunya saling berinteraksi dengan memanfaatkan internet
K	
Kode	: tanda (kata-kata, tulisan) yang disepakati untuk maksud tertentu
Konsumen	: pemakai jasa (pelanggan dan sebagainya)
Komputer	: sistem elektronik yang digunakan untuk mengolah dan memanipulasi data dengan cepat dan tepat
Konstanta	: entitas di dalam program yang nilainya tidak berubah selama eksekusi program
L	
Lisensi	: surat izin untuk mengangkut barang dagangan, usaha, dan sebagainya
M	
<i>Maintenance</i>	: pemeliharaan
<i>Multitasking</i>	: keterampilan dalam menjalankan beberapa tugas dalam waktu yang bersamaan
<i>Multi user</i>	: beberapa pengguna
O	
<i>Omni directional</i>	: mikrofon yang menerima suara dari semua arah

P

Perangkat Lunak

: perangkat program, prosedur, dan dokumen yang berkaitan dengan suatu system (misalnya system computer)

Pemasaran

: proses penting agar bisnis bisa berhasil

Pemrograman

: proses memprogram

Print out

: mencetak

Profesi

: bidang pekerjaan yang dilandasi pendidikan keahlian (keterampilan, kejuruan, dan sebagainya) tertentu

Proses Bisnis

: aktivitas yang terukur dan terstruktur untuk memproduksi output tertentu untuk kalangan pelanggan tertentu. Terdapat di dalamnya penekanan yang kuat pada "bagaimana" pekerjaan itu dijalankan di suatu organisasi, tidak seperti fokus dari produk yang berfokus pada aspek "apa". Suatu proses oleh karenanya merupakan urutan spesifik dari aktivitas kerja lintas waktu dan ruang, dengan suatu awalan dan akhiran, dan secara jelas mendefinisikan input dan output

Prosedur

: metode langkah demi langkah secara pasti dalam memecahkan suatu masalah

Punched card

: kartu berlubang-lubang

R

Rekayasa

: penerapan kaidah-kaidah ilmu dalam pelaksanaan

Realtime

: simulasi yang mendekati keadaan aslinya

S

Sains

: ilmu pengetahuan pada umumnya

Software

: data yang diprogram, disimpan, dan diformat secara digital dengan fungsi tertentu

T

Teknologi

: sarana berbentuk aneka macam peralatan atau sistem yang berfungsi untuk memberikan kenyamanan serta kemudahan bagi manusia

Tekstur	: komponen desain grafis yang merupakan tampilan/karakteristik/gambaran/representasi sifat dari suatu permukaan
<i>Tool</i>	: alat
V	
Visi	: pandangan atau wawasan ke depan
Video	: hasil rekaman proses kejadian yang berisikan gambar berurutan disertai suara
W	
<i>Webcam</i>	: perangkat yang berupa sebuah kamera digital yang dihubungkan ke komputer atau laptop
Y	
<i>Youtube</i>	: situs web yang dirancang untuk berbagi video
Z	
<i>Zoom</i>	: gerakan lensa mendekati atau menjauhi obyek secara optic

PROFIL PENULIS

Nama Lengkap : Yustiana Amita Utama,S.ST
Email : yustiana.amita@gmail.com
Handphone : 085735397639
Instansi : SMK Negeri 8 Malang
Bidang Keahlian: Teknologi Informasi /
Rekayasa Perangkat Lunak



- **Riwayat Pekerjaan/Profesi dalam 10 Tahun Terakhir:**
 1. Guru Produktif TKJ SMK Negeri 8 Malang (2007-2015)
 2. Guru Produktif RPL SMK Negeri 8 Malang (2015-sekarang)
 3. Ketua Kompetensi Keahlian RPL SMK Negeri 8 Malang (2018-sekarang)
- **Riwayat Pendidikan Tinggi dan Tahun Belajar:**
 1. D4 Teknologi Informasi, Politeknik Elektronika Negeri Surabaya (lulus tahun 2006)
- **Judul Buku dan Tahun Terbit (10 Tahun Terakhir):**
 -
- **Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):**
 -
- **Informasi lain dari Penulis**

Penulis saat ini tinggal di Kota Malang. Penulis telah menyelesaikan Pendidikan Dasar di MIN Malang I, kemudian melanjutkan pendidikan menengah di SMP Negeri 3 Malang dan SMA Negeri 4 Malang. Dilanjutkan dengan jenjang pendidikan tinggi di Program Studi Teknologi Informasi Politeknik Elektronika Negeri Surabaya. Selama pendidikan menengah penulis aktif dalam beberapa organisasi siswa. Saat ini penulis bekerja di SMK Negeri 8 Malang.

PROFIL PENULIS

Nama Lengkap : Esa Hergatama, S.Kom., Gr
Email : esa.hergatama@gmail.com
Handphone : 082121218513
Instansi : SMK Bina Informatika
Bidang Keahlian: Rekayasa Perangkat Lunak



- **Riwayat Pekerjaan/Profesi dalam 10 Tahun Terakhir:**
 1. Guru Produktif dan Kepala Program Keahlian Rekayasa Perangkat Lunak, SMK Bina Informatika (2015 – Sekarang).
 2. Chief Executive Officer, PT. Invensi Kreatif Lab. (2020 – Sekarang).
 3. Programmer, PT. Bentala Piranti Global (2011 – 2015).
- **Riwayat Pendidikan Tinggi dan Tahun Belajar:**
 1. Program Profesi Guru (PPG) Universitas Negeri Yogyakarta (lulus tahun 2018).
 2. S1 Mobile Application Technology Universitas Bina Nusantara (lulus tahun 2015).
- **Judul Buku dan Tahun Terbit (10 Tahun Terakhir):**
 -
- **Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):**
 1. Aplikasi Game Arcade Banana Catch Pada Platform Ios Dengan Menggunakan Cocos2d (2015)
- **Informasi lain dari Penulis**

Penulis saat ini tinggal di Kota Tangerang. Selama Pendidikan penulis aktif dalam kegiatan organisasi sekolah (OSIS) sehingga memotivasi penulis untuk membangun startup sendiri. Di sekolah penulis juga aktif dalam berbagai kegiatan pendampingan Lomba Keterampilan Siswa (LKS) hingga mendapatkan kesempatan untuk mendampingi siswa pada kegiatan Sakura Science Program di Jepang pada tahun 2018.

PROFIL PENELAAH

Nama Lengkap : Dr. Eko Subiyantoro, S.Pd., S.ST., MT.
Email : ekovedc@gmail.com,
Handphone : 08123308016
Instansi : BPPMPV – BOE Malang
Bidang Keahlian : Teknologi Informasi



- **Riwayat Pekerjaan/Profesi dalam 10 Tahun Terakhir:**
 1. Pembantu Ketua I Joint Program VEDC (2009-2013)
 2. Tim Pengembang JENI PPPPTK BOE Malang (2007-2010)
 3. Kepala Departemen Teknologi Informasi PPPPTK BOE Malang (2012-2015)
 4. Widyaiswara BBPPMPV – BOE Malang (2010-sekarang)
- **Riwayat Pendidikan Tinggi dan Tahun Belajar:**
 1. S1 Pendidikan Teknik Elektronika, IKIP Yogyakarta (lulus tahun 1998)
 2. D4 Teknologi Informasi, Politeknik Elektronika Negeri Surabaya (PENS), Surabaya (lulus tahun 2002)
 3. S2 Game Technology, Institut Teknologi Sepuluh Nopember (ITS), Surabaya (lulus tahun 2009)
 4. S3 Ilmu Komputer, Universitas Gadjah Mada (UGM), Yogyakarta (lulus tahun 2020)
- **Judul Buku dan Tahun Terbit (10 Tahun Terakhir):**
 1. 2013. Buku Pemrograman Berorientasi Objek Kelas XI SMK/MAK. ISBN 978-602-04-8796-0. PT. Elex Media Komputindo : Jakarta.
 2. 2018. Buku Pemrograman Dasar. ISBN : 978-602-04-6194-6. PT. Elex Media Komputindo : Jakarta.
 3. 2019. Buku Pemrograman Dasar (C2) Kelas X SMK/MAK. ISBN : 978-623-7398-30-1. PT. Kuantum Buku Sejahtera : Malang.
- **Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):**
 1. 2011. Penentuan Kualitas Daun Tembakau Dengan Perangkat Mobile Berdasarkan Fitur Rata-Rata RGB Menggunakan Algoritma K-Nearest Neighbor , SNATIKA Proceeding APTIKOM, STIKI Malang
 2. 2011. Pengembangan Aplikasi m-Learning Sebagai Prototipe Pembuatan Bahan Ajar Berbasis Mobile, Seminar PPPPTK BOE/VEDC Malang
 3. 2012. Aplikasi VEDC Mobile Map Explorer Sebagai Penunjang Informasi Peserta Pelatihan Dengan Menggunakan Teknologi J2ME, Seminar PPPPTK BOE/VEDC Malang
 4. 2012. Design Mobile Game Base Learning With Single Loop Learning Methode, International Conferen on Vocational Education an Training(ICVET),Proceeding Yogyakarta University
 5. 2017. Autonomous Cognitive Leveling Game Pada Serious Game Menggunakan Particle Swarm Optimization di SENTIKA <https://fti.uajy.ac.id/sentika>)

6. Learning Path Model based on Revised Bloom's Taxonomy and Domain Ontologies using Discrete Particle Swarm Optimization di International Journal of Engineering Research & Technology (IJERT) <http://www.ijert.org> ISSN: 2278-0181 www.ijert.org Vol. 8 Issue 08, August 2019
7. Learning Path Model Based On Cognitive Classification Using Hybrid Discrete Particle Swarm Optimization di Journal of Theoretical and Applied Information Technology 31st December 2019. Vol.97. No 24 ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195.
8. Cognitive Classification Based on Revised Bloom's Taxonomy Using Learning Vector Quantization di CENIM 2020 is an IEEE 2020 International Conference on Computer Engineering, Network and Intelligent Multimedia This conference will be organized by the Department of Computer Engineering - Faculty of Intelligent Electrical and Informatics Technology - Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.
9. Learning Path Recommendation using Hybrid Particle Swarm Optimization, di Advances in Science, Technology and Engineering Systems Journal (ASTESJ) 28 January 2021 Volume 6, Issue 1, Page No 570-576, 2021

■ **Informasi lain dari penelaah**

Eko Subiyantoro menjalani pendidikan dasar di SD Negeri Sraten III, kemudian melanjutkan pendidikan menengah di SMP Negeri I Benculuk dan STM Negeri I Banyuwangi. Dilanjutkan dengan jenjang pendidikan tinggi di IKIP Yogyakarta dan PENS. Pada tahun 2000-2009 pernah menjadi instruktur di PPPGT Malang dan lulus dari program pasca sarjana pada tahun 2009. Kemudian menjadi widyausaha pada tahun 2010 sampai dengan sekarang. Lalu melanjutkan program doctor di UGM dan lulus pada tahun 2020.

PROFIL EDITOR

Nama Lengkap : Tri FirmanSyah, M.Pd
Email : tri.firsyh06@gmail.com
Handphone : 085755359696
Instansi : SMK Negeri 8 Malang
Bidang Keahlian: Teknologi Informasi /
Rekayasa Perangkat Lunak



- **Riwayat Pekerjaan/Profesi dalam 10 Tahun Terakhir:**
 1. Guru KKPI di SMKS Qodiriyah Sulaimaniyah Singosari (2013-2016)
 2. Guru Produktif RPL di SMK Negeri 8 Malang (2016-Sekarang)
- **Riwayat Pendidikan Tinggi dan Tahun Belajar:**
 1. S1. Pendidikan Teknik Informatika Universitas Negeri Malang (lulus tahun 2013)
 2. S2. Pendidikan Kejuruan Universitas Negeri Malang (lulus tahun 2018)
- **Judul Buku dan Tahun Terbit (10 Tahun Terakhir):**
 -
- **Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):**
 -
- **Informasi lain dari editor**

Editor menyelesaikan pendidikan dasar dan menengah di SD Negeri Pagentan 1 Singosari, SMP Negeri 3 Singosari, SMA Islam Al Ma'arif Singosari. Editor melanjutkan pendidikan tinggi di Fakultas Teknik Universitas Negeri Malang, jurusan Teknik Elektro lulus tahun 2013. Editor melanjutkan untuk mengambil gelar Magister pada Fakultas Pascasarjana Universitas Negeri Malang lulus pada tahun 2018. Saat ini editor bekerja sebagai pengajar di Kompetensi Keahlian Rekayasa Perangkat Lunak SMK Negeri 8 Malang

PROFIL DESAINER

Nama Lengkap : Hendro Soemarno, S.T.
Email : hendro@smktelkom-mlg.sch.id
Handphone : 085101442463
Instansi : SMK Telkom Sandhy Putra Malang
Bidang Keahlian: Teknologi Informasi /
Rekayasa Perangkat Lunak



- **Riwayat Pekerjaan/Profesi dalam 10 Tahun Terakhir:**
1. Guru Produktif Rekayasa Perangkat Lunak (2012 - sekarang)
- **Riwayat Pendidikan Tinggi dan Tahun Belajar:**
1. S1 Teknik Elektro Fakultas Teknik Universitas Brawijaya (lulus tahun 2007)
- **Judul Buku dan Tahun Terbit (10 Tahun Terakhir):**
-
- **Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):**
-
- **Informasi lain dari desainer**
Hendro Soemarno menyelesaikan Pendidikan Dasar di SDN 3 Brebes, Jawa Tengah, kemudian pendidikan menengah di SMPN 23 Surabaya, Jawa Timur dan SMAN 1 Surabaya, Jawa Timur. Dilanjutkan dengan jenjang pendidikan tinggi di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya. Selama pendidikan tinggi penulis menjadi asisten laboratorium elektronika.