

## ALGORITMA DAN FLOWCHART

Algoritma memiliki peranan yang sangat penting dalam sebuah program, terutama untuk memahami alur dari program yang dibuat, jika sebuah program tidak dibuat dengan algoritma yang benar maka output yang dihasilkan juga akan tidak benar. Algoritma itu sendiri dapat dikatakan urutan langkah- langkah logis penyelesaian masalah yang disusun secara sistematis dan logis, sedangkan Program adalah Kumpulan pernyataan komputer, sedangkan metode dan tahapan sistematis dalam program adalah algoritma.

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis.

### JENIS FLOWCHART

Flowchart sendiri terdiri dari lima jenis, masing-masing jenis memiliki karakteristik dalam penggunaannya. Berikut adalah jenis-jenisnya:

#### 1. Flowchart dokumen

Pertama ada flowchart dokumen (document flowchart) atau bisa juga disebut dengan paperwork flowchart. Flowchart dokumen berfungsi untuk menelusuri alur form dari satu bagian ke bagian yang lain, termasuk bagaimana laporan diproses, dicatat, dan disimpan.

#### 2. Flowchart program

Selanjutnya kita akan membahas flowchart program. Flowchart ini menggambarkan secara rinci prosedur dari proses program. Flowchart program terdiri dari dua macam, antara lain: flowchart logika program (program logic flowchart) dan flowchart program komputer terinci (detailed computer program flowchart).

#### 3. Flowchart proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

#### 4. Flowchart sistem

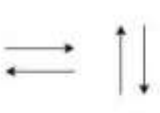


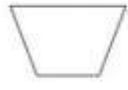


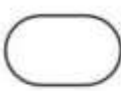
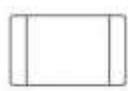



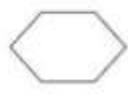
Yang keempat ada flowchart sistem. Flowchart sistem adalah flowchart yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh. Selain itu flowchart sistem juga menguraikan urutan dari setiap prosedur yang ada di dalam sistem.

#### 5. Flowchart skematik

Terakhir ada flowchart skematik. Flowchart ini menampilkan alur prosedur suatu sistem, hampir sama dengan flowchart sistem. Namun, ada perbedaan dalam penggunaan simbol-simbol dalam menggambarkan alur. Selain simbol-simbol, flowchart skematik juga menggunakan gambar-gambar komputer serta peralatan lainnya untuk mempermudah dalam pembacaan flowchart untuk orang awam.

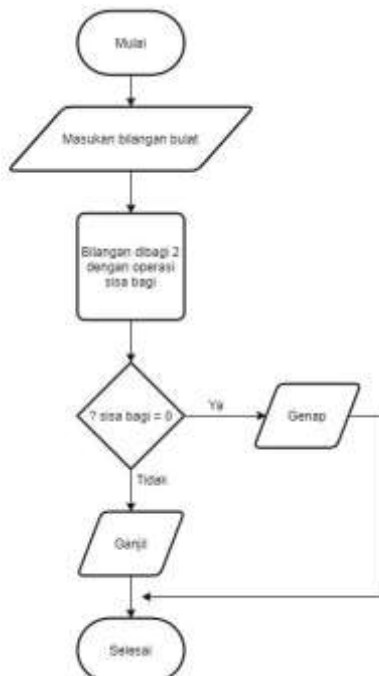
## SIMBOL FLOWCHART

Pada dasarnya simbol-simbol dalam flowchart memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan flowchart.

	<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		<b>Input/output</b> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	<b>On-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		<b>Manual Operation</b> Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	<b>Off-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		<b>Document</b> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.		<b>Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.		<b>Display</b> Simbol yang menyatakan peralatan output yang digunakan.
	<b>Decision</b> Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

## CONTOH FLOWCHART

Untuk membuat sebuah flowchart sederhana kamu diharuskan untuk mengetahui setiap simbol dan juga fungsinya. Nah, di bawah ini saya akan memberikan sebuah contoh flowchart sederhana untuk menentukan apakah bilangan yang dimasukan ganjil atau genap. Berikut adalah contohnya:



Pembahasan:

- Pertama pengguna menginput data yang berupa nilai dari bilangan bulat.
- Kemudian nilai yang dimasukan diproses dengan cara dibagi dengan angka 2.
- Jika sisa bagi sama dengan 0 berarti bilangan yang dimasukan adalah bilangan genap.
- Jika sisa bagi tidak sama dengan 0 berarti bilangan yang dimasukan adalah bilangan ganjil.
- Selesai.

## PEMROGRAMAN DASAR MENGGUNAKAN C#

### A. Struktur program C#

Ada lima bagian pada struktur program c#, bisa dilihat dari contoh gambar di bawah ini ;

```
1  using System;
2  using System.Collections.Generic; // Using
3  using System.Linq;
4  using System.Text;
5
6  namespace IWTekno_dot_com // Namespace
7  {
8      class Program // Class
9      {
10         static void Main(string[] args) // Method Utama
11         {
12             Console.WriteLine("IWTekno.com"); // Statement & Ekspresi
13             Console.ReadLine();
14         }
15     }
16 }
```

#### 1. Using (Resource atau library)

Pada struktur pertama merupakan pendefinisian library apa saja yang digunakan pada program Anda atau library apa yang akan diimpor. Apabila kita ingin menambah class database (menggunakan: using System.Data, maka kita bisa menambahkan dibawahnya. Sebuah program umumnya memiliki beberapa pernyataan using, tergantung kebutuhan program yang dimaksud.

#### 2. Namespace

Untuk struktur kedua ini adalah nama dari project. Dimana berisi beberapa kumpulan class dan objek yang terkait. Misal pada project "Hewan" berisi class "Singa", "Harimau", "Serigala". Dimana untuk awal penulisan diawali dengan Namespace baru nama project-nya. Untuk penulisan, dalam satu blok harus diawali tanda kurung kurawal ( {} ).

#### 3. Class

Struktur ketiga adalah class. Dalam Pemrograman berorientasi Object atau OOP ( Object Oriented Programming ) Class merupakan suatu tipe data yang mendefinisikan seperangkat variable dan method untuk object yang di deklarasikan. Sama dengan namespace, class juga diawali dan di akhiri dengan tanda kurung kurawal ( {} ).

#### 4. Method

Yang keempat, Method adalah sekelompok pernyataan yang dieksekusi bersama untuk melakukan tugas tertentu. Setiap program C# akan berisi setidaknya satu kelas dengan setidaknya satu Method Utama. Dimana method utama dapat memanggil beberapa class dan beberapa method tambahan untuk dieksekusi dalam tugas.

## 5. Statement & Ekspresi

Bagian kelima, berisi sekelompok pernyataan dan kemudian diekspresikan dalam sebuah hasil tugas / program. Misalnya kita ingin menampilkan tulisan IWTekno.com pada sebuah console c#.

Berikut Kode Programnya:

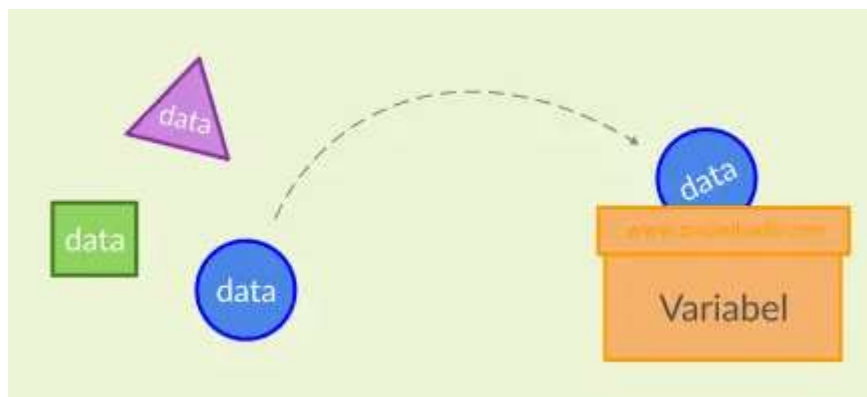
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace IWTekno_dot_com
7  {
8      class Program
9      {
10         static void Main(string[] args)
11         {
12             Console.WriteLine("IWTekno.com");
13             Console.ReadLine();
14         }
15     }
16 }
```

Hasil Pernyataan dan ekspresi di atas adalah sebagai berikut:



## B. Variabel, Konstanta, dan Tipe Data

Variabel adalah sebuah wadah penyimpanan data pada program yang akan digunakan selama program itu berjalan. Data atau nilai yang kita simpan dalam variabel, akan disimpan dalam memori (RAM). Semakin banyak variabel yang dibuat, semakin banyak pula ruang memori yang dibutuhkan. Secara teknis, variabel bisa kita sebut sebagai nama tempat untuk menggantikan alamat memori.



Tipe data adalah jenis-jenis data yang akan disimpan di dalam variabel. Berikut ini beberapa jenis tipe data yang sering digunakan:

- string adalah tipe data yang berupa teks, contoh: "Saya hebat!";
- int adalah tipe data yang berupa angka, Contoh: 3, 9, 0;
- float adalah tipe data yang berupa angka pecahan, contoh 1.2f, 2.4f, 5.5f (huruf f artinya float);
- bool adalah tipe data boolean yang hanya berisi true dan false.

### Cara Membuat Variabel Pada C#

Ada dua cara membuat variabel di C#:

- Pertama (**eksplisit**), dengan menuliskan tipe data lalu diikuti nama variabel.

```
[tipe data] namaVariabel;
```

Contoh:

```
// membuat variabel kosong
string alamat;
int umur;
float beratBadan;
bool isMenikah;

// membuat variabel dan langsung mengisinya
string nama = "Petani Kode";
int umur = 18;
float beratBadan = 59.34;
```

- Kedua (**implicit**), apabila kita tidak tahu tipe data yang akan digunakan, maka membuat variabel bisa menggunakan kata kunci var.

```
var namaVariabel = "isi variabel";
```

Pembuatan variabel dengan var harus kita isi nilainya, karena kalau tidak.. akan terjadi error seperti ini:

**An implicitly typed local variable declarator must include an initializer**

Contoh:

```
// membuat variabel dan langsung mengisinya
var namaWeb = "Petani Kode";
var alamatWeb = "https://www.petanikode.com";
var bahasaPemrograman = "C#";
var umur = 18;
var isMenikah = false;
```

Konstanta hampir sama seperti variabel, bedanya konstanta bersifat immutable artinya nilainya tidak bisa diisi ulang alias konstan. Cara membuat konstanta cukup gunakan kata kunci const, lalu diikuti dengan tipe data dan nilainya.

Contoh:

```
const string NamaKonstanta = "isi atau nilai";
```

nama konstanta dianjurkan menggunakan huruf kapital di awal namanya. Tujuannya untuk membedakannya dengan variabel. Jika di variabel menggunakan huruf kecil di awal, maka di konstanta disarankan menggunakan huruf besar.

## Soal Latihan

1. Perhatikan kode program berikut:

```
using System;

class AritmatikaPenjumlahan
{
    public static void Main (string[] args)
    {
        int mangga, apel, hasil = 0;

        Console.Write("mangga = ");
        mangga = int.Parse(Console.ReadLine());
        Console.Write("appel = ");
        apel = int.Parse(Console.ReadLine());

        // operasi penjumlahan dengan operator +
        hasil = mangga + apel;

        Console.WriteLine ($"Hasil mangga + appel = {hasil}");
    }
}
```

Apabila diberikan input untuk variabel mangga adalah 5 dan apel adalah 7, maka hasilnya adalah...

**Jawaban :** 5+7 hasilnya akan 12.

2. Perhatikan kode program berikut:

```
using System;

class AritmatikaPembagian
{
    public static void Main (string[] args)
    {
        int mangga, orang, hasil = 0;

        Console.Write("jumlah mangga = ");
        mangga = int.Parse(Console.ReadLine());
        Console.Write("jumlah orang = ");
        apel = int.Parse(Console.ReadLine());

        hasil = mangga / orang;

        Console.WriteLine ($"Hasil mangga / orang = {hasil}");
    }
}
```

Apabila diberikan input untuk variabel mangga adalah 20 dan orang adalah 3, maka hasilnya adalah...

**Jawaban :** 6

Loh kok 6!

Bukannya 6.666666?

Ini disebabkan karena kita menggunakan tipe data integer untuk variabel mangga dan orang.

3. Perhatikan kode program berikut:

```
using System;

class ProgramConditional
{
    static void Main(string[] args)
    {
        Console.Write("Berapa mumurmu: ");
    }
}
```

```
int age = int.Parse(Console.ReadLine());

// menggunakan conditional operator
string message = age < 18 ? "Belum cukup umur" : "sudah cukup umur
untuk menikah";

Console.WriteLine(message);
}
}
```

Apabila age diisi dengan nilai 19 maka hasilnya adalah ...

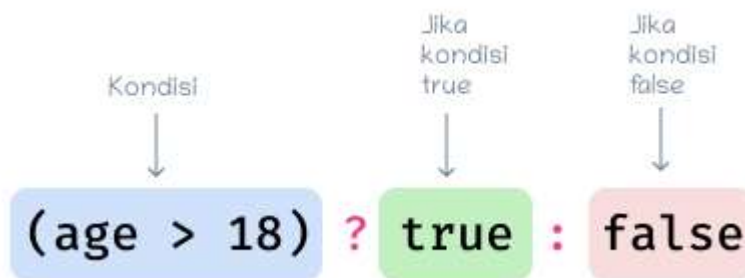
**Jawaban:**

```
Berapa umurmu: 19
sudah cukup umur untuk menikah
```

Perhatikan Conditional Operator (Ternary)!

Conditional operator adalah operator yang membentuk logika jika/maka atau if/else.

Conditional Operator disebut juga operator ternary karena memiliki tiga operan.



Operan pertama adalah kondisi yang akan dicek. Pada bagian ini, kita bisa membuat ekspresi dengan operator perbandingan dan logika. Lalu operan berikutnya adalah ekspresi jika kondisi benar, dan sisanya ekspresi yang akan dipakai jika kondisi salah.