

The background features a decorative graphic consisting of three blue circles of varying sizes, each with a darker blue center and a lighter blue outer ring. These circles are arranged in a triangular pattern. Two thin, light blue lines intersect at the top left, forming a large 'V' shape that frames the circles and the text.

Modul Algoritma Dan Pemrograman Pascal

MODUL ALGORITMAN DAN PEMROGRAMAN

TURBO PASCAL

I. DASAR PEMROGRAMAN

Untuk membuat sebuah program kita pastinya membutuhkan sebuah atau beberapa software yang mendukung pembuatan program tersebut dan sesuai dengan kemampuan kita dalam bahasa pemrograman. Bahasa pemrograman diklasifikasikan menjadi 3 jenis yaitu :

↗ Low - level programming language

Tingkat bahasa pemrograman ini disebut "*rendah*" (low level) dikarenakan kurangnya abstraksi atau kurangnya penggambaran oleh kode instruksi yang ada antara bahasa natural (manusia) dan bahasa mesin. Dan karena itu bahasa ini sering disebut bahasa mesin.

↗ High - level programming language

Bahasa pemrograman di tingkat ini sudah memiliki abstraksi yang luas dan memiliki kemiripan dengan bahasa natural (manusia) terutama bahasa inggris. Dan bahasa ini sangat mudah di gunakan dan biasanya bahasa pemrograman ini digunakan untuk orang awam yang ingin belajar bahasa pemrograman.

↗ Very high - level programming language.

Bahasa ini memiliki abstraksi yang sangat tinggi dari pada High - level language dan biasanya digunakan untuk menunjang produktivitas para programmer profesional.








Nah .. untuk pembelajaran tahap awal kita akan membahas tentang salah satu bahasa pemrograman tingkat tinggi yaitu bahasa pemrograman PASCAL. Sebelum itu kita harus belajar tentang algoritma pemrograman yang biasanya diimplementasikan dengan **Flowchart** atau **Pseudocode**. Pembuatan algoritma ini bertujuan untuk mempermudah pembuatan program, algoritma juga bisa dikatakan sebagai kerangka awal sebuah program.

I.1 Flowchart

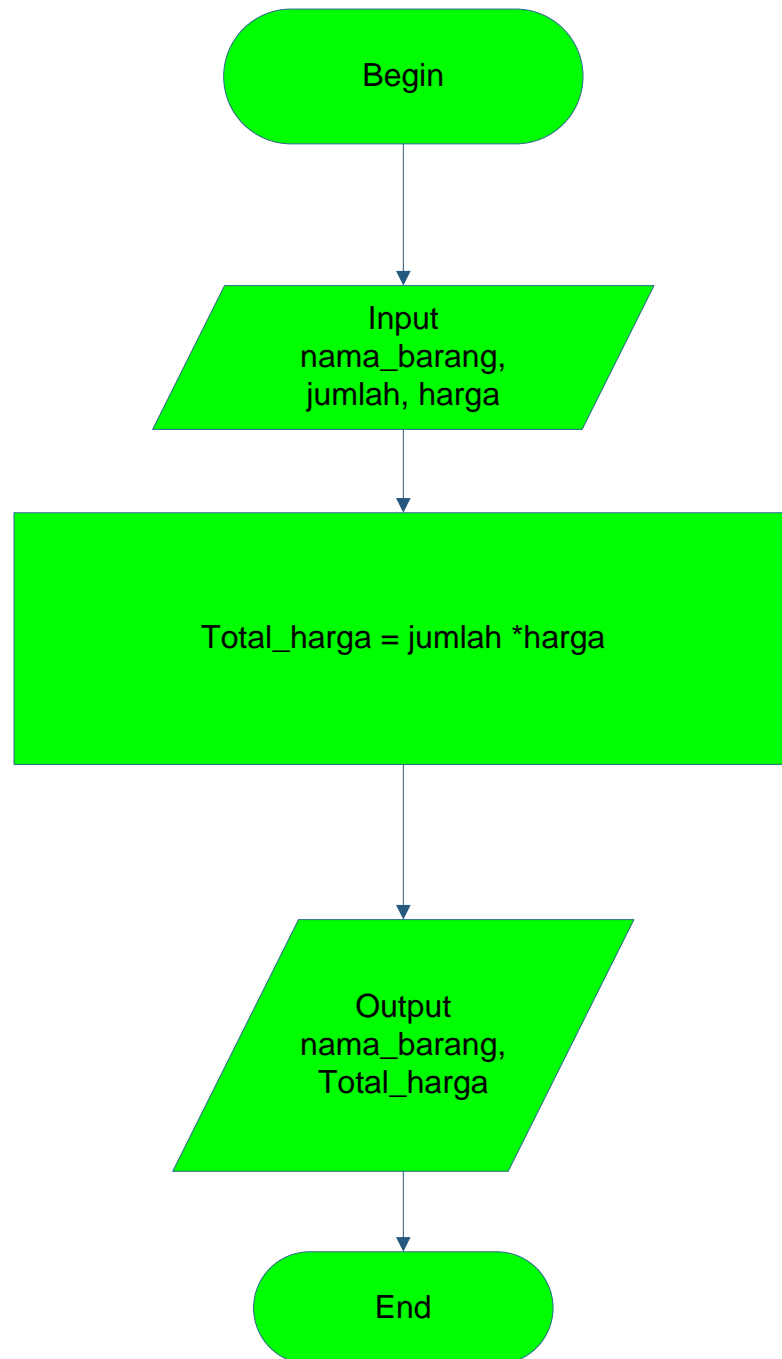
Flowchart merupakan skema penggambaran tahap - tahap esekusi dalam sebuah program yang akan kita buat dalam bentuk simbol. Beberapa macam simbol itu adalah :



Terminator
Sebagai "Start" atau "End" pada Sebuah Flowchart

	<p>Input/Output Untuk menerima inputan atau menampilkan sebuah output hasil proses</p>
	<p>Proses Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi aritmatika ($a = b + c$)</p>
	<p>Conditional/Decision Digunakan untuk menyatakan proses yang membutuhkan keputusan (ya/tidak)</p>
	<p>Preparation Digunakan untuk memberikan nilai awal</p>
	<p>Arrow Sebagai penunjuk alur proses</p>
	<p>Connector (On-page) Digunakan untuk menyatukan beberapa arrow</p>
	<p>Connector (Off-page) Digunakan untuk menghubungkan flowchart yang harus digambarkan pada halaman yang berbeda. Biasanya simboli ini diberi nomor sebagai penanda.</p>

Contoh dari penggunaan flowchart secara sederhana adalah seperti gambar dibawah ini, merupakan flowchart menghitung harga barang. Dan satu lagi hal yang penting dalam flowchart, yaitu dalam penulisan jika terdapat spasi maka digunakan tanda "_" underscore.



Dengan menggunakan flowchart maka tahapan - tahapan yang akan digunakan atau dilaksanakan akan lebih mudah dipahami. Tapi flowchart juga memiliki kelemahan, jika kita akan membuat algoritma sebuah kasus yang besar maka akan menghabiskan kertas atau jika menggunakan komputer maka menggunakan banyak page.

I.2 Pseudocode

Selain flowchart kita juga dapat menyusun algoritma dengan menggunakan *pseudocode*. Pseudocode adalah bentuk informal untuk menggambarkan struktur bahasa pemrograman. Tujuan digunakannya pseudocode adalah :

1. Lebih mudah dipahami
2. Lebih mudah dibaca oleh manusia
3. Lebih mudah dalam pencurahan ide/ gagasan
4. Dan lebih mudah dalam pentransformasian kedalam bahasa pemrograman

Karena penggunaannya lebih simpel maka pseudocode sangat dianjurkan untuk programmer pemula dalam menyusun sebuah algoritma sebuah kasus. Dan biasanya pseudocode menggunakan campuran antara bahasa pemrograman dan bahasa manusia, terutama dengan bahasa pascal. Jika dibandingkan dengan bahasa pascal, pseudocode memiliki kemiripan dalam menuliskan sintaks program dengan pascal sehingga lebih mudah dipelajari. Berikut contoh dari penggunaan pseudocode.

Pascal	Pseudocode
<pre> Program hitungGaji; Var Nama: string; jk, tarif, gaji : integer; Begin {input} Write ('masukan nama : '); readln(Nama); Write ('masukan Jamkerja : '); readln(jk); Write ('masukan Tarif : '); readln(tarif); {proses} gaji := jk*tarif; {output} Write('Nama : '); </pre>	<pre> Algoritma menghitung gaji Kamus nama : text jamkerja, tarifperjam, gaji : numerik Algoritma Input (nama, jamkerja, tarifperjam) Gaji ← jamkerja * tarif perjam Output (nama, Gaji) </pre>

```
Writeln>Nama)
Write('Gaji : ');
Writeln(gaji);
End.
```

I.3 Input/Output (I/O)

Perintah inputan dalam bahasa pascal ada 2 yaitu, *Read* dan *Readln*. Namun yang sering digunakan pada proses input adalah *Readln*. *Readln* dan *Read* memiliki sedikit perbedaan. *Readln* digunakan untuk memasukan data perbaris, artinya setelah menekan tombol Enter maka apa yang kita inputan akan segera masuk lalu berganti baris ke baris yang selanjutnya. Perintah *Read* sama seperti *Readln* hanya saja tidak beraganti baris kebaris selanjutnya, melainkan tetap pada baris yang sama.

Perintah output dalam bahasa pascal ada 2 yaitu, *Write* dan *Writeln*. Seperti halnya *Read* dan *Readln*, *Write* akan mengoutputkan data dalam baris yang sama dan *Writeln* akan mengoutputkan pada suatu baris lalu output yang selnjutnya akan tertulis pada baris setelahnya.

Untuk lebih memahami silahkan tuliskan program ini kedalam program turbo pascal :

```
Program InputOutput;
Var
  nama1, nama2 : String;

Begin
  {Input}
    Write('Masukkan Nama ke 1 : ');read(nama1);
    Writeln('Masukkan Nama ke 2 : ');readln(nama2);

  {Output}
    Writeln(nama1);
    Write(nama2);

  Readln;

End.
```

II. Tipe Data, Variabel dan Konstanta

II.1 Tipe Data

Tipe data dapat di kelompokkan menjadi 2 macam : tipe data dasar dan tipe data bentukan. Tipe data dasar adalah tipe data yang dapat langsung dipakai, sedangkan tipe data bentukan adalah tipe yang telah dihasilkan dari kumpulan tipe data dasar.

A. Tipe data dasar

1. Integer

Tipe data integer digunakan untuk variabel yang menyimpan nilai atau data bilangan bulat, rentang nilainya secara teoritis tidak terhingga dari minus tak hingga ($-\infty$) plus tak hingga (∞). Agar dalam pembuatan program tidak memakan memori yang banyak maka tipe data integer dibagi menjadi 5 macam, yaitu :

Tipe	Rentang nilai
Byte	0..255
Shortint	-128..127
Word	0..65535
Integer	-32768..32767
Longint	-2147483648..2147483647

Contoh : 1, 2, 89, -900, 10000099, -9988787

2. Real

Tipe data real merupakan tipe data yang digunakan untuk variabel yang menyimpan data berupa bilangan desimal atau pecahan. Pemisah desimal dalam bahasa pascal menggunakan titik ("."). Berikut merupakan macam - macam tipe data integer :

Tipe	Ketelitian
Real	$2.9 \times 10^{-39} .. 1.7 \times 10^{38}$
Single	$1.5 \times 10^{-45} .. 3.4 \times 10^{38}$
Double	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$
Extended	$3.4 \times 10^{-4932} .. 1.1 \times 10^{4932}$

Dalam ke 4 tipe tersebut tidak menunjukkan rentang nilai melainkan tingkat ketelitian.

3. Char

Tipe data Char digunakan untuk variabel yang akan menyimpan data berupa karakter. Yang termasuk kedalam tipe data karakter adalah huruf - huruf alfabet, tanda baca, angka dan karakter - karakter khusus seperti '#', '@', '&'...dll. Penulisan dalam bahasa pascal adalah *Char*.

4. String

Tipe data string digunakan untuk variabel yang menyimpan lebih dari satu karakter. Pengertian string sendiri merupakan deretan karakter dengan panjang tertentu. Contoh : ' halo ', 'HAI', 'LaGi ngapain ?'.. dll.

5. Boolean

Tipe data boolean digunakan untuk variabel yang akan menyimpan nilai True atau False. Berdasarkan deskripsi tersebut maka dapat disimpulkan bahwa tipe data boolean hanya memiliki rentang nilai True dan False.

B. Tipe data bentukan

Tipe data bentukan merupakan sekumpulan tipe data dasar yang didefinisikan sendiri oleh sang programmer dan digunakan hanya pada program yang sedang ia buat. Contohnya penggunaan tipe data bentukan adalah :

Dalam Algoritma	Dalam Pascal
type NamaTipe :< filed1 : tipeData; filed2 : tipeData ; ... filedN: Tipedata; >	type NamaTipe=record filed1 : tipeData; filed2 : tipeData ; ... filedN: Tipedata; end;

Contoh Penggunaan

Algoritma Pendataan_mahasiswa; Kamus type mahasiswa :< nim : longint; nama : string; umur : word;	program pendataan_mahasiswa; uses crt; type mahasiswa = record nim : longint; nama : string; umur : word; end;
----------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------


```

>
mhs : mahasiswa;
Algoritma
{input}
  mhs.nim := 30208079;
  mhs.nama:= 'Luna Maya';
  mhs.umur:= 18;
{output}
  output(mhs.nim);
  output (mhs.nama);
  output (mhs.umur);

var
  mhs : mahasiswa;
begin
  clrscr;
  {input}
    mhs.nim := 30208079;
    mhs.nama:= 'Luna Maya';
    mhs.umur:= 18;
  {output}
    writeln(mhs.nim);
    writeln(mhs.nama);
    writeln(mhs.umur);
    readln;
end.

```

Penggunaan tipe data bentukan atau sering disebut Record biasanya digunakan bersama Array... (akan dibahas nanti).

II.2 Variabel

Variabel atau peubah dapat dianalogikan sebagai sebuah tempat untuk menyimpan nilai suatu data atau menyimpan nilai suatu hasil eksekusi program. Nilai yang ditempatkan kedalam variabel tentunya harus sama dengan tipe data variabel tersebut. Nilai suatu variabel dapat berubah - ubah sesuai dengan konteks dan keperluan proses. Contoh :

```

Var  X      : integer;
     Nama   : string;

```

Variabel X bertipe integer berarti hanya bisa diisi oleh nilai bilangan bulat saja dan variabel Nama bertipe string berarti hanya bisa diisi oleh deretan karakter. Dalam penulisan variabel harus diperhatikan beberapa hal berikut :

1. Nama variabel tidak mengandung spasi kosong tau karakter khusus !@#%\$%^&*()-+{}:"<>?[];',./ dan ~. Biasanya sebagai pengganti spasi para programmer menggunakan '_', contoh : Nama_pasien, Kode_produk,dll
2. Cara lain untuk memisahkan kata adalah dengan cara setiap awal kata diberikan huruf kapital. Contoh : JenisKelamin, NilaiAkhir, HasilUjian,dll
3. Nama variabel harus cukup jelas menunjukan tujuan kegunaan variabel tersebut.

II.3 Konstanta

Jika sebuah variabel nilainya dapat berubah - ubah dalam program, lain halnya dengan konstanta memiliki nilai yang tetap dari awal dideklarasikan hingga akhir program. Artinya setelah konstanta telah diberikan nilai maka nilainya langsung dapat dipergunakan dan nilainya selalu tetap. Deklarasi konstanta dalam flowchart menggunakan simbol proses (persegi). Contoh :

Dalam Flowchart	Dalam Pascal
<pre>Phi = 3.14 Maks = 200</pre>	<pre>Const Maks = 200; Phi = 3.14; End;</pre>

III. Operator

Dalam sebuah operasi perhitungan dibutuhkan sebuah atau beberapa operator untuk dapat mengeksekusi nilai yang diinputkan. Nah dalam bahasa pemrograman pascal terdapat beberapa operator, operator itu dibagi 3 jenis yaitu operator aritmatika, operator relasional dan operator logika.

III.1 Operator Aritmatika

Operator ini merupakan operator perhitungan aritmatik. Dan ada yang membutuhkan nilai integer atau real. Yang termasuk tipe operator ini adalah :

Simbol	Deskripsi	Contoh
+	Penjumlahan	$X \leftarrow Y + Z$
-	Pengurangan	$Z \leftarrow K - A$
*	Perkalian	$M \leftarrow X * S$
/	Pembagian	$R \leftarrow O / P$
Div	Hasil bagi	$G \leftarrow E \text{ Div } R$
Mod	Modulo (sisah bagi)	$C \leftarrow A \text{ Mod } B$

Mungkin untuk orang biasa masih awam dengan operan *Div* dan *Mod*. Untuk itu akan saya jelaskan. Untuk operan *Div* merupakan Hasil bagi dan *Mod* merupakan Sisah Bagi, contohnya :

$C \leftarrow 13 / 2$, 13 di bagi 2 hasilnya adalah 6 dan sisah 1. Nah angka 6 merupakan hasil operasi *Div* dan 1 merupakan hasil operasi *Mod*.

$C \leftarrow 13 \text{ Div } 2$, maka hasilnya 6 dan $C \leftarrow 13 \text{ Mod } 2$, maka hasilnya 1.

III.2 Operator Relasional

Operator Relasional akan membandingkan 2 nilai variabel. Yang termasuk tipe operan relasional adalah :

Simbol	Deskripsi	Contoh
=	Sama dengan	$C = A$
≠	Tidak sama dengan	$L \neq M$ atau $K \neq L$
>	Lebih dari	$S > R$
<	Kurang dari	$O < P$
>=	Lebih besar sama dengan	$Y \geq T$
<=	Lebih kecil sama dengan	$J \leq F$

III.3 Operator Logika

Operator logika adalah operator yang akan menghasilkan nilai Boolean (Benar atau Salah). Berikut adalah tipe operator logika :

Simbol	Deskripsi	Contoh
AND	Dan	$(X > 9) \text{ AND } (y < 9)$
OR	Atau	$(K \geq 1) \text{ OR } (P \geq 5)$
NOT	Tidak / Bukan	Not $X > 7$

III.4 Urutan Operasi

Jika dalam sebuah proses terdapat lebih dari satu operan seperti :

$$\text{Hasil} \leftarrow x * 2 \text{ Mod } 2 > y \text{ And } (x \neq 3)$$

Maka terdapat prioritas dalam proses penghitungan yang dilakukan, untuk itu dibuatlah pelevelan operan yaitu tingkat prioritas eksekusi operan yang terdiri dari 5 level. Berikut adalah levelnya :

Simbol	Deskripsi	Level
()	Kurung	1
NOT	Tidak / Bukan	2

*	Perkalian	3
/	Pembagian	
Div	Hasil bagi	
Mod	Modulo (sisah bagi)	
+	Penjumlahan	4
-	Pengurangan	
<	Lebih kecil	5
<=	Lebih kecil sama dengan	
>=	Lebih besar sama dengan	
>	Lebih besar	
=	Sama dengan	6
<>	Tidak sama dengan	
AND	Dan	7
OR	Atau	8

Jadi soal : Hasil $\leftarrow x * 2 \text{ Mod } 2 > y \text{ And } (x \leftrightarrow 3)$ dapat diselesaikan misalnya jika inputannya $x \leftarrow 5$ dan $y \leftarrow 3$ maka tahap pengerjaanya adalah :

Tahap	Proses
1	Hasil $\leftarrow 5 * 2 \text{ Mod } 2 > y \text{ And } (5 \leftrightarrow 3)$
	Hasil $\leftarrow 5 * 2 \text{ Mod } 2 > y \text{ And } (\text{True})$
2	Hasil $\leftarrow 5 * 2 \text{ Mod } 2 > y \text{ And } (\text{True})$
	Hasil $\leftarrow 10 \text{ Mod } 2 > y \text{ And } (\text{True})$
	Hasil $\leftarrow 0 > y \text{ And } (\text{True})$
3	Hasil $\leftarrow 0 > 3 \text{ And } (\text{True})$
	Hasil $\leftarrow \text{False} \text{ And } (\text{True})$
4	Hasil $\leftarrow \text{False}$

