

## **Bab 2**

# **DASAR-DASAR ALGORITMA**

Pada bab ini anda akan mempelajari

1. Nama (pengenal)
2. Nilai
3. Variabel dan Konstanta
4. Penugasan (Assignment)
5. Jenis-jenis tipe data
6. Jenis-jenis operasi dan kaitannya dengan tipe data
7. Operasi logika dan cara menggunakannya
8. Input
9. Output

---

### **Nama (Pengenal)**

Nama atau pengenal adalah sesuatu yang digunakan untuk identifikasi, yaitu sebagai pembeda antara obyek yang satu dengan obyek lain. Di dalam algoritma, obyek yang harus diberi nama adalah

- a. Judul algoritma
- b. konstanta
- c. variabel
- d. tipe
- e. prosedur
- f. fungsi

Karena fungsinya sebagai pembeda, maka nama obyek haruslah unik. Artinya, satu nama hanya untuk satu obyek. Tidak diperkenankan memberikan satu obyek dengan dua nama berbeda atau dua buah obyek yang berbeda dengan satu nama. Misalnya, di satu bagian anda menamai sebuah variabel untuk menyimpan nilai akhir dengan nama **NilAkhir**, sementara di bagian lain anda menamai obyek tersebut dengan **NilaiAkhr**.

Penulisan nama di dalam algoritma pada dasarnya tidak memiliki batasan. Namun untuk kemudahan beradaptasi dengan bahasa pemrograman Pascal, penulisan pengenal dibatasi dengan aturan berikut ini :

- Terdiri dari huruf (A-Z, a-z), angka (0-9), dan tanda garis bawah ( \_ )
- Dimulai dengan huruf
- Tidak ada spasi
- Tidak mengandung simbol khusus seperti &, %, \*, @, #, !, ~, \$, dll
- Bukan merupakan kata baku (*reserved word*) yang dikenal dalam bahasa pemrograman Pascal. Contoh **write**, **read**, **if**, **then**, **for**, **while**, **do**, **repeat**, **until**, dll.
- Sedapatnya singkat tapi mudah diingat (*optional*). Misalnya, untuk pengenal **Gaji Pokok** dengan **Gapok**, **Jumlah** dengan **Jum**, **Keterangan** dengan **Ket**, dll

Berikut ini adalah contoh penulisan pengenal yang benar.

Nama / Pengenal
Ket1
AorPkL
Nama_Mahasiswa
GajiPokok

Contoh pemberian nama yang salah

Nama / Pengenal	Keterangan
1Angka	Dimulai dengan angka
N@m@	Mengandung simbol khusus (@)
Nama Mahasiswa	Mengandung spasi
Oke!	Mengandung simbol khusus (!)
Begin	Merupakan kata baku

## Nilai

Nilai adalah suatu besaran untuk menyatakan suatu hal. Nilai ini dapat disimpan di dalam konstanta atau variabel. Ada beberapa tipe nilai, yaitu diantaranya yang paling sederhana adalah boolean, numerik (numerik bulat dan pecahan) dan karakter (char dan string). Contoh

- Pernyataan  $4 > 5$  adalah **salah/false** (nilai boolean)
- Jumlah buku di perpustakaan adalah **1493** (nilai numerik bulat)
- Tinggi badan Rusli **173,4** (nilai numerik pecahan)
- Nilai Kamil dalam kuliah Kalkulus adalah **'A'** (nilai char)
- Saya tinggal di **'Bandung'** (nilai string)

## Variabel dan Konstanta

Untuk dapat mengolah data (nilai), komputer membutuhkan media penyimpanan. Ada dua media penyimpanan data sederhana, yaitu konstanta dan variabel. Sebelum menggunakannya, kedua media penyimpanan data ini harus didefinisikan dahulu namanya di bagian deklarasi algoritma.

Variabel adalah media penyimpanan data yang nilai datanya dapat berubah selama proses eksekusi. Deklarasi variabel menyatakan nama variabel dan jenis data yang dapat disimpannya. Bentuk umum deklarasinya adalah sebagai berikut

## Deklarasi

**<Variabel1> : tipeData1**

**<Variabel2> : tipeData2**

**<Variabel3> : tipeData3**

Jika tipe datanya sama, deklarasi variabel dapat ditulis

## Deklarasi

**<Variabel2>, <Variabel3>, ... : tipeData2**

Contoh.

Deklarasi

```
var
  nama : string
  usia : integer
  panjang, lebar, tinggi : real
```

Konstanta adalah media penyimpanan data yang nilai datanya tidak dapat berubah selama proses eksekusi (diluar proses eksekusi dapat saja dirubah). Berbeda dengan deklarasi variabel, deklarasi konstanta menyatakan nama konstanta dan nilai data yang disimpannya. Bentuk umum deklarasinya adalah sebagai berikut

## Deklarasi

**<Konstanta1> : Nilai1**

**<Konstanta2> : Nilai2**

Contoh

Deklarasi

```
const
  Nama = 'Anna'
  Usia = 25
  NilaiAkhir = 24.11
```

## Tipe Data

Setiap variabel harus memiliki kejelasan tentang data apa yang dapat disimpannya karena variabel tersebut tidak bisa menyimpan semua jenis data, melainkan hanya bisa menyimpan satu jenis tipe data. Dalam konteks algoritma ini, ada lima jenis tipe data yang akan dibahas, yaitu :

✓ **Boolean**

Tipe ini hanya mengenal dua nilai yaitu true (benar) dan false (salah).

✓ **Bilangan bulat (integer)**

Tipe ini hanya mengenal bilangan bulat ..., -3, -2, -1, 0, 1, 2, 3, ... Di dalam Turbo Pascal tipe data bilangan bulat memiliki rentang nilai berhingga. Berikut adalah beberapa varian tipe bilangan bulat, jangkauan nilai, serta ukuran memori untuk menyimpan nilai tersebut.

**Tabel 2.1.**

<b>Tipe</b>	<b>Jangkauan</b>	<b>Ukuran (byte)</b>
Byte	0..255	1
Shortint	-128...127	1
Integer	-32768...32767	2
Word	0...65535	2
Longint	-2147483648...2147483647	4

✓ **Bilangan real**

Tipe ini hanya mengenal data bilangan real (gabungan dari bilangan pecahan dan bulat). Contoh : -2.3419, 394.0023, 1.1, 0.002, -11.00, 1.25E+5 (artinya  $1.25 \times 10^5$ ), 2.78E-8 (artinya  $2.78 \times 10^{-8}$ ), dll. Seperti juga tipe bilangan bulat, bilangan real juga memiliki rentang nilai yang berhingga dan memiliki beberapa varian. Berikut tabel varian tipe bilangan bulat, jangkauan nilai, serta ukuran memori untuk menyimpan nilai tersebut.

**Tabel 2.2.**

<b>Tipe</b>	<b>Jangkauan</b>	<b>Ukuran (byte)</b>	<b>Digit</b>
Real	$\pm 2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	6	11-12
Single	$\pm 1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	4	7-8
Double	$\pm 5 \times 10^{-324} \dots 1.7 \times 10^{308}$	8	15-16
Extended	$\pm 3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$	10	19-20
Comp	$\pm 9.2 \times 10^{18} \dots 9.2 \times 10^{18}$	8	19-20

**Catatan** : variabel bertipe real dapat digunakan untuk menyimpan nilai bertipe integer. Namun, setelah nilai tersebut tersimpan, maka nilai itu bukan lagi nilai integer lagi, melainkan real.

✓ **Char**

Tipe ini untuk menampung nilai berupa simbol-simbol karakter yang kita kenal seperti huruf 'A' sampai 'Z', 'a' sampai 'z', '0' sampai '9', '@', '#', '\$', '&', '+', '/', dst. Perhatikan: setiap kali kita menulis karakter selalu diapit oleh tanda ( ' )

✓ **String**

Tipe ini merupakan gabungan dari beberapa karakter. Misalnya 'lemari', 'mesin cuci', 'Indonesia tanah airku', '#4dfjhver elnrejkne567^PJK', dll. Perhatikan : penulisan string selalu diapit dengan ( ' ' )

**Penugasan (Assignment)**

Penugasan merupakan perintah untuk menyimpan suatu nilai pada sebuah variabel. Penugasan memiliki dua komponen yang dipisahkan oleh operator penugasan berbentuk panah dengan arah ke kiri ( $\leftarrow$ ). Bagian sebelah kiri adalah bagian tempat menyimpan nilai hasil operasi. Bentuknya berupa variabel. Sedangkan yang di kanan tanda panah merupakan bagian pernyataan operasinya sendiri, bentuknya dapat berupa variabel, konstanta, nilai atau gabungan ketiganya.

Perintah Teks Umum	Perintah Teks Standar
Kalikan variabel <b>x</b> dan <b>y</b> , simpan pada <b>z</b>	$z \leftarrow x * y$
- Tambahkan <b>a</b> dengan satu simpan lagi di <b>a</b> , atau - Naikkan nilai <b>a</b> sebanyak satu	$a \leftarrow a + 1$

Contoh penulisan dalam bentuk flowchart



Penugasan atau pemberian nilai pada suatu variabel dapat dilakukan dengan beberapa cara, yaitu :

**a. NamaVariabel ← Nilai**

Contoh : JumlahBuku ← 20  
MerkMobil ← 'Honda'  
Keterangan ← false

**b. NamaVariabel ← Konstanta**

Contoh : Suhu ← SuhuAwal  
Nama ← NamaDepan  
Konstan ← Phi

**SuhuAwal, NamaDepan, Phi** merupakan konstanta

**c. NamaVariabel ← Variabel**

Contoh : X ← Y  
SuhuCelcius ← SuhuKelvin  
KecepatanAwal ← KecepatanAkhir

**Y, Suhu Celcius, SuhuKelvin, KecepatanAkhir** merupakan variabel.

#### d. NamaVariabel ← Ekspresi

Contoh : Jumlah ← Jumlah + 1  
Luas ← Phi\*Jari\*Jari  
Pilih ← A > B

Dengan operasi penugasan ini, maka nilai baru yang disimpan akan menghapus nilai yang lama. Misalnya, **X** diketahui menyimpan nilai 5. Kemudian dilakukan operasi penugasan berikut :

$x \leftarrow 10$

Maka **X** kini tidak lagi bernilai 5, tapi 10.

Yang harus diingat adalah bahwa nilai, konstanta, dan variabel yang berada di sebelah kanan operator penugasan ( ← ) memiliki tipe data yang sama dengan variabel yang ada di sebelah kiri. Jika di sebelah kanan operator adalah ekspresi, maka ekspresi itu harus menghasilkan nilai yang bertipe sama dengan variabel yang di sebelah kanan. Misalnya, diketahui variabel A (integer), B (integer), C (string), maka operasi

$C \leftarrow A + B$

adalah operasi yang **salah** karena sisi kanan dan kiri memiliki tipe data yang tidak bersesuaian. Tipe data **C** yang benar adalah integer atau real, karena dapat menyimpan hasil operasi penjumlahan **A** dengan **B**.

**Catatan** : bentuk operator penugasan di dalam pemrograman Pascal adalah ( := )

#### Operasi

Operasi adalah proses memanipulasi atau pengolahan terhadap data. Misalnya, untuk menghitung luas suatu persegi panjang, data diolah atau dimanipulasi dengan cara mengalikan data variabel **panjang** dengan **lebar**, hasilnya disimpan pada variabel **luas**. Atau dapat ditulis :

luas ← panjang \* lebar

Jika pada variabel **panjang** dan **lebar** masing-masing disimpan nilai 5 dan 3, maka hasil perkalian keduanya, yaitu 15 akan disimpan pada variabel **luas**.

Deklarasi variabel selain untuk menyatakan nama variabel, juga berfungsi untuk memberi tahu kepada *compiler* mengenai operasi apa saja yang bisa dilakukan oleh variabel bersangkutan. Untuk suatu variabel bertipe tertentu, berlaku jenis operasi tertentu pula. Misalnya, untuk dua variabel bertipe string tidak dikenal operasi perkalian atau penjumlahan.

Berikut beberapa jenis operasi, operator dan tipe hasil dalam berkaitan dengan tipe variabel yang terlibat di dalam operasi tersebut.

**a. Operasi antara variabel bertipe boolean dengan boolean**

**Tabel 2.3.**

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Logika	logika and (and) logika atau (or) logika negasi (not) logika xor (xor)	Boolean

Contoh.

Variabel	C	←	A	operator	B
Nilai	False	←	true	and	false
	true	←	true	or	false

**b. Operasi antara variabel bertipe integer dengan integer**

**Tabel 2.4.**

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Aritmatika	penjumlahan (+) pengurangan (-) perkalian (*)	Bulat

	pembagian bulat (div) sisa pembagian (mod)	
Arimatika	pembagian (/)	Real
Perbandingan	lebih kecil sama dengan ( $\leq$ ) lebih kecil ( $<$ ) sama dengan ( $=$ ) lebih besar ( $>$ ) lebih besar sama dengan ( $\geq$ ) tidak sama dengan ( $\neq$ )	Boolean

### Keterangan

**div** : operator pembagian bulat

Contoh :  $5 \text{ div } 2 = 2$

$15 \text{ div } 3 = 5$

**mod** : operator sisa pembagian

Contoh :  $5 \text{ mod } 2 = 1$

$15 \text{ mod } 3 = 0$

Contoh.

Variabel	C	←	A	operator	B
Nilai	20	←	4	*	5
	true	←	4	>	5
	2	←	5	mod	3
	1	←	5	div	3

Catatan : angka 20 di atas merupakan bilangan bulat

### c. Operasi antara variabel bertipe real dengan real

**Tabel 2.5.**

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Arimatika	penjumlahan (+) pengurangan (-) perkalian (*) pembagian (/)	Real

Perbandingan	lebih kecil sama dengan	(<=)	Boolean
	lebih kecil	(<)	
	lebih besar	(>)	
	lebih besar sama dengan	(>=)	
	tidak sama dengan	(<>)	

Catatan : Operasi kesamaan (=) untuk operasi real-real tidak dikenal

Contoh.

Variabel	C	←	A	operator	B
Nilai	20	←	4	*	5
	true	←	4	>	5

Catatan : angka 4, 5, dan 20 di atas merupakan bilangan real

#### d. Operasi antara variabel bertipe char dengan char

Tabel 2.6.

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Penyambungan	+	String

Contoh.

Variabel	C	←	A	operator	B
Nilai	'OK'	←	'O'	+	'K'
	'47'	←	'4'	>	'7'

#### e. Operasi antara variabel bertipe string dengan string

Tabel 2.7.

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Penyambungan	+	String

Contoh.

Variabel	C	←	A	operator	B
Nilai	'Kota Bandung'	←	'Kota '	+	'Bandung'

**f. Operasi antara variabel bertipe string dengan char**

**Tabel 2.8.**

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Penyambungan	penyambungan (+)	String

Contoh.

Variabel	C	←	A	operator	B
Nilai	'SIM A'	←	'SIM '	+	'A'

**g. Operasi antara variabel bertipe real dengan integer**

**Tabel 2.9.**

Operasi yang bisa dilakukan	Operator	Tipe Hasil
Aritmatika	penjumlahan (+) pengurangan (-) perkalian (*) pembagian (/)	Real
Perbandingan	lebih kecil sama dengan (<=) lebih kecil (<) sama dengan (=) lebih besar (>) lebih besar sama dengan (>=) tidak sama dengan (<>)	Boolean

Contoh.

Variabel	C	←	A	operator	B
Nilai	2.5	←	2	+	0.5
	false	←	3	>	3.5

**Kasus 1**

A dan B adalah variabel bertipe boolean. Nilai **A** dan **B** masing-masing adalah false dan true. Pada **A** dan **B** dilakukan operasi **and** yang hasilnya disimpan pada **C**. Operasinya dapat dinyatakan sebagai berikut :

$$C \leftarrow A \text{ and } B$$

Karena hasil operasi tersebut menghasilkan nilai bertipe boolean, yaitu true, maka variabel C harus bertipe boolean juga.

## Kasus 2

Misal, diketahui **X** dan **Y** adalah variabel bertipe bilangan bulat (integer) maka padanya dapat dilakukan operasi penjumlahan

$$Z \leftarrow X + Y;$$

dengan **Z** merupakan hasil operasi, **X** dan **Y** merupakan operan (variabel yang dioperasikan) dan tanda '+' merupakan operator penjumlahan. Hasil penjumlahan ini berupa bilangan integer, sehingga Z haruslah merupakan variabel bertipe integer. Nilai integer juga bisa disimpan oleh variabel bertipe real, sehingga Z juga bisa bertipe real. Selain itu Z, yang bertipe integer atau real juga bisa menampung hasil perkalian X dengan Y seperti berikut :

$$Z \leftarrow X * Y$$

Tetapi tidak demikian halnya dengan operasi pembagian berikut :

$$Z \leftarrow X / Y$$

karena meskipun operasi pembagian dapat dilakukan pada dua buah variabel bertipe integer, hasil operasi pembagian adalah data bertipe real, sehingga **Z** yang menampung hasil pembagian **X** dengan **Y** harus bertipe real juga, dan tidak boleh bertipe integer.

### Kasus 3

Misal, P dan Q adalah variabel bertipe char dengan nilai masing-masing '1' dan '2'. Perhatikan bahwa '1' dan '2' di sini bukan merupakan angka (numerik), sehingga kita tidak bisa melakukan operasi perkalian

$$R \leftarrow P \times Q$$

layaknya perkalian dari dua buah angka.

Jika antara P dan Q dilakukan operasi penggabungan (+) (perhatikan sekali lagi, operator (+) di sini bukanlah penjumlahan, tetapi penggabungan!) dengan ekspresi berikut

$$R \leftarrow P + Q$$

maka hasil operasi tersebut akan menghasilkan nilai '12', sehingga R dengan demikian haruslah merupakan variabel bertipe string.

Selain antara variabel dengan variabel, operasi juga dapat dilakukan antara variabel-konstanta, variabel-nilai, dan konstanta-nilai.

### Latihan

Isilah kolom **Keterangan** dengan **benar** jika operasi sesuai dengan tipe data atau **salah** jika operasi tidak sesuai dengan tipe data.

Variabel	Tipe Data	Operasi	Keterangan
(x, y, z)	(real, real, real)	$x \leftarrow y \text{ div } z$	
(x, y, z)	(string, char, char)	$x \leftarrow y + z$	
(x, y, z)	(integer, integer, integer)	$x \leftarrow y/z$	
(x, y, z)	(char, char, char)	$x \leftarrow y+z$	
(x, y, z)	(boolean, boolean, boolean)	$x \leftarrow y/z$	

## Operator Logika

Ada empat operator logika yang akan dibahas, yaitu : **not**, **and**, **or**, **xor**. Yang perlu diingat adalah operator logika bekerja pada operan (variabel yang dioperasikan) bertipe boolean, dan tipe hasilnya pun bertipe boolean. Misalnya,

$A \leftarrow B \text{ and } C$

maka A, B dan C semuanya bertipe boolean Misalnya, diketahui tiga variabel bertipe boolean, yaitu A, B, dan C. Maka kita dapat melakukan operasi :

1.  $C \leftarrow \text{not } A$
2.  $C \leftarrow A \text{ and } B$
3.  $C \leftarrow A \text{ or } B$
4.  $C \leftarrow A \text{ xor } B$

Maka tabel solusinya untuk operasi no 1 adalah

**Tabel 2.10**

<b>A</b>	<b>C <math>\leftarrow</math> not A</b>
<b>true</b>	<b>False</b>
<b>false</b>	<b>True</b>

Dan tabel tabel solusi operasi untuk no 2-4 :

**Tabel 2.11.**

<b>A</b>	<b>B</b>	<b>C <math>\leftarrow</math> A and B</b>	<b>C <math>\leftarrow</math> A or B</b>	<b>C <math>\leftarrow</math> A xor B</b>
<b>true</b>	<b>true</b>	<b>true</b>	<b>true</b>	<b>false</b>
<b>true</b>	<b>false</b>	<b>false</b>	<b>true</b>	<b>true</b>
<b>false</b>	<b>true</b>	<b>false</b>	<b>true</b>	<b>true</b>
<b>false</b>	<b>false</b>	<b>false</b>	<b>false</b>	<b>false</b>

Bagaimana cara mengingatnya?

- C ← not A** : C bernilai kebalikan dari nilai A
- C ← A and B** : C bernilai benar jika A dan B bernilai benar
- C ← A or B** : C bernilai benar jika salah satu dari B bernilai benar
- C ← A xor B** : C bernilai benar jika nilai A dan B berbeda nilai

**Derajat Operator**

Jika diberikan operasi berikut ini :

$$X \leftarrow 4 + 5 * 2$$

maka, berapa nilai yang disimpan oleh x ? Apakah 18, hasil dari 4 ditambah 5 kemudian dikali 2 ? Atau 14, hasil dari 5 dikali 2 kemudian ditambah 4 ? Dalam kaitan ini, maka kita perlu mengenal derajat operator sehingga dapat mengetahui operasi mana yang didahulukan bila berhadapan dengan lebih dari satu operator atau operasi.

**Tabel 2.12.**

<b>Operator</b>						<b>Derajat</b>
<b>Not</b>						<b>Tertinggi</b>
*	/	<b>div</b>	<b>mod</b>	<b>and</b>		
+	-	<b>or</b>	<b>xor</b>			
=	<>	<	>	<=	>=	<b>Terendah</b>

Berdasarkan tabel 2.12 maka dapat disimpulkan bahwa nilai yang disimpan X adalah 14, karena operasi perkalian (\*) lebih tinggi derajatnya dari operasi penjumlahan (+) sehingga operasi perkalian lebih didahulukan daripada operasi penjumlahan. Namun demikian, jika kita tidak ingin dipusingkan dengan derajat operator maka kita cukup menambahkan kurung pada operasi tersebut sesuai dengan keinginan kita. Jadi, jika yang dimaksud adalah 4 ditambah 5 baru kemudian dikali 2, maka operasi tersebut cukup ditulis dengan

$$X \leftarrow (4 + 5) * 2$$

Pertanyaan : Jika X menyimpan hasil operasi berikut ini

$$X \leftarrow B * C > D + E \text{ mod } 3$$

- a. Jika B, C, D dan E adalah variabel bertipe integer, tentukanlah tipe data X !
- b. Jika B, C, D, E bernilai masing-masing 2, 3, 7 dan 5, tentukanlah nilai yang disimpan X !

Jawab :

Berdasarkan derajat operatornya, maka operasi tersebut dapat kita perjelas dengan

$$X \leftarrow (B * C) > (D + (E \text{ mod } 3))$$

- a. Operasi  $(B * C)$  menghasilkan nilai integer,  $(D + (E \text{ mod } 3))$  juga menghasilkan nilai integer, maka operasi tersebut pada dasarnya adalah operasi perbandingan antara dua nilai integer, yang hasilnya adalah nilai boolean. Dengan demikian tipe data X adalah boolean. maka tipe data X adalah boolean.
- b. Nilai yang disimpan X dari operasi tersebut adalah false. (silakan buktikan sendiri!)

### **Masukan (Input)**

Masukan atau input adalah proses menyimpan suatu nilai ke dalam suatu variabel melalui piranti masukan. Kata perintahnya dalam algoritma standar adalah **read** atau **readln** (pemakaian **readln** lebih banyak digunakan di dalam program Pascal)

Bentuk umum perintahnya adalah :

$$\text{read}(\text{argumen1}, \text{argumen2}, \dots)$$

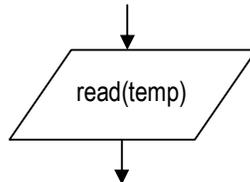
Argumen dalam hal ini adalah media penyimpanan nilai input tersebut, yaitu variabel. Banyaknya bisa satu atau lebih. Berikut ini adalah contoh perintah input dalam bentuk teks standar dan umum.

1. Perintah teks umum :Masukkan nilai temperatur

Perintah teks standar : read (temp)

Keterangan : **temp** merupakan variabel yang disiapkan untuk menyimpan nilai temperatur

Bentuk flowchart :

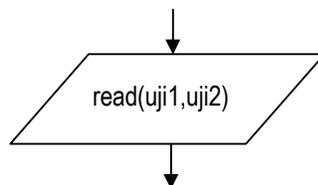


2. Perintah teks umum :Masukkan nilai ujian1 dan ujian2

Perintah teks standar : read (uji1, uji2)

Keterangan : **uji1** dan **uji2** adalah variabel yang disiapkan untuk menyimpan data Ujian 1 dan Ujian 2

Bentuk flowchart



### Keluaran (Output)

Keluaran atau output adalah proses menampilkan nilai pada media keluaran. Media keluaran pada komputer dapat berupa layar monitor, printer, ploter, speaker, dll. Namun dalam konteks ini media keluaran yang kita maksud adalah layar / monitor komputer. Ekspresi perintahnya secara umum adalah

**write(argumen1, argumen2, ...)**

**writeln(argumen1, argumen2,...)**

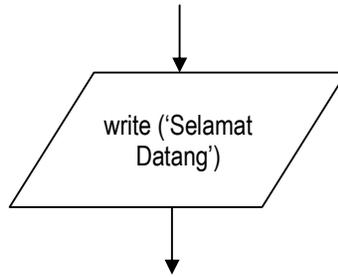
Banyaknya argumen dapat lebih dari satu. Bentuknya bisa berupa nilai, variabel, konstanta, dan ekspresi. Contoh :

Perintah teks umum :Tampilkan nilai string 'Selamat Datang'

Perintah teks standar :write('Selamat Datang')

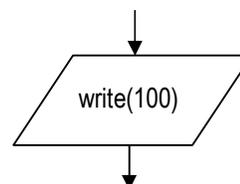
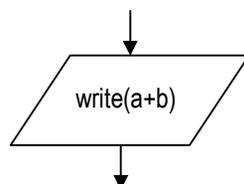
Bentuk keluaran : Selamat Datang

Bentuk flowchart :



PerintahTeks Umum	Perintah Teks Standar	Keterangan	Bentuk Keluaran
Tampilkan nilai 'Selamat Datang'	<b>write ('Selamat Datang')</b>	Perintah menampilkan nilai (string) ' <b>Selamat Datang</b> '. Perhatikan, penulisan nilai string selalu diapit oleh tanda ('')	Selamat Datang
Tampilkan nilai variabel temperatur	<b>write (temp)</b>	Perintah menampilkan nilai yang terkandung pada variabel temp. Misalnya nilai yang dikandung temp = 25	25
Tampilkan nilai a dan b secara horizontal	<b>write (a, b)</b>	Perintah menampilkan nilai a dengan b. Misalnya nilai a=7, dan b =11	7 11
Tampilkan nilai penjumlahan a dengan b	<b>write(a+b)</b>	Perintah menampilkan hasil operasi penjumlahan variabel a dengan b. Misalnya nilai a=7, dan b =11	18
Tampilkan nilai 100	<b>write (100)</b>	Perintah menampilkan nilai numerik <b>100</b>	100
Tampilkan konstanta phi	<b>write (phi)</b>	Tampilkan nilai konstanta phi (3.14)	3.14

Contoh penulisan perintah output dalam bentuk flowchart



Perintah **write** digunakan jika tampilan berikutnya hendak ditulis di sebelah kanan tampilan sebelumnya, sedangkan **writeln** digunakan jika tampilan berikutnya hendak ditulis satu baris di bawah tampilan sebelumnya. Contoh.

Perintah	Keluaran
write(100) write(100)	100 100
writeln(100) write(100)	100 100

### Algoritma Sederhana

Berikut ini adalah beberapa contoh algoritma sederhana

- a. Algoritma menampilkan tulisan 'Selamat Datang'

#### Algoritma 2.1.

---

```

Algoritma SelamatDatang1
{Menampilkan tulisan Selamat Datang ke media keluaran}

Deklarasi
  {tidak ada}

Deskripsi
  Write('Selamat Datang')

```

---

Maka keluaran algoritma : **Selamat Datang**

- b. Algoritma menampilkan tulisan 'Selamat datang, <nama>', dengan <nama> adalah variabel string yang nilainya diperoleh dari input.

#### Algoritma 2.2.

---

```

Algoritma SelamatDatang2
{Menampilkan tulisan 'Selamat Datang' dengan input nama
seseorang ke media keluaran}

```

```
Deklarasi
var
    nama : string

Deskripsi
Read(nama)
Write('Selamat Datang, ', nama)
```

---

Jika variabel nama diinput dengan nilai 'Ahmad' maka keluarannya menjadi

**Selamat Datang, Ahmad**

- c. Algoritma menghitung penjumlahan dua buah bilangan

### **Algoritma 2.3.**

---

Algoritma Jumlah  
*{Menginput nilai dua buah bilangan kemudian menampilkan hasil penjumlahannya}*

```
Deklarasi
var
    Bil1, Bil2, Jum : integer
```

```
Deskripsi
Read(Bil1, Bil2)
Jum ← Bil1 + Bil2
Write(Bil1, ' ditambah ', Bil2, ' = ', Jum)
```

---

Bila input **Bil1** dan **Bil2** masing-masing adalah **4** dan **5** maka keluaran algoritma adalah

**4 ditambah 5 = 9**

### **SOAL**

1. Jelaskan perbedaan antara konstanta dengan variabel.
2. Jelaskan pengertian input dan output
3. Apa artinya operator penugasan ('←')
4. Jelaskan perbedaan antara perintah **write** dan **writeln**
5. Sebutkan tipe data untuk menyatakan hal berikut ini

- a. Banyaknya pengunjung perpustakaan
  - b. Ukuran baju (s, m, l, xl)
  - c. Luas sebuah perkebunan
  - d. Nama-nama kota di Indonesia
  - e. Jenis kelamin
6. Diketahui **A**, **B**, dan **C** masing-masing bertipe real, integer, integer. Manakah di antara operasi berikut yang salah. Jelaskan juga alasannya.
- a.  $A \leftarrow B + C$
  - b.  $A \leftarrow B / C$
  - c.  $C \leftarrow A + B$
  - d.  $A \leftarrow B \text{ div } C$
  - e.  $B \leftarrow A \text{ mod } C$
7. Tuliskan tipe data variabel berikut yang sesuai dengan operasinya
- a.  $C \leftarrow A / B$
  - b.  $P \leftarrow Q \text{ or } R$
  - c.  $Z \leftarrow A = B$
  - d.  $I \leftarrow J \text{ or } (X = Y)$
8. Nyatakanlah perintah-perintah di bawah ini dengan teks algoritma standar berikut bentuk flowchartnya
- a. Masukkan nilai variabel N
  - b. Jumlahkan Q dan R, simpan sebagai P
  - c. Isikan variabel Nama dengan 'Taufik'
  - d. Tambahkan A dengan B, kemudian bagi 2, lalu simpan sebagai C
  - e. Tampilkan variabel P dan Q
  - f. Naikkan variabel A sebanyak 2
  - g. Tampilkan tulisan 'Halo, apa kabar?'
9. Tuliskan variabel input yang dibutuhkan untuk membuat algoritma penyelesaian masalah berikut ini. Sebutkan juga tipe datanya masing-masing.
- a. Menghitung luas segitiga

- b. Menghitung luas permukaan balok
- c. Menentukan status kelulusan mahasiswa berdasarkan nilai UTS (Ujian Tengah Semester), UAS (Ujian Akhir Semester), dan Tugas
- d. Menghitung jarak dari dua buah titik