

# Algoritma Dan Pemrograman

## Pertemuan 2

# TIPE DATA, OPERATOR DAN EKSPRESI

Program Komputer memanipulasi data (variabel dan konstanta) di dalam memori.

## ▶ TIPE DATA :

- ▶ Untuk menyatakan tipe data dari sebuah variabel (peubah) pada Deklarasi.

## ▶ OPERATOR :

- ▶ Menspesifikasikan operasi apa yang dapat dilakukan terhadap peubah (*variabel*) dan konstanta.

## ▶ EKSPRESI :

- ▶ Mengkombinasikan peubah-peubah dan konstanta untuk menghasilkan hasil baru.

# TIPE DATA

## 1. TIPE DATA DASAR :

- ▶ Tipe yang dapat langsung dipakai.

## 2. TIPE DATA BENTUKAN :

- ▶ Tipe dasar atau dari tipe bentukan lain yang sudah didefinisikan sebelumnya.

# 1. TIPE DATA DASAR

- ▶ Dalam pemrograman, yang termasuk ke dalam tipe dasar adalah : Bilangan logik, bilangan bulat, karakter, bilangan riil, dan string
- ▶ Bilangan logik, bilangan bulat, dan karakter disebut juga dengan TIPE ORDINAL (nilainya ada dalam urutan).

# BILANGAN LOGIK

## ▶ NAMA TIPE :

- ▶ Boolean

## ▶ RANAH NILAI :

- ▶ Dua buah nilai : Benar (*true*) dan Salah (*false*)
- ▶ bilangan logik : benar  $\rightarrow$  1, salah  $\rightarrow$  0

## ▶ KONSTANTA :

- ▶ True dan False

## ▶ OPERASI :

- ▶ Operasi Logika atau operasi boolean
- ▶ Operasi logika menghasilkan nilai : *true* atau *false*
- ▶ Operator logika : AND, OR dan XOR

# BILANGAN LOJIK - next

a	b	not a	a and b	a or b	a xor b
True	True	False	True	True	False
True	False	False	False	True	True
False	True	True	False	True	True
False	false	True	false	False	False

# BILANGAN LOGIK - next

- ▶ Contoh operasi logika :
- ▶ Misalkan X, Y, dan Z adalah peubah (*variabel*) bertipe *boolean*.
- ▶ Dimana :
  - ▶ X bernilai *true*, Y bernilai *false*, dan Z bernilai *true*
- ▶ Maka :

operasi logika	hasil
<hr/>	
( x and y) or z	true
A and ( y or z)	true
Not (x and z)	false
(y xor z) and y	false

# BILANGAN BULAT

- ▶ Bilangan yang tidak mengandung pecahan desimal, misal : 34, 8, 0, -17, 45678901, dsb
- ▶ NAMA TIPE :
  - ▶ Integer
- ▶ RANAH NILAI :

tipe	Rentang nilai	Format
Byte	0 .. 255	8 bit
Shortint	-128 .. 127	8 bit
Word	0 .. 65535	16 bit
Integer	-32768 .. 32767	16 bit
Longint	-2147483648 .. 2147483647	32 bit



# BILANGAN BULAT - next

- ▶ Contoh ;
- ▶ deklarasi :
  - ▶ X : byte
  - ▶ Y : integer
- ▶ Maka :
  - ▶ Peubah X tidak dapat dioperasikan untuk nilai-nilai di atas 255
  - ▶ Peubah Y tidak dapat dioperasikan untuk nilai-nilai di atas 32767

# Contoh program

Program salahtipe;

Uses

    wincrt;

Var

    a,b,x : shortInt;

Begin

    a:=10;

    b:=20;

    X:=a\*b; {x akan bernilai -56, bukan 200}

    Writeln('Nilai x=', x);

End.

# BILANGAN BULAT - next

## ▶ KONSTANTA :

- ▶ Harus ditulis tanpa mengandung titik desimal :
- ▶ Contoh : 78, -14, 7654, 0, 5, 9999, dsb

## ▶ OPERASI :

### 1. Operasi aritmetika :

<b>+</b>	<b>(tambah)</b>
<b>-</b>	<b>(kurang)</b>
<b>*</b>	<b>(kali)</b>
<b>Div</b>	<b>(hasil bagi bilangan bulat)</b>
<b>Mod</b>	<b>(siswa hasil bagi)</b>

Contoh :

$3 + 10 \rightarrow \text{hasil : } 13$

$10 \text{ DIV } 3 \rightarrow \text{hasil : } 3$

$10 \text{ MOD } 3 \rightarrow \text{hasil : } 1$

# BILANGAN BULAT - next

## 2. Operasi Perbandingan :

- Menghasilkan nilai *boolean* (*true* dan *false*)

$<$	Lebih kecil
$\leq$	Lebih kecil atau sama dengan
$>$	Lebih besar
$\geq$	Lebih besar atau sama dengan
$=$	Sama dengan
$\neq$	Tidak sama dengan

# Contoh : operasi perbandingan

$3 < 8$	True
$74 > 101$	False
$17 = 17$	True
$(24 \text{ div } 3) \neq 8$	false

# BILANGAN RIIL

- ▶ Bilangan yang mengandung pecahan desimal : 3.65, 0.003, 29.0, .24, dll
- ▶ NAMA TIPE :
  - ▶ Real
- ▶ RANAH NILAI :

type	Rentang nilai	Format
Real	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	6 byte
Single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	4 byte
Double	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$	8 byte
extended	$3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$	10 byte

# BILANGAN RIIL - next

## ► **KONSTANTA :**

harus ditulis dengan tanda titik desimal  
contoh : 0.78, -14.2376, 0.0, .5, 99.0

## ► **OPERASI :**

### **1. operasi Aritmetika**

+ (tambah, -(kurang), \*(kali), / (bagi)

Contoh:

$$6.4 + 5.7 \rightarrow \text{hasil : } 12.1$$

$$8.0 - 2.8 \rightarrow \text{hasil: } 5.2$$

$$10/2.5 \rightarrow \text{hasil: } 4.0 \text{ (operasi bilangan campuran)}$$

$$7.2 * 0.5 \rightarrow \text{hasil : } 3.6$$

# BILANGAN RIIL - next

## 2. Operasi Perbandingan :

menghasilkan nilai *boolean* (*true* dan *false*)

<	Lebih kecil
≤	Lebih kecil atau sama dengan
>	Lebih besar
≥	Lebih besar atau sama dengan
≠	Tidak sama dengan

Tipe bilangan riil tidak mengenal operator kesamaan (=), karena bilangan riil tidak bisa disajikan secara tepat oleh komputer,

Misal :  $1/3$  tidak sama dengan  $0.33333$ , sebab  $1/3 = 0.33333\dots$  (dg angka 3 yg tidak pernah berhenti).



# KARAKTER

- ▶ Semua huruf-huruf abjad, semua tanda baca, karakter khusus, karakter kosong (*null*) ''.

- ▶ **NAMA TIPE :**

- ▶ Char

- ▶ **RANAH NILAI :**

- ▶ Adalah semua huruf di dalam alfabet ('a'..'z', 'A'..'Z', angka desimal ('0'..'9'), tanda baca('.',':','!',dll), operator aritmetika('+','- ',dll), karakter khusus('\$','#','@',dll)

- ▶ **KONSTANTA :**

- ▶ Karakter harus diapit oleh tanda petik tunggal.
    - ▶ Contoh : 'h', 'y', '.', ' ', '9', '\$'

# KARAKTER - next

## ▶ OPERASI :

### ▶ Hanya Operasi Perbandingan :

<	Lebih kecil
≤	Lebih kecil atau sama dengan
>	Lebih besar
≥	Lebih besar atau sama dengan
=	Sama dengan
≠	Tidak sama dengan

Contoh :

'a' = 'a' → hasil: true

'T' = 't' → hasil: false

'y' ≠ 'y' → hasil: false

'm' < 'z' → hasil: true

'q' > 'z' → hasil : false

# STRING

- ▶ Adalah untaian karakter dengan panjang tertentu.
- ▶ **NAMA TIPE :**
  - ▶ String
- ▶ **RANAH NILAI :**
  - ▶ Deretan karakter yg telah didefinisikan pada ranah karakter.
- ▶ **KONSTANTA :**
  - ▶ Semua konstanta string harus diapit oleh tanda petik tunggal.  
Contoh: 'BANDUNG', 'ganesha', 'Jl. Pahlawan no. 76',  
'.....', 'k768532', dll.

# STRING - next

## ► OPERASI :

### 1. Operasi Penyambungan (*Concatenation*):

- Operator : + (penyambungan, bukan tambah)

Contoh :

'Teknik' + 'Informatika' → hasil : 'Teknik Informatika'

'aaa' + ' bbb' → hasil: 'aaa bbb'

'1' + '2' → hasil: '12'

### 2. Operasi Perbandingan :

# STRING - next

## 2. Operasi Perbandingan

menghasilkan nilai *boolean* (*true* dan *false*)

<	Lebih kecil
≤	Lebih kecil atau sama dengan
>	Lebih besar
≥	Lebih besar atau sama dengan
=	Sama dengan
≠	Tidak sama dengan

Contoh:

'abcd' = 'abc' → hasil: *false*

'aku' < 'AKU' → hasil: *true*

Syarat penamaan (variabel, konstanta, nama type  
bentukan, nama field, nama fungsi, nama prosedur)

- Harus unik (tidak boleh dua buah nama yang sama)
- Harus dimulai dengan huruf alfabet (tidak boleh dimulai dg angka, spasi, atau karakter khusus lainnya)
- Huruf besar dan huruf kecil tidak dibedakan
- Karakter penyusun nama hanya boleh huruf alfabet, angka, dan '\_' (*underscore*)
- Tidak boleh mengandung operator aritmetika, relasional, tanda banca, dan karakter khusus lainnya
- Karakter tidak boleh dipisah dengan spasi
- Panjang nama tidak dibatasi

# Contoh penamaan NAMA

Contoh yg salah	Contoh yg benar
6titik	Titik6 atau titik_6
Nilai ujian	NilaiUjian atau Nilai_ujian
PT-1	PT_1 atau PT1
Hari!	Hari
A 1	A1

# Yg diberi nama dalam Algoritma :

## 1. Peubah (*variable*) :

Tempat menyimpan nilai yg isinya dapat diubah

Contoh: X, nilai\_ujian, jumlah : real

k : integer

## 2. Konstanta (*constant*)

Tempat penyimpanan di dalam memori yg nilainya tetap selama pelaksanaan program dan. Notasi yg digunakan adalah const

Contoh ;

const

phi = 3.14, Nmaks = 100, sandi = 'xyz', dll



# Yg diberi nama dalam Algoritma -next:

## 3. Tipe Bentuk :

Nama tipe bentuk diberikan oleh pemrogram  
contoh:

Type

```
titik : record <x:real, y : real>
```

Type

```
Jam : record
```

```
  < hh : integer      {0..23}
```

```
    mm : integer      {0..59}
```

```
    ss : integer      {0..59} >
```

```
P : titik
```

```
J1, J2 : Jam
```

Titik dan Jam adalah nama tipe, sedangkan P adalah variabel yg bertipe Titik, J1 dan J2 adalah variabel yg bertipe Jam

# Yg diberi nama dalam Algoritma -next:

## 4. Nama Fungsi :

Function Maksimum (input A,B : integer) →  
integer

*{ mengembalikan nilai terbesar antara A dan B }*

Maksimum adalah nama fungsi

## 5. Nama prosedur :

Procedure Tukar(input/output A, B : integer)  
*{ mempertukarkan nilai A dan B }*

Tukar adalah nama prosedur

# NILAI dalam Algoritma

- ▶ Nilai adalah besaran dari tipe data yang terdefinisi (tipe dasar atau tipe bentukan)
- ▶ Dapat berupa data yg disimpan di dalam peubah atau konstanta, nilai dari hasil hitung, atau nilai yg dikirim oleh fungsi
- ▶ Pada ALgoritma, memanipulasi nilai di dalam peubah/variabel (yg bertipe sama)

# Pengisian Nilai ke dalam Peubah

## 1. *(Variable)*: **Pengisian secara langsung (assignment):**

*Mengisi sebuah nilai ke dalam peubah secara langsung (nilai yg diisikan harus bertipe sama dg tipe peubah)*

contoh :

$M \leftarrow 16$

$P \leftarrow M * 2$

$M \leftarrow P + 100$

$P \leftarrow P + M$

## 2. **Pembacaan :**

*Nilai peubah dapat diisi secara eksternal dari piranti masukan, misalnya dari keyboard, dari file, mouse, dsb.*

*Akan membaca sesuai apa yg kita masukkan (inputkan).*

Contoh :

**Read (M)**

*: komputer membaca nilai M yg diinputkan dari luar (mis: keyboard), jika nilai yg diketik adalah 78, maka lokasi memori yg bernama M sekarang berisi 78*

# EKSPRESI

- ▶ Transformasi nilai menjadi keluaran dilakukan melalui suatu perhitungan (komputasi)
- ▶ Cara perhitungan itu dinyatakan dalam suatu ekspresi
- ▶ Ekspresi terdiri atas :*OPERAND* dan *OPERATOR*

# Tiga macam ekspresi :

1. Ekspresi Aritmetik (*operator aritmetik*):

Contoh :  **$a*b$ ,  $a*b/c$ ,  $a \text{ div } b * c$ ,  $a+b*c$**

**:  $d \leftarrow a*b$  (tipe data  $d$  harus sama dengan hasil hitungan ekspresi  $a*b$ )**

2. Ekspresi Relasional (*operator relasional*):

Contoh :  **$\text{not ada}$ ,  $\text{ada or ketemu}$ ,  $x < 5$ ,  
 $\text{ada or } (x=y)$ , dll.**

3. Ekspresi String (*operator penyambungan (+)*)

Contoh :

**'Jl. Ganesa' + 'No.12'**

# Soal

Sebuah Mobil melaju dengan kecepatan tetap  $V$  km/jam jika mobil berjalan dalam  $t$  jam tulislah algoritma untuk menghitung jarak yang sudah ditempuh algoritma tersebut membaca masukan  $v$  dan  $t$  dengan rumus  $s=v*t$  lalu mencetak jarak tersebut