

# Variabel & Tipe Data



# Perhatikan!

$$X = 10 + Y$$

variabel

konstanta

- Berapakah nilai X, jika  $Y=3$ ?
- Berapakah nilai X, jika  $Y=5$ ?
- Berapakah nilai Y, jika  $X=15$  ?



# Variabel

- Tempat untuk menyimpan data
- Sifatnya hanya **sementara**
- Nilai di dalamnya **dapat berubah**



# Konstanta

- Tempat untuk menyimpan data
- Sifatnya hanya **permanen**
- Nilai didalamnya **tetap**

# Syarat penamaan variabel

Untuk C++ dan Java,  
nama variabel  
bersifat **case  
sensitive**.

Harus **diawali**  
dengan huruf

Boleh diikuti dengan  
huruf/angka/symbol

Tidak boleh ada  
spasi

Bukan merupakan  
**keyword** dalam  
sebuah bahasa  
pemrograman

misalkan read, write (pascal), private, protected (java), cout, cin (c++)



# Jawablah!

Manakah yang tepat untuk nama variabel?

- a) Bilangan
- b) 2013nama\_mahasiswa
- c) @email
- d) Alamat rumah
- e) Main
- f) Kode\_arsip



# Tipe data

- Tipe data mengacu pada **jenis data** yang disimpan dalam variabel atau konstanta
- Setiap tipe data membutuhkan **jumlah memori yang berbeda** untuk menyimpannya
- Untuk menentukan jumlah memori yang akan diproses, program harus tahu jenis data apa yang dibutuhkan



# Jenis/Tipe Data (Data Type)

- Terdiri dari
  - Set nilai data
  - Set operasi yang bisa diterapkan pada nilai tersebut





# Klasifikasi Jenis Data

- Simple Data Type (Jenis Data Sederhana) / Tipe dasar
  - Item data individual
- Data Structures / data aggregates (struktur data) / Tipe bentukan
  - Kombinasi dari item data individual
  - Membentuk item data lain



# Jenis Data Sederhana

- Numerik, terdiri dari :
  - Numerik integer (bilangan bulat)
  - Numerik real (bilangan riil)
- Karakter, terdiri dari :
  - Alfabet : a .. z, A .. Z
  - Angka : 0 .. 9
  - Simbol khusus : + ? ' ! [ ] { } ... dll
- Boolean (logika), terdiri dari :
  - True
  - False



# Deklarasi Tipe Data

- Jenis data konstanta dan variabel harus didefinisikan dalam program sehingga :
  - Operasi yang tepat dapat dijalankan pada nilai data dan
  - Jumlah ruang penyimpanan yang tepat bisa ditentukan
- Statement untuk mendefinisikan jenis data disebut **declarative statement**
- Beberapa bahasa pemrograman memiliki sintaks pendeklarasian yang berbeda
- Beberapa contoh program (pendeklarasian data) yang akan diberikan ditulis dalam *pseudo-code*



# Contoh

## Constants

```
pi = 3.141592654
```

## Variables

```
i, qty                : integer  
harga_satuan, harga_beli : real  
status                : boolean  
nama                  : character(25)
```



# Struktur Data

- Kelompok item data yang terorganisasi yang dianggap sebagai suatu unit
- Disebut juga sebagai jenis data kompleks (*complex data type*) atau *data aggregates*
- Beberapa struktur data :
  - Array (larik)
  - String
  - Record
  - List (daftar)
  - Tree

# Array (Larik)

- Set item data yang disusun secara baik menjadi sebuah rangkaian
- Contoh : Nilai = (56 42 89 65 48)
- Item data individual dalam array bisa ditunjuk secara terpisah dengan menyatakan posisinya dalam array itu
  - Nilai(0) menunjuk 56
  - Nilai(1) menunjuk 42
- Bilangan yang ditulis dalam tanda kurung menandakan **posisi** item individual dalam array (disebut juga *subscript / indeks*)

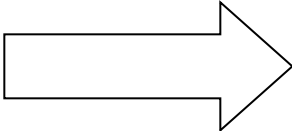
# Array (Larik)

- Variabel bisa digunakan sebagai subscript, misalnya Nilai(**i**).
  - Jika **i** = 2 maka menunjuk ke Nilai(**2**) yaitu 42
  - Jika **i** = 4 maka menunjuk ke Nilai(**4**) yaitu 65
- Item data individual dalam suatu array sering disebut **elemen**
- Matriks
  - Array yang hanya berisi bilangan dan tidak ada data alfabetisnya
- Klasifikasi Array
  - Array 1 dimensi
  - Array multi dimensi

# Array Multi Dimensi

- Mempunyai elemen-elemen yang disusun ke dalam baris dan kolom dan digunakan sebagai tabel data
- Contoh : Nilai ujian dari mahasiswa satu kelas untuk beberapa mata kuliah bisa ditempatkan dalam array 2 dimensi

Siswa ke (no. baris)	B. Inggris (kolom 1)	Matematik a (kolom 2)
1	$A(1,1) = 56$	$A(1,2) = 44$
2	$A(2,1) = 42$	$A(2,2) = 36$
3	$A(3,1) = 89$	$A(3,2) = 73$
4	$A(4,1) = 65$	$A(4,2) = 86$
5	$A(5,1) = 48$	$A(5,2) = 51$


$$A = \begin{bmatrix} 56 & 44 \\ 42 & 36 \\ 89 & 73 \\ 65 & 86 \\ 48 & 51 \end{bmatrix}$$



# Deklarasi Array

## ■ Array 1 dimensi

Variables

```
Nilai: array [1..5] of integer  
A      : array [1..4] of real
```

## ■ Array 2 dimensi

Variables

```
A      : array [1..5, 1..2] of integer
```



# String

- Rangkaian karakter yang ditangani sebagai unit data tunggal
- Contoh (string literal) :
  - “ABC, 32f12. 3h”
  - “Kucing dalam karung”
- Contoh (variabel string) :
  - A = “Sandi”
  - B = “Rumput”
- Berada dalam bentuk array karakter 1 dimensi



# String

- *Fixed-length* string (String yang panjangnya tetap)
  - Mempunyai jumlah tempat karakter yang tetap yang tersedia (bisa digunakan) untuk penyimpanan data
- *Variable-length* string (String yang panjangnya berubah-ubah)
  - Memberi data sejumlah spasi (ruang) sesuai yang ia perlukan

# String

## ■ *Fixed-length* string

posisi karakter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
isi	A	N	D	R	I	A	M	R	I		I	N	A			J	O	K	O		D	E	D	I	
komentar	string ke 1					string ke 2					string ke 3					string ke 4					string ke 5				

## ■ *Variable-length* string

posisi karakter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
isi	A	N	D	R	I	*	A	M	R	I	*	I	N	A	*	J	O	K	O	*	A	L	*		
komentar	string ke 1					string ke 2					string ke 3					string ke 4					string ke 5		tempat sisa		



# Deklarasi String

## ■ *Fixed-length* string

Variables

```
nama : string[5]
```

## ■ *Variable-length* string

Variables

```
nama : string
```

# Operasi pada String

## ■ Concatenation

- Penggabungan dua atau lebih string
- Contoh :

A = "Sandi"

B = "Rumput"

C = A + B

maka

C = "SandiRumput"

# Operasi pada String

## ■ Substring

- Mengambil bagian dari suatu string

- Contoh

A = "Sandi"

B = "Rumput"

C = Left(A, 3)

C = Substr(A, 0, 3)

D = Right(B, 5)

E = Substr(B, 3, 3)

maka

C = "San"

D = "umput"

E = "put"



# Record

- Seperti array 1 dimensi
- Terdiri dari serangkaian item data yang terkait
- Item data berurutan yang ada dalam record bisa mempunyai jenis yang berbeda
- Contoh : Mengorganisasikan 3 item data yang berbeda ke dalam struktur data tunggal
  - NIP : string(8)
  - Nilai : real
  - Lulus : boolean



# Deklarasi Record

```
mahasiswa      : record
                  NIM      : string(8)
                  Nilai    : real
                  Lulus    : boolean
                end record
```

- Setiap elemen memiliki *identifier* sendiri
- Elemen dari suatu record disebut **field**



Penunjukan ke setiap field dari suatu record bisa dilakukan dengan :

- Notasi “**dot**” (titik)

Begin

```
mahasiswa.NIM := '51292215'  
mahasiswa.Nilai := 90.5  
mahasiswa.Lulus := True
```

End

- Notasi “**with**”

Begin

```
with mahasiswa  
do  
    NPM := '51292215'  
    Nilai := 90.5  
    Lulus := True  
end with
```

End

# Array Record (Tabel)

- Kumpulan dua atau lebih record
- Deklarasi Array Record

Variable

Mahasiswa : Array [1..5] of record

NIM : string(8)

Nilai : real

Lulus : boolean

End record