

PERTEMUAN 3

Tipe Data

Hermanto, S.Kom. M.Kom.



Tipe Data

Merupakan klasifikasi variabel untuk menentukan bentuk keadaan data yang akan disimpan dalam memori dan jenis operasi apa yang dapat dilakukan oleh variabel tersebut.

```
3  int jumlah;  
4  jumlah = 10;
```

Dua pernyataan diatas dapat disingkat menjadi :

```
3  int jumlah = 10;
```

Jenis Tipe Data

- Sebelum mendeklarasikan sebuah variabel atau konstanta harus diketahui terlebih dahulu tipe data yang tepat
- Ketepatan pemilihan tipe data pada variabel atau konstanta akan sangat menentukan pemakaian sumber daya komputer
- Salah satu tugas penting seorang **programmer** adalah **memilih tipe data** yang sesuai untuk menghasilkan program yang efisien dan berkinerja tinggi

Jenis Tipe Data

- Banyaknya tipe data tergantung pada bahasa pemrograman
- Secara umum dapat dikelompokkan menjadi tiga :
 1. **Tipe Data Dasar (Primitive Data Type)**

Tipe data yang sudah ada dan dijadikan standar dalam bahasa pemrograman tertentu. Sebagai Contoh : tipe data integer dan char
 2. **Tipe Data Bentukan (Composit Data Type)**

Tipe data yang dibentuk dari tipe data dasar dengan maksud mempermudah pekerjaan programmer. Contoh : tipe data string

Jenis Tipe Data

3. Tipe Data Abstrak (Abstract Data Type)

Model matematika dari obyek data yang menyempurnakan tipe data dengan cara mengaitkannya dengan fungsi-fungsi yang beroperasi pada data yang bersangkutan.

Tipe data abstrak adalah tipe data yang didefinisikan sendiri oleh pemrogram untuk suatu keperluan tertentu yang tidak memungkinkan untuk mendeklarasikan dari tipe data yang sudah ada.

Contoh tipe data abstrak adalah stack, queue, list, tree, graph.

Tipe Data Dasar (Primitive Data Type)

- ▶ Ada 3 jenis tipe data dasar dalam bahasa pemrograman:
 - Tipe Data Numerik (angka) – untuk angka dan berhubungan dengan aritmetika
 - Tipe Data Text (karakter) – untuk karakter dan angka bukan
 - ▶ untuk operasi aritmetika
 - Tipe Data Boolean (logika) – untuk logika benar (*true*) atau salah (*false*)

Tipe Data Numerik (angka)

1. Bilangan Bulat

- Memungkinkan kita untuk menyimpan data dalam bentuk bilangan bulat
- Bilangan bulat adalah bilangan yang tidak mengandung pecahan decimal.
- Contoh : -1, -2, -3, 0, 1, 2, 3, 4

int	2 bytes	-32,768 to 32,767
unsigned int	2 bytes	0 to 65,535
short	2 bytes	-32,768 to 32,767
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

2. Floating Point [float] atau Double Precision [double]

- Memungkinkan untuk menyimpan nilai dalam memori bersifat bilangan pecahan atau real, maupun eksponensial.

“18” dianggap bilangan bulat [int]

“18.0” bilangan real [floating point]

$2.60200\text{E-}6 = 2.60200 \times 10^{-6}$


float	4 bytes	1.2E-38 to 3.4E+38 (6 decimal places)
double	8 bytes	2.3E-308 to 1.7E+308 (15 decimal places)
Long double	10 bytes	3.4E-4932 to 1.1E+4932 (19 decimal places)

Tipe Data Text (Karakter)

- Memungkinkan kita untuk memesan memori berformat text (huruf, angka, dan simbol) dengan karakter tunggal
- Char (Karakter) adalah semua huruf yang ada di dalam alphabet ('a' 'z', 'A' 'Z'), tanda baca ('.', ':', '?', '!' dan lain-lain) maupun karakter special ('\$' , '#' , '@' , '^' , '~' dan lain-lain). Karakter ditulis diantara dua tanda petik tunggal. Contoh : 'A'.

char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127

Tipe Data Boolean (Logika)

- Termasuk tipe bilangan logika atau boolean (diambil dari nama seorang ahli matematika Inggris, George Boole).
 - Boolean hanya memiliki dua pilihan yaitu Benar (*true*) atau Salah (*false*). Karena nilai tipe data Boolean ini hanya beranggotakan dua buah nilai, maka konstanta (constant) yang terdapat pada variable ini adalah True (1) dan False (0).
 - Digunakan untuk memberikan kondisi pada program atau bisa juga memastikan kebenaran dari sebuah operasi
- 

Tipe Data Boolean (Logika)

- Misalkan a dan b adalah dua variable yang bertipe *Boolean*, jika dioperasikan oleh keempat operator logika akan dinyatakan oleh table kebenaran (*truth table*) sebagai berikut :

A	not a
true (1)	false (0)
false (0)	true (1)

A	b	a and b	a or b	a xor b
true (1)	true (1)	true (1)	true (1)	false (0)
true (1)	false (0)	false (0)	true (1)	true (1)
false (0)	true (1)	false (0)	true (1)	true (1)
false (0)	false (0)	false (0)	false (0)	false (0)

Tipe Data Bentukan (Composit Data Type)

Tipe data bentukan disusun oleh satu atau lebih tipe data dasar. Ada tiga macam tipe data bentukan :

1. Tipe Dasar yang Diberi Nama Tipe Baru

Kita dapat memberi nama baru untuk tipe dasar tersebut dengan kata kunci **type**. Ruang nilai, cara menulis konstanta, dan operasi-operasi terhadap tipe baru tersebut tidak berubah, tetap sama dengan tipe dasar aslinya.

Contoh :

DEKLARASI:

Type BilanganBulat : integer

P : BilanganBulat

2. Tipe Terstruktur

- Tipe terstruktur adalah tipe yang berupa rekaman (*record*). Rekaman disusun oleh satu atau lebih *field*. Tiap *field* menyimpan data dari tipe dasar tertentu atau dari tipe bentukan lain yang sudah didefenisikan sebelumnya.

Didefenisikan tipe terstruktur yang mewakili tanggal dalam kalender Masehi. Hari dinyatakan sebagai tanggal (dd), bulan (mm), dan tahun (yy), misalnya 10-8-1999. Misalkan tipe bentukan tersebut di beri nama Tanggal:

dd	mm	yy
----	----	----

Cara mendeklarasikan tipe Tanggal:

Contoh:

DEKLARASI:

```
type Tanggal : record <dd : integer,  
                        mm : integer,  
                        yy : integer  
                        >
```

D : Tanggal //Jika D adalah peubah bertipe Tanggal

Maka cara mengacu tiap field di dalam D adalah:

D.dd

D.mm

D.yy

Contoh Konstanta: <12,7,1997>, <31,12,1980>, <29,2,1980>

Operasi: -Operasi aritmetik bilangan bulat terhadap tiap field

-Operasi perbandingan terhadap masing-masing field

3. String

- String adalah untaian karakter dengan panjang tertentu. String sebenarnya bukan tipe dasar murni karena ia disusun dari elemen-elemen bertipe karakter yang berurutan
- Penggunaan sehari-hari string adalah tipe data yang menampung nilai text atau kalimat.
- Misalkan i dan j adalah dua peubah (*variable*) yang bertipe string, maka cara menuliskannya di bagian Deklarasi adalah :

Contoh :

DEKLARASI:

i, j : **string**

3. String

Konstanta :

Semua konstanta string harus diapit oleh tanda petik tunggal. Contoh-contoh konstanta string misalnya :

'BANDUNG'

'ganesha'

'Jl. Pahlawan No.76'

'Jurusan Teknik Informatika'

'.....'

'ABCD765'

'm'

'Semua konstanta string harus diapit oleh tanda petik tunggal'

3. String

<code>\a</code>	Alarm or Beep
<code>\b</code>	Backspace
<code>\f</code>	Form Feed
<code>\n</code>	New Line
<code>\r</code>	Carriage Return
<code>\t</code>	Tab (Horizontal)
<code>\v</code>	Vertical Tab
<code>\\</code>	Backslash
<code>\'</code>	Single Quote
<code>\"</code>	Double Quote
<code>\?</code>	Question Mark
<code>\nnn</code>	octal number
<code>\xhh</code>	hexadecimal number
<code>\0</code>	Null

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 bytes	-32,768 to 32,767
unsigned int	2 bytes	0 to 65,535
short	2 bytes	-32,768 to 32,767
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295
float	4 bytes	1.2E-38 to 3.4E+38 (6 decimal places)
double	8 bytes	2.3E-308 to 1.7E+308 (15 decimal places)
Long double	10 bytes	3.4E-4932 to 1.1E+4932 (19 decimal places)

Variabel

Variable (peubah) adalah objek yang nilainya dapat di ubah-ubah oleh intruksi di dalam algoritma. Nama peubah harus didefenisikan tipenya di dalam bagian DEKLARASI.

Untuk peubah bertipe bentukan, tipe bentukannya harus didefenisikan sebelum pendeklarasian peubah. Contohnya seperti di bawah ini :

DEKLARASI:

X, nilai_ujian, jumlah : **real**

k : **integer**

c : **char**

DEKLARASI:

type Titik : **record** <x, y : **real**>

type Jam : **record** <hh : **integer**,

mm: **integer**,

ss : **integer**

>

P : Titik

J1, J2 : Jam

Konstanta

Konstanta adalah objek yang nilainya tetap selama pelaksanaan program dan tidak boleh diubah. Notasi untuk menyatakan konstanta adalah ***const***.

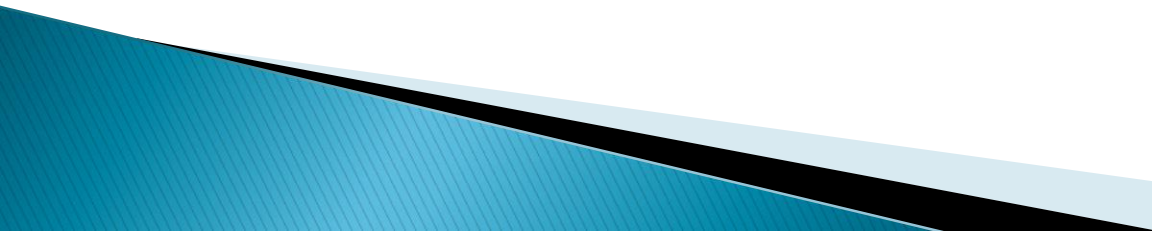
DEKLARASI:

const phi = 3.14


const Nmaks = 100

const sandi = 'xyz123'

Nilai konstanta yang sudah diisi ke dalam nama konstanta tidak dapat di ubah oleh instruksi di dalam algoritma.



Ekspresi dan Operator

- Transformasi nilai menjadi keluaran dilakukan melalui suatu perhitungan (komputasi). Cara perhitungan itu dinyatakan dalam suatu ekspresi. Ekspresi terdiri atas operand dan operator.
 - Operand adalah nilai yang dioperasikan dengan operator tertentu. Operand dapat berupa konstanta, nama variabel atau hasil dari suatu fungsi.
 - Dikenal tiga macam ekspresi : ekspresi numerik, ekspresi relasional, dan ekspresi string.
- 

Ekspresi Numerik

- Ekspresi numerik adalah ekspresi yang baik operand nya bertipe numerik dan hasilnya juga bertipe numerik.
- Contoh ekspresi numerik misalnya :

$a * c$

Pada ekspresi ini, operandnya adalah a dan b, sedangkan operatornya adalah “*”.

Contoh ekspresi numerik:

$a/2 + c*b$

$i+(j*k)-10*(d+k)$

$100.0/(a+c)-2.5$

$(i+j) \text{ div } 2$

$P^3 + 2*a-b \rightarrow ^ : \text{operator perpangkatan}$

Tingkatan operator (dari tertinggi ke terendah) ;

1. $^$
2. div, mod
3. /, *
4. +, -

Ekspresi Relasional

- Ekspresi relasional adalah ekspresi dengan operator
- $<$, \leq , $>$, \geq , $=$, \neq , not, and, or, dan xor.
- Hasil evaluasi ekspresinya adalah nilai bertipe boolean (**true** atau **false**), sehingga ekspresi relasional kadang-kadang di sebut juga ekspresi Boolean. Misalkan dideklarasikan nama-nama peubah sebagai berikut ini :

DEKLARASI:

ada, ketemu, besar : **boolean**

x, y : **integer**

- Misalkan ketemu berisi nilai **false**, ada bernilai **true**, x bernilai 8 dan y bernilai 12, maka. Contoh-contoh ekspresi relasional adalah :

not ada {hasil : false}

ada or ketemu {hasil : true}

ada and true {hasil : true}

x<5 {hasil : false}

ada or (x=y) {hasil : true}

Ekspresi String

- Ekspresi string adalah ekspresi dengan operator “+” (operator penyambungan)

Contoh :

‘Jl. Ganesha’ + ‘No 10’ {hasil : Jl. Ganesha No 10}

‘AA’ + ‘123’ {hasil : AA 123}

‘Neneng Rachmalia’ + ‘Feta’ {hasil : Neneng Rachmalia Feta}



Terima
kasih

Latihan Soal

1. Defenisikan sebuah tipe terstruktur untuk menyatakan data nasabah di sebuah bank. Data nasabah terdiri atas field : nomor account, nama nasabah, alamat nasabah, kota nasabah, nomor telepon nasabah. Untuk setiap field, defenisikan tipe data yang cocok.
2. Defenisikan sebuah tipe terstruktur untuk menyatakan data penerbangan di sebuah bandara. Data penerbangan terdiri atas : nomor penerbangan (misal GA101), bandara (kota) asal, bandara tujuan, tanggal keberangkatan, jam keberangkatan, jam datang. Untuk setiap field, defenisikan tipe data yang cocok.
3. Apa perbedaan $n \leftarrow n + 2$ dari sudut pandang algoritma dan $n = n + 2$ dari sudut pandang matematika?
4. Berapa nilai A dan B setelah intruksi berikut ini?

DEKLARASI:

A, B : integer

ALGORITMA:

$A \leftarrow 10$

$B \leftarrow 2 * A - 5$

$A \leftarrow B$

$A \leftarrow A + 2$

$B \leftarrow A + B$