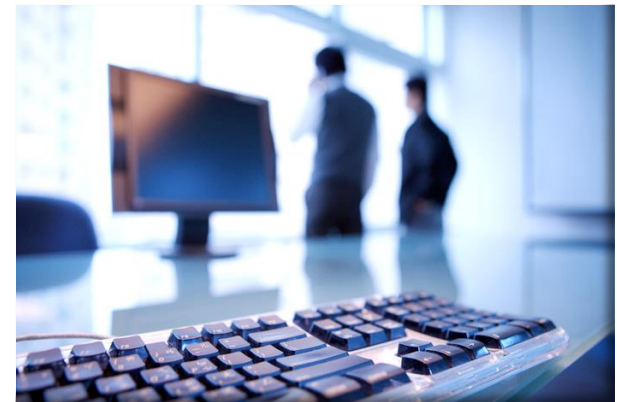


# Nama, Tipe, Ekspresi, dan Nilai

## ALGORITMA DAN PEMROGRAMAN [IS6110102]

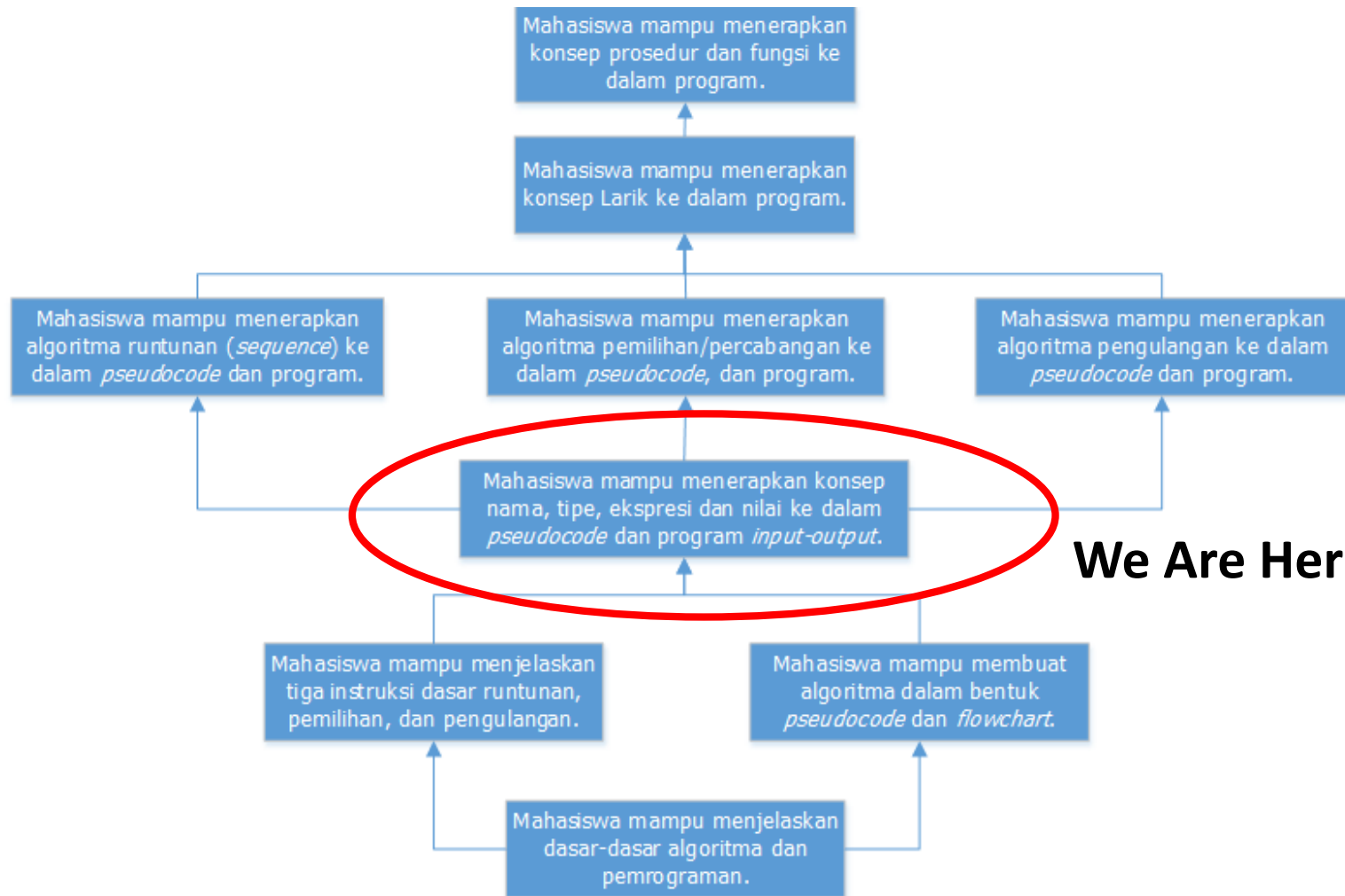
Dosen: Condro Kartiko, S. Kom., M. T. I.



## Sub-Capaian Pembelajaran MK

**“Mahasiswa mampu menerapkan konsep nama, tipe, ekspresi, nilai ke dalam pseudocode dan program input-output.”**

# Sub-Capaian Pembelajaran



**We Are Here !!! 😊**

# Pendahuluan

## Tinjau kembali Algoritma Euclidean

Nama

### Algoritma EUCILEDEAN

*{ Dibaca dua buah bilangan bulat positif m dan n ( $m \geq n$ ). Carilah pembagi bersama terbesar, pbt, dari kedua bilangan tersebut, yaitu bilangan bulat positif terbesar yang habis membagi m dan n }*

### DEKLARASI:

m, n : integer { bilangan bulat yang akan dicari pbt-nya }  
r : integer { sisa hasil bagi }

Tipe

### DESKRIPSI:

read(m,n)  
while n  $\neq$  0 do {  $m \geq n$  }  
    r  $\leftarrow$  m MOD n { hitung sisa hasil pembagian }  
    m  $\leftarrow$  n  
    n  $\leftarrow$  r  
endwhile  
    { kondisi selesai pengulangan n=0, maka pbt (m, n = m ) }  
write(m)

Ekspresi

Nilai

[illegible]

# Nama/Penamaan

- Pada dasarnya program adalah proses memanipulasi objek-objek di dalam memori, maka objek tersebut harus diberi nama.
- Objek diberi nama supaya mudah diidentifikasi, diacu, dan dibedakan dari objek lainnya.

## Aturan Penamaan dalam Pseudocode:

1. Nama harus dimulai dengan huruf alphabet, tidak boleh dimulai dengan angka, spasi atau karakter khusus lainnya.
2. Huruf besar atau huruf kecil tidak dibedakan. Jadi suatu nama yang ditulis dalam huruf besar atau huruf kecil dianggap sama.
3. Karakter penyusun nama hanya boleh huruf alphabet, angka, dan “\_”. Karakter garis bawah/underscore dihitung sebagai sebuah huruf.
4. Nama tidak boleh mengandung operator aritmetika, operator relasional, tanda baca, dan karakter khusus lainnya.
5. Karakter-karakter di dalam nama tidak boleh dipisah dengan spasi.
6. Panjang nama tidak dibatasi.

# Contoh Penamaan dalam Pseudocode

## Contoh nama yang salah:

<i>6titik</i>	{karena dimulai dengan angka}
<i>nilai ujian</i>	{karena dipisahkan dengan spasi}
<i>PT-1</i>	{karena mengandung operator aritmetik kurang}
<i>hari!</i>	{karena mengandung karakter khusus}
<i>A 1</i>	{karena mengandung spasi}

## Contoh nama yang benar:

*Titik6* atau *titik\_6*  
*nilai\_ujian* atau *nilaiujian*  
*PT\_1* atau *PT1*  
*hari*  
*A1* atau *A\_1*





# Tipe Data

- Pada umumnya, program komputer bekerja dengan memanipulasi objek (data) di dalam memori. Objek yang akan diprogram bermacam-macam tipenya, misalnya tipe numerik, karakter, string dsb. Pada algoritma Euclidean, sebagai contoh,  $m$  dan  $n$  adalah objek yang dimanipulasi yang bertipe integer (bilangan bulat).
- Tipe data dapat dikelompokkan menjadi atas dua macam: **tipe data dasar dan tipe data bentukan**. Tipe data dasar adalah tipe data yang dapat langsung dipakai, sedangkan tipe data bentukan dibentuk dari tipe data dasar atau dari tipe data bentukan lain yang sudah didefinisikan.

# Tipe Data Dasar

- Tipe data dasar sudah dikenal dalam kehidupan sehari-hari.
- Dalam dunia pemrograman yang termasuk ke dalam tipe data dasar adalah: **bilangan bulat**, **bilangan riil**, **bilangan logika**, **karakter**, dan **string**.

## A. Bilangan Logika

**Nama Tipe:** Boolean → diambil dari nama seorang matematikawan inggris, George Boole

**Ranah Nilai:** hanya mengenal dua buah nilai yaitu benar (true) dan salah (false). Istilah bilangan dikarenakan kita dapat menyatakan “benar” dengan angka 1 dan “salah” dalam angka 0.

**Konstanta:** true dan false

**Operasi:** not, and, or, dan xor

# Tabel Kebenaran

A	B	A AND B
False	False	False
False	True	False
True	False	False
True	True	True

A	B	A OR B
False	False	False
False	True	True
True	False	True
True	True	True

A	NOT A
False	True
True	False

A	B	A XOR B
False	False	False
False	True	True
True	False	True
True	True	False

# Tipe Data Dasar

## B. Bilangan Bulat

**Nama Tipe:** integer

**Contoh penerapan:** `DEKLARASI:`  
                  `m, n : integer`  
                  `r : integer`

**Ranah Nilai:** secara teoritis, tipe bilangan bulat memiliki ranah nilai yang tidak terbatas dari minus tak hingga sampai plus tak hingga.

**Konstanta:** harus ditulis tanpa mengandung titik desimal, contoh:

78    -14    7649    0    5    99999

**Operasi:** aritmetika dan perbandingan

# Tipe Data Dasar (2)

## Operasi aritmetika

**Operand** → Objek yang dioperasikan

**Operator** → simbol yang digunakan untuk memanipulasi operand  
(+, -, \*, div, mod)

Contoh Operasi aritmetika:

$6+4$

$87-10$

$5*10$

$10 \text{ div } 3$

$12 \text{ mod } 3$

# Tipe Data Dasar (3)

Operasi perbandingan → Menghasilkan nilai *boolean*

**Operator:**

< (lebih kecil)

≤ (lebih kecil sama dengan)

> (lebih besar)

≥ (lebih besar sama dengan)

= (sama dengan)

≠ (tidak sama dengan)

Contoh Operasi perbandingan:

3<8                      (true)

75>101                (false)

17=17                  (true)

(24 div 3) ≠ 8        (false)

# Tipe Data Dasar (4)

## C. Bilangan Riil

**Nama Tipe:** real

**Contoh penerapan:** DEKLARASI

x, nilai\_ujian, jumlah : real

**Ranah Nilai:** secara teoritis, tipe bilangan bulat memiliki ranah nilai yang tidak terbatas dari minus tak hingga sampai plus tak hingga.

**Konstanta:** harus ditulis dengan tanda titik desimal, contoh:

0.78      -14.234      7649+E8      0.001      5.33      99999.0

**Operasi:** aritmetika dan perbandingan

**Contoh :** 6.4+5.7      10.0/2.5      0.0003<-0.3      3.0 ≠ 3.5

# Tipe Data Dasar (5)

## C. Karakter

**Nama Tipe:** char

**Contoh penerapan:** `DEKLARASI:`  
`g,h: char`

**Ranah Nilai:** semua huruf di dalam alphabet ('a'...'z', 'A'...'Z'), angka decimal (0...9), tanda baca (.,!?), operator aritmetika (+,-,\*,/), dan karakter khusus seperti (!#\$%&@)

**Konstanta:** semua karakter harus diapit tanda petik tunggal, contoh:

'h' 'Y' ':' 'p' '+' '9' 'o'

**Operasi:** perbandingan

Contoh:

'a' = 'a' (true)

'T' = 't' (false)

'm' < 'z' (true)



# Tipe Data Dasar (6)

## D. String

**Nama Tipe:** string

**Contoh penerapan:** Kamus

```
nama : string           {variabel nama bertipe string}  
nim: integer           {variabel nim bertipe integer / bilangan bulat}
```

**Ranah Nilai:** deretan karakter yang telah didefinisikan

**Konstanta:** semua konstanta string harus diapit oleh tanda petik tunggal, contoh:

'Bandung'                      'ganesha'                      'Jl. Pahlawan No. 76'

**Operasi:** penyambungan (***concatenation***) dan perbandingan

# Tipe Data Dasar (7)

## Operasi penyambungan (*concatenation*)

Operator → +

Contoh:

'Teknik' + 'INFORMATIKA' = 'TeknikINFORMATIKA'

'Software' + ' ENGINEERING' = 'Software ENGINEERING'

'aaa' + ' bbb' + 'cc' = 'aaa bbbcc'

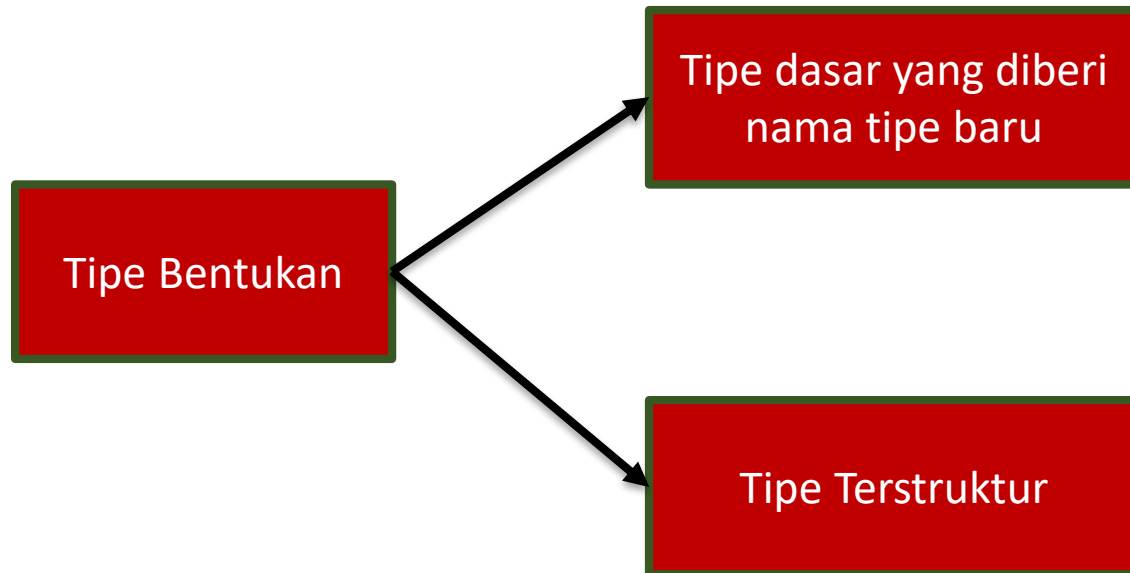
## Operasi perbandingan

Contoh:

'abcd' = 'abc' (false)

# Tipe Data Bentukan

- Tipe bentukan adalah tipe yang didefinisikan sendiri oleh pemrogram (user-defined data type).



# Tipe Dasar yang Diberi Nama Tipe Baru

- Kadang-kadang pemrogram ingin memberi nama baru terhadap tipe dasar yang sudah dikenal. Tujuannya supaya nama baru tersebut lebih mudah diinterpretasi oleh orang yang membaca teks algoritma.

**Nama Tipe:** type

**Contoh penerapan:**

DEKLARASI:

```
type BilanganBulat : integer
```

```
P: BilanganBulat
```

# Tipe Terstruktur

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

<i>Field 1</i>	<i>Field 2</i>	<i>Field 3</i>	<i>...</i>	<i>Field N</i>
----------------	----------------	----------------	------------	----------------

# Contoh Tipe Terstruktur

Titik dalam koordinat kartesian dinyatakan sebagai (x,y), dengan x adalah nilai absis dalam arah sumbu-x dan y adalah nilai ordinat dalam arah sumbu-y



DEKLARASI:

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

**Cara Penggunaan :**

P : Titik

**Cara Mengacu tiap field :**

P.x

P.y

## Contoh Tipe Terstruktur (2)

Didefinisikan tipe terstruktur yang mewakili jam. Jam dinyatakan sebagai jam (hh), menit (mm), dan detik (ss), contohnya 12:45:10 (jam 12 lewat 45 menit lewat 10 detik).



DEKLARASI:

```
type Jam : record < hh : integer, mm: integer, ss: integer>
```

```
type Jam : record < hh, mm, ss: integer>
```

**Cara Penggunaan :**

J : Jam

**Cara Mengacu tiap field :**

J.hh

J.mm

J.ss

# Peubah/Variabel

- Peubah (*variable*) adalah objek yang nilainya dapat diubah-ubah oleh instruksi di dalam algoritma. Nama peubah harus didefinisikan tipenya dalam bagian **DEKLARASI**

## Contoh:

DEKLARASI:

x, nilai\_ujian, jumlah: **real**

k: **integer**

c: **char**

Peubah/Variabel



# Konstanta

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

## Contoh:

DEKLARASI :

```
const phi = 3.14
```

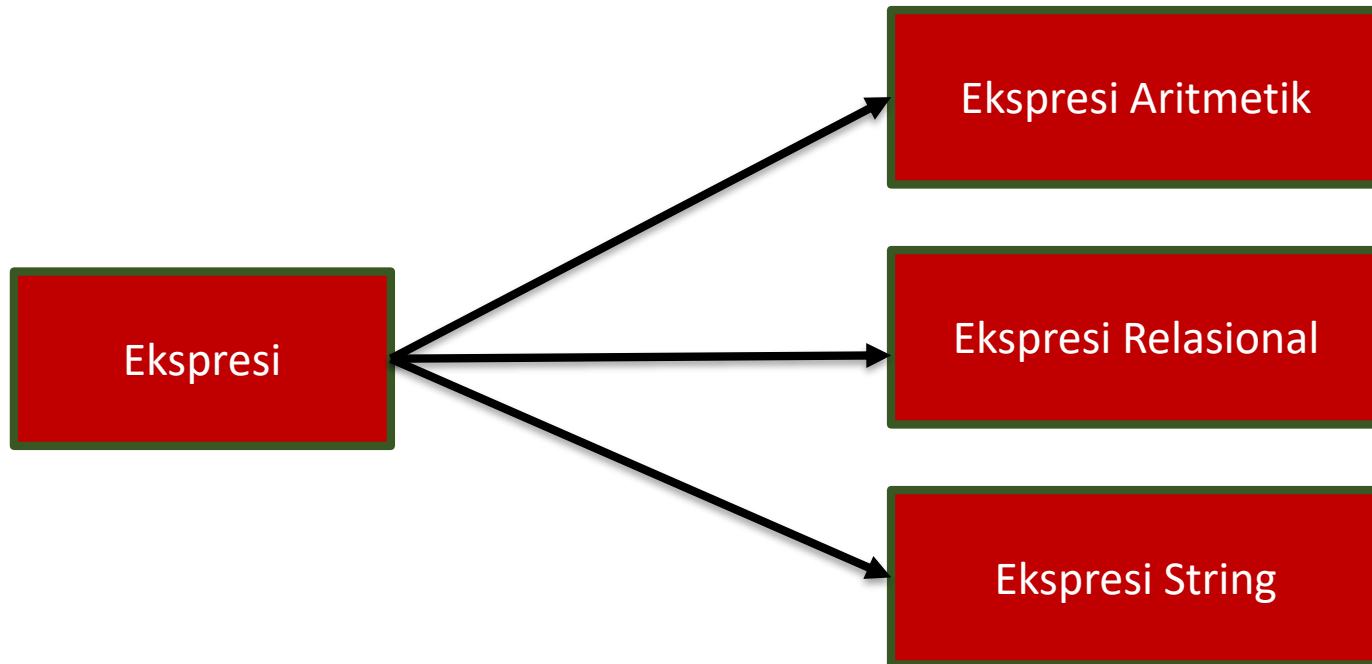
```
const Nmaks = 100
```

```
const sandi = xyz123
```

# Ekspresi

# Ekspresi

- Transformasi nilai menjadi keluaran dilakukan melalui suatu perhitungan (komputasi). Cara perhitungan tersebut dinyatakan dalam **ekspresi**.



# Ekspresi Aritmetik

- Ekspresi aritmetik adalah ekspresi dengan operand-nya bertipe numeric dan hasilnya juga bertipe numeric

**Contoh:**

$$a * c$$

Pada ekspresi tersebut, operand-nya adalah a dan c, sedangkan operatornya adalah “\*”.

# Ekspresi Relasional

- Ekspresi relasional adalah ekspresi dengan operator  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $\neq$ ,  $=$ , `not`, `and`, `or`, dan `xor`.

## Contoh:

DEKLARASI :

`ada, ketemu, besar: boolean`

`x, y: integer`

Misalkan `ketemu` bernilai **false**, `ada` bernilai **true**, `x` bernilai 8 maka:

`not ada`                      (**false**)

Ada **or** `ketemu`            (**true**)

Ada **and** `true`              (**true**)

`x < 5`                        (**false**)

# Ekspresi String

- Ekspresi string adalah ekspresi dengan operator “+”

**Contoh:**

```
`Jl Panjaitan' + `No 128'
```

```
{Hasil: Jl PanjaitanNo 128}
```

NILAI

# Nilai

- Nilai adalah besaran dari tipe data yang terdefinisi (tipe dasar atau tipe bentukan). Nilai dapat berupa data yang disimpan di dalam peubah atau konstanta, nilai dari hasil perhitungan, atau nilai yang dikirim oleh fungsi.

## a. Mengisi Nilai ke dalam Peubah/Variabel

- Nilai yang diisi ke dalam peubah/variable dapat berupa konstanta

peubah  $\leftarrow$  konstanta

**Contoh:**  $A \leftarrow 10$

- Peubah diisi dengan nilai dari peubah lain

peubah1  $\leftarrow$  peubah2

**Contoh:**  $M \leftarrow N$



# Nilai (2)

- Peubah diisi dengan hasil evaluasi dari sebuah ekspresi:

peubah  $\leftarrow$  ekspresi

**Contoh:**  $C \leftarrow (A+B)/2$  {A berisi hasil evaluasi dari  $(A+B)/2$ }

## b. Membaca Nilai dari Piranti Masukan

Selain dengan pengisian secara langsung, nilai peubah juga dapat diisi dari luar piranti masukan (misalkan dari papan ketik atau keyboard).

**read** (nama<sub>1</sub>, nama<sub>2</sub>, ..... , nama<sub>n</sub>)

Contoh: **read** (N)

Instruksi di atas artinya meminta pengguna (user) untuk memasukkan nilai untuk peubah N.

## Nilai (3)

### c. Mencetak Nilai ke Piranti Keluaran

Nilai konstanta, peubah, dan hasil ekspresi dapat ditampilkan ke piranti luaran (monitor), Notasi pencetakan nilai dilakukan dengan notasi **write**.

**write**(nama<sub>1</sub>, nama<sub>2</sub>, .... , nama<sub>n</sub>)

**write**(konstanta)

**write**(ekspresi)

Contoh:

**write**('A')                      {Mencetak karakter A}

**write**(A)                        {mencetak nilai yang disimpan peubah A}

**write**(A + 2)                    {mencetak ekspresi yaitu nilai A ditambah 2}

**write**('Hello', A)            {mencetak 'Hello' dan nilai peubah A}

# TUGAS

## Petunjuk pengerjaan:

Kerjakan beberapa latihan soal di bawah ini, diketik dalam **format PDF** dan dikumpulkan via email ke [condro.kartiko@ittelkom-pwt.ac.id](mailto:condro.kartiko@ittelkom-pwt.ac.id) paling lambat hari Selasa, tanggal 10 Oktober 2017 pukul 23:59 WIB.  
**Format file dan subjek email: [ALPRO]S1IF05A  
Tugas2\_NIM1\_Nama1,NIM2\_Nama2,NIM3\_Nama3**

1. **[poin 10]** Misalkan X, Y, dan Z adalah peubah bertipe boolean, X bernilai true, Y bernilai false, dan Z bernilai true, tentukan hasil dari:

Operasi logika	Hasil
$(X \text{ and } Y) \text{ or } Z$	?
$X \text{ and } (Y \text{ or } Z)$	?
$\text{Not } (X \text{ and } Z)$	?
$(Y \text{ Xor } Z) \text{ and } Y$	?

# Tugas

2. **[poin 20]** Buatlah pseudocode untuk mengkonversi waktu ke detik! Gunakan tipe data bentukan!
3. **[poin 20]** Buatlah pseudocode untuk menampilkan, nama, nim dan nilai mahasiswa sesuai inputan! Gunakan tipe data bentukan!
4. **[poin 20]** Definisikan sebuah tipe terstruktur untuk menyatakan data nasabah di sebuah bank. Data nasabah terdiri dari field: nomor account, nama nasabah, alamat nasabah, kota nasabah, nomor telepon nasabah. Untuk setiap field, definisikan tipe data yang cocok!
5. **[poin 20]** Definisikan 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 (departure time), jam datang (arrival time). Untuk setiap field, definisikan tipe data yang cocok!

# Tugas

6. [poin 10] Tentukan nilai A dan B setelah instruksi 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

# Kesimpulan

- Tipe data dapat dikelompokkan menjadi dua macam yaitu tipe dasar dan tipe bentukan. Tipe dasar adalah tipe yang dapat langsung dipakai, sedangkan tipe bentukan diturunkan dari tipe dasar atau tipe bentukan lain yang sudah didefinisikan sebelumnya.
- Penamaan adalah konsep yang penting dalam pemrograman karena pada dasarnya program adalah proses memanipulasi objek yang ada di dalam memori, maka objek tersebut haruslah diberi nama.
- Tipe data dasar diantaranya adalah bilangan logika, bilangan bulat, bilangan riil, karakter, dan string.
- Ada dua macam tipe bentukan yaitu tipe dasar yang diberi nama dengan nama tipe baru dan tipe terstruktur.
- Peubah adalah objek yang nilainya dapat diubah-ubah oleh instruksi dalam algoritma.
- Konstanta adalah objek yang nilainya tetap selama pelaksanaan program dan nilainya tidak boleh diubah.
- Ekspresi dibedakan menjadi ekspresi aritmetik, relasional, dan string.
- Nilai adalah besaran dari tipe data yang terdefinisi. Pengisian nilai dapat melalui peubah, atau

