

KUG1C3

Dasar Algoritma dan Pemrograman



**Writing the Algorithm :
Variables and Input/Output**

Writing Algorithm

- ▶ Change from **graphic** to **text**
- ▶ More efficient to store and read
- ▶ Algorithm is **not** a programming code



A Good Algorithm Must Be...

1. **Stop** at some point
2. Precisely defined and **unambiguous**
3. It's **Simple** yet **Effective**



Algorithm Text

Consists of 3 parts

1. Title
2. Dictionary
3. Algorithm

JUDUL

```
{ Ini adalah teks dalam bahasa Indonesia untuk memudahkan pembacaan  
teks algoritma }  
{ Spesifikasi teks algoritmik secara umum }
```

KAMUS

```
{Pada bagian ini, dilakukan pendefinisian nama konstanta, nama variabel,  
spesifikasi prosedur, spesifikasi fungsi}
```

ALGORITMA

```
{ Pada bagian ini, semua teks yang tidak dituliskan di antara tanda kurung  
kurawal buka dan kurung kurawal tutup harus dianggap sebagai notasi  
algoritmik }
```

JUDUL

Bagian teks algoritma tempat mendefinisikan apakah teks tsb adalah program, prosedur, fungsi, atau modul

Setelah judul tuliskan spesifikasi singkat dari teks algoritma.

Bagian judul identik dengan judul buku dan intisari pada suatu teks ilmiah dalam suatu makalah berbahasa Indonesia

KAMUS

Definisi

Bagian teks algoritma tempat mendefinisikan:

- Nama type
- Nama konstanta
 - Nama informasi (nama variabel)
- Nama prosedur, sekaligus spesifikasi
- Nama fungsi, sekaligus spesifikasi

Contoh Pendefinisian Kamus

```
KAMUS
{Nama Type, hanya untuk type yang bukan type dasar }
  type Point : <X:real,Y:real> { koordinat pada sumbu kartesian }

{Nama Konstanta, harus menyebutkan type dan nilai }
  constant PI: real = 3.14159
  constant Ka: character = 'K'
  constant Faktor: integer = 3
  constant Benar: boolean = true

{Nama Informasi, menyebutkan type}
  NMax : integer {jumlah maksimum elemen tabel}
  Found : boolean { Hasil pencarian, true jika ketemu }
  P : Point { Posisi pena pada bidang kartesian }
  CC : character { Current character }

{Spesifikasi Fungsi, menyebutkan nama fungsi, domain dan range}
  Function RealToInt (i:real)→ integer
  {Mengkonversi harga i yang bertype real menjadi harga ekivalen
  yang bertype integer }

{Spesifikasi Prosedur, menyebutkan Nama, parameter, Initial State, Final
State dan Proses}
  procedure INISIALISASI
  {I.S. Sembarang
  F.S. Semua nama global dalam kamus terdefinisi nilainya
  Proses : Menginisialisasi semua nama informasi global}
  procedure Tulis (Input Pesan: string)
  {I.S. sembarang
  F.S. Pesan tertulis di layar
  Proses : Menulis isi Pesan ke layar }
  procedure TUKAR(Input/Output : A,B:real)
  {I.S. A dan B terdefinisi, A=a dan B=b
  F.S. A=b dan B=a
  Proses : Menukar isi nama informasi A dan B}
```

ALGORITMA

Definisi

Bagian teks algoritma yang berisi instruksi atau pemanggilan aksi yang telah didefinisikan

Komponen Algoritma

Komponen teks algoritmik dalam pemrograman prosedural dapat berupa

- Instruksi dasar, seperti I/O, assignment
 - *Sequence*
 - Analisa kasus
 - Pengulangan

NAMA

Definisi

Sesuatu yang dipakai sebagai identifikasi:

- Modul program, algoritma, skema program dsb
 - Fungsi
 - Prosedur
 - Type
- Tempat penyimpanan (nama variabel)
 - Konstanta

Aturan pemberian NAMA

- ▶ Dalam bahasa pemrograman, setiap **nama** mempunyai aturan penulisan. Pada teks algoritma tidak ada aturan ketat.
- ▶ Pemilihan **nama** harus **interpretatif**, tidak menimbulkan kerancuan, jika singkat harus disertai penjelasan.

contoh: $x-y$ akan membingungkan sebab mungkin "x minus y"

- ▶ **Nama** merupakan satu kesatuan *leksikal*, maka sebuah nama harus dituliskan secara utuh (tidak dipisah **blank**) supaya satu **nama** dapat dibedakan dengan **nama** lainnya (besaran *leksikal* lain)

Aturan pemberian NAMA

- ▶ Semua nama dalam program harus unik, artinya suatu nama hanya didefinisikan satu kali saja
- ▶ Semua nama yang dipakai dalam teks algoritma harus sudah didefinisikan pada kamus
- ▶ Nama bukan merupakan reserved word/ Keyword
- ▶ Nama tidak diawali angka

Variable and Type of Data


- ▶ Variable is a storage to store a **data**
- ▶ Variable type must **match** the data type that is stored



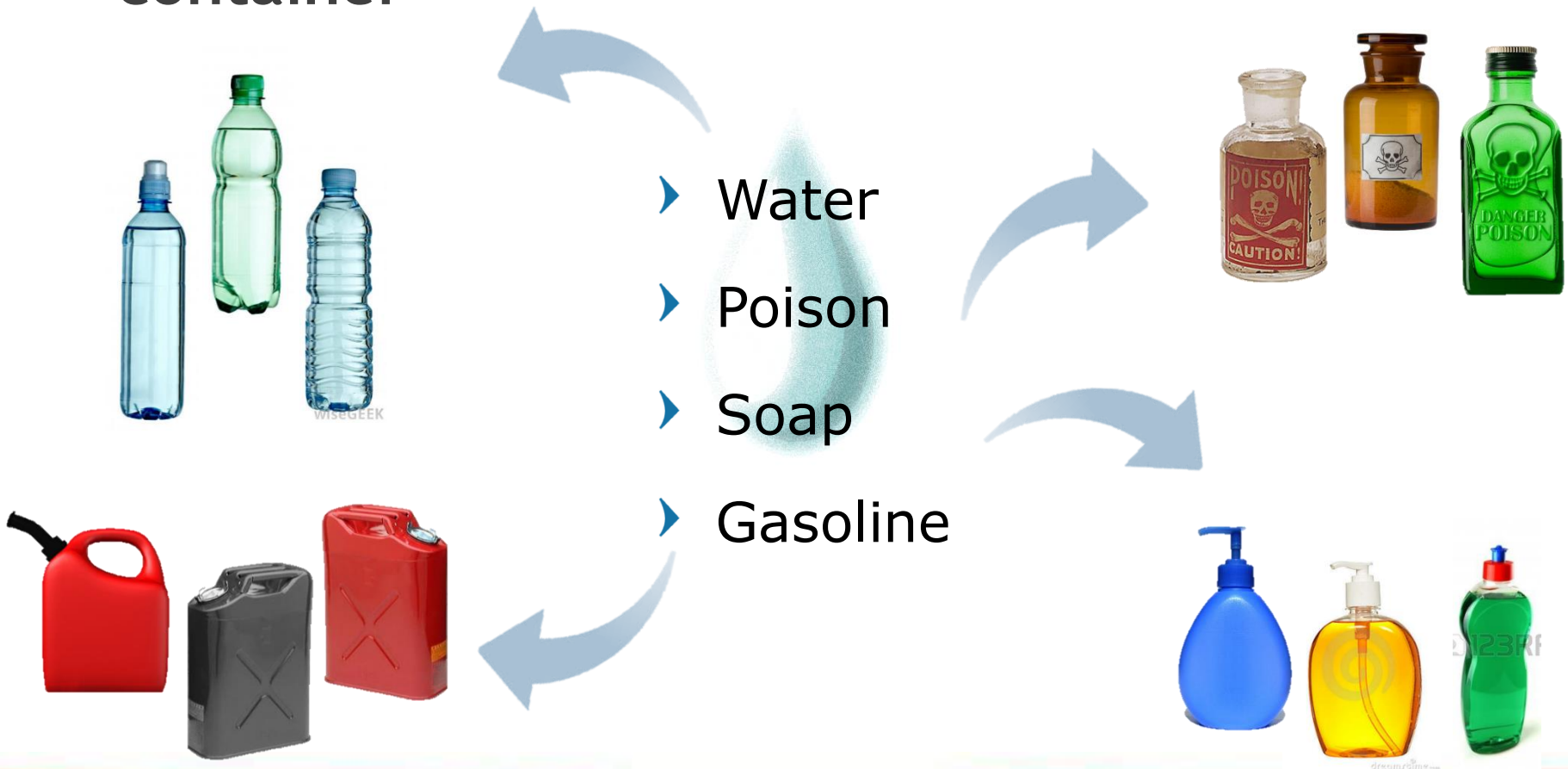
Data Type

- ▶ Pattern representation of data
- ▶ To define the programmed object
- ▶ To Understand the difference between data type, data, and variable, let's see the following illustration..

There are many types of fluid

- 
- ▶ Water
 - ▶ Poison
 - ▶ Soap
 - ▶ Gasoline

Each type of fluid has a suitable type of container



Should not be put where it doesn't belong



- ▶ Water
- ▶ Poison
- ▶ Soap



- ▶ Gasoline

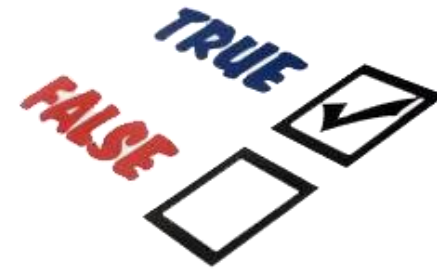


What was shown by the illustration

- ▶ The Fluid = The Data
- ▶ Type of fluid = Data Type
- ▶ The containers = Variable

Basic Type

- ▶ Logic / Boolean
- ▶ Integer
- ▶ Character and String
- ▶ Real



Boolean Operator

Operator	Output
And	boolean
Or	boolean
Xor	boolean
Not	boolean

operation	result	operation	result
T and T	T	T or T	T
T and F	F	T or F	T
F and T	F	F or T	T
F and F	F	F or F	F
T Xor T	F	F Xor T	T
T Xor F	T	F Xor F	F
Not T	F	Not F	T

Integer operator

operator	Result type
+	Integer
-	Integer
*	Integer
/	real
Mod	Integer
Div	Integer
Abs	Integer
^	Integer

operator	Result type
<	Boolean
>	Boolean
≤	Boolean
≥	Boolean
=	Boolean
≠	Boolean

Let's Try it

Example	Output
$3 + 10$	=
$87 - 31$	=
$5 * 10$	=
$10 \text{ div } 3$	=
$10 \text{ mod } 3$	=
$20 \text{ mod } 5$	=

Example	Output
$3 < 10$	=
$87 > 31$	=
$5 \leq 5$	=
$5 < 5$	=
$5 = 5$	=
$20 \neq 5$	=

Real operator

operator	Result type
+	real
-	real
*	real
/	real
^	real

operator	Result type
<	Boolean
>	Boolean
≤	Boolean
≥	Boolean
≠	Boolean

Character operator

operator	Result type
=	Boolean
≠	Boolean

String operator

operator	Result type
=	Boolean
≠	Boolean
& (concatenate)	String
+ (concatenate)	String

Let's Try it

Example	Output
$6.4 + 5.7$	=
$10/3$	=
$10/2.5$	=
$7.2 * 0.5$	=
$0.003 < 0.3$	=
$8.0 \geq 5$	=

Example	Output
$3.0 \neq 3.5$	=
$1/3 = 0.333$	=
<code>'a' = 'a'</code>	=
<code>'T' = 't'</code>	=
<code>'m' < 'z'</code>	=
<code>'Q' > 'z'</code>	=

Writing Algorithm

- ▶ Keyword, data type, must be **underlined**
- ▶ Pay attention to the **indentation**
- ▶ Highly recommended to write using pencil first



Writing Title

Program Calculator_Add
{this program will calculate sum of two variable}

Dictionary

Algorithm

Declaring Variable

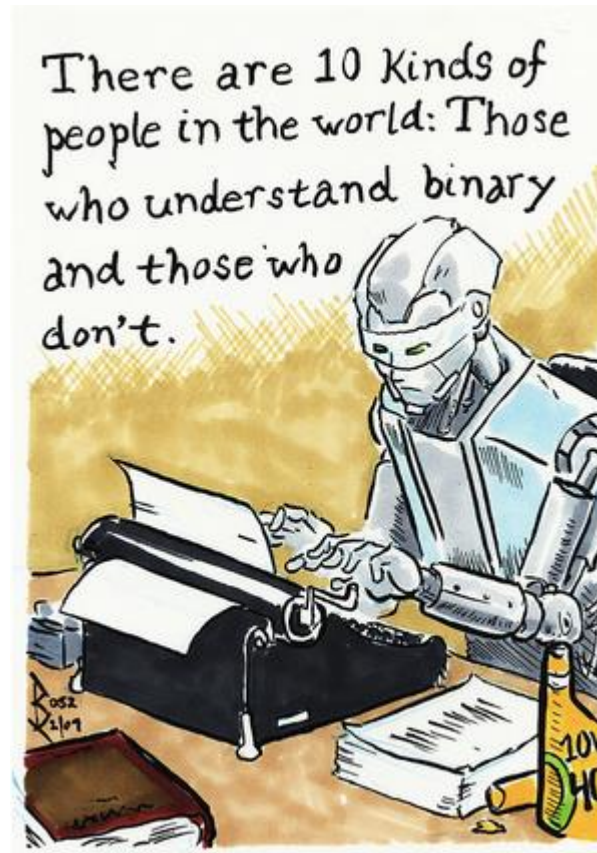
- ▶ Write all the variable that you need in Dictionary
- ▶ Each variable **must have** a name
- ▶ Variable's name must be **unique**
- ▶ Cannot use keyword as variable name
- ▶ Short yet clear naming
- ▶ Starts with alphabet character
- ▶ **No space allowed**, variable must be 1 word only,
(use underscore for spacing)

Declaring Variable

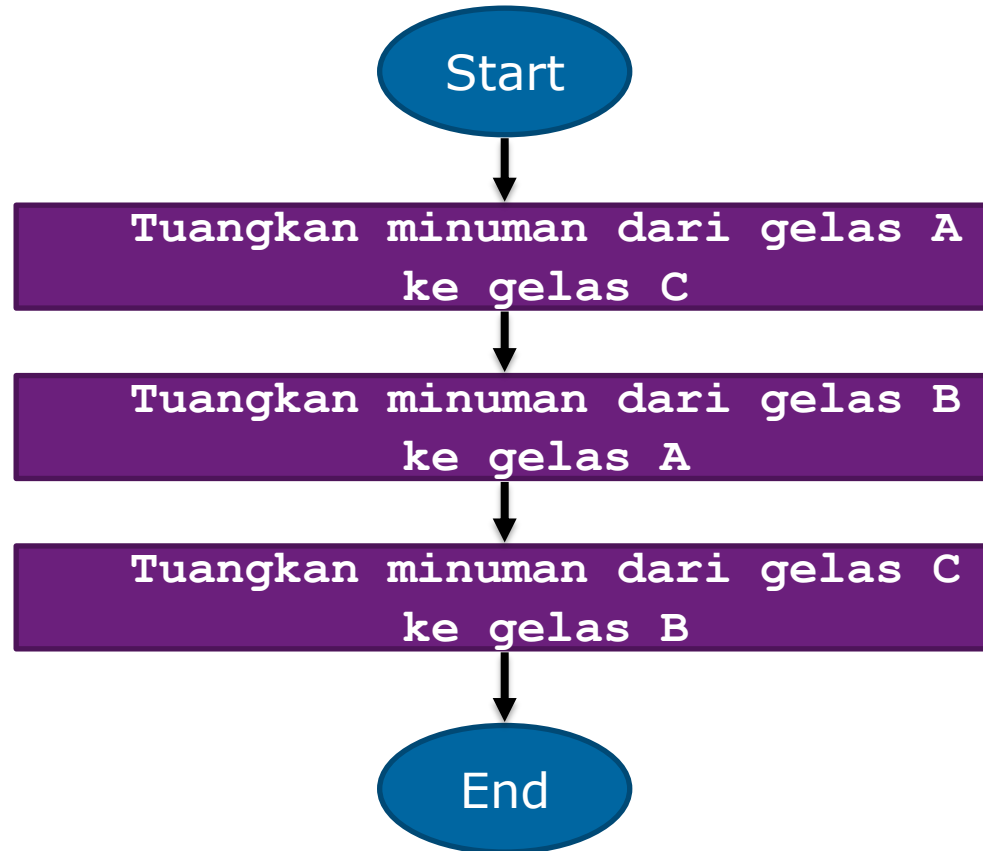
- ▶ To declare a variable:
 - Variable_name : data_type
- ▶ Example

<u>Title</u>
<u>Dictionary</u> Text : <u>string</u> Height : <u>real</u> Number_of_student : <u>integer</u>
<u>Algorithm</u>

Let's start writing algorithm



Remember..



Assignment

- ▶ Assigning a value to a variable
- ▶ Use **left arrow** symbol (\leftarrow)
- ▶ To store a value to a variable, write
– variable_name \leftarrow value
- ▶ To copy the value of a variable and store it to another variable, write:

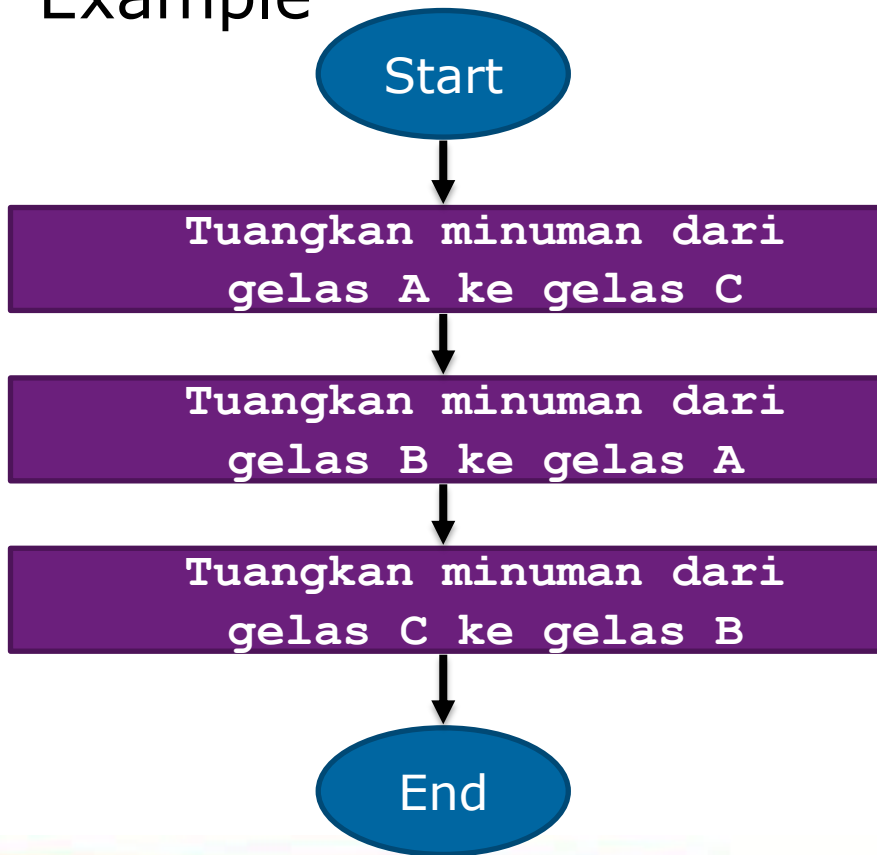
– Var_A \leftarrow Var_B



Var Tujuan Var Asal

Assignment

Example



Program TukarMinuman

Kamus

A, B, C: minuman

Algoritma

$C \leftarrow A$

$A \leftarrow B$

$B \leftarrow C$

Assignment

▶ Example

Title

Dictionary

Number : integer

text: string

Distance : real

Algorithm

Number \leftarrow 10

text \leftarrow 'hello world'

Distance \leftarrow 5.67

Let's Try It

Program var_trial

Dictionary

num_1 : **integer**

num_2 : **integer**

num_3 : **integer**

Algorithm

num_1 \leftarrow 3

num_3 \leftarrow 10

num_2 \leftarrow num_3

num_3 \leftarrow num_1 + num_2

num_1 \leftarrow num_2

num_1 \leftarrow num_1 + 1

num_3 \leftarrow num_2

Line	Num_1	Num_2	Num_3
1	3	0	0
2	3	0	10
3			
4			
5			
6			
7			

Constant

- ▶ Designed so that its initial value can not be changed
- ▶ Use keyword **constant** or **const** before declaring
- ▶ Value of constant is set when declared

Constant

▶ Example

<u>Title</u>
<u>Dictionary</u> <u>Constant</u> Phi : <u>real</u> = 3.14 <u>Constant</u> max_speed : <u>integer</u> = 100
<u>Algorithm</u>

Input from User

- ▶ Sometimes we need to design that a value of a variable is determined by the user who run the program
- ▶ Write a mechanism to receive user input and store it to a variable
 - input(variable_name)



Output result

- ▶ We sometimes also need to display the value of a variable or text as an output or result of our program to the output device such as monitor, printer, etc.
- ▶ Write a mechanism to display the value of a variable
 - output(variable_name)
- ▶ Input/output is a way to communicate with user

Example: HELLOX

Tuliskanlah algoritma untuk membaca sebuah nama, dan menulis "HELLO" yang diikuti dengan nama yang diketikkan.
Contoh :

jika dibaca 'ALI', maka keluaran adalah : 'HELLO ALI'

jika dibaca 'SINTA', maka keluaran adalah : 'HELLO SINTA'

Spesifikasi

Input : nama

Output : 'HELLO nama'

Proses : menulis 'HELLO' diikuti nama yang dibaca

Program HELLOX

Kamus

nama: String

Algoritma

input(nama)

output('Hello ', nama)

Exercise 1 & 2

- ▶ Determine the output of the following algorithm

Exercise 1

Program COBA2

{ di dalam program akan dilakukan pertukaran nilai-nilai. Program akan mengoutputkan nilai dari setiap variabel setelah ditukar)

Kamus

n : integer
m : integer
l : integer

Algoritma :

```
n ← 7  
m ← 12  
l ← m  
m ← n  
n ← l  
l ← n  
m ← l  
output('Nilai l=', l)  
output('Nilai m=', m)  
output('Nilai n=', n)
```

Exercise 2

Program COBA3

{ di dalam program akan dilakukan operasi div dan mod. Program akan mengoutputkan nilai dari setiap variabel setelah operasi dilakukan)

Kamus

p : integer
q : integer
o : integer
n : integer
m : integer
l : integer

Algoritma :

```
n ← 7
m ← 12
l ← n mod m
o ← n div m
p ← m mod n
q ← m div n
output ( 'Nilai l=', l)
output ( 'Nilai o=', o)
output ( 'Nilai p=', p)
output ( 'Nilai q=', q)
```

Exercise 3

- ▶ Buat flowchart dan algoritma untuk menjumlahkan dua buah bilangan!
 - User menginputkan bilangan 1 dan bilangan 2
 - Outputkan hasil penjumlahan bilangan 1 dan bilangan 2

Question?



Home Task

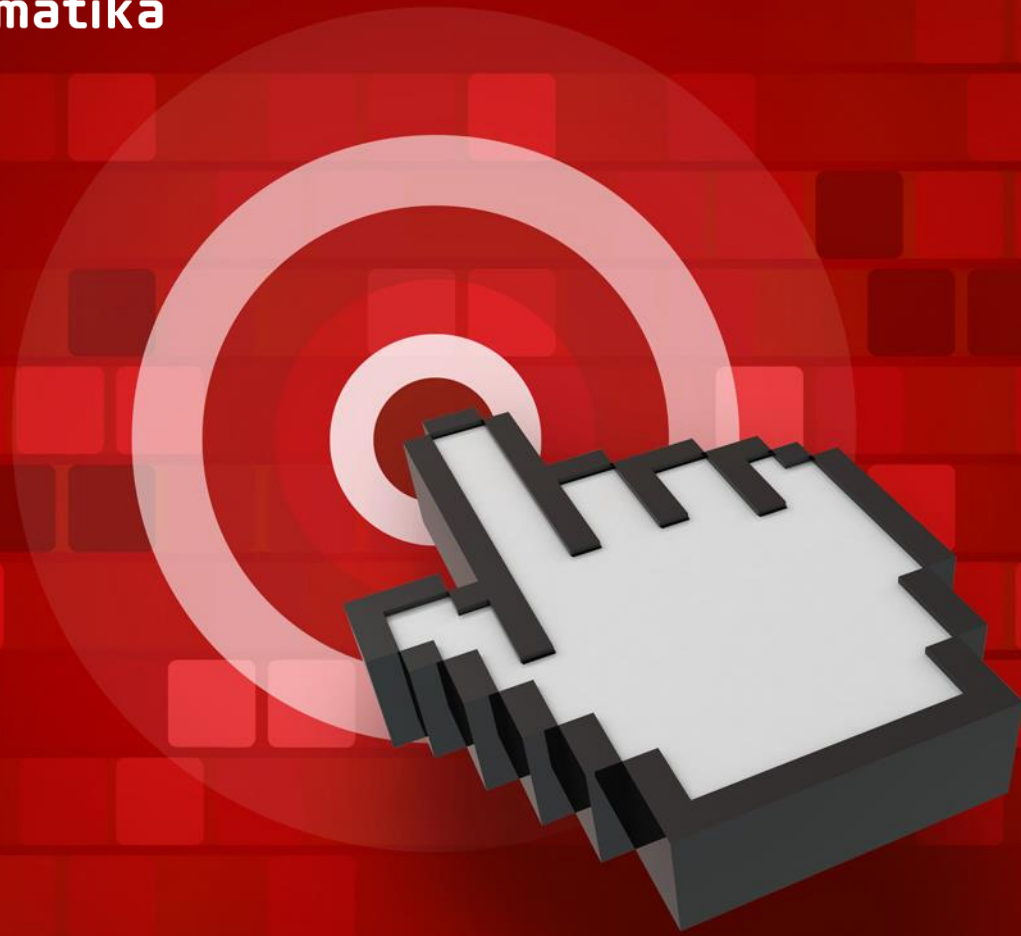
1. Dibaca dua buah harga yang dihasilkan dari pengukuran Arus (Ampere) dan Tahanan (Ohm), harus dihitung tegangan yang dihasilkan. Buat flowchart dan algoritmanya!
2. Dibaca sebuah besaran riil, yang mewakili hasil pengukuran temperatur dalam derajat Celcius. Hitung ekivalensinya dalam derajat Fahrenheit, Rheamur dan Kelvin.

Buat flowchart dan algoritmanya!





Fakultas Informatika
School of Computing
Telkom University



THANK YOU