

# **CSG2A3**

## **STRUKTUR DATA**



## **Review Dasar Algoritma dan Pemrograman**

## **Mengapa harus review DAP?**

- ▶ Struktur data merupakan bagian dari algoritma dan pemrograman
- ▶ Penulisan/penjabaran solusi dari sebuah masalah menggunakan ketentuan penulisan algoritma

# Variable and Constant

- ▶ Variable : nilai bisa berubah
- ▶ Constant: nilai tidak dapat berubah
- ▶ Deklarasi :  
nama\_variable: type\_data  
**constant** nama\_cons: type\_data = nilai

# Type Data Dasar

- ▶ Integer
- ▶ Real
- ▶ Boolean
- ▶ Character
- ▶ String

# Type Data Bentukan

- ▶ Pembuatan type data:

Type nama : string

- ▶ Deklarasi variable:

var1 : nama

Type JAM : < HH: integer[0..23], {jam}  
MM: integer[0..59], {menit}  
SS: integer[0..59] {detik}  
>

J: JAM {artinya: J adalah sebuah JAM}

J.HH {menghasilkan nilai bagian jam}

J.MM {menghasilkan nilai bagian menit}

J.SS {menghasilkan nilai bagian detik}

## Seleksi Kondisi

- ▶ Dalam menyelesaikan suatu permasalahan di dunia nyata sering kali kita berhadapan dengan kondisi percabangan/ harus melakukan pengambilan keputusan sesuai dengan kondisi/ keadaan
- ▶ **If ... then**
- ▶ **Case ... of**

## If ... then

Contoh notasi if dengan 1 aksi

```
if (condition) then  
    action
```

Contoh notasi if...else dengan 2 aksi

```
if (condition) then  
    action_1  
else  
    action_2
```



## Case... of

Program eat\_food

Kamus

day : string

Algoritma

day  $\leftarrow$  friday

case day of

'Sunday' : output('eat Apple')

'Tuesday' : output('eat Choco')

'Friday' : output('eat Orange')

else : output('do not eat anything')

## Perulangan (Looping)

- ▶ by stop condition (repeat until) (lakukan perulangan sampai tercapai suatu kondisi)
- ▶ based on the condition of repetition (while do) (lakukan perulangan selama kondisi terpenuhi)
- ▶ based enumerator (for to do) (jumlah perulangan sudah ditentukan)

# Array

## KAMUS

**TabNamaHari:** array [1..7] of string

**TabJumlahHari :** array [1..12] of integer

## ALGORITMA

**TabNamaHari[1] ← 'Senin'**

**TabJumlahHari[2] ← 4**

**Output (TabNamaHari[1], TabJumlahHari[2])**

# Prosedur

Program simple\_calculation

Kamus

A, B, C : integer

procedure

simple\_calc(I:B, C:integer;

O:A:integer)

Algoritma

input(B, C)

simple\_calc(B, C, A)

output(A, B, C)

Procedure simple\_calc(

Input:B, C:integer;

Output:A:integer)

Kamus

Algoritma

A  $\leftarrow$  10 + B - C

C  $\leftarrow$  C + 4

output(A)

# Fungsi

<p><b><u>Function</u></b> CelciusToReamur(x: <u>real</u> ) <math>\rightarrow</math> <u>real</u> {diberikan x berupa suhu dalam derajat celcius, menghasilkan nilai suhu dalam derajat reamur}</p>
<p><b>Kamus Lokal</b></p>
<p><b>Algoritma :</b>     <math>\rightarrow 0.8 * x</math></p>

# Search

- ▶ Sequential Search
- ▶ Binary Search

# Sequential Search

## Kamus

i: integer

## Algoritma

$i \leftarrow 1$

while ( $i < N$ ) and ( $\text{TabInt}[i] \neq X$ ) do

$i \leftarrow i + 1$

{ $I = N$  or  $\text{TabInt}[i] = X$ }

$\text{Found} \leftarrow \text{TabInt}[i]=X$

# Binary Search

## Kamus

atas, tengah, bawah: integer

## Algoritma

Found  $\leftarrow$  false

atas  $\leftarrow$  1

bawah  $\leftarrow$  N - 1

while (atas  $\leq$  bawah) and (not Found) do

    tengah  $\leftarrow$  (atas + bawah) div 2

if X = TabInt[tengah] then

        Found  $\leftarrow$  true

else if X < TabInt[tengah] then

        bawah  $\leftarrow$  tengah - 1

else

        atas  $\leftarrow$  tengah + 1

{atas > bawah or Found}



# Sort

- ▶ Selection Sort
- ▶ Insertion Sort ?

**Program MAXSORT** (Input/Output TabInt: TabelInteger, Input N : integer)  
{ mengurut tabel integer [1..N] terurut mengecil dengan maksimum suksesif }

**Kamus**

i : integer { indeks untuk traversal tabel }  
Pass : integer { tahapan pengurutan }  
Temp : integer { memorisasi harga untuk penukaran }  
IMax : integer { indeks, dimana TabInt [1..pass] berharga maksimum }

**Algoritma**

```

Pass traversal [1..N-1]
{ Tentukan Maximum [Pass..N] }
IMax ← pass
i traversal [pass+1..N]
    if (TabIntIMax < TabInti ) then
        IMax ← i
{ TabIntIMax adalah maximum TabInt[pass..N] }
{Tukar TabIntIMax dengan TabIntPass }
Temp ← TabIntIMax
TabIntIMax ← TabIntPass
TabIntPass ← Temp
{ TabInt[1..Pass] terurut: TabInt1 ≥ TabInt2 ≥ TabInt3 .. ≥ TabIntPass }
{Seluruh tabel terurut, TabInt1 ≥ TabInt2 ≥ TabInt3 ..... ≥ TabIntN }

```

# Question?



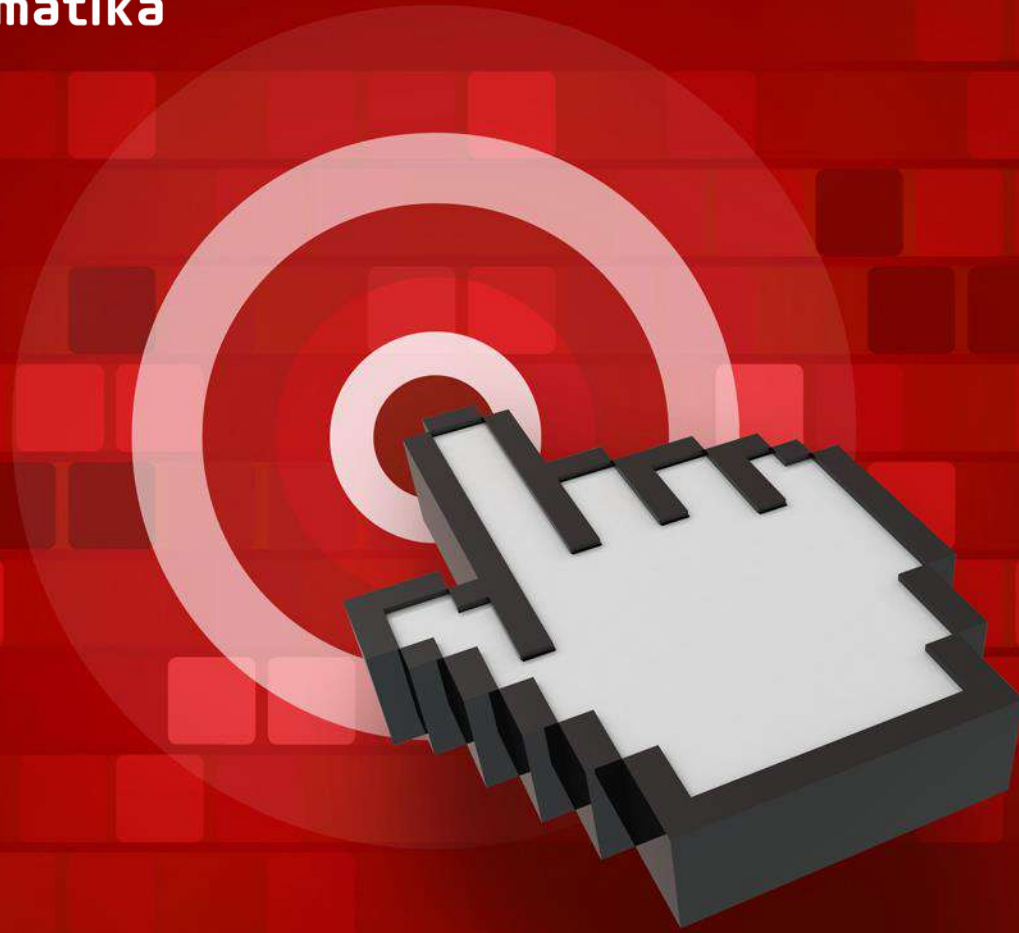
## Tugas Hari Ini

Terdapat sebuah array yang berisi bilangan bilangan integer positif dan negative. Banyaknya data pada array tersebut adalah N-data. Isi array masih belum terurut.

- a. Buatlah prosedur/fungsi untuk mengurutkan array diatas secara descending menggunakan metode Insertion Sort.
- b. Buatlah prosedur/fungsi untuk mencari suatu nilai (mis: X) pada array menggunakan Sequential Search.
- c. Buatlah main program untuk memanggil kedua fungsi/prosedur di atas.



Fakultas Informatika  
School of Computing  
Telkom University



THANK YOU