

KUG1C3

Dasar Algoritma dan Pemrograman



Procedure

What is Procedure?

- ▶ A modular algorithm that can be executed at any time

Definition

- ▶ Sederetan instruksi algoritmik yang diberi nama, dan akan menghasilkan efek neto yang terdefinisi.
- ▶ Spesifikasi
 - nama prosedur dan parameter (jika ada)
 - initial state (**IS**) dan final state (**FS**)
 - Prosedur didefinisikan dalam **kamus**
- ▶ Ketika “dipanggil”
 - akan terjadi asosiasi parameter (jika ada)
 - IS dan FS menjamin bahwa eksekusi program akan menghasilkan efek neto yang diharapkan

Definition [2]

- ▶ Jadi, setiap prosedur harus
 - Dibuat spesifikasinya dan dituliskan kode programnya
 - Dipanggil, pada saat eksekusi

Writing a Procedure

May be empty,
If any, must be defined
the variable and the type

```
Procedure procedure_name( parameter1, parameter2, ... )  
{  
  IS : .....  
  FS : ..... }  
}
```

Local Dictionary

```
{local variable}  
var1, var2 : tipe_var
```

Algorithm

```
{body procedure}
```

Calling a Procedure

Program main_program

Dictionary


Procedure procedure_name(parameter1, parameter2, ...)
var1, var2 : tipe_var

Algorithm

{body of main program}

procedure_name(var1, var2)

Executing a procedure by
calling its name



Local Variable vs. Global Variable

- ▶ Local variables are variables declared in the procedure's dictionary
- ▶ Global variables are variables declared in the main program's dictionary

Local Variable vs. Global Variable

- ▶ Declaring variable in Global Dictionary will make the variable available for the program overall and for all procedures/functions defined in main algorithm
- ▶ Declaring variable in Local Variable Dictionary will make the variable only available within the scope of the functions/procedures itself

Example Global Variable

Program simple_calculation

Kamus

A : integer

B : integer

C : integer

procedure simple_calc

Algoritma

B ← 3, C ← 5

simple_calc

output (A, B, C)

B ← 2

simple_calc

output (A, B, C)

Procedure simple_calc

Kamus

Algoritma

A ← 10 + B - C

C ← C + 4

Will use **A,B,C** from Global variable
All changes will affect global variable

A	B	C
8	2	13

>> 8, 3, 9

>> 2, 3, 13

Example Local Variable

Program simple_calculation

Kamus

A : integer
 B : integer **Global**
 C : integer

procedure simple_calc

Algoritma

B ← 3, C ← 5

simple_calc

output (A, B, C)

B ← 2

simple_calc

output (A, B, C)

Procedure simple_calc

Kamus

A, B, C : integer

Algoritma

A ← 10 + B - C

C ← C + 4

Procedure will use **A,B,C** from Local variable. All changes will stay inside the procedure and will not affect global variable.

A	B	C
	2	5

A	B	C
10		4

>> 0, 3, 5

>> 0, 2, 5

A = 10 stays in local
 Will not be outputted,
 And will be lost once the
 Procedure finished

Example Global and Local Variable

Program simple_calculation

Kamus

A : integer

B : integer

C : integer

procedure simple_calc

Algoritma

B ← 3, C ← 5

simple_calc

output (A, B, C)

B ← 2

simple_calc

output (A, B, C)

Output
variable
from Global

Procedure simple_calc

Kamus

A : integer

Algoritma

A ← 10 + B - C

C ← C + 4

Will use **B,C** from Global Variable,
and **A** from Local Variable

A	B	C
	2	13

A
8

>> 0, 3, 9

>> 0, 2, 13

Example Global and Local Variable

Program simple_calculation

Kamus

A : integer

B : integer

C : integer

procedure simple_calc

Algoritma

B ← 3, C ← 5

simple_calc

B ← 2

simple_calc

Procedure simple_calc

Kamus

A : integer

Algoritma

A ← 10 + B - C

C ← C + 4

output (A, B, C)

Output **A** from Local,
output **B** and **C** from Global

A	B	C
	2	13

A
8

>> 8, 3, 9

>> 3, 2, 13

Example Global and Local Variable

Program simple_calculation

Kamus

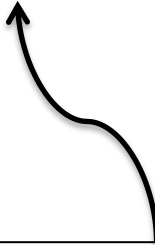
procedure simple_calc

Algoritma

B ← 3, C ← 5
simple_calc

B ← 2
simple_calc

Error !! Main
program can't use
local variable.



Procedure simple_calc

Kamus

A, B, C : integer

Algoritma

A ← 10 + B - C

C ← C + 4

output (A, B, C)

A	B	C

>> Error. Undefined variable
B and C.

Passing value using Parameter

- ▶ Sometimes we want to use various variable for the input and output for the same procedure
 - Same procedure will produce a different output from different input
- ▶ To transfer a value from program to a procedure, we used a parameter

Parameter

▶ Without Parameter

- using variable in main program as global variable
- must be careful because anyone can change the value of global variable

▶ With Parameter

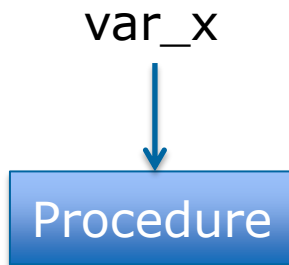
- With the same code, will produce different result when executed according to the input parameter
- Parameter defined in procedure is called **formal parameter**
- Parameter written when calling the procedure is called **actual parameter**

Formal Parameter

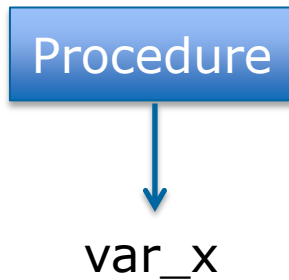
- ▶ List of variable name to define the procedure, so that it can be executed using different variable
- ▶ type parameter :
 - input, receive input value from the given variable to the procedure
 - output, return output value from procedure to the assigned variable
 - Input/output, receive input value from the given variable, and output the return value to the same variable

Formal Parameter

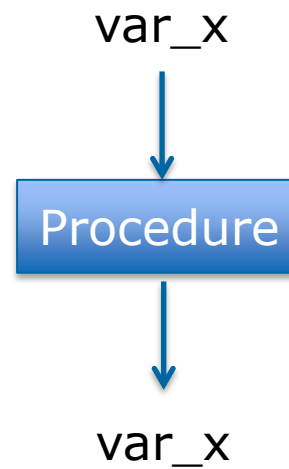
› input



› output



› input/ output



Actual Parameter

- ▶ The assigned variable or value that used when calling a procedure
 - for input parameter: can be variable or direct value
 - for output parameter: must be variable, so that the result from procedure can be stored in that variable
- ▶ When calling a procedure, the type of actual parameter must match the formal parameter
 - input parameter: the value must be defined
 - output parameter: the value doesn't need to be defined
 - input/output parameter: the value must be defined
- ▶ At the time of the call, there was an association between formal and actual parameters corresponding position (position based), then type var or value (price) should be compatible

Calling the Procedure

- ▶ Actual parameter for input can use
 - Variables or constants that have been defined in Dictionary
 - Direct value or value from other function or expression
- ▶ Actual Parameter for output must use variable name
- ▶ Actual Parameter for input/output also must use variable name

Example Using Parameter

Procedure simple_calc (Input:B; Output:A:integer)

Kamus

Algoritma

A ← 5 * B

Formal Parameter



Program try_procedure

Kamus

var1, var2, out : integer

Procedure simple_calc (I:B; O:A:integer)

Algoritma

var1 ← 4

var2 ← -2

simple_calc(var1, out)

simple_calc(var2, out)

simple_calc(10, out)

Actual
Parameter



Example Using Parameter

Program simple_calculation

Kamus

A, B, C : integer

proced

O: A: in

I: B, C:

Algoritma

B ← 3, C ← 5

simple_calc(A, B, C)

output (A, B, C)

B ← 2

simple_calc(A, B, C)

output (A, B, C)

Procedure simple_calc(
Output: A: integer;
Input: B, C: integer)

Kamus

Algoritma

A ← 1

C ← C + 4

Local **A** will be returned to main program as output parameter and received by Global **A**

Local **A, B, C** will use values from parameter

A	B	C
8	3	5

A	B	C
8	3	9

>> 8, 3, 5

>> 7, 2, 5

Careful Using Parameter

Program simple_calculation

Kamus

A, B, C : integer

procedure simple_calc(
 O:A:integer;
 I:B,C:integer)

Algoritma

B ← 3, C ← 5

simple_calc(A, B, C)

output (A, B, C)

simple_calc(A, C, B)

output (A, B, C)

simple_calc(C, B, A)

output (A, B, C)

Procedure simple_calc(
Output:A:integer;
Input:B,C:integer)

Output:A:integer;

Input:B,C:integer)

Kamus

Algoritma

A ← 10 + B - C

C ← C + 4

A	B	C
12	3	5

A	B	C
12	3	19

>> 8, 3, 5

>> 12, 3, 5

>> 12, 3, 1

Contoh 1. Voltage

Contoh 1-Prosedur: VOLTAGE

Tuliskanlah program yang membaca tahanan (Ohm) dan arus (Ampere), kemudian menghitung tegangan yang dihasilkan dan menuliskan hasilnya. Perhitungan tegangan harus dituliskan menjadi suatu prosedur bernama PROSES, supaya struktur program jelas : Input - Proses - Output.

Input : R : integer, tahanan (Ohm) dan A : integer, arus (Ampere)

Proses : menghitung $V = R * A$

Output : V : integer, tegangan (Volt)

Pelajarilah dua buah solusi yang diberikan berikut ini, dan berikan komentar anda.

Contoh Solusi 1: Voltage (Prosedur tanpa Parameter)

Program VOLTAGE1

{Program yang membaca tahanan dan arus, menghitung Voltage dan mencetak hasil perhitungan}

Kamus :

R : integer { tahanan dalam ohm}
A : integer { arus dalam ohm}
V : integer { tegangan dalam ohm}

procedure PROSES1

{ Prosedur untuk "memproses" tahanan dan arus menjadi tegangan }

Algoritma :

input (R,A)
PROSES1
output (V)

procedure PROSES1

0.1

{ I.S : diberikan harga R dan A yang telah terdefinisi}

{ FS : memproses R dan A sehingga dihasilkan V yaitu tegangan dengan rumus : $V = R * A$ }

Kamus lokal :

Algoritma :

$V \leftarrow R * A$

Contoh Solusi 2: Voltage (Solusi dengan Parameter)

Program VOLTAGE2	0.0
{Program yang membaca tahanan dan arus, menghitung Voltage dan mencetak hasil perhitungan}	
Kamus :	
R : <u>integer</u>	{ tahanan dalam ohm}
A : <u>integer</u>	{ arus dalam ohm}
V : <u>integer</u>	{ ttegangan dalam ohm}
procedure PROSES2 (<u>Input</u> : R,A : <u>integer</u> ; <u>Output</u> V: <u>integer</u>)	
{ Prosedur untuk "memproses" tahanan R dan arus A menjadi tegangan V}	

Algoritma :

```
input (R,A)
PROSES2 (R,A,V)
output (V)
```

procedure PROSES2 (<u>Input</u> : R,A : <u>integer</u> ; <u>Output</u> V: <u>integer</u>)	0.1
{ I.S : diberikan harga R dan A yang telah terdefinisi}	
{ FS : memproses R dan A sehingga dihasilkan V yaitu tegangan dengan rumus : $V = R * A$ }	
Kamus lokal :	
Algoritma :	
V ← R * A	

Komentar Solusi Voltage

- ▶ Prosedur dg parameter lebih menjamin modularitas program. Sedapat mungkin semua prosedur diparameterisasi dengan baik
- ▶ Prosedur tanpa parameter bekerja dg nama variabel global. Hanya boleh dipakai untuk kasus yg sangat khusus.
- ▶ Prosedur tanpa parameter **TIDAK BOLEH** dipakai jika alasannya hanya karena pemrogram **MALAS** menuliskan parameter !

Exercise 1 – Jumlah hari dalam bulan

Coba tuliskan prosedur dari kasus dimana diinputkan nomor bulan lalu ditentukan jumlah hari dalam bulan itu (abaikan tahun kabisat, anggap Februari adalah 28 hari)

Exercise 2 – Bonus Karyawan

Pernyataan

Tuliskan prosedur untuk menentukan bonus yang diberikan kepada karyawan PT ABC berdasarkan ketentuan yang diberikan oleh bagian personalia dan keuangan sebagai berikut:

Aturan bonus:

1. Bonus 1 juta rupiah untuk karyawan yang telah bekerja paling tidak selama 5 tahun dan umurnya sudah mencapai 50 tahun
2. Bonus 800 ribu rupiah untuk karyawan yang telah bekerja paling tidak selama 5 tahun dan umurnya belum mencapai 50 tahun
3. Bonus 500 ribu rupiah untuk karyawan yang telah bekerja kurang dari 5 tahun berapapun umurnya

Data karyawan disimpan dalam tipe bentukan **Karyawan** yang terdiri dari NIK, nama, umur, dan lama kerja.

Input: Data karyawan

Output: Bonus

Exercise 3 – Kalkulator

- ▶ Akan dibangun sebuah sistem untuk mensimulasikan kalkulator sederhana, prosedur yang diperlukan adalah:
 - Input Bilangan: menginputkan dua buah bilangan
 - Pilih Menu: menampilkan dan memilih salah satu menu kalkulator. Menu yang disediakan terdiri dari penjumlahan, pengurangan, perkalian, pembagian, dan tukar bilangan.
 - Penjumlahan: menjumlahkan dua buah bilangan
 - Pengurangan: mengurangkan dua buah bilangan
 - Perkalian: mengalikan dua buah bilangan
 - Pembagian: menghitung pembagian dua buah bilangan
 - Tukar Bilangan: menukar dua buah bilangan

Contoh Running Program

>> Menu Kalkulator

>> 1. Penjumlahan

>> 2. Pengurangan

>> 3. Perkalian

>> 4. Pembagian

>> 5. Tukar

>> Pilih menu: 1

>> Bilangan 1: 10

>> Bilangan 2: 12

>> Jumlah: 22

>> Menu Kalkulator

>> 1. Penjumlahan

>> 2. Pengurangan

>> 3. Perkalian

>> 4. Pembagian

>> 5. Tukar

>> Pilih menu: 5

>> Bilangan 1: 10

>> Bilangan 2: 12

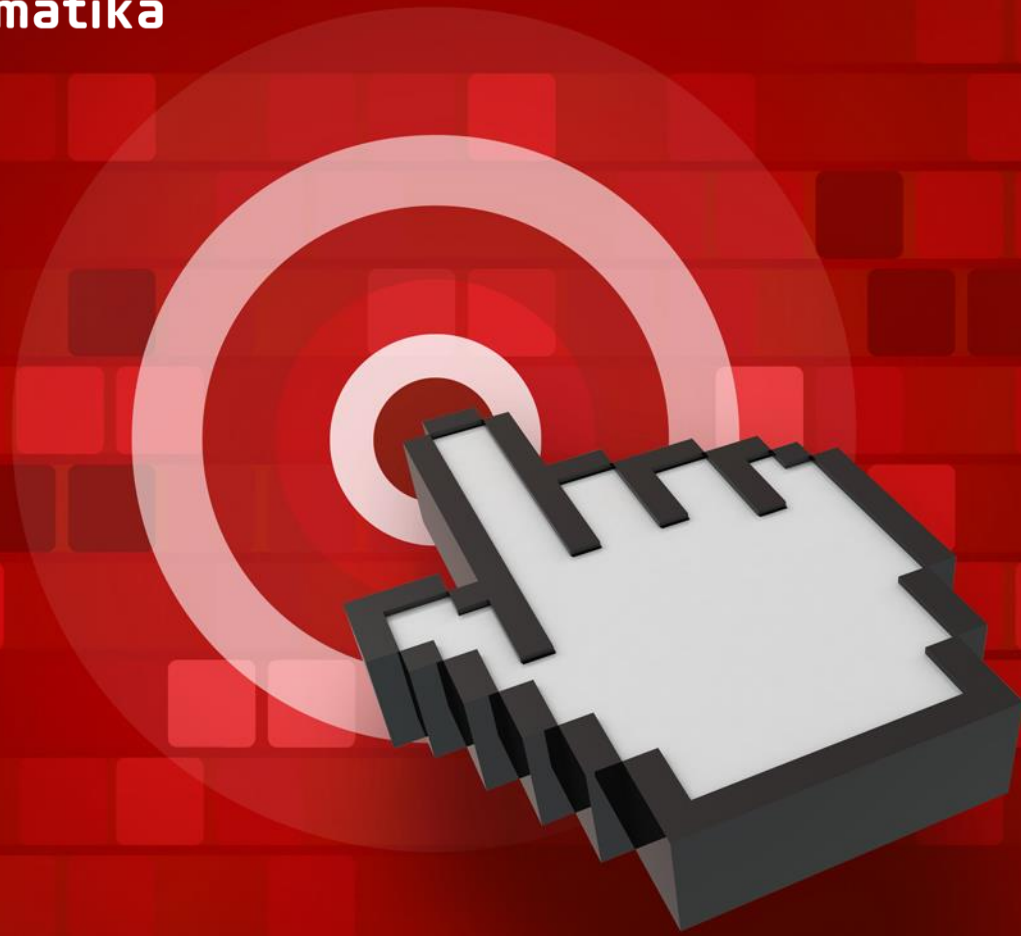
>> Tukar

>> Bilangan 1: 12

>> Bilangan 2: 10



Fakultas Informatika
School of Computing
Telkom University



THANK YOU