



# ALGORITMA DAN STRUKTUR DATA 1

---

Mia Fitriawati, M.Kom



## PENDAHULUAN

Seringkali dalam membuat program besar, pemrogram perlu memecah program menjadi beberapa subprogram yang lebih kecil.

Tiap subprogram(modul) dapat dirancang oleh pemrogram selain orang yang mengembangkan program utama.

## KEUNTUNGAN MODULARISASI

1. Untuk aktivitas yang harus dilakukan lebih dari sekali, cukup ditulis sekali sehingga dapat mengurangi panjang program.
2. Kemudahan menulis dan mencari kesalahan(*debug*) program.

## CONTOH KEUNTUNGAN MODULARISASI

<pre> Program ABCD Deklarasi A, B, C, D, temp : integer Algoritma ..... {Pertukarkan nilai A dan B} temp ← A A ← B B ← temp ..... if C &gt; D then {pertukarkan nilai C dan D} temp ← C C ← D D ← temp endif ... </pre>	<pre> Procedure TUKAR(input/output P, Q : integer) {mempertukarkan nilai P dan Q} Deklarasi Temp : integer Algoritma Temp ← P P ← Q Q ← Temp <b>Program ABCD</b> Deklarasi A, B, C, D, temp : integer Procedure TUKAR(input/output P, Q : integer) Algoritma ..... {Pertukarkan nilai A dan B} TUKAR(A,B) {panggil prosedur TUKAR} ..... if C &gt; D then {pertukarkan nilai C dan D} TUKAR(C,D) {panggil prosedur TUKAR} endif </pre>
---	--

## CONTOH ILUSTRASI

- a. Prosedur URUS PASPOR (di kantor imigrasi)
  - Isi formulir permintaan paspor dengan lampiran foto copy KTP, Kartu keluarga, pas foto
  - Serahkan formulir yang sudah diisi beserta biaya pembuatan paspor
  - Wawancara dengan petugas imigrasi
  - Terima paspor
- b. Prosedur URUS VISA (di kantor kedutaan besar)
  - Isi formulir permintaan visa dengan lampiran foto copy KTP, paspor, pas foto, tiket pesawat
  - Serahkan formulir yang sudah diisi beserta biaya pembuatan visa
  - Terima visa
- c. Prosedur BERANGKAT DARI BANDARA
  - Datang ke bandara satu jam sebelum keberangkatan
  - Jika sudah disuruh naik ke pesawat, tunjukkan tiket, paspor, dan visa ke petugas
  - Naik ke pesawat
  - Selamat jalan...

### Algoritma PERGI\_KE\_LUAR\_NEGERI

- a. Urus Paspor
- b. URUS VISA
- c. BERANGKAT DARI BANDARA

## STRUKTUR PROSEDUR

1. Bagian *Header* → nama prosedur dan komentar yang menjelaskan spesifikasi prosedur
2. Bagian Deklarasi
3. Badan Prosedur (Algoritma)

Nama prosedur sebaiknya diawali kata kerja, misalnya Hitung\_Luas, Tukar, CariMaks

---

## STRUKTUR PROSEDUR 2#

### **Procedure Nama\_Prosedur**

{Spesifikasi prosedur, penjelasan yang berisi uraian singkat mengenai apa yang dilakukan prosedur}

### **Deklarasi**

{semua nama yang dipakai dalam prosedur dan hanya berlaku lokal didefinisikan disini}

### **Algoritma**

{Badan prosedur, berisi kumpulan instruksi}

---

## CONTOH PROSEDUR 1#

Procedure CETAK\_HALLO

{mencetak string "Hallo, Dunia"}

Deklarasi

Algoritma

write('Hallo, Dunia')

## CONTOH PROSEDUR 2#

Procedure HIT\_LUAS\_SEGITIGA

{menghitung luas segitiga dengan rumus  $L = \frac{1}{2}$   
(alas x tinggi)}

Deklarasi

a, t, L: real

Algoritma

read(a, t)

$L \leftarrow a * t / 2$

write(L)

## CONTOH PROSEDUR 3#

Procedure HIT\_LUAS\_PERSEGI\_PANJANG

{menghitung luas empat persegi panjang dengan  
rumus  $L = \text{panjang} \times \text{lebar}$ }

Deklarasi

p, l, Luas: real

Algoritma

read(p, l)

$\text{Luas} \leftarrow p * l$

write('Luas = ', Luas)

## CONTOH PROSEDUR 4#

Procedure HIT\_LUAS\_LINGKARAN

{menghitung luas segitiga dengan rumus  $L = \frac{1}{2} (a \times t)$ }

Deklarasi

const phi = 3.14

r , L: real

Algoritma

read(r)

$L \leftarrow \text{phi} * r * r$

write(L)

## PEMANGGILAN PROSEDUR

Prosedur tidak bisa dieksekusi langsung. Instruksi pada prosedur bisa dilaksanakan jika prosedur diakses.

Prosedur diakses dengan memanggil namanya dari program pemanggil (program utama atau modul program lain)

**NAMA\_PROSEDUR**

---

## CONTOH PROSEDUR 1#

Program HALLO

{program utama untuk mencetak string "Hallo,  
Dunia"}

Deklarasi

Procedure CETAK\_HALLO

Algoritma

CETAK\_HALLO {panggil prosedur  
CETAK\_HALLO}

---

## CONTOH PROSEDUR 2#

Program HITUNG\_LUAS\_SEGITIGA

{program utama untuk menghitung luas segitiga}

Deklarasi

Procedure HIT\_LUAS\_SEGITIGA

Algoritma

write('Menghitung Luas Sebuah Segitiga')

HIT\_LUAS\_SEGITIGA {panggil prosedur  
HIT\_LUAS\_SEGITIGA}

write('Selesai')

Program HITUNG\_LUAS

{program utama untuk menampilkan menu perhitungan luas segitiga, luas persegi panjang, dan lingkaran, memilih menu, dan melakukan proses perhitungan sesuai pilihan menu}

Deklarasi

nomor\_menu : integer

Procedure HIT\_LUAS\_SEGITIGA

Procedure HIT\_LUAS\_PERSEGI\_PANJANG

Procedure HIT\_LUAS\_LINGKARAN

Algoritma

Repeat

{cetak menu ke layar}

write(' # Menu Pilihan Menghitung Luas # ')

write(' 1. Menghitung Luas Segitiga ')

write(' 2. Menghitung Luas Persegi Panjang ')

write(' 3. Menghitung Luas Lingkaran ')

write(' 4. Keluar Program ')

write(' Masukkan Pilihan Menu (1 / 2 / 3 / 4) : ')

read(nomor\_menu)

case nomor\_menu

1 : HIT\_LUAS\_SEGITIGA

2 : HIT\_LUAS\_PERSEGI\_PANJANG

3 : HIT\_LUAS\_LINGKARAN

4 : write('Keluar Program...Sampai Jumpa')

endcase

until nomor\_menu = 4

## CONTOH PROSEDUR 3#

## VARIABEL GLOBAL DAN VARIABEL LOKAL

- Variabel Lokal ← Ditulis pada bagian deklarasi dan hanya dapat digunakan oleh prosedur
- Variabel Global ← Ditulis pada bagian deklarasi program utama, dapat digunakan di dalam program utama maupun prosedur.



## CONTOH PROSEDUR 1#

```

Procedure Hitung_Rata_Rata
{program utama untuk menghitung rata-rata N buah bilangan bulat
yang dibaca dari keyboard}
Deklarasi
x : integer {data bilangan bulat yang dibaca dari keyboard}
k : integer {pencacah banyak bilangan}
jumlah : real {jumlah seluruh bilangan}
Algoritma
k ← 1
jumlah ← 0
while k ≤ N do
  read(x)
  jumlah ← jumlah + x
  k ← k + 1
endwhile
rata ← jumlah / N

```

## CONTOH PROSEDUR 1# (lanjutan)

```

Program Rata_Rata_Bilangan_Bulat
{program utama untuk menghitung rata-rata N buah
bilangan bulat yang dibaca dari keyboard }
Deklarasi
N : integer {banyaknya bilangan bulat }
rata : real {nilai rata-rata bilangan bulat}
Procedure Hitung_Rata_Rata
Algoritma
read(N)
write('Menghitung rata-rata bilangan bulat')
Hitung_Rata_Rata
write('Nilai rata-rata : ',rata)

```

## Parameter

- **Parameter adalah** nama- nama peubah yang dideklarasikan pada bagian *header* prosedur.
- **Parameter actual ( argument ) adalah** parameter yang disertakan pada waktu pemanggilan prosedur.
- **Parameter formal adalah** parameter yang diDeklarasikan di dalam bagian *header* prosedur itu sendiri.

## STRUKTUR PROSEDUR (PARAMETER)

**Procedure Nama\_Prosedur(daftar parameter formal)**

{Spesifikasi prosedur, penjelasan yang berisi uraian singkat mengenai apa yang dilakukan prosedur}

**Dekripsi**

{semua nama yang dipakai dalam prosedur dan hanya berlaku lokal didefinisikan disini}

**Algoritma**

{Badan prosedur, berisi kumpulan instruksi}

## **PEMANGGILAN PROSEDUR (PARAMETER)**

### **NAMA\_PROSEDUR(daftar parameter actual)**

Tiap item data ditransfer antara parameter aktual(yang disertakan pada waktu pemanggilan) dengan parameter formal(yang dideklarasikan di prosedur).

Aturan penting korespondensi satu-satu antara parameter formal dengan parameter aktual :

1. Jumlah parameter aktual harus sama dengan jumlah parameter formal
2. Tipe parameter aktual harus sama dengan tipe parameter formal
3. Tiap parameter aktual harus diekspresikan dengan cara yang benar dengan parameter formal bersesuaian, tergantung jenis parameter formal

## PARAMETER FORMAL

1. Parameter Masukan(input parameter) = parameter nilai(*value parameter*) dalam bahasa pemrograman → nilainya berlaku sebagai masukan untuk prosedur.

Nilai parameter aktual diisikan (*assign*) ke parameter formal bersesuaian. Nilai tersebut digunakan dalam badan prosedur tetapi tidak dapat dikirimkan ke titik pemanggilan.

Perubahan nilai parameter dalam badan prosedur tidak mengubah nilai parameter aktual. Nama parameter aktual boleh berbeda dengan nama parameter formal.

## CONTOH 1#

```

Procedure SATU ( input x : integer , input y : real)
{Contoh prosedur dengan parameter formal jenis parameter masukan}
Deklarasi
Algoritma
x ← x + 1
y ← y + 1
write(x)
write(y)

Program PQR
{Contoh program utama yang memanggil prosedur SATU}
Deklarasi
a , b : integer
c , d : real
Procedure SATU ( input x : integer , input y : real)
Algoritma
SATU(4,10.5)
read(a,b,c,d)
SATU(a,c)
SATU(b,d)
SATU(a+5,c/d)

```

## CONTOH 2#

Procedure HIT_LUAS_SEGITIGA(input a , t : real)	
{menghitung luas segitiga dengan rumus $L = \frac{1}{2}$ (alas x tinggi)}	
Deklarasi	Program HITUNG_LUAS_SEGITIGA
L: real	{program utama untuk menghitung luas segitiga}
Algoritma	Deklarasi
$L \leftarrow a * t / 2$	alas , tinggi : real
write('Luas Segitiga = ',L)	Procedure HIT_LUAS_SEGITIGA(input a , t : real)
	Algoritma
	write('Menghitung Luas Sebuah Segitiga')
	read(alas,tinggi)
	HIT_LUAS_SEGITIGA(alas,tinggi)
	write('Selesai')

## PARAMETER FORMAL

- Parameter Keluaran(output parameter) → menampung keluaran yang dihasilkan oleh prosedur.

Bila prosedur menghasilkan satu atau lebih nilai yang akan digunakan oleh program pemanggil, maka nilai keluaran ditampung di dalam parameter keluaran.

Bila prosedur dengan parameter keluaran dipanggil, nama parameter aktual dalam program pemanggil akan menggantikan nama parameter formal yang bersesuaian dalam prosedur.

Parameter formal  $\leftarrow$  Parameter aktual

## CONTOH 1#

Procedure DUA( input x : integer , output y : real)

{Contoh prosedur dengan parameter formal jenis parameter masukan dan jenis parameter keluaran}

Deklarasi

Algoritma

$x \leftarrow x + 1$

$y \leftarrow x * 10$

Program PQR

{Contoh program utama yang memanggil prosedur DUA}

Deklarasi

a , b : integer

Procedure DUA ( input x : integer , output y : real)

Algoritma

DUA (4,b)

write(b)

read(a)

DUA (a,b)

write(b)

DUA (a+5,b)

write(b)

## CONTOH 2#

Procedure HIT\_LUAS\_SEGITIGA( input a , t : real, output

L : real)

{menghitung luas segitiga dengan rumus  $L = \frac{1}{2}$  (alas x tinggi)}

Deklarasi

Program HITUNG\_LUAS\_SEGITIGA

Algoritma

{program utama untuk menghitung luas segitiga}

$L \leftarrow a * t / 2$

Deklarasi

Procedure HIT\_LUAS\_SEGITIGA (input alas , tinggi: real, output Luas : real)

Algoritma

write('Menghitung Luas Sebuah Segitiga')

read(alas,tinggi)

HIT\_LUAS\_SEGITIGA(alas,tinggi, Luas)

write('Luas segitiga = ', Luas)

## PARAMETER FORMAL

- Parameter masukan/keluaran(input/output parameter)  
→ sebagai masukan dan keluaran bagi prosedur

Parameter masukan digunakan pada situasi dimana informasi dikirim dari titik pemanggilan → prosedur

Parameter keluaran digunakan pada situasi dimana informasi dikirim dari titik pemanggilan → prosedur

Pada kebanyakan aplikasi, informasi harus dikirim dalam dua arah → Parameter masukan / keluaran

### CONTOH 1#

Procedure TIGA (input/output x,y : integer)

{Menambahkan nilai x dengan dan mengurangi nilai y dengan 2 }

Deklarasi

Algoritma

$x \leftarrow x + 2$

$y \leftarrow y - 2$

write('Nilai x dan y di akhir Prosedur TIGA :')

write('x = ',x)

write('y = ',y)

Program FGH

{Contoh program utama yang memanggil prosedur TIGA}

Deklarasi

a , b : integer

Procedure TIGA (input/output x, y : integer)

Algoritma

$a \leftarrow 15$

$b \leftarrow 10$

write('Nilai a dan b sebelum pemanggilan :

')

write('a = ',a)

write('b = ',b)

TIGA(a,b)

write('Nilai a dan b sesudah pemanggilan :

')

write('a = ',a)

write('b = ',b)

---

- ***Program dengan Prosedur atau tanpa Prosedur ?***

Program yang modular menunjukkan teknik pemrograman yang baik dan terstruktur

- ***Prosedur dengan parameter atau tanpa parameter?***

Parameter digunakan sebagai media komunikasi antara prosedur dengan program pemanggil dan dapat mengurangi kebutuhan penggunaan variabel global

- ***Parameter Masukan atau Parameter Keluaran?***

Bila prosedur menghasilkan keluaran yang dibutuhkan program pemanggil, gunakan parameter keluaran untuk menampung keluaran tersebut.

Bila prosedur tidak menghasilkan keluaran atau keluarannya hanya digunakan di dalam prosedur, gunakan parameter masukan. Jika prosedur menerima masukan sekaligus keluaran pada parameter yang sama, gunakan parameter masukan/keluaran

