

KUG1C3

Dasar Algoritma dan Pemrograman



Function

Pengantar

Untuk mencapai suatu tujuan besar, maka tujuan tersebut harus dibagi-bagi menjadi tujuan kecil sampai tujuan kecil itu merupakan tujuan yang dapat dicapai berdasarkan kondisi dan potensi yang dimiliki saat itu.

(Al-Khuwarizmi)



MODULARITAS

Modularitas



Keuntungan Modularitas

Program daily_activity

Kamus

Algoritma

```
output('i'm hungry')  
output('get a plate')  
output('grab any food')  
output('eat the food')
```

```
output('go to school')  
output('play game')  
output('studying')
```

```
output('i'm hungry')  
output('get a plate')  
output('grab any food')  
output('eat the food')
```

```
output('doing homework')  
output('doing laundry')
```

```
output('i'm hungry')  
output('get a plate')  
output('grab any food')  
output('eat the food')
```

Procedure EATING

Kamus

Algoritma

```
output('i'm hungry')  
output('get a plate')  
output('grab any food')  
output('eat the food')
```

- Menghindari penulisan teks program yang sama secara berulang kali
- Kemudahan membaca program yang besar
- Kemudahan men-debug program
- Kemudahan pengembangan program yang dikerjakan secara tim

Bentuk Modularitas dalam Algoritma

- ▶ Fungsi
- ▶ Prosedur

Definisi Fungsi [1]

- ▶ Fungsi adalah sebuah transformasi akibat pemetaan suatu nilai ke nilai lain
- ▶ Secara algoritmik, fungsi akan menerima harga yang diberikan lewat parameter formal dan menghasilkan nilai sesuai dengan definisi dalam spesifikasinya

Definisi Fungsi [2]

- ▶ Dalam penulisannya fungsi terdiri dari:
 - Nama fungsi
 - Parameter formal
 - Type hasil
- ▶ Fungsi yang telah didefinisikan dapat “dipanggil” untuk dieksekusi lewat namanya dan diberikan parameter aktual

Definisi Fungsi [3]

▶ Contoh fungsi:

$$f(x) = 2x+4$$

- Nilai fungsi f tergantung dari nilai parameter x :
 - Jika $x=4$ maka $f(x)$ akan menghasilkan 12
 - Jika $x=1$ maka $f(x)$ akan menghasilkan 6

$$f(x, y) = \frac{2x}{y}$$

- Nilai fungsi f tergantung dari nilai parameter x dan y :
 - Jika $x=2$ dan $y=2$, maka $f(x, y)$ akan menghasilkan 2
 - Jika $x=1$ dan $y=5$, maka $f(x, y)$ akan menghasilkan 0.4

Notasi Algoritmik Fungsi [1]

function NAMA F (*<list-parameter input>*) → *<type hasil>*

{Spesifikasi fungsi}

Kamus lokal:

{ semua NAMA yang dipakai dalam algoritma dari fungsi}

Algoritma :

{deretan instruksi algoritmik :

 pemberian harga, input, output, analisa kasus, pengulangan)

{ Pengiriman harga di akhir fungsi, harus sesuai dengan type hasil}

 → **hasil**

Notasi Algoritmik Fungsi [2]

dengan syarat :

- *list parameter input* boleh tidak ada (kosong), dalam hal ini di fungsi tidak membutuhkan apa-apa dari pemakainya untuk menghasilkan harga.
- jika *list parameter input* (parameter **formal**) tidak kosong, minimal mengandung satu nama, maka nama tersebut harus berupa **nama informasi** beserta typenya.
- instruksi “terakhir” yang harus ada pada fungsi harus merupakan pengiriman harga yang dihasilkan oleh fungsi (dituliskan seperti pada notasi di atas, dengan type hasil boleh type dasar atau type terstruktur). Type hasil boleh dinyatakan oleh suatu **nama type**. Dengan catatan, bahwa instruksi “terakhir” belum tentu dituliskan pada baris terakhir, misalnya jika hasil merupakan sebuah nilai yang dikirimkan berdasarkan analisa kasus.

Penulisan Fungsi [1]

- ▶ Menulis Fungsi hampir sama dengan menulis sebuah program, yaitu ada bagian header, kamus dan algoritma
- ▶ Keyword penulisan fungsi:
 - No I/O statement
 - Process only
 - RETURN VALUE is a MUST
- ▶ Misalkan akan dibuat fungsi $f(x) = 2x+4$

Penulisan Fungsi [2]

function f (x: integer) → integer

Kamus Lokal

fx: integer

Algoritma

fx ← 2x+4

→ fx

- ▶ NB: Kamus yang ada pada deklarasi fungsi merupakan kamus lokal (nama-nama yang didefinisikan hanya dikenali dalam blok fungsi)

Penulisan Fungsi [3]

function f (x: integer) → integer

Kamus Lokal

Algoritma
→ $2x+4$

- ▶ Fungsi dapat langsung me-return nilai dari suatu ekspresi

Pemanggilan Fungsi [1]

Program POKOKPERSOALAN

{Spesifikasi : Input, Proses, Output}

Kamus :

{ semua NAMA yang dipakai dalam algoritma }

function NAMA_F (<list nama parameter formal/input>) → <type hasil>
{Spesifikasi fungsi}

Algoritma :

{ deretan instruksi pemberian harga, input, output, analisa kasus,
pengulangan yg memakai fungsi }

{ Harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi }

nama ← **NAMA_F** (<list parameter aktual>)

output (**NAMA_F** (<list parameter aktual>))

{ Harga yang dihasilkan fungsi jugadapat dipakai dalam ekspresi }

Pemanggilan Fungsi [2]

Catatan :

1. Pada waktu pemanggilan terjadilah asosiasi antara parameter formal/input dengan parameter aktual sesuai dengan urutan penulisan dalam list-nama parameter input.
2. List parameter input dapat berupa nama informasi atau nama konstanta yang telah terdefinisi dalam kamus atau konstanta; dapat juga berupa harga konstanta, atau harga yang dihasilkan oleh suatu ekspresi atau fungsi.
3. List parameter aktual harus sama jumlah, urutan dan typenya dengan list parameter formal pada pendefinisian fungsinya.
4. Harga yang dihasilkan oleh fungsi dapat didefinisikan domainnya dengan lebih rinci.
5. Pada akhir dari eksekusi fungsi, harga yang dihasilkan oleh fungsi dikirimkan ke pemakainya.
6. Fungsi boleh dipakai oleh program utama, prosedur atau fungsi lain

Contoh pemanggilan fungsi

Program PanggilF

Kamus

x, y, a, c : integer
 cek : boolean
function f (x : integer) \rightarrow integer

Algoritma

input (x)
output ($f(x)$)
 $x \leftarrow 5$
output ($f(x)$)
output ($f(9)$)
input (a)
output ($f(a)$)
 $c \leftarrow f(a)+8$
 $cek \leftarrow f(a) > 5$

Notes:

- Pemanggilan Fungsi bisa langsung DIOUTPUTKAN atau DIOPERASIKAN
- NAMA parameter aktual BOLEH BERBEDA dengan nama parameter formal, asal TIPE DATANYA SAMA

Proses Eksekusi Fungsi

function **f** (**x: integer**) → integer

Kamus Lokal

Algoritma
→ $2x+4$

Parameter Formal

Terjadi proses asosiasi antara parameter formal dengan aktual

Program PanggilF

Kamus

x, y, a, c : integer

cek : boolean

function f (x: integer) → integer

Algoritma

input (a)

output (f(a))

Parameter Aktual

Bagaimana???

Sudah cukup jelas dengan cara penulisan fungsi dan pemanggilannya??

Ingat bahwa Fungsi akan terlihat kegunaannya apabila fungsi telah dipanggil!!

Fungsi Terdefinisi

- ▶ Fungsi terdefinisi adalah fungsi yang sudah diberikan / disediakan oleh sistem, dan tinggal dipakai (dipanggil)
- ▶ Fungsi terdefinisi selalu diberikan daftar dan spesifikasinya (domain+range)
- ▶ Nb: Setiap bahasa pemrograman akan menyediakan fungsi terdefinisi terutama pada proses/operasi yang umum digunakan seperti konversi tipe data, mencari nilai sinus

Contoh Fungsi Terdefinisi [1]

- Fungsi Terdefinisi untuk melakukan konversi tipe

Function RealToInteger (x:real) → integer

{ mengkonversi harga x yang bertipe real menjadi harga lain yang bertipe integer dengan pemotongan, yaitu menghilangkan bagian di belakang titik desimalnya }

Function IntToReal (i:integer) → real

{ mengkonversi harga i yang bertipe integer menjadi harga ekuivalen yang bertipe real }

- Fungsi Terdefinisi untuk memperoleh informasi tentang sebuah string

function LONG (S:string) → integer

{Menghasilkan panjang string S}

function KOSONG (S: string) → boolean

{true jika S adalah string kosong }

Contoh Fungsi Terdefinisi [2]

- ▶ Fungsi Terdefinisi untuk seleksi terhadap sebuah string

function AWAL (S: string) → string
{mengambil sisa string S tanpa karakter terakhir}

function AKHIR (S: string) → string
{mengambil sisa string S tanpa karakter pertama}

function FIRSTCHAR (S:string , tidak kosong) → character
{menghasilkan Karakter pertama string S}

function LASTCHAR (S: string, tidak kosong) → character
{menghasilkan Karakter terakhir string S}

Contoh Fungsi Terdefinisi [3]

- ▶ Fungsi Terdefinisi untuk perhitungan Matematika

function Sin (x: real) → real
{Memberikan $\sin(x)$, x dalam radian}

function Cos (x: real) → real
{Memberikan $\cos(x)$, x dalam radian}

function Abs (x: real) → real
{Memberikan $|x|$ }

Contoh Fungsi Terdefinisi [4]

- ▶ Fungsi Terdefinisi untuk suksesor dan predesesor integer

function Succ ($x: \text{integer}$) \rightarrow integer

{Memberikan suksesor x , $x + 1$.

Contoh: Succ(0) = 1
 Succ(-1) = 0
 Succ(5) = 6}

function Pred ($x: \text{integer}$) \rightarrow integer

{Memberikan predesesor x , $x - 1$

Contoh: Pred(-1) = -2
 Pred(3) = 2 }

Fungsi dengan > 1 Nilai Balik [1]

Tuliskanlah sebuah fungsi yang jika diberikan sebuah angka centi meter, akan menghasilkan pasangan harga $\langle m, \text{cm} \rangle$ yang menyatakan hasil konversi centi meter menjadi meter dan centi meter.

Contoh: $f(150) = \langle 1, 50 \rangle$

**function konversiCM (cm: integer) → <integer, integer>
{diberikan cm, mengubahnya menjadi berapa meter dan centi meter}**

Kamus Lokal

meter, centi: integer

Algoritma

meter ← cm div 100

centi ← cm mod 100

→ $\langle \text{meter}, \text{centi} \rangle$

Fungsi dengan > 1 Nilai Balik [2]

function konversiCM2 (cm: integer) → <integer, integer>
{diberikan cm, mengubahnya menjadi berapa meter dan centi meter}

Kamus Lokal

Algoritma

→ <cm div 100, cm mod 100>

Summary

- Dari contoh-contoh penulisan fungsi yang telah ditunjukkan, tidak ada perintah untuk menampilkan hasil di dalam fungsi (output). Perintah/ tahapan untuk menampilkan hasil ada pada program pemanggil fungsi.
- Fungsi ditujukan untuk melakukan proses transformasi/ perubahan tanpa proses input/output
- Fungsi dapat dipanggil dari program utama atau fungsi yang lain (ingat aturan pemanggilan!!)

Exercise 1

- ▶ Tulislah sebuah fungsi `IsGenap` untuk menentukan apakah sebuah angka yang diberikan kepadanya adalah angka genap atau bukan!

Harga yang dihasilkan adalah `TRUE` jika genap, dan `FALSE` jika bukan genap.

- ▶ Buat program utama yang memanggil fungsi tersebut!

Exercise 2

- ▶ Tulislah sebuah fungsi toDetik yang menerima masukan jam, menit, detik, kemudian mengkonversikannya menjadi detik!
- ▶ Buat program utama yang memanggil fungsi tersebut!

Exercise 3

- ▶ Tulislah sebuah fungsi `IsVokal` untuk menentukan apakah sebuah karakter yang diberikan kepadanya adalah huruf vokal atau bukan!

Harga yang dihasilkan adalah 'TRUE' jika huruf vokal, dan 'FALSE' jika bukan huruf vokal.

- ▶ Buat program utama yang memanggil fungsi tersebut!

Exercise 4

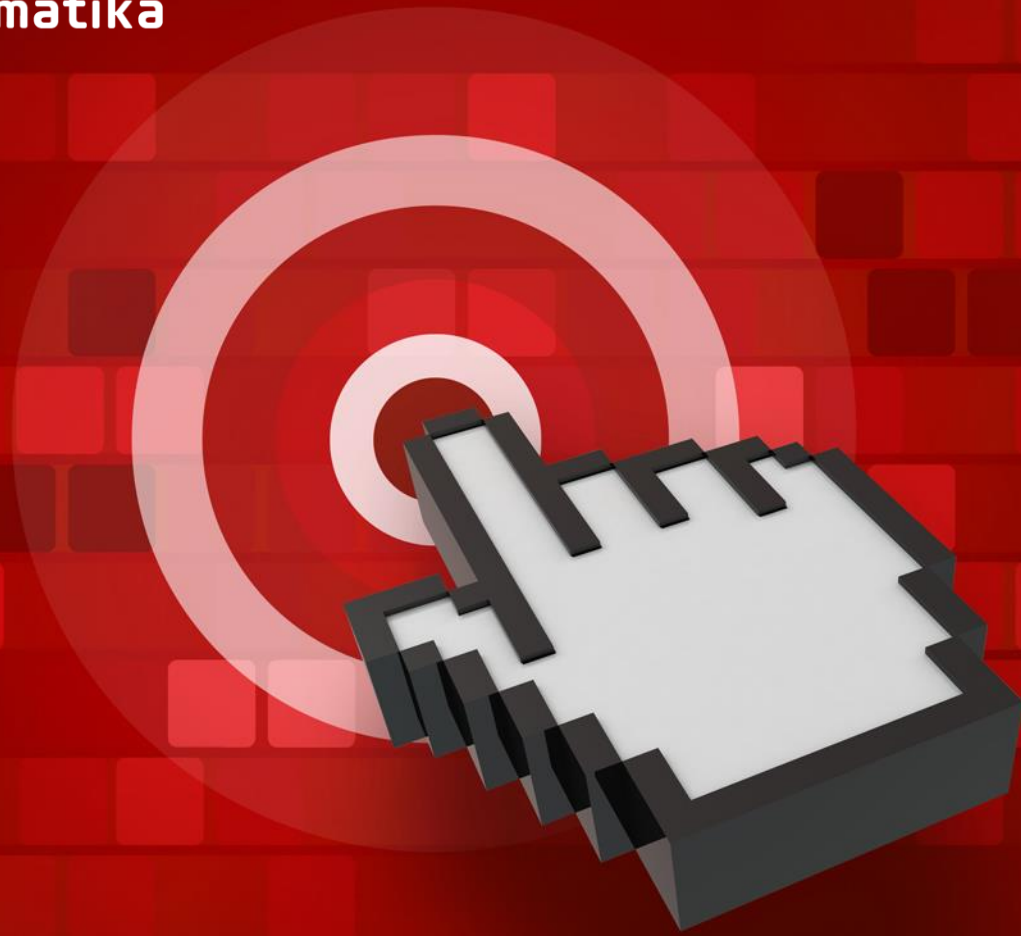
- ▶ Tuliskanlah sebuah fungsi untuk memecah uang menjadi:
 - a lembar 10000
 - b lembar 5000
 - c lembar 1000
- Contoh: $f(126000) = \langle 12, 1, 1 \rangle$
- ▶ Buat program utama yang memanggil fungsi tersebut!

Exercise 5

- ▶ Definisikan sebuah type pecahan yang terdiri dari pembilang dan penyebut berupa bilangan integer, kemudian buat fungsi kaliP yang menerima dua buah pecahan dan menghasilkan hasil kali berupa pecahan!
- ▶ Buat program utama yang memanggil fungsi tersebut!



Fakultas Informatika
School of Computing
Telkom University



THANK YOU