

Pemrograman Prosedural

FUNGSI

(Subprogram)

Tim Pengajar KU1071

Sem. 1 2009 - 2010

Tujuan Perkuliahan

- Mahasiswa memahami makna dan kegunaan fungsi sebagai salah satu sub program
- Mahasiswa dapat menggunakan notasi fungsi dengan benar
- Mahasiswa dapat membuat program dengan menggunakan fungsi

Definisi Fungsi

- Fungsi adalah sebuah transformasi akibat pemetaan suatu nilai (dari **domain**) ke nilai lain (dalam **range**)
- Fungsi diberi **nama**, dan **parameter formal** (harga masukan yang diberi nama dan dijelaskan type-nya)
- Fungsi harus didefinisikan dalam kamus

Contoh Fungsi

- Fungsi bernama $f(x)$ memiliki satu parameter x didefinisikan sebagai $f(x) = x^2 + 3x - 5$
 - jika diberi harga $x = 4$ maka $f(x)$ akan menghasilkan 23
 - jika diberi harga $x = 1$ maka $f(x)$ akan menghasilkan -1
- Fungsi $f(x,y)$ memiliki dua parameter x dan y , didefinisikan sebagai $f(x,y) = x^2 + 3xy - 5y - 1$
 - jika diberi harga $x = 0$ dan $y = 0$ maka $f(x,y)$ akan menghasilkan -1
 - jika diberi harga $x = 1$ dan $y = 0$ maka $f(x,y)$ akan menghasilkan 0

Notasi Algoritmik untuk Pendefinisian Fungsi (1)

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

{Spesifikasi fungsi}

Kamus lokal:

{semua NAMA yang dipakai dalam algoritma dari fungsi}

Algoritma:

{deretan fungsi algoritmik:

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

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

→**hasil**

Notasi Algoritmik untuk Pendefinisian 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**) ada (tidak kosong, minimal satu nama), maka harus berupa satu atau beberapa nama INFORMASI beserta typenya (bisa type dasar atau terstruktur)
- pada akhir dari FUNGSI, harga yang dihasilkan oleh fungsi dituliskan seperti pada notasi di atas, type hasil boleh type dasar atau type terstruktur

Notasi Algoritmik untuk Pemanggilan Fungsi (1)

Program POKOKPERSOALAN

{Spesifikasi: Input, Proses, Output}

Kamus global
(vs kamus lokal)

Kamus :

{semua NAMA yang dipakai dalam algoritma}

Hasil atau output dari fungsi

Parameter formal

function NAMA^F ([list nama parameter input]) → [type hasil]
{Spesifikasi fungsi}

Algoritma:

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

{harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi}

nama ← **NAMA^F ([list parameter aktual])**

output (**NAMA^F ([list parameter aktual])**)

Parameter aktual

{harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi}

Notasi Algoritmik untuk Pemanggilan Fungsi (2)

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

Contoh-contoh Fungsi

Fungsi FX_Kuadrat

function FX_Kuadrat (| x: integer) \rightarrow integer

{Diberikan x, integer, menghitung $f(x) = x^2 + 2x - 5$ }

Kamus lokal:

Algoritma: {penghitungan}

$\rightarrow (x * x + 2 * x - 5)$

Program CONTOH1

{input: x; output: hasil penghitungan $x^2 + 3x - 5$; proses: dibaca x, menghitung: $f(x) = x^2 + 3x - 5$; menuliskan hasil perhitungan }

Kamus :

x : integer { data }

FX : integer { Hasil perhitungan $f(x) = x^2 + 3x - 5$ }

function FX_KUADRAT (x : integer) \rightarrow integer
{ diberikan x, menghitung $f(x) = x^2 + 3x - 5$ }

Algoritma :

input (x)

FX \leftarrow FX_KUADRAT(x)

output (FX)

Algoritma:

input (x)

FX \leftarrow 3*FX_KUADRAT(x)

output (x)

Algoritma:

input (x)

output (FX_KUADRAT(x))

Fungsi Konversi

- **Persoalan:**

- Tuliskanlah sebuah fungsi, yang mengkonversikan harga karakter angka (nol sampai dengan 9) menjadi harga numerik sesuai dengan karakter yang tertulis.

Contoh :

- '0' \rightarrow 0
- '8' \rightarrow 8

- **Spesifikasi :**

- Fungsi KarakterToInteger :
- Domain : $x : \text{character ['0'..'9']}$
- Range : integer [0..9]
- Proses : analisis kasus terhadap x , untuk setiap harga x diasosiasikan integer yang sesuai.

function KarakterToInteger (x:character['0'..'9']) → integer [0..9]

{diberikan x berupa karakter, menghasilkan harga integer yang sesuai dengan penulisan pada karakter}

Kamus lokal:

Algoritma:

depend on (x)

x = '0' : → 0

x = '1' : → 1

x = '2' : → 2

x = '3' : → 3

x = '4' : → 4

x = '5' : → 5

x = '6' : → 6

x = '7' : → 7

x = '8' : → 8

x = '9' : → 9

Fungsi IsAnA

- **Persoalan :**

- Tuliskanlah fungsi IsAnA yang mentest apakah sebuah karakter yang diberikan kepadanya adalah sebuah huruf 'A'. Harga yang dihasilkan adalah benar jika huruf itu 'A', salah jika huruf itu bukan 'A'
- Contoh : IsAnA('A') → true
- IsAnA ('X') → false
- IsAnA ('Y') → false

- **Spesifikasi :**

- Fungsi IsAnA :
- Domain : x (karakter)
- Range : boolean
- Proses : menghasilkan true jika x adalah 'A', false jika tidak

function IsAnA (x:character) → boolean

{Menghasilkan true jika x adalah 'A'; dengan menuliskan ekspresi boolean }

Kamus lokal:

Algoritma:

→ (x = 'A')

Algoritma:

if (x = 'A') then

→ true

else { x ≠ 'A' }

→ false

Hitung m dan cm

- **Persoalan :**

- Tuliskanlah sebuah fungsi, yang jika diberikan sebuah angka Cm yang menyatakan panjang dalam cm, akan menghasilkan pasangan harga $\langle x1, x2 \rangle$ sesuai dengan rumus ukuran metris ($1\text{ m} = 100\text{ cm}$). sehingga $x1 * 100 + x2 = \text{Cm}$
- Contoh : $F(100) = \langle 1, 0 \rangle$
- $F(355) = \langle 3, 55 \rangle$

- **Spesifikasi :**

- Fungsi KonversiCm :
- Domain : Cm : integer
- Range : pasangan harga integer
- Proses : menghitung Meter dan SisaCm sehingga $\text{Cm} = 100 * \text{Meter} + \text{SisaCm}$

function KonversiCm (Cm : integer) \rightarrow \langle integer, integer \rangle
{diberikan cm, mengubahnya menjadi berapa meter dan cm}

function KonversiCm1 (Cm:integer) → <integer, integer>
{diberikan Cm, mengubahnya menjadi berapa meter dan cm }

Kamus lokal:

meter: integer { meter }
sisacm: integer { sisa cm }

Algoritma:

meter ← Cm div 100
sisacm ← Cm mod 100
→ <meter, sisacm>

function KonversiCm2 (Cm:integer) → <integer, integer>
{diberikan Cm, mengubahnya menjadi berapa meter dan cm }

Kamus lokal:

Algoritma:

→ < Cm div 100, Cm mod 100 >

MAX2 dan MAX3

- Tuliskan fungsi MAX2, yang menerima masukan dua buah bilangan real dan menghasilkan sebuah bilangan real yang merupakan bilangan terbesar
 - Cth: $\text{MAX2}(1,2) \rightarrow 2$
- Tuliskan fungsi MAX3 yang memanfaatkan fungsi MAX2. Fungsi MAX3 menerima input 3 bilangan real
 - Cth: $\text{MAX3}(10,2,3) \rightarrow 10$

function MAX2 (a, b: real) → real
{diberikan a dan b, menghasilkan a jika $a \geq b$, dan b jika $b > a$ }

Kamus lokal

Algoritma
 if ($a \geq b$) then
 → a
 else
 → b

function MAX3 (a, b, c: real) → real
{ diberikan a,b dan c, menghasilkan a jika $a \geq b$ dan $a \geq c$, menghasilkan b jika $b \geq a$ dan $b \geq c$, menghasilkan c jika $c \geq a$ dan $c \geq b$ }

Kamus lokal

Algoritma
 → MAX2(MAX2(a,b),c)

Algoritma
 → MAX2(a,MAX2(b,c))

Penanggalan & Nextday

- Input dan output merupakan type bentukan
- Didefinisikan sebuah type DATE yang memiliki informasi tanggal, bulan dan tahun
- Tuliskan algoritma untuk:
 - membaca sebuah tanggal dan sebuah kode penulisan (1=Inggris, 2=Indonesia, 3=Perancis)
 - Menuliskan tanggal keesokan harinya sesuai kode bahasa. Dalam bahasa Inggris, DATE ditulis dalam Bulan '/' Tanggal '/' Tahun. Dalam bahasa Indonesia, DATE ditulis dalam Tanggal '-' Bulan '-' Tahun. Dalam bahasa Perancis, DATE ditulis dalam Tanggal '/' Bulan '/' Tahun
 - Proses menghitung hari esok dilakukan oleh sebuah fungsi NextDay yang menerima masukan sebuah tanggal dan menghasilkan tanggal keesokan harinya. Contoh pemanggilan dan hasil fungsi adalah
 - NextDay(<13,4,1990>) → <14,4,1990>
 - NextDay(<31,12,1990>) → <1,1,1991>

Program PENANGGALAN

{diberikan input sebuah DATE dan kode penulisan, menghasilkan tanggal hari berikutnya sesuai format dari kode penulisan}

Kamus lokal:

type Tanggal: integer [1..31]

type Bulan: integer [1..12]

type Tahun: integer > 0

type DATE: <DD: Tanggal, MM: Bulan, YY: Tahun>

Harilni, Esok: DATE

KodeBahasa: integer [1..3]

function NEXTDAY (Now: DATE) → DATE {mengirimkan keesokan hari dari Now}

Algoritma:

input (Harilni, KodeBahasa)

Esok ← NEXTDAY (Harilni)

depend on KodeBahasa

KodeBahasa = 1 : output(Esok.MM,' / ',Esok.DD,' / ',Esok.YY)

KodeBahasa = 2 : output(Esok.DD,' - ',Esok.MM,' - ',Esok.YY)

KodeBahasa = 3 : output(Esok.DD,' / ',Esok.MM,' / ',Esok.YY)

Fungsi NEXTDAY

- Buatlah fungsi NEXTDAY dengan mengecek informasi bulan dari input DATE
 - Bulan = 2 → jumlah hari bisa 28 atau 29; gunakan fungsi ISKABISAT
 - Bulan = 12 → untuk tgl 31, hari keesokannya adalah tahun baru
 - Bulan = 1, 3, 5, 7, 8, 10 → jumlah hari adalah 31
 - Bulan = 4, 6, 9, 11 → jumlah hari adalah 30

Translasi ke Pascal

Pendefinisian/Spesifikasi Fungsi

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

```
function NAMA F (<list-parameter input>) : <type hasil>
```

```
(* Spesifikasi fungsi: diberikan ... menghasilkan ... *)
```

```
(* dalam bahasa Pascal standard, type hasil harus type primitif, *)
```

```
(* atau untuk array diberi nama *)
```

```
(* Kamus lokal: boleh mengandung VAR, TYPE, CONST *)
```

VAR

```
(* semua NAMA yang dipakai dalam algoritma/realisasi fungsi *)
```

```
(* ALGORITMA *)
```

begin

```
{deretan instruksi algoritmik:
```

```
    pemberian harga, input, output, analisis kasus, pengulangan}
```

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

```
    Nama F := <ekspresi hasil>
```

end;

Pemanggilan Fungsi

(* Program POKOKPERSOALAN *)

Program POKOKPERSOALAN;

(* Spesifikasi: Input, Proses, Output *)

(* KAMUS *)

(* semua NAMA yang dipakai dalam program *)

(* TYPE *)

(* CONST *)

(* VAR *)

(* spesifikasi dan body fungsi langsung dituliskan di bagian ini *)

(* ALGORITMA *)

begin

{ deretan instruksi pemberian harga, input, output, analisis kasus,
pengulangan yang memakai fungsi }

{ Harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi }

nama := **NAMAF** (<list parameter aktual>)

write(**NAMAF** (<list parameter aktual>))

(* parameter aktual dapat berupa nilai, konstanta, ekspresi *)

(* Harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi *)

end.

Latihan Soal

- Pecahan (hal 75)
 - Definisikan sebuah type pecahan yang terdiri dari pembilang dan penyebut bilangan integer, dan sekumpulan fungsi yang merupakan realisasi dari operator pecahan:
 - JumlahP: menerima dua pecahan, menghasilkan jumlah berupa pecahan
 - KurangP: menerima dua pecahan, menghasilkan selisih berupa pecahan
 - KaliP: menerima dua buah pecahan, menghasilkan hasil kali berupa pecahan
 - BagiP: menerima dua buah pecahan, menghasilkan hasil bagi berupa pecahan