

FUNGSI

DASAR PEMROGRAMAN

suprayogi@dsn.dinus.ac.id

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)

functionNAMAF (*<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

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

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 NMAF ([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 ← **NMAF ([list parameter aktual])**

output (**NMAF ([list parameter aktual])**)

Parameter aktual

{harga yang dihasilkan fungsi juga dapat dipakai dalam ekspresi}

NOTASI ALGORITMIK UNTUK PEMANGGILAN FUNGSI

- 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 FUNGSI

FUNGSI

FX_KUADRAT

LATIHAN 1

Program FungsiKuadrat

{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) → integer

{ diberikan x, menghitung $f(x) = x^2 + 3x - 5$ }

Algoritma:

input(x)

FX=FX_KUADRAT(x)

output(FX)

Function FX_KUADRAT(x:integer) → integer

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

Kamus Lokal:

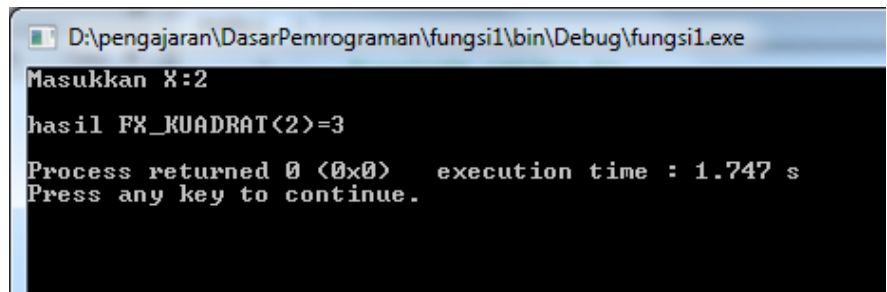
Algoritma:

→ $(x*x) + (2*x) - 5$

Input(x)
Output(FX_KUADRAT(x))

Input(x)
FX=3*FX_KUADRAT(x)
Output(FX)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /*
5  program fungsi kuadrat
6  {input: x; output: hasil penghitungan  $x^2 + 3x - 5$ ; proses: dibaca x,
7  menghitung:  $f(x) = x^2 + 3x - 5$ ; menuliskan hasil perhitungan }
8  */
9
10 int main()
11 {
12     //kamus
13     int x,FX;
14     int FX_KUADRAT(int x);
15     //algoritma
16     printf("Masukkan X:");scanf("%d",&x);
17     FX=FX_KUADRAT(x);
18     printf("\n");
19     printf("hasil FX_KUADRAT(%d)=%d\n",x,FX);
20     return 0;
21 }
22 int FX_KUADRAT(int x){
23     return (x*x)+(2*x)-5;
24 }
25
```



```
D:\pengajaran\DasarPemrograman\fungsi1\bin\Debug\fungsi1.exe
Masukkan X:2
hasil FX_KUADRAT(2)=3
Process returned 0 (0x0) execution time : 1.747 s
Press any key to continue.
```

LATIHAN 2

Translasikan program menghitung kecepatan dengan masukan berupa **S** (jarak dalam Meter) dan **t** (waktu dalam detik) yang menggunakan **fungsi** `hitungKecepatan` dengan rumus $V=S/t$.

Program Menghitung kecepatan
{menghitung kecepatan dengan masukan berupa S (jarak dalam Meter) dan t (waktu dalam detik) dan mencetak hasil perhitungan}

Kamus:

S:real {jarak dalam meter}

t:real {waktu dalam detik}

V:real {kecepatan dalam m/d}

FunctionhitungKecepatan(input S,t:real):real

{fungsi untuk memproses jarak S dan waktu t menjadi kecepatan V}

Input (S,t)

V=hitungKecepatan(S,t)

output(V)

FunctionhitungKecepatan(input S,t:real):real

{memproses S dan t sehingga dihasilkan kecepatan(V) yaitu $V=S/t$ }

Kamus Lokal:

V

→ $V= S/t$

LATIHAN 3

Buat *notasi algoritma* (*dengan translasinya*) untuk program konversi suhu dengan masukan berupa suhu C dalam derajat celcius bertipe real dan kode_konversi (F,K,R) bertipe char, dengan menggunakan fungsi konversiSuhu dengan rumus:

jika kode_konversi='F' maka dihasilkan hasil= $(9/5 \times C) + 32$

jika kode_konversi='R' maka dihasilkan hasil= $4/5 \times C$

jika kode_konversi='K' maka dihasilkan hasil= $C + 273$

LATIHAN 4

Translasikan program menghitung selisih dua buah jam yang menggunakan fungsi SelisihJam dengan:

Fungsi selisihJam memproses jam_awal dan jam_akhir sehingga dihasilkan jam_selisih yaitu:

$$\text{jam_selisih} = \text{jam_akhir} - \text{jam_awal}$$

Program Selisih Jam

Kamus:

Type jam : <jj:int,mm:int,dd:int>

jam_awal,jam_akhir,jam_selisih : jam

Function selisihJam(input jam1,jam2:jam):jam

{menghitung selisih jam_awal dengan jam_akhir }

input(jam_awal.jj, jam_awal.mm, jam_awal.dd)

input(jam_akhir.jj, jam_akhir.mm, jam_akhir.dd)

,jam_selisih=selisihJam(jam_awal,jam_akhir)

output(jam_selisih,jj,jam_selisih.mm,jam_selisih.dd)

Function selisihJam(input jam1,jam2:jam):jam

{memproses jam1 dan jam2 sehingga dihasilkan selisih jam yaitu jam3 =jam2-jam1 }

Kamus Lokal:

selisih,sjam,detik_awal,detik_akhir:integer

Jam_hasil : jam

detik_awal= (Jam1.jjX3600)+(Jam1.mmx60)+Jam1.dd

detik_akhir= (Jam2.jjX3600)+(Jam2.mmx60)+Jam2.dd

selisih=detik_akhir-detik_awal

Jam_hasil.jj=selisih/3600

sjam=selisih mod 3600

jam_hasil.mm=sjam/60

jam_hasil.dd=sjam mod 60

→ jam_hasil

```
27 | return jam_hasil;  
28 | }
```

LATIHAN5

Buat *notasi algoritma* (*dengan translasinya*) untuk program menghitung jarak dengan masukan berupa point P1 dan P2 (yang memiliki [absis,ordinat]), dengan menggunakan fungsi hitungJarak yang memproses P1 dan P2 sehingga dihasilkan

$$\text{jarak} = \sqrt{(P1.x - P2.x)^2 + (P1.y - P2.y)^2}$$