

PROSEDUR

DASAR PEMROGRAMAN

suprayogi@dsn.dinus.ac.id

PROSEDUR

Prosedur adalah sederetan instruksi algoritmik yang diberi nama, dan akan menghasilkan efek neto yang terdefinisi

Mendefinisikan prosedur

- menentukan nama prosedur beserta parameternya (jika ada)
- Mendefinisikan keadaan awal (initial state) dan keadaan akhir (final state)
- Mendefinisikan prosedur tersebut dalam kamus

PARAMETER FORMAL

Parameter Formal: nama-nama variabel (list nama) yang dipakai dalam mendefinisikan prosedur, dan membuat prosedur tersebut dapat dilaksanakan dengan nama-nama yang berbeda ketika dipanggil.

Parameter formal adalah list nama yang akan dipakai pada prosedur, yang nantinya akan diasosiasikan terhadap nama variabel lain pada saat pemanggilan

Sesuai dengan ketentuan nilainya, ada 3 type:

- **Parameter Input:** parameter yang diperlukan prosedur sebagai masukan untuk melakukan aksi yang efektif
- **Parameter Output:** parameter yang nilainya akan dihasilkan oleh prosedur. Hasil nilai akan disimpan pada parameter output ini.
- **Parameter Input/Output:** parameter yang nilainya diperlukan prosedur sebagai masukan untuk melakukan aksi, dan pada akhir prosedur akan dihasilkan nilai yang baru

PARAMETER AKTUAL

Parameter Aktual: nama-nama informasi yang dipakai ketika prosedur itu dipakai (“dipanggil”).

- Parameter aktual dapat berupa nama atau harga, tetapi harus berupa nama jika parameter tersebut adalah parameter output (karena hasilnya akan disimpan dalam nama tersebut)
- Sesuai dengan jenis parameter formal, parameter aktual pada saat pemanggilan:
 - Parameter Input harus terdefinisi nilainya (karena dibutuhkan oleh prosedur untuk menghasilkan nilai)
 - Parameter Output tidak perlu terdefinisi nilainya, tetapi justru setelah pemanggilan prosedur akan dimanfaatkan oleh deretan instruksi berikutnya, karena nilainya akan dihasilkan oleh prosedur
 - Parameter Input/Output harus terdefinisi nilainya dan nilai baru yang diperoleh karena eksekusi prosedur akan dimanfaatkan oleh deretan instruksi berikutnya

PEMANGGILAN PROSEDUR

Memakai atau “memanggil” prosedur adalah menuliskan nama prosedur yang pernah didefinisikan, dan memberikan harga-harga yang dibutuhkan oleh prosedur itu untuk dapat melaksanakan suatu aksi tertentu

Sebuah prosedur juga boleh “memakai” atau memanggil prosedur

Pada saat eksekusi, terjadi asosiasi nama parameter formal dengan nama parameter aktual

Pada notasi algoritmik, asosiasi dilakukan dengan cara “by position”, urutan nama parameter aktual akan diasosiasikan sesuai dengan urutan parameter formal. Karena itu, type harus kompatibel.

KAMUS LOKAL

Prosedur dapat mempunyai kamus lokal, yaitu pendefinisian nama yang dipakai dan hanya berlaku dalam ruang lingkup prosedur tersebut

Jika nama yang dipakai di dalam prosedur tidak terdefinisi dalam list parameter formal atau dalam kamus lokal, maka nama tersebut harus sudah terdefinisi pada prosedur yang memakainya

Penulisan kamus lokal = kamus global, bedanya adalah lingkup berlakunya nama yang didefinisikan

- – Pada kamus “global”, nama berlaku untuk program dan semua prosedur/fungsi yang didefinisikan
- – Pada kamus lokal, nama berlaku untuk prosedur/fungsi yang bersangkutan dan prosedur/fungsi yang didefinisikan di dalamnya

PROGRAM MODULAR

Adalah program yang dibagi-bagi menjadi modul-modul yang terdefinisi dengan baik dalam bentuk prosedur-prosedur

Setiap prosedur harus jelas definisi dan ruang lingkungannya, supaya dapat dipanggil secara independen

Pembagian program besar dalam prosedur-prosedur akan mempermudah pembagian kerja di antara beberapa pemrogram

Penulisan prosedur juga akan memudahkan program untuk dibaca oleh “manusia” karena kita tidak perlu terpaku pada detail kode prosedur untuk mengerti efek neto yang dihasilkannya

Dalam beberapa hal, pemrogram tidak perlu tahu sama sekali “isi” atau kode dari prosedur dengan mengetahui spesifikasinya

Beberapa bahasa pemrograman menyediakan prosedur terdefinisi yang sering dipakai dalam memrogram sehingga pemrogram tidak perlu lagi menuliskan kodenya

PROSEDUR TANPA PARAMETER

Program Voltase

{program yang membaca tahanan dan arus, menghitung voltase dan mencetak hasil perhitungan}

Kamus:

R:integer {tahanan dalam ohm}

A:integer {arus dalam ampere}

V:integer {tegangan dalam volt}

Procedure hitungVoltase

{ prosedur untuk memproses tahanan dan arus menjadi tegangan }

Algoritma:

Input (R,A)

hitungVoltase

output(V)

Procedure hitungVoltase

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

{F.S:memproses R dan A sehingga dihasilkan voltase(V) yaitu $V=R \times A$ }

Kamus Lokal:

Algoritma:

$V = R \times A$

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //Program Voltase
4  //{program yang membaca tahanan dan arus, menghitung voltase dan mencetak hasil perhitungan}
5  //kamus global
6  int R; //{tahanan dalam ohm}
7  int A; //{arus dalam ampere}
8  int V; //{tegangan dalam volt}
9  void hitungVoltase(); //{ prosedur untuk memproses tahanan dan arus menjadi tegangan }
10 int main()
11 {
12     printf("R:");scanf("%d",&R);
13     printf("A:");scanf("%d",&A);
14     hitungVoltase();
15     printf("V=%d\n",V);
16     return 0;
17 }
18 void hitungVoltase(){
19     //{I.S:diberikan harga R dan A yang telah terdefinisi }
20     //{F.S:memproses R dan A sehingga dihasilkan voltase(V) yaitu V=R*A }
21     //kamus lokal
22
23     //algoritma
24     V=R*A;
25 }
26
```

```
D:\pengajaran\DasarPemrograman\prosedur1\bin\Debug\prosedur1.exe
R:10
A:13
V=130
Process returned 0 (0x0)   execution time : 2.995 s
Press any key to continue.
```


PROSEDUR DENGAN PARAMETER

Program Voltase

{program yang membaca tahanan dan arus, menghitung voltase dan mencetak hasil perhitungan}

Kamus:

R:integer {tahanan dalam ohm}

A:integer {arus dalam ampere}

V:integer {tegangan dalam volt}

Procedure hitungVoltase(input R,A:integer;output

V:integer)

{prosedur untuk memproses tahanan R dan arus A menjadi tegangan V}

Input (R,A)

hitungVoltase(R,A,V)

output(V)

Procedure hitungVoltase(input R,A:integer;output

V:integer)

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

{F.S:memproses R dan A sehingga dihasilkan voltase(V) yaitu $V=R \times A$ }

Kamus Lokal:

$V = R \times A$

```
main.c X main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3  //Program Voltase
4  //{program yang membaca tahanan dan arus, menghitung voltase dan mencetak hasil perhitungan}
5  void hitungVoltase (); //{ prosedur untuk memproses tahanan dan arus menjadi tegangan }
6  int main()
7  {
8      //kamus global
9      int R; //(tahanan dalam ohm)
10     int A; //(arus dalam ampere)
11     int V; //(tegangan dalam volt)
12     //algoritma
13     printf("R:");scanf("%d",&R);
14     printf("A:");scanf("%d",&A);
15     hitungVoltase(R,A,&V);
16     printf("V=%d\n",V);
17     return 0;
18 }
19 void hitungVoltase(int R,int A,int *V){
20     //{I.S:diberikan harga R dan A yang telah terdefinisi }
21     //{F.S:memproses R dan A sehingga dihasilkan voltase(V) yaitu  $V=R \times A$  }
22     //kamus lokal
23     //algoritma
24     *V=R*A;
25 }
```

```
D:\pengajaran\DasarPemrograman\prosedur1\bin\Debug\prosedur1e
R:13
A:10
V=130
Process returned 0 (0x0)   execution time : 4.985
Press any key to continue.
```

LATIHAN

Translasikan program menghitung kecepatan dengan masukan berupa S (jarak dalam Meter) dan t (waktu dalam detik) yang menggunakan prosedur hitungKecepatan dengan

I.S: diberikan harga t dan S yang telah terdefinisi , dan

F.S:memproses t dan S sehingga dihasilkan kecepatan(V) yaitu $V=S/t$.

```
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}

Procedure hitungKecepatan(input S,t:real;output V:real)
{prosedur untuk memproses jarak S dan waktu t menjadi
kecepatan V}

Input (S,t)

hitungKecepatan(S,t,V)

output(V)

```
Procedure hitungKecepatan(input S,t:real;output V:real)
```

```
{I.S:diberikan harga S dan t yang telah terdefinisi }
```

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

Kamus Lokal:

$V= S/t$

LATIHAN

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

I.S: diberikan harga C dan kode_konversi yang telah terdefinisi , dan

F.S:memproses C dan kode_konversi sehingga

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

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

I.S: diberikan harga Jam_awal dan Jam_akhir yang telah terdefinisi, dan

F.S:memproses jam_awal dan jam_akhir sehingga dihasilkan jam_selisih yaitu:

$jam_selisih = jam_akhir - jam_awal$

Program Selisih Jam

Kamus:

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

jam_awal,jam_akhir,jam_selisih : jam

Procedure selisihJam(input jam1,jam2:jam;output jam3: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)

selisihJam(jam_awal,jam_akhir,jam_selisih)

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

Procedure selisihJam(input jam1,jam2:jam;output jam3:jam)

{I.S:diberikan harga jam1 dan jam2 yang telah terdefinisi }

{F.S:memproses jam1 dan jam2 sehingga dihasilkan selisih jam yaitu jam3 =jam2-jam1 }

Kamus Lokal:

selisih,sjam,detik_awal,detik_akhir:integer

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

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

selisih=detik_akhir-detik_awal

jam3.jj=selisih/3600

sjam=selisih mod 3600

jam3.mm=sjam/60

jam3.dd=sjam mod 60

LATIHAN

Buat notasi algoritma untuk program menghitung jarak dengan masukan berupa point P1 dan P2 (yang memiliki [absis,ordinat]), dengan menggunakan prosedur hitungJarak dengan:

I.S: diberikan harga P1 dan P2 yang telah terdefinisi , dan

F.S:memproses P1 dan P2 sehingga dihasilkan

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