

PERTEMUAN 11
PROSEDUR

Algoritma dan Pemrograman I

Teknik Informatika

UNIKOM

Prosedur dan struktur prosedur

PROSEDUR :

Modul program yang mengerjakan tugas / aktivitas yang spesifik dan menghasilkan suatu efek netto.

Suatu efek netto diketahui dengan membandingkan keadaan awal dan keadaan akhir pada pelaksanaan sebuah prosedur.

Mendefinisikan prosedur :

- **Menuliskan nama prosedur**
- **Mendeklarasikan nama-nama konstanta, peubah (variabel) dan tipe (jika ada).**
- **Menjabarkan rangkaian yang dilakukan.**

Prosedur dan struktur prosedur (lanjutan)

Struktur Prosedur :

- **Bagian Judul (*header*)**

  Terdiri atas Nama prosedur dan deklarasi parameter (jika ada).

- **Bagian deklarasi (kamus)**

  Untuk mengumumkan nama-nama.

- **Bagian deskripsi (badan algoritma)**

  Disebut juga badan prosedur.

Nama prosedur sebaiknya diawali dengan kata kerja karena prosedur berisi suatu aktivitas, misalnya HitungLuas, Tukar, CariMaks, Inisialisasi, AktifkanMenu, dan lain sebagainya.

Prosedur yang baik adalah

- Prosedur yang *independent* dari program pemanggilannya. Prosedur yang tidak menggunakan peubah- peubah global di dalam badan prosedurnya.
- Jika program utama perlu mengkomunikasikan nilai peubah global ke dalam prosedur, maka ada satu cara untuk melakukannya yaitu dengan menggunakan parameter.

Parameter

Parameter adalah :Nama- nama peubah yang dideklarasikan pada bagian *header* prosedur

1. **Parameter aktual (argument) adalah :**
   Parameter yang disertakan pada waktu pemanggilan prosedur.
2. **Parameter formal adalah :**
   Parameter yang dideklarasikan di dalam bagian *header* prosedur itu sendiri.

Parameter actual dan Parameter formal

Aturannya :

- Jumlah parameter aktual pada pemanggilan prosedur harus sama dengan jumlah parameter formal pada deklarasi rosedurnya.
- Tiap parameter aktual harus bertipe sama dengan tipe parameter formal yang bersesuaian.
- Tiap parameter aktual harus diekspresikan dalam cara yang taat-asas dengan parameter formal yang bersesuaian, bergantung pada jenis parameter formal.

Tiga Jenis Parameter formal

1. Parameter masukan (Input parameter).

   Parameter yang nilainya berlaku sebagai masukan untuk prosedur.

   Dalam bahasa pemrograman, istilah parameter masukan ini sering dinamakan **parameter nilai** (*value parameter atau parameter by value*).

2. Parameter keluaran (Output parameter).

   Parameter yang menampung keluaran yang dihasilkan oleh prosedur, parameter ini sering juga disebut parameter referensi (*parameter by reference*).

3. Parameter masukan / keluaran (Input/Output parameter).

   Parameter yang berfungsi sebagai masukan sekaligus keluaran bagi prosedur tersebut. Ini pun termasuk ke dalam *parameter by reference*.

Program dengan Prosedur atau Tanpa Prosedur?
Sangat dianjurkan menulis program yang modular.
Program yang dipecah menjadi beberapa prosedur
(fungsi) menunjukan teknik pemrograman yang baik
dan terstruktur.

Prosedur dengan Parameter atau Tanpa Parameter?
Sangat dianjurkan menulsikan **prosedur dengan parameter**.
Parameter berfungsi sebagai media komunikasi antara modul
dengan program pemanggil. Selain itu, parameter dapat mengurangi
kebutuhan penggunaan peubah global.

Parameter Masukan atau Parameter Keluaran?
Tergantung pada kasus yang ada.

Procedure pada Algoritma

- Prosedur Dengan Parameter Masukan
- Prosedur Dengan Parameter Keluaran
- Prosedur Dengan Parameter Masukan/Keluaran

Prosedur Dengan Parameter Masukan Pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

```
Kamus :
    Procedure  namaprosedur(input namaparameter : type)
Algoritma :
    namaprosedur(parameter_aktual)
```

2. Cara mendifinisikan:

```
Procedure  namaprosedur(input namaparameter : type)
{ I.S.(kondisi awal) : … }
{ F.S.(kondisi akhir) : … }
Kamus :
    {deklarasi/kamus lokal}
Algoritma :
    {aksi-aksi dalam satu prosedur}
EndProcedure
```

Prosedur Dengan Parameter Keluaran pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

   Kamus :

   Procedure namaprosedur(output namaparameter : type)

   Algoritma :

   namaprosedur(parameter_aktual)

2. Cara mendifinisikan:

   Procedure *n*amaprosedur(output namaparameter : type)

   { I.S.(kondisi awal) : … }

   { F.S.(kondisi akhir) : … }

   Kamus :

   {deklarasi/kamus lokal}

   Algoritma :

   {aksi-aksi dalam satu prosedur}

   EndProcedure

Prosedur Dengan Parameter Masukan/Keluaran Pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

```
Kamus :
    Procedure  namaprosedur(input/output namaparameter : type)
Algoritma :
    namaprosedur(parameter_aktual)
```

2. Cara mendifinisikan:

```
Procedure  namaprosedur(input/output namaparameter : type)
{ I.S.(kondisi awal) : … }
{ F.S.(kondisi akhir) : … }
Kamus :
    {deklarasi/kamus lokal}
Algoritma :
    {aksi-aksi dalam satu prosedur}
EndProcedure
```

1. Contoh translasi prosedur dengan parameter masukan

```
procedure HitungLuasSegitiga(input alas, tinggi : real)
{ I.S. : alas dan tinggi sudah terdefinisi nilainya }
{ F.S. : menampilkan luas segitiga  }
Kamus :
  luas      : real      { luas segitiga, dalam cm2 }

Algoritma :
  luas ← (alas * tinggi)/2
  output(luas)

EndProcedure
```

Algoritma utama yang memanggil prosedur HitungLuasSegitiga

ALGORITMA :
{Algoritma utama}

```
Algoritma_Segitiga
{ Menghitung luas N buah segitiga. }
Kamus :
  i, N : integer
  a, t : real
  Procedure HitungLuasSegitiga(input alas, tinggi : real)
  { Menghitung luas segitiga dengan rumus L =(alas x tinggi)/2 }
Algoritma :
  input (N)          { tentukan banyaknya segitiga }
  for  i ← 1  to  N  do
          input (a,t)
          HitungLuasSegitiga(a,t)
  endfor

EndProcedure
```

2. Contoh translasi prosedur dengan parameter keluaran

```
Procedure HitungLuasSegitiga(input alas, tinggi : real,  output luas : real)
{ I.S. : alas dan tinggi sudah terdefinisi nilainya }
{ F.S. : menghasilkan luas segitiga. }
Kamus :
  { tidak ada }

Algoritma :
  luas ← (alas * tinggi)/2

EndProcedure
```

Algoritma utama yang memanggil prosedur HitungLuasSegitiga

ALGORITMA :
{Algoritma utama}

```
Algoritma_Segitiga
{ Menghitung luas N buah segitiga. }
Kamus :
  i, N : integer
  a, t : real
  L        : real  { luas segitiga }
  Procedure HitungLuasSegitiga(input alas, tinggi : real , output luas : real)
Algoritma :
  input (N)          { tentukan banyaknya segitiga }
  for i ← 1 to N do
          input (a,t)
          HitungLuasSegitiga(a,t,L)
          output (L)
  endfor

EndProcedure
```

3. Contoh translasi prosedur dengan parameter masukan / keluaran

```
Procedure Tukar(input/output A,B : integer)
{ I.S. : nilai A dan B sudah terdefinisi. }
{ F.S : A berisi nilai B, B berisi nilai A semula. }
Kamus :
  temp : integer              { peubah Bantu }

Algoritma :
  temp ← A          { simpan nilai A ke dalam temp }
  A ← B             { isikan nilai B ke dalam A }
  B ← temp          { isikan nilai temp ke dalam B }

EndProcedure
```

Algoritma utama yang memanggil prosedur Tukar

{ Algoritma utama}

```
Algoritma_SelisihXY
{Prog. untuk menghitung selisih nilai X dan Y,dengan syarat X ≥ Y jika X < Y,ma
 ka X dan Y dipertukarkan dengan memanggil prosedur Tukar. }
Kamus :
  X, Y, Z : integer
  Procedure Tukar(input/output A,B : integer)
  { mempertukarkan nilai A dan B }

Algoritma :
  input(X,Y)          { baca nilai X dan Y terlebih dahulu }
  {Jika X < Y, pertukaran nilai X dan Y dengan memanggil prosedur Tukar }
  If (X < Y)
    then
          Tukar(X,Y)
  endif
  Z ← X - Y { hitung selisih x dan Y }
  output(Z)

EndProcedure
```

Tugas

Buat Algoritma untuk menghitung faktorial dengan menggunakan dua prosedur (prosedur memasukkan harga N dan prosedur menampilkan faktorial dari N) beserta pengendalinya (algoritma utama)