PERTEMUAN 12
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 dan tipe (jika ada).**
- **Menjabarkan rangkaian yang dilakukan.**

Prosedur dan struktur prosedur



Struktur Prosedur :

- **Bagian Judul (header).**
  Terdiri atas Nama prosedur dan deklarasi parameter (jika ada).
- **Bagian deklarasi.**
  Untuk mengumumkan nama-nama**.**
- **Bagian deskripsi.**
  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 dan struktur prosedur



Prosedur yang baik adalah

- Prosedur yang *independent* dari program pemanggilannya. Prosedur yang tidak menggunakan peubah- peubah global didalam 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 actual (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 actual pada pemanggilan prosedur harus sama dengan jumlah parameter formal pada deklarasi prosedurnya.
- Tiap parameter actual harus bertipe sama dengan tipe parameter formal yang bersesuaian.
- Tiap parameter actual harus diekspresikan dalam cara yang taat-asas dengan parameter formal yang bersesuaian, bergantung pada jenis parameter formal.

3 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.
3. Parameter masukan / keluaran (Input/Output parameter).
   Parameter yang berfungsi sebagai masukan sekaligus keluaran bagi prosedur tersebut.

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



1. Prosedur Tanpa Parameter
2. Prosedur Dengan Parameter Masukan
3. Prosedur Dengan Parameter Keluaran
4. Prosedur Dengan Parameter Masukan/Keluaran

Prosedur Tanpa Parameter



1. Cara memanggil (dari algoritma/program utama)

   DEKLARASI

   Procedure namaprosedur

   DESKRIPSI

   namaprosedur
2. Cara mendifinisikan:

   Procedure namaprosedur

   { kondisi awal : … }

   { kondisi akhir : … }

   deklarasi lokal

   deskripsi

Prosedur Dengan Parameter Masukan Pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

   DEKLARASI

   Procedure namaprosedur(input namaparameter : type)

   DESKRIPSI

   namaprosedur(namaparameter)

2. Cara mendifinisikan:

   Procedure namaprosedur(input namaparameter : type)

   { kondisi awal : … }

   { kondisi akhir : … }

   deklarasi lokal

   deskripsi

Prosedur Dengan Parameter Keluaran Pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

   DEKLARASI

   Procedure namaprosedur(output namaparameter : type)

   DESKRIPSI

   namaprosedur(namaparameter)

2. Cara mendifinisikan:

   Procedure namaprosedur(output namaparameter : type)

   { kondisi awal : … }

   { kondisi akhir : … }

   deklarasi lokal

   deskripsi

Prosedur Dengan Parameter Masukan/Keluaran Pada Algoritma

1. Cara memanggil (dari algoritma/program utama)

   DEKLARASI

   Procedure namaprosedur(input/output namaparameter : type)

   DESKRIPSI

   namaprosedur(namaparameter)

2. Cara mendifinisikan:

   Procedure namaprosedur(input/output namaparameter : type)

   { kondisi awal : … }

   { kondisi akhir : … }

   deklarasi lokal

   deskr

Aturan Translasi Notasi algoritma prosedur ke dalam Notasi bahasa C++

1. Pendefinisian prosedur ditulis di **luar blok program utama (*main*).**
2. Dalam bahasa C, tidak dikenal istilah prosedur.
3. Dalam bahasa C, bila prosedur tidak mempunyai parameter, maka tanda kurung ‘(‘ dan ‘)’ tetap ditulis setelah setelahnamaprosedur tersebut
4. Semua parameter dalam bahasa C adalah parameter masukan

ALGORITMA

```
procedure NamaProsedur(deklarasi parameter,jika ada)
{spesifikasi prosedur,berisi penjelasan tentang apa yang dilakukan
oleh prosedur ini.
K.Awal  : keadaan sebelum prosedur dilaksanakan.
K.Akhir : kedaan setelah prosedur dilaksanakan. }
DEKLARASI
 { semua nama yang dipakai didalam prosedur dan hanya berlaku
   local didalam prosedur didefinisikan disini }
DESKRIPSI :
   { badan prosedur, barisi urutan instruksi}
```

- Cara memanggil prosedur :
  NamaProsedur
  Jika prosedur tidak mempunyai parameter,
- atau
  NamaProsedur(parameter actual)
  Jika prosedur mengandung parameter.

BAHASA C++

```
void NamaProsedur(deklarasi parameter,jika ada)
/* Spesifikasi prosedur, berisi penjelasan tentang apa yang dila
      kukan oleh prosedur ini. */
/* K.Awal  : keadaan sebelum prosedur dilaksanakan. */
/* K.Akhir : keadaan setelah prosedur dilaksanakan. */
{
 /* DEKLARASI */
 /* semua nama yang dipakai dalam prosedur dan hanya berlaku lok
    al di dalam prosedur didefinisikan di sini */
 /* DESKRIPSI */
 /* badan prosedur, berisi kumpulan instruksi */
}
```

- Cara memanggil prosedur :
  NamaProsedur
  Jika prosedur tidak mempunyai parameter,
- atau
  NamaProsedur(parameter actual)
  Jika prosedur mengandung parameter.

1.Contoh translasi prosedur tanpa parameter

Algoritma :
prosedur

```
procedure HitungLuasSegitiga
{ Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2 }
{ K.Awal  : sembarang }
{ K.Akhir : luas segitiga tercetak. }
DEKLARASI
  alas    : real    { panjang alas segitiga, dalam cm }
  tinggi  : real    { tinggi segitiga, dalam cm }
  luas    : real    { luas segitiga, dalam cm2 }
DESKRIPSI
  read(alas,tinggi)
  luas ← (alas * tinggi)/2
  write(luas)
```


1.Contoh translasi prosedur tanpa parameter

Algoritma :
Program utama

```
Algoritma Segitiga
{ Menghitung luas N buah segitiga. }
DEKLARASI
  i, N : integer
  procedure HitungLuasSegitiga
  { Menghitung luas segitiga dengan rumus L =(alas x tinggi)/2 }
DESKRIPSI:
  read(N)          { tentukan banyaknya segitiga }
  for i ← 1 to N do
          HitungLuasSegitiga
  endfor
```

1.Contoh translasi prosedur tanpa parameter

Bahasa C / C++ dengan include <stdio.h> :

```
#include <stdio.h>
void HitungLuasSegitiga();
main()
{
  int i, N;
  printf("banyaknya segitiga? "); scanf("%d",&N);
  for(i=1;i<=N;i++)
       HitungLuasSegitiga();
}
void HitungLuasSegitiga()
{
  float alas;
  float tinggi;
  float luas;
  printf("\nPanjang alas segitiga? "); scanf("%f",&alas);
  printf("Tinggi segitiga? "); scanf("%f",&tinggi);
  luas = ( alas * tinggi )/2.0;
  printf("Luas segitiga = %5.2f\n",luas);
}
```


1.Contoh translasi prosedur tanpa parameter

Bahasa C / C++ dengan include <iostream.h> :

```
#include <iostream.h>
void HitungLuasSegitiga();
void main()
{
   int i, N;
   cout << "banyaknya segitiga? ";
   cin >> N;
   for(i=1;i<=N;i++)
     HitungLuasSegitiga();
}
void HitungLuasSegitiga()
{
   float alas;
   float tinggi;
   float luas;
   cout << "\nPanjang alas segitiga? ";
   cin >> alas ;
   cout << "Tinggi segitiga? ";
   cin >> tinggi;
   luas = (alas * tinggi)/2.0;
   cout << "Luas = " << luas << endl;
}
```

2.Contoh translasi prosedur dengan parameter masukan

ALGORITMA :
Prosedur

```
procedure HitungLuasSegitiga(input alas, tinggi : real)
{ Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2 }
{ K.Awal  : alas dan tinggi sudah terdefinisi nilainya }
{ K.Akhir : luas segitiga tercetak. }
DEKLARASI
  luas     : real     { luas segitiga, dalam cm2 }
DESKRIPSI
  luas ← (alas * tinggi)/2
  write(luas)
```


2.Contoh translasi prosedur dengan parameter masukan

ALGORITMA :
Program utama

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

2.Contoh translasi prosedur dengan parameter masukan

Bahasa C / C++ dengan menggunakan #include <stdio.h>

```
#include <stdio.h>
void HitungLuasSegitiga(float a, float t);
main()
{
  int i, N;
  float a, t;
  printf("banyaknya segitiga? "); scanf("%f",&N);
  for(i=1;i<=N;i++)
   {
      printf("\nPanjang alas segitiga? "); scanf("%f",&a);
      printf("Tinggi segitiga? "); scanf("%f",&t);
      HitungLuasSegitiga(a,t);
   }
}
void HitungLuasSegitiga(float alas, float tinggi)
{
  float luas;
  luas = ( alas * tinggi )/2.0;
  printf("Luas segitiga = %f\n",luas);
}
```


2.Contoh translasi prosedur dengan parameter masukan

Bahasa C / C++ dengan menggunakan #include <iostream.h>

```
#include <iostream.h>
void HitungLuasSegitiga(float a, float t);
void main()
{
   int i, N;
   float a, t;
   cout << "banyaknya segitiga? ";
   cin >> N;
   for(i=1;i<=N;i++)
    {
       cout << "\nPanjang alas segitiga? ";
       cin >> a;
       cout << "Tinggi segitiga? ";
       cin >> t;
       HitungLuasSegitiga(a,t);
    }
}
void HitungLuasSegitiga(float alas, float tinggi)
{
  float luas;
  luas = ( alas * tinggi )/2.0;
  cout << "Luas=" << luas << endl;
}
```

3. Contoh translasi prosedur dengan parameter keluaran

ALGORITMA :
Prosedur

```
procedure HitungLuasSegitiga(input alas, tinggi : real,
  output luas : real)
{ Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2 }
{ K.Awal  : alas dan tinggi sudah terdefinisi nilainya }
{ K.Akhir : luas berisi luas segitiga. }
DEKLARASI
  { tidak ada }
DESKRIPSI
  luas ← (alas * tinggi)/2
```


3. Contoh translasi prosedur dengan parameter keluaran

ALGORITMA :
Program utama

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

3. Contoh translasi prosedur dengan parameter keluaran

Bahasa C / C++ dengan menggunakan #include <stdio.h>

```
#include <stdio.h>
void HitungLuasSegitiga(float a, float t, float *L);
main()
{
  int i, N;
  float a, t;
  float L;
  printf("banyaknya segitiga? "); scanf("%d",&N);
  for(i=1;i<=N;i++)
    {
      printf("\nPanjang alas segitiga? "); scanf("%f",&a);
      printf("Tinggi segitiga? "); scanf("%f",&t);
      HitungLuasSegitiga(a,t,&L);
      printf("Luas segitiga = %5.2f\",L);
    }
}
void HitungLuasSegitiga(float alas, float tinggi, float *luas)
{
   *luas = ( alas * tinggi )/2.0;
}
```


3. Contoh translasi prosedur dengan parameter keluaran

Bahasa C / C++ dengan menggunakan #include <iostream.h>

```
#include <iostream.h>
void HitungLuasSegitiga(float a, float t, float *L);
void main()
{
  int i, N;
  float a, t;
  float L;
  cout << "banyaknya segitiga? ";
  cin >> N;
  for(i=1;i<=N;i++)
    {
         cout << "\nPanjang alas segitiga? ";
         cin >> a;
         cout << "Tinggi segitiga? ";
         cin >> t;
         HitungLuasSegitiga(a,t, &L);
         cout << "Luas segitiga = " << L <<endl;
    }
}
void HitungLuasSegitiga(float alas, float tinggi, float *luas)
{
  *luas = ( alas * tinggi )/2.0;
}
```

4. Contoh translasi prosedur dengan parameter masukan / keluaran

Algoritma :
prosedur

```
procedure Tukar(input/output A,B : integer)
{ Mempertukarkan nilai A dan B. }
{ K.Awal  : nilai A dan B sudah terdefinisi. }
{ K.Akhir : A berisi nilai B, B berisi nilai A semula. }
DEKLARASI
  temp : integer              { peubah Bantu }
DESKRIPSI
  temp ← A          { simpan nilai A ke dalam temp }
  A ← B             { isikan nilai B ke dalam A }
  B ← temp          { isikan nilai temp ke dalam B }
```


4. Contoh translasi prosedur dengan parameter masukan / keluaran

Algoritma :
Program 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. }
DEKLARASI
  X, Y, Z : integer
  procedure Tukar(input/output A,B : integer)
  { mempertukarkan nilai A dan B }
DESKRIPSI:
  read(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 }
  write(Z)
```

4. Contoh translasi prosedur dengan parameter masukan / keluaran

Bahasa C / C++ dengan menggunakan #include <stdio.h>

```
#include <stdio.h>
void Tukar(int *X, int *Y)
main()
{
    int X, Y, Z;
    printf("X = ?"); scanf("%d",&X);
    printf("Y = ?"); scanf("%d",&Y);
    if(X < Y)
    Tukar(&X,&Y);
    Z = X – Y;
    printf("Z = %d \n",Z);
}
void Tukar(int *A, int *B)
{
    int temp;
    temp = *A;
    *A = *B;
    *B = temp;
}
```


4. Contoh translasi prosedur dengan parameter masukan / keluaran

Bahasa C / C++ dengan menggunakan #include <iostream.h>

```
#include <iostream.h>
void Tukar(int *X, int *Y);
void main()
{
   int X, Y, Z;
   cout << "X = ?"; cin >> X;
   cout << "Y = ?"; cin >> Y;
   if(X < Y)
      Tukar(&X,&Y);
   Z = X – Y;
   cout << "Z = " << Z;
}
void Tukar(int *A, int *B)
{
  int temp;
  temp = *A;
  *A = *B;
  *B = temp;
}
```