

Algoritma dan Pemrograman

Pertemuan Ke-10 Prosedur



**Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.**

**Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional “Veteran”
Yogyakarta**

Algoritma dan Pemrograman

Judul Materi : Prosedur dan fungsi

Deskripsi Materi : Materi ini membahas procedure dan parameter-parameter yang terlibat.

Tujuan Instruksional Khusus :

1. Menjelaskan pemrograman modular
2. Menjelaskan perbedaan procedure dan function
3. Mendeskripsikan parameter pada procedure dan function
4. Membandingkan parameter input dan parameter output pada procedure dan function

BAB VIII

PROSEDUR

1. PENDAHULUAN

Bab 9 dan 10 memiliki kompetensi dasar untuk memahami dan mengimplementasikan procedure dan function dengan memperhatikan parameter dan variabel-variabel yang akan dipakai.

Ketika kita ingin membuat suatu program yang kompleks, kita dapat memanfaatkan konsep pemrograman modular, yaitu masalah dibagi-bagi permodul untuk tiap permasalahan yang lebih spesifik. Procedure dan function di dalam algoritma memiliki perbedaan. Tetapi di bahasa pemrograman C++ semua disebut function dengan perbedaan sintak untuk membedakan antara fungsi sebagai procedure dan sebagai function yang dimaksud pada algoritma.

2. PENYAJIAN

2.1. Prosedur Dan Struktur Prosedur

Prosedur adalah :

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. Pada prosedur hasil akhir bergantung pada return.

Mendefinisikan prosedur

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

Struktur Prosedur :

1. Bagian Judul (header).

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

2. Bagian deklarasi.

Untuk mengumumkan nama-nama.

3. Bagian deskripsi.

Disebut juga **badan prosedur**.

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

Parameter adalah :

Nama- nama peubah yang dideklarasikan pada bagian *header* prosedur.

Penggunaan parameter menawarkan mekanisme pertukaran informasi antar prosedur (atau fungsi). Tiap item data ditransfer antara **Parameter actual** dan **Parameter formal** yang bersesuaian.

Parameter actual (argument) adalah :

Parameter yang disertakan pada waktu pemanggilan prosedur.

Parameter formal adalah :

Parameter yang dideklarasikan di dalam bagian *header* prosedur itu sendiri.

Ketika prosedur dipanggil, parameter actual menggantikan parameter formal. Tiap-tiap parameter actual berpasangan dengan parameter formal yang bersesuaian.

Cara pemanggilan prosedur dengan parameter adalah :

NamaProsedur(*parameter aktual*)

pendeklarasian parameter di dalam prosedur bukanlah keharusan. Dengan kata lain **parameter boleh ada atau tidak ada**.

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.

Aturan yang harus diperhatikan dalam korespondensi satu-satu antara parameter actual dan parameter formal adalah :

1. Jumlah parameter actual pada pemanggilan prosedur harus sama dengan jumlah parameter formal pada deklarasi prosedurnya.
2. Tiap parameter actual harus bertipe sama dengan tipe parameter formal yang bersesuaian.
3. Tiap parameter actual harus diekspresikan dalam cara yang taat-asas dengan parameter formal yang bersesuaian, bergantung pada jenis parameter formal.

Berdasarkan maksud penggunaannya, ada 3 jenis parameter formal yang disertakan ke dalam prosedur :

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) menunjukkan teknik pemrograman yang baik dan terstruktur.

Prosedur dengan Parameter atau Tanpa Parameter?

Sangat dianjurkan menuliskan **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.

Bila prosedur menghasilkan keluaran yang digunakan oleh program pemanggil, gunakan **parameter keluaran** untuk menampung keluaran tersebut.

Bila prosedur tidak menghasilkan keluaran ataupun jika ada dan keluaran tersebut hanya digunakan didalam prosedur itu saja, maka gunakan **parameter masukan**.

Bila prosedur menerima masukan sekaligus pengeluaran pada parameter yang sama, gunakan **parameter masukan / keluaran**.

2. 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 pada Algoritma**1. Cara memanggil (dari algoritma/program utama)**

```

DEKLARASI
    Procedure namaprosedur
DESKRIPSI
    Namaprosedur
  
```

2. Cara mendefinisikan:

```

Procedure namaprosedur
{ kondisi awal : ... }
{ kondisi akhir : ... }

```

DEKLARASI LOKAL

DESKRIPSI

- Prosedur tanpa parameter hanya merupakan perintah untuk melaksanakan program, dan dalam pemberian tugas melaksanakan program, nama variabel tidak mempengaruhi jalannya program karena yang diperhatikan hanyalah nilai dan urutannya.

Prosedur Dengan Parameter Masukan Pada Algoritma**1. Cara memanggil:**

```

DEKLARASI
    Procedure namaprosedur(input namaparameter : type)
DESKRIPSI
    namaprosedur(namaparameter)

```

2. Cara mendefinisikan:

```

Procedure namaprosedur(input namaparameter : type)
{ kondisi awal : ... }
{ kondisi akhir : ... }

```

DEKLARASI LOKAL

DESKRIPSI

Prosedur Dengan Parameter Keluaran Pada Algoritma

1. Cara memanggil:

```
DEKLARASI
    Procedure namaprosedur(output namaparameter : type)
DESKRIPSI
    namaprosedur(namaparameter)
```

2. Cara mendefinisikan:

```
Procedure namaprosedur(output namaparameter : type)
{ kondisi awal : ... }
{ kondisi akhir : ... }
```

DEKLARASI LOKAL

DESKRIPSI

Prosedur Dengan Parameter Masukan/Keluaran Pada Algoritma

1. Cara memanggil:

```
DEKLARASI
    Procedure namaprosedur(input/output namaparameter:type)

DESKRIPSI
    namaprosedur(namaparameter)
```

2. Cara mendefinisikan:

```
Procedure    namaprosedur(input/output    namaparameter    :
type)
{ kondisi awal : ... }
{ kondisi akhir : ... }
```

DEKLARASI LOKAL

DESKRIPSI

3. Beberapa aturan Translasi Notasi algoritma prosedur ke dalam Notasi bahasa C++ :

1. Pendefinisian prosedur ditulis di **luar blok program utama (*main*)**. Kecuali jika direalisasikan sebagai *file include*. **Prosedur umumnya diletakkan sesudah blok *main()***. Sedangkan **deklarasi prosedur ditulis sebelum blok *main()*** sebagai purwarupa saja (purwarupa prosedur hanya berupa *header* dari prosedur tersebut), maka pendeklarasian purwarupa prosedur tidak diperlukan lagi.
2. Dalam bahasa C, tidak dikenal istilah prosedur. Semua modul program adalah fungsi. Prosedur adalah fungsi yang tidak mengembalikan nilai apapun. Karena itu, *return value* untuk prosedur adalah void, (yang artinya ‘kosong’).
3. Dalam bahasa C, bila prosedur tidak mempunyai parameter, maka tanda kurung ‘(‘ dan ‘)’ tetap ditulis setelah setelah nama prosedur tersebut.
4. Semua parameter dalam bahasa C adalah parameter masukan.

Oleh karena itu, semua argument parameter actual dilewatkan sebagai “*by value*”. Ini artinya nilai parameter actual disalin ke parameter formal yang bersesuaian.

Suatu cara untuk memperoleh efek parameter keluaran maupun efek parameter masukan / keluaran adalah melewatkan *pointer* terhadap parameter actual dengan menambahkan karakter “&” di awal nama parameter actual yang berjenis parameter keluaran atau masukan / keluaran.

Tanda “&” menyatakan alamat dari peubah.

Sedangkan parameter formal yang berjenis keluaran atau masukan / keluaran ditambahkan dengan karakter “*” didepannya.

Tanda “*” menyatakan operator *indirect*.

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 : keadaan setelah prosedur dilaksanakan. }

DEKLARASI
{ semua nama yang dipakai didalam prosedur dan hanya berlaku
local didalam prosedur didefinisikan disini }

DESKRIPSI :
{ badan prosedur, berisi 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  $L = (\text{alas} \times \text{tinggi})/2$  }
{ K.Awal : sembarang }
{ K.Akhir : luas segitiga tercetak. }

DEKLARASI
  alas      : riil      { panjang alas segitiga, dalam cm }
  tinggi    : riil      { tinggi segitiga, dalam cm }
  luas      : riil      { luas segitiga, dalam  $\text{cm}^2$  }

DESKRIPSI
  read(alas,tinggi)
  luas  $\leftarrow$  (alas * tinggi)/2
  write(luas)

```

Program utama :

```

Algoritma Segitiga
{ Menghitung luas N buah segitiga. }

DEKLARASI
  i, N : integer
  procedure HitungLuasSegitiga
  { Menghitung luas segitiga dengan rumus  $L = (\text{alas} \times \text{tinggi})/2$  }

DESKRIPSI:
  read(N) { tentukan banyaknya segitiga }
  for i  $\leftarrow$  1 to N do
    HitungLuasSegitiga
  endfor

```

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

```
/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga();

main()
{ /* DEKLARASI */
    int i, N;

    /* PROGRAM UTAMA */
    printf("banyaknya segitiga? "); scanf("%d",&N);
    for(i=1;i<=N;i++)
        HitungLuasSegitiga();
    /*endfor*/
    return 0;
}

void HitungLuasSegitiga()
/* Menghitung luas segitiga dengan rumus  $L = (\text{alas} \times \text{tinggi}) / 2$  */
/* K.Awal : sembarang */
/* K.Akhir : luas segitiga tercetak. */

{ /*DEKLARASI LOKAL*/
    float alas; /* panjang alas segitiga, dalam cm */
    float tinggi; /* tinggi segitiga, dalam cm */
    float luas; /* luas segitiga, dalam cm2 */

    /*DESKRIPSI*/
    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);
}
```

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

```
/* Program Segitiga */
/* Menghitung luas N buah segitiga. */

#include <iostream.h>

/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga();

void main()
{ /* DEKLARASI */
  int i, N;
  /* PROGRAM UTAMA */
  cout << "banyaknya segitiga? ";
  cin >> N;
  for(i=1;i<=N;i++)
    HitungLuasSegitiga();
  /*endfor*/
}

void HitungLuasSegitiga()
/* Menghitung luas segitiga dengan rumus  $L = (\text{alas} \times \text{tinggi}) / 2$  */
/* K.Awal : sembarang */
/* K.Akhir : luas segitiga tercetak. */

{ /*DEKLARASI*/
  float alas; /* panjang alas segitiga, dalam cm */
  float tinggi; /* tinggi segitiga, dalam cm */
  float luas; /* luas segitiga, dalam cm2 */
  /*DESKRIPSI*/
  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 : riil)
{ Menghitung luas segitiga dengan rumus  $L = (alas \times tinggi)/2$  }
{ K.Awal : alas dan tinggi sudah terdefinisi nilainya }
{ K.Akhir : luas segitiga tercetak. }

```

DEKLARASI

```

luas      : riil      { luas segitiga, dalam  $cm^2$  }

```

DESKRIPSI

```

luas  $\leftarrow$  (alas * tinggi)/2
write(luas)

```

Program utama :

Algoritma Segitiga

```

{ Menghitung luas N buah segitiga. }

```

DEKLARASI

```

i, N : integer

```

```

a, t : riil

```

```

procedure HitungLuasSegitiga(input alas, tinggi : riil)
{ Menghitung luas segitiga dengan rumus  $L = (alas \times tinggi)/2$  }

```

DESKRIPSI:

```

read(N) { tentukan banyaknya segitiga }

```

```

for i  $\leftarrow$  1 to N do

```

```

    read(a,t)

```

```

    HitungLuasSegitiga(a,t)

```

```

endfor

```

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

```
/* Program Segitiga */
/* Menghitung luas N buah segitiga */

#include <stdio.h>

/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga(float a, float t);

main()
{ /* DEKLARASI */
    int i, N;
    float a, t;
    /* PROGRAM UTAMA */
    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);
    }
}

void HitungLuasSegitiga(float alas, float tinggi)
/* Menghitung luas segitiga dengan rumus  $L = (\text{alas} \times \text{tinggi}) / 2$  */
/* K.Awal : alas dan tinggi sudah terdefinisi nilainya */
/* K.Akhir : luas segitiga tercetak. */

{ /*DEKLARASI*/
    float luas; /* luas segitiga, dalam cm2 */
    /*DESKRIPSI*/
    luas = ( alas * tinggi )/2.0;
    printf("Luas segitiga = %5.2f\n",luas);
}
```

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

```
/* Program Segitiga */
/* Menghitung luas N buah segitiga. */

#include <iostream.h>

/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga(float a, float t);

void main()
{ /* DEKLARASI */
  int i, N;
  float a, t;
  /* PROGRAM UTAMA */
  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)
/* Menghitung luas segitiga dengan rumus  $L=(\text{alas} \times \text{tinggi})/2$  */
/* K.Awal : alas dan tinggi sudah terdefinisi nilainya */
/* K.Akhir : luas segitiga tercetak. */

{ /*DEKLARASI*/
  float luas; /* luas segitiga, dalam cm2 */
  /*DESKRIPSI*/
  luas = ( alas * tinggi )/2.0;
  cout << "Luas=" << luas << endl;
}
```


3. Contoh translasi prosedur dengan parameter keluaran

ALGORITMA :

Prosedur :

```

procedure HitungLuasSegitiga(input alas, tinggi : riil,
                             output luas : riil)
{ Menghitung luas segitiga dengan rumus  $L = (alas \times tinggi)/2$  }
{ K.Awal : alas dan tinggi sudah terdefinisi nilainya }
{ K.Akhir : luas berisi luas segitiga. }

```

DEKLARASI

{ tidak ada }

DESKRIPSI

luas \leftarrow (alas * tinggi)/2

Program utama :

Algoritma Segitiga

{ Menghitung luas N buah segitiga. }

DEKLARASI

i, N : integer

a, t : riil

L : riil { luas segitiga }

procedure HitungLuasSegitiga(input alas, tinggi : riil)

{ Menghitung luas segitiga dengan rumus $L = (alas \times tinggi)/2$ }

DESKRIPSI:

read(N) { tentukan banyaknya segitiga }

for i \leftarrow 1 to N do

read(a,t)

HitungLuasSegitiga(a,t,L)

Write(L)

endfor

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

```
/* Program Segitiga */
/* Menghitung luas N buah segitiga. */

#include <stdio.h>

/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga(float a, float t, float *L);

main()
{ /* DEKLARASI */
    int i, N;
    float a, t;
    float L;          /* luas segitiga, dalam cm2 */
    /* PROGRAM UTAMA */
    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\n ",L);
    }
}

void HitungLuasSegitiga(float alas, float tinggi, float *luas)
/* Menghitung luas segitiga dengan rumus  $L = (alas \times tinggi) / 2$  */
/* K.Awal : alas dan tinggi sudah terdefinisi nilainya */
/* K.Akhir : luas segitiga tercetak. */

{ /*DEKLARASI*/
    /*DESKRIPSI*/
    *luas = ( alas * tinggi )/2.0;
}
```

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

```
/* Program Segitiga */
/* Menghitung luas N buah segitiga. */

#include <iostream.h>

/* Deklarasi purwarupa prosedur yang digunakan */
void HitungLuasSegitiga(float a, float t, float *L);

void main()
{
    /* DEKLARASI */
    int i, N;
    float a, t;
    float L;          /* luas segitiga, dalam cm2 */

    /* PROGRAM UTAMA */
    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)
/* Menghitung luas segitiga dengan rumus  $L=(\text{alas} \times \text{tinggi})/2$  */
/* K.Awal : alas dan tinggi sedah terdefinisi nilainya */
/* K.Akhir : luas segitiga tercetak. */

{
    /*DEKLARASI*/
    /*DESKRIPSI*/

    *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  $\leftarrow$  A { simpan nilai A ke dalam temp }
    A  $\leftarrow$  B      { isikan nilai B ke dalam A }
    B  $\leftarrow$  temp { isikan nilai temp ke dalam B }

```

Program utama :

```

Algoritma SelisihXY
{ Prog. untuk menghitung selisih nilai X dan Y, dengan syarat  $X \geq Y$ 
jika  $X < Y$ , maka 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  $\leftarrow$  X - Y { hitung selisih x dan Y }
    write(Z)

```

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

```

/* PROGRAM SelisihXY */
/* {Prog. utama untuk menghitung selisih nilai X dan Y,dengan
syarat  $X \geq Y$  jika  $X < Y$ ,maka X dan Y dipertukarkan dengan
memanggil prosedur Tukar. } */

#include <stdio.h>
void Tukar(int *X, int *Y)
/* Mempertukarkan nilai X dan Y */

main()
{ /* DEKLARASI */
  int X, Y, Z;
  /* PROGRAM UTAMA */
  printf("X = ?"); scanf("%d",&X);
  printf("Y = ?"); scanf("%d",&Y);

  /* jika  $X < Y$ ,pertukaran nilai X dan Y dengan memanggil
prosedur Tukar */
  if(X < Y)
    Tukar(&X,&Y);
  /*endif*/
  Z = X - Y;    /* hitung selisih X dan Y */
  printf("Z = %d \n",Z);
}

void Tukar(int *A, int *B)
/* Mempertukarkan nilai B dan B */
/* K.Awal : nilai A dan B sudah terdefinisi */
/* K.Akhir : A berisi nilai B, B berisi nilai A semula. */
{ /*DEKLARASI*/
  int temp;    /* peubah Bantu */
  /*DESKRIPSI*/
  temp = *A;    /* simpan nilai A didalam temp */
  *A = *B;    /* isi A dengan nilai B */
  *B = temp;    /* isi B dengan nilai temp */
}

```

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

```
/* PROGRAM SelisihXY */
/* {Prog. utama untuk menghitung selisih nilai X dan Y,dengan
syarat  $X \geq Y$  jika  $X < Y$ ,maka X dan Y dipertukarkan dengan
memanggil prosedur Tukar. } */

#include <iostream.h>
void Tukar(int *X, int *Y);
/* Mempertukarkan nilai X dan Y */

void main()
{ /* DEKLARASI */
    int X, Y, Z;
    /* PROGRAM UTAMA */
    cout << "X = ?"; cin >> X;
    cout << "Y = ?"; cin >> Y;

    /* jika  $X < Y$ ,pertukaran nilai X dan Y dengan memanggil
prosedur Tukar */
    if(X < Y)
        Tukar(&X,&Y);
    /*endif*/
    Z = X - Y;      /* hitung selisih X dan Y */
    cout << Z ;
}

void Tukar(int *A, int *B)
/* Mempertukarkan nilai B dan B */
/* K.Awal : nilai A dan B sudah terdefinisi */
/* K.Akhir : A berisi nilai B, B berisi nilai A semula. */
{ /*DEKLARASI*/
    int temp;      /* peubah Bantu */
    /*DESKRIPSI*/
    temp = *A;      /* simpan nilai A didalam temp */
    *A = *B;        /* isi A dengan nilai B */
    *B = temp;      /* isi B dengan nilai temp */
}
```

PENUTUP

Procedure dan function di dalam algoritma memiliki perbedaan. Tetapi di bahasa pemrograman C++ semua disebut function dengan perbedaan sintak untuk membedakan antara fungsi sebagai procedure dan sebagai function yang dimaksud pada algoritma

SOAL-SOAL

Buatlah program dengan menggunakan procedure untuk menghitung volume silinder (dengan berbagai bentuk parameter).

REFERENSI :

- Buku Teks
 1. Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 1, Edisi Ketiga, Penerbit Informatika Bandung.
 2. Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman I*, Edisi Kedua, Yogyakarta
- Buku Acuan/Referensi
 1. Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
 2. Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.
 3. Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
 4. Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
 5. Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

