

Algoritma Pemrograman dan Struktur Data

Materi 3: FUNGSI DAN PROSEDUR

Dosen pengampu:

Suamanda Ika Novichasari, M.Kom.

Imam Adi Nata, M.Kom



**Kampus
Merdeka**
INDONESIA JAYA

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR

Jl. Kapten Suparman No.39, Tuguran,
Potrobangsari, Kec. Magelang Utara, Kota
Magelang, Jawa Tengah 56116



Learning Objective

Mahasiswa mampu menjelaskan konsep fungsi



Mahasiswa mampu menjelaskan konsep prosedur

Course Material

Fungsi

Prosedur

Parameter
formal dan
aktual

Variabel
Global dan
Lokal

BAB MATERI



Konsep Sub Program

Subbab ini mempelajari konsep sub program



Pendahuluan

Program Aplikasi terdiri dari dua bagian yaitu Main Program dan Sub Program

Main Program: Sekumpulan perintah (kode) yang merupakan bagian utama dari program aplikasi.

- ☐ Main program dieksekusi paling awal ketika aplikasi digunakan

Sub Program: Sekumpulan perintah (kode) yang berdiri sendiri

- ☐ Sub program dijalankan ketika dipanggil oleh main program atau sub program lain



Main Program
memanggil **sub
program**

sekumpulan
sub program



Sebuah
Aplikasi
terdiri dari
satu buah
**Main
Program** dan
beberapa **sub
program**



Apa itu Sub Program ?

Sub Program ataudikenal juga sebagai ***Sub-Routine***.

Sub Program : sekumpulan perintah (kode) yang ditujukan untuk mengerjakan tugas tertentu (spesifik) dan dikelompokkan dalam satu unit.



Mengapa Butuh Sub Program?

Aplikasi yang kompleks (**skala besar**) membutuhkan banyak sekali baris kode program.

Jumlah Baris kode program bisa terdiri lebih dari jutaan.

Semakin banyak baris kode, semakin sulit dalam mencari letak kesalahan program.

Tanpa **sub program**, maka pemeriksaan kesalahan dimulai dari baris pertama sampai terkahir.

Kurang efisien dalam pelacakan kesalahan program



Manfaat Sub Program

- Memanajemen kode-kode program
- Penulisan kode program lebih pendek
- Penulisan kode program lebih efisien karena tidak perlu mengulangi penulisan kode yang sama
- Meminimalisir kesalahan penulisan kode program
- Memudahkan pelacakan kesalahan kode program



Simbol Sub Program

Pada penulisan flowchart, simbol sub program seperti dibawah ini :





Jenis Sub Program

Subbab ini mempelajari jenis-jenis sub program



Jenis Sub Program

Fungsi

- Sub Program yang **mengembalikan** suatu nilai

Prosedur

- Sub Program yang **tidak mengembalikan** suatu nilai



Menyatakan Fungsi

```
tipeDataReturn namaFungsi()  
{  
    .... insturksi 1 .... ;  
    .... instruksi 2 .... ;  
    return variabel;  
}
```

Pseudocode

```
int hitung()  
{  
    int A = 2;  
    int B = 5;  
    int jum;  
    jum = A + B;  
    return jum;  
}
```

Kode Bahasa C



Menyatakan Prosedur

```
void namaFungsi()  
{  
    ..... insturksi 1 ..... ;  
    ..... instruksi 2 ..... ;  
    print variabel;  
}
```

Pseudocode

```
void hitung()  
{  
    int A = 2;  
    int B = 5;  
    int jum;  
    jum = A + B;  
    printf("%d, jum");  
}
```

Kode Bahasa C



Memanggil Sub Program

```
int main()  
{  
    namaFungsi();  
    namaProsedur();  
    return 0;  
}
```

namaFungsi();

namaProsedur();

return 0;

Memanggil **FUNGSI**

Memanggil **PROSEDUR**



Keterangan pada Sub Program

Void artinya Sub-program yang dibuat **berjenis prosedur** (tidak memiliki nilai kembalian).

tipeDataReturn artinya Sub-Program yang dibuat **berjenis fungsi** dengan jenis data kembalian tertentu.

namaProsedur dan **namaFungsi** menyatakan nama dari Sub-Program.

return merupakan perintah untuk mengembalikan suatu nilai sesuai dengan **tipeDataReturn**



Konsep Parameter

Subbab ini mempelajari jenis-jenis sub program



Apa itu parameter ?

Parameter adalah variabel yang digunakan menampung nilai yang akan diproses di dalam Sub-Program.

Parameter berperan sebagai **input** untuk Sub-Program



Aturan Penulisan Parameter

Parameter ditulis di dalam tanda kurung (...).

Setiap parameter harus memiliki tipe data.

Bila memiliki lebih dari satu parameter, maka penulisannya dipisahkan dengan tanda koma (,)



Penulisan Fungsi dengan Parameter

```
tipeData namaFungsi(tipeData parameter1, tipeData parameterN) {  
    ..... instruksi 1 ..... ;  
    ..... instruksi 2 ..... ;  
    return variabel;  
}
```



Pseudocode

```
float luasSegitiga(float a, float t) {  
    float alas = a;  
    float tinggi = t;  
    float luas;  
    luas = 1/2 (a x t);  
    return luas;  
}
```

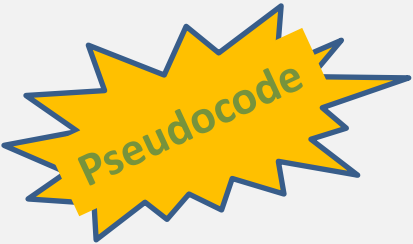


Contoh
Kode C



Penulisan Prosedur dengan Parameter

```
void namaFungsi(tipeData parameter1, tipeData parameterN) {  
    ..... instruksi 1 ..... ;  
    ..... instruksi 2 ..... ;  
    print variabel;  
}
```



Pseudocode

```
void luasSegitiga(float a, float t) {  
    float alas = a;  
    float tinggi = t;  
    float luas;  
    luas = 1/2 (a x t);  
    printf("%.2f", luas);  
}
```



Contoh
Kode C



Variabel Lokal & Global

Subbab ini mempelajari perbedaan variabel lokal dan global



Review Lagi “Variabel”

Variabel merupakan sebuah “**wadah**” yang digunakan untuk menampung data/nilai dengan tipe data tertentu.

Terdapat 2 tempat untuk mendeklarasikan variabel:

- 1) Di dalam fungsi atau blok yang disebut variabel lokal.
- 2) Di luar semua fungsi yang disebut variabel global.



Variabel Lokal

- **Variabel lokal** adalah variabel yang **dideklarasikan** dalam suatu **blok fungsi**.
- **Variabel lokal** **hanya terlihat** dan **dapat digunakan** di dalam **blok fungsi** dimana **variabel tersebut dideklarasikan**.
- **Variabel lokal** akan **dibuat** saat *compiler* melawati deklarasi yang ada di dalam suatu lingkup fungsi.
- **Variabel lokal** **dihancurkan** setelah *compiler* keluar dari lingkup fungsi.



Sifat “Variabel Lokal”

- Segala sesuatu yang **berada** di dalam ‘{’ dan ‘}’ berarti berada di dalam blok fungsi.
- Variabel lokal **berada** di dalam **blok suatu fungsi**.
- Variabel lokal **tidak dapat** diakses atau digunakan **di luar blok**.
- Variabel lokal **hanya bisa digunakan** oleh kode program yang ada di dalam fungsi.

Contoh “Variabel Lokal”

```
float penjumlahan(float a, float b) {
```

```
    float c;
```

```
    c = a + b;
```

```
    return c;
```

```
}
```

```
int main ()
```

```
{
```

```
    float d, x;
```

```
    d = penjumlahan(4,5);
```

```
    x = d + 1;
```

```
    printf ("%f", x);
```

```
    return 0;
```

```
}
```

Variabel **a**, **b**, dan **c** adalah variabel lokal milik **fungsi penjumlahan()**

Variabel **d** dan **x** adalah variabel lokal milik **fungsi main()**



Variabel Global

- **Variabel global** adalah variabel yang memiliki *ruang lingkup file*.
- **Variabel global** dideklarasikan **di luar fungsi** apapun dalam file.
- **Variabel global** dideklarasikan **di bagian atas file**, **di bawah keyword `include`** dan di atas kode apapun



Sifat “Variabel Global”

- Diakses dari titik awal hingga akhir file di mana variabel tersebut dideklarasikan.
- Digunakan di mana saja di dalam file dimana variabel tersebut dideklarasikan



Contoh “Variabel GLobal”

```
int umur=20;

void cetakumur()
{
    printf("Umur = %d\n", umur);
}

int main(){
    cetakumur();
    printf("Umur = %d", umur);
    getchar();
    return 0;
}
```

Variabel **umur** adalah variabel global,

Sehingga dapat digunakan oleh **fungsi cetakumur()**

dan dapat digunakan oleh **fungsi main()**



Contoh “Variabel GLobal”

```
int umur=20;

void cetakumur()
{
    printf("Umur = %d\n", umur);
}

int main(){
    cetakumur();
    printf("Umur = %d", umur);
    getchar();
    return 0;
}
```

Variabel **umur** adalah variabel global,

Sehingga dapat digunakan oleh **fungsi cetakumur()**

dan dapat digunakan oleh **fungsi main()**

Tugas

Jelaskan alur dari pseudocode berikut dan tentukan outputnya jika $s = 5$!

```
//PROCEDURE P_Lp (input x : integer; output Lp :
integer)→ integer
// Prosedur untuk menghitung luas persegi dengan
parameter

//IMPLEMENTATION
    Lp ← x * x
```

```
//FUNCTION F_LP (a : integer)→ integer
// Fungsi untuk menghitung luas persegi

//DECLARATION

//IMPLEMENTATION
    →(a * a)
```

```
//PROGRAM Beda_FP
// Mengimplementasikan Perbedaan Fungsi dan
Prosedur

//DECLARATION
    s : integer
    L : integer

    procedure P_Lp (input x : integer; output
Lp : integer)→ integer
    function F_LP (a : integer)→ integer

//IMPLEMENTATION / DEFINITION
    input (s)
    P_Lp (s,L)
    output (L)
    output (F_LP(a))
```



TERIMA KASIH

#NEXT... FUNGSI&PROSEDUR