

PARAMETER PADA FUNGSI

Pada program dengan fungsi dan prosedur, seringkali terjadi pertukaran data atau informasi antara fungsi atau prosedur yang dipanggil dengan program pemanggil atau sub program lainnya. Pertukaran atau pengiriman data tersebut melalui antar muka atau penghubung yang disebut parameter.

Dengan parameter kita dapat meneruskan data atau informasi ke dalam fungsi, dan akan diterima oleh fungsi maupun prosedur sebagai nilai masukan. Penggunaan parameter menjadikan fungsi lebih dinamis, dengan parameter satu fungsi dan prosedur dapat digunakan untuk melakukan satu tugas yang sama dengan nilai yang berbeda-beda sekalipun. Parameter dapat dideklarasikan pada saat pemanggilan dan pendefinisian fungsi maupun prosedur. Jenis-jenis parameter pada program antara lain:

A. PARAMETER FORMAL

Parameter formal adalah daftar variabel yang dituliskan pada saat pendefinisian fungsi, parameter formal menerima data atau informasi yang diteruskan dari program pemanggil. Cara penulisan parameter formal pada fungsi maupun prosedur adalah sebagai berikut:

```
tipe-data-hasil nama-subprogram (tipe-data-  
parameter1 nama-parameter1, tipe-data-  
parameter2 nama-parameter2,....., tipe-data-  
parameter-n nama-parameter-n)
```

Keterangan:

1. Parameter formal dituliskan setelah nama fungsi maupun prosedur
2. Daftar parameter dituliskan dalam tanda kurung "(" dan dituliskan setelah nama fungsi maupun prosedur
3. Daftar parameter ditulis dengan menuliskan nama parameter dan tipe datanya
4. Parameter dapat berjumlah lebih dari satu, untuk daftar parameter yang lebih dari satu dipisahkan dengan tanda koma ", "

B.PARAMETER AKTUAL

Parameter aktual atau sering disebut **argumen**, adalah parameter yang dituliskan pada saat pemanggilan fungsi atau prosedur. Parameter aktual meneruskan data atau nilai ke suatu fungsi ketika dipanggil. Cara penulisan parameter aktual pada program adalah sebagai berikut:

```
nama-sub-program (nama-parameter1, nama-  
parameter2,....., nama-parmaeter-n)
```

Keterangan:

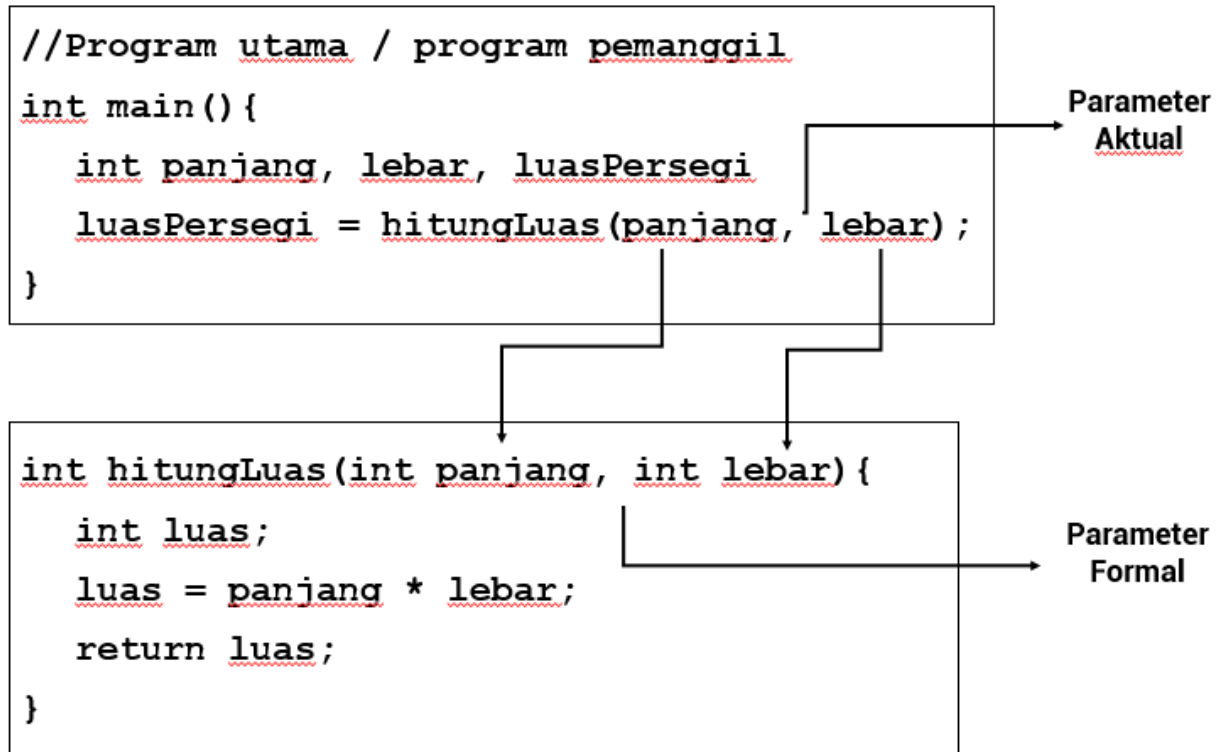
1. Parameter dituliskan dalam tanda kurung “()” dan dituliskan setelah nama fungsi atau prosedur yang dipanggil
2. Daftar parameter ditulis dengan menuliskan **nama parameter** saja
3. Untuk daftar parameter yang lebih dari satu dipisahkan dengan tanda koma “ , ”

Pada saat fungsi maupun prosedur dipanggil maka parameter aktual berkoresponden atau berpasangan dengan parameter formal. Tiap-tiap parameter aktual berpasangan dengan parameter formal yang bersesuaian.

Aturan yang harus diperhatikan dalam pertukaran data atau informasi antara parameter formal dan aktual yang bersesuaian adalah:

1. Tiap parameter aktual harus bertipe data sama dengan tipe data parameter formal yang bersesuaian.
2. Jumlah parameter aktual pada pemanggilan fungsi maupun prosedur harus sama dengan jumlah parameter formal.
3. Urutan penulisan parameter formal dan aktual yang bersesuaian harus sama.

Sebagai contoh, perhatikan proses korespondensi parameter formal dan aktual berdasarkan fungsi `hitungLuas()` yang telah dibahas pada sub materi fungsi berikut:



Pada saat fungsi `hitungLuas` dipanggil, parameter aktual `panjang` dan `lebar` mengirimkan nilai pada parameter formal dalam fungsi. Selanjutnya, nilai yang didapat dari parameter aktual digunakan fungsi untuk melakukan perhitungan.

KEGUNAAN PARAMETER

Dengan parameter kita dapat meneruskan data atau informasi ke dalam fungsi maupun prosedur yang nilainya dapat kita atur, parameter akan diterima oleh fungsi maupun prosedur sebagai nilai masukan. Penggunaan parameter menjadikan fungsi lebih dinamis, karena dengan parameter satu fungsi dan prosedur dapat digunakan untuk melakukan satu tugas yang sama dengan nilai yang berbeda-beda sekalipun.

Sebagai contoh dalam suatu prosedur ingin menampilkan suatu data, tapi tidak memiliki nilai yang persis sama persis seperti berikut:

1.

Nama Lengkap: Tiara

Jurusan: RPL

Alamat Asal: Bandung

2.

Nama Lengkap: Dini

Jurusan: Multimedia

Alamat Asal: Bogor

3.

Nama Lengkap: Tia

Jurusan: TKJ

Alamat Asal: Cimahi

Jika dilihat format teksnya seperti berikut:

Nama Lengkap: <nama_lengkap>

Jurusan sekolah: <jurusan>

Alamat Asal: <alamat>

Ketiganya mencetak data hanya dengan nilai yang berbeda-beda. Nilai seperti (nama_lengkap), (jurusan), (alamat) memiliki nilai yang berbeda-beda. Nilai tersebut dapat diatur dengan menggunakan **parameter**.

Pada program dengan fungsi dan prosedur, kerap kali kita ingin melakukan tugas yang sama dengan nilai yang berbeda-beda seperti contoh 1, hal tersebut dapat dilakukan dengan menggunakan dengan parameter

Dengan parameter kita dapat meneruskan data atau informasi ke dalam fungsi maupun prosedur yang nilainya dapat kita atur, dan parameter akan diterima oleh fungsi maupun prosedur sebagai nilai masukan. Penggunaan parameter menjadikan fungsi lebih dinamis, karena dengan parameter satu fungsi dan prosedur dapat digunakan untuk melakukan satu tugas yang sama dengan nilai yang berbeda-beda sekalipun.

Contoh 1(Prosedur dengan Parameter)

Nama Lengkap: Tiara

Jurusan: RPL

Alamat Asal: Bandung

Nama Lengkap: Dini

Jurusan: Multimedia

Alamat Asal: Bogor

Nama Lengkap: Tia

Jurusan: TKJ

Alamat Asal: Cimahi

Karena nilai dari nama, alamat dan jurusan berbeda-beda, maka **<nama_lengkap>** **<jurusan>** dan **<alamat>** dijadikan parameter agar nilainya dapat diatur:

```
#include<iostream>
```

```
using namespace std;
```

PARAMETER FORMAL

```
void biodata(string nama, string jurusan, string alamat){
```

```
    cout<<"Nama Lengkap : "<< nama_lengkap <<endl;
```

```
    cout<<"Jurusan      : "<< jurusan <<endl;
```

```
    cout<<"Alamat Asal   : "<< alamat <<endl;
```

```
}
```

```
int main(){
```

```
    biodata("Tiara","RPL","Bandung");
```

```
    biodata("Dini","Multimedia","Bogor");
```

```
    biodata("Tia","TKJ","Cimahi");
```

PARAMETER AKTUAL

Berdasarkan program, prosedur biodata dipanggil dari program utama dengan menyertakan parameter aktual berikut:

```
biodata("Tiara", "RPL", "Bandung");  
biodata("Dini", "Multimedia", "Bogor");  
biodata("Tia", "TKJ", "Cimahi");
```

Ketika prosedur biodata dipanggil, maka nilai parameter aktual tersebut diisikan ke dalam parameter formal **nama_lengkap**, **jurusan** dan **alamat**. Selanjutnya nilai tersebut digunakan dalam proses yang terdapat pada prosedur

```
int main() {  
    biodata("Tiara", "RPL", "Bandung");  
    biodata("Dini", "Multimedia", "Bogor");  
    biodata("Tia", "TKJ", "Cimahi");  
}
```

```
void biodata(string nama, string jurusan, string alamat) {  
    cout<<"Nama Lengkap   : " << nama <<endl;  
    cout<<"Jurusan         : " << jurusan <<endl;  
    cout<<"Alamat Asal      : " << alamat <<endl;  
}
```

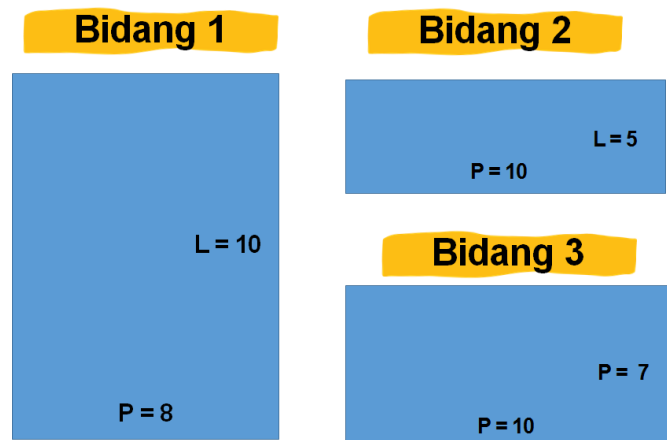
Ketika program tersebut dijalankan, maka nilai keluaran program adalah:

```
Nama Lengkap   : Tiara  
Jurusan        : RPL  
Alamat Asal    : Bandung  
  
Nama Lengkap   : Dini  
Jurusan        : Multimedia  
Alamat Asal    : Bogor  
  
Nama Lengkap   : Tia  
Jurusan        : TKJ  
Alamat Asal    : Cimahi  
-----
```

Dengan parameter satu fungsi maupun prosedur dapat digunakan untuk melakukan satu tugas yang sama dengan nilai yang berbeda-beda.

Contoh 2 (Fungsi dengan Parameter)

Membuat sebuah Program dengan fungsi yang bertugas untuk menghitung luas persegi panjang dengan ukuran panjang dan lebar seperti yang tertera pada gambar.



Karena fungsi tersebut melakukan tugas yang sama yaitu menghitung luas, hanya dengan nilai yang berbeda-beda. Oleh karena itu panjang dan lebar dapat digunakan sebagai parameter agar nilainya dapat kita atur.

```
#include<iostream>
using namespace std;

int hitungLuas(int panjang, int lebar){
    int luas;
    luas = panjang * lebar;
    return luas;
}

int main(){
    int p, l, luasPersegi;
    p = 8; l = 10;
    luasPersegi = hitungLuas (p, l);
    luasPersegi = hitungLuas (10, 5);
    luasPersegi = hitungLuas (10, 7);
}
```

parameter formal

parameter aktual

Berdasarkan program, fungsi `hitungLuas` dipanggil dari program utama dengan menyertakan nilai panjang dan lebar dalam parameter aktual `p` dan `l`

```
hitungLuas(p, l);
```

yang nantinya akan diterima nilainya oleh parameter formal panjang dan lebar.

Karena yang dipentingkan adalah nilainya maka nama parameter aktual tidak harus sama dengan parameter formal yang bersesuaian. Selain itu, parameter aktual dapat berupa ekspresi atau konstanta. Jadi pada program tersebut, pemanggilan fungsi dengan menyertakan parameter aktual berikut adalah benar.

```
hitungLuas(10, 5);  
hitungLuas(10, 7);
```

Maka ketika fungsi `hitungLuas` dipanggil, maka nilai parameter aktual diisikan ke dalam parameter formal panjang dan lebar. Selanjutnya nilai panjang dan lebar digunakan untuk melakukan perhitungan pada fungsi.

```
int main(){  
    int p, l, luasPersegi;  
    p = 8; l = 10;  
    luasPersegi = hitungLuas (p, l);  
    luasPersegi = hitungLuas (5, 10);  
    luasPersegi = hitungLuas (7, 10);  
}
```

```
int hitungLuas(int panjang, int lebar){  
    int luas;  
    luas = panjang * lebar;  
    return luas;  
}
```


Ketika program tersebut dijalankan, maka nilai keluaran program adalah:

Hasil Eksekusi

Luas = 80

Luas = 50

Luas = 70

Pada contoh program, fungsi hitungLuas dapat digunakan untuk menghitung satu tugas yang sama yaitu menghitung luas persegi panjang dengan nilai panjang dan lebar yang berbeda-beda sekalipun.

Berdasarkan beberapa contoh tersebut menunjukkan bahwa penggunaan parameter pada program dapat menjadikan suatu fungsi maupun prosedur menjadi lebih dinamis. Dengan parameter satu fungsi maupun prosedur dapat digunakan untuk melakukan satu tugas yang sama dengan nilai yang berbeda-beda.