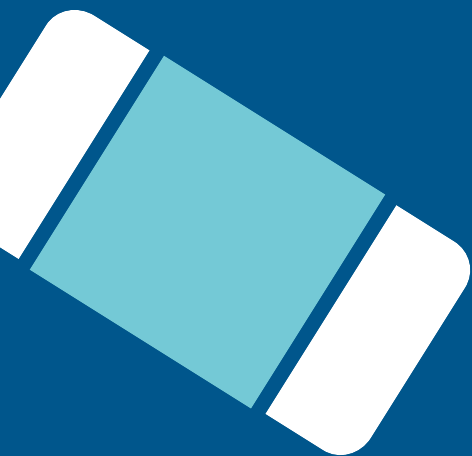
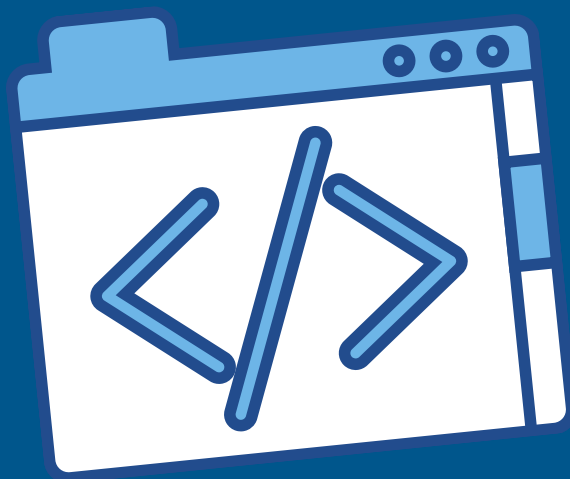
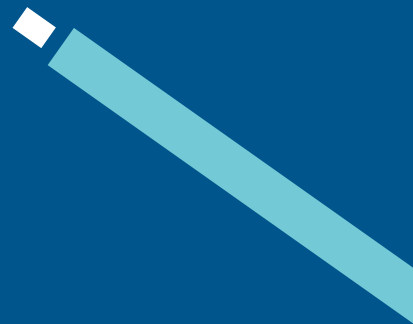
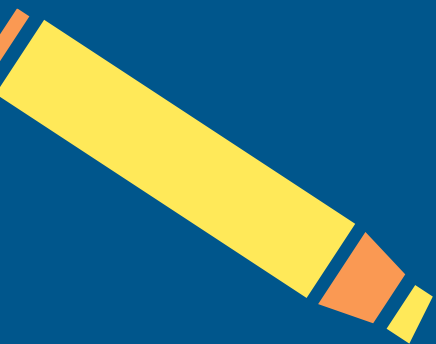


PARAMETER PADA FUNGSI



PARAMETER

Pada program dengan fungsi dan prosedur, seringkali terjadi pertukaran data atau informasi antara fungsi atau prosedur yang dipanggil dengan program pemanggil. 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:

1. Parameter Formal
2. Parameter Aktual

PARAMETER FORMAL

Parameter formal adalah variabel yang ada pada daftar parameter dalam definisi fungsi, parameter formal menerima data atau informasi yang diteruskan dari program pemanggil. Parameter formal dituliskan pada saat **pendefinisian** fungsi maupun prosedur.

Cara penulisan parameter formal pada program adalah sebagai berikut:

```
tipe-data-hasil namaFungsi  
  (tipe-data-parameter1 nama-parameter1,  
   tipe-data-parameter-2, nama-parameter2,  
   tipe-data-parameter-n, nama-parameter-N  
  )
```

Keterangan:

- Parameter dituliskan dalam tanda kurung “()” dan dituliskan setelah nama fungsi.
- Daftar parameter ditulis dengan menuliskan tipe data serta nama parameter
- Untuk parameter yang lebih dari satu dapat dipisahkan dengan tanda koma “ , ”

PARAMETER AKTUAL

Parameter aktual atau sering disebut **argumen**, adalah nilai yang dituliskan pada saat pemanggilan fungsi atau prosedur. Parameter aktual adalah data atau nilai yang diteruskan ke suatu fungsi ketika dipanggil. Parameter aktual dituliskan pada saat pemanggilan fungsi maupun prosedur.

Cara penulisan parameter aktual pada program adalah sebagai berikut:

```
namaFungsi (nama-parameter1,  
nama-parameter2,.....,nama-  
parameter-N)
```

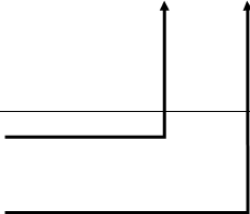
Keterangan:

- Parameter dituliskan dalam tanda kurung “()” dan dituliskan setelah nama fungsi.
- Daftar parameter ditulis dengan menuliskan nama parameter
- Untuk parameter yang lebih dari satu dapat dipisahkan dengan tanda koma “ , ”

Contoh:

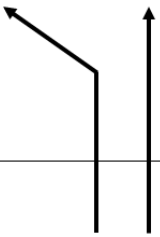
```
main{
    .....
    c = jumlah(a, b);
    .....
}
```

parameter
aktual

Two arrows originate from the labels 'parameter' and 'aktual' below the main function. They point upwards to the arguments 'a' and 'b' in the function call 'jumlah(a, b)' inside the main function's body.

```
float jumlah (int x,int y){
    return (x+y);
}
```

parameter
formal

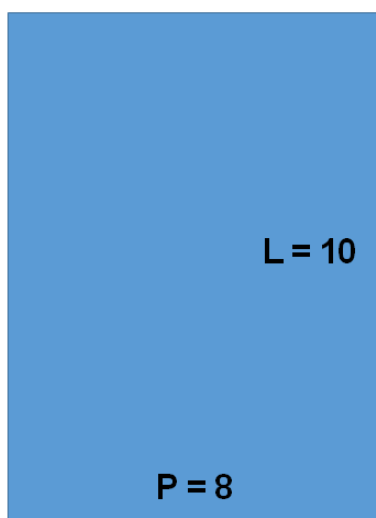
Two arrows originate from the labels 'parameter' and 'formal' below the jumlah function. They point upwards to the parameters 'x' and 'y' in the function signature 'float jumlah (int x,int y)'. A diagonal arrow also points from the 'return' statement to the 'parameter' label.

Pada contoh program di atas, dalam fungsi jumlah() variabel x dan y dinamakan sebagai parameter formal. Adapun parameter aktual adalah variabel a dan b, yang disertakan pada saat pemanggilan fungsi dan meneruskan nilai ke dalam fungsi.

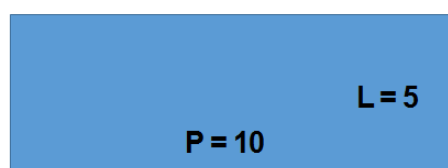
Perhatikan contoh berikut!

Terdapat tiga buah persegi panjang dengan ukuran panjang dan lebar yang **berbeda-beda**. Buat sebuah program dengan fungsi untuk menghitung luas ketiga persegi panjang tersebut.

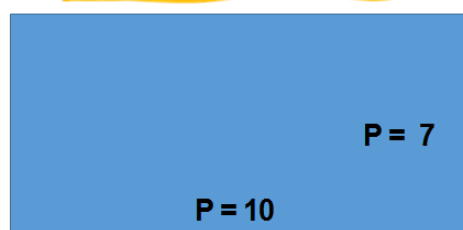
Bidang 1



Bidang 2



Bidang 3



```

#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) ;
```

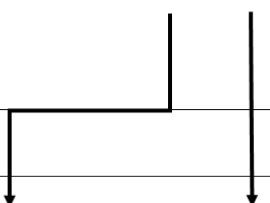
Karena yang dipentingkan adalah nilainya, maka nama parameter aktual **tidak harus sama** dengan parameter formal yang bersesuaian.

Selain itu, karena yang dipentingkan adalah nilainya, maka 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) ;
```

Ketika fungsi `hitungLuas` dipanggil, maka **nilai parameter aktual p dan l** diisikan ke dalam **parameter formal panjang dan lebar**. Selanjutnya nilai panjang dan lebar digunakan untuk menghitung luas persegi panjang dalam fungsi `hitungLuas`.

```
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;  
}
```

The diagram shows two vertical arrows pointing from the function calls in the `main` function to the parameters of the `hitungLuas` function. The first arrow connects the `p` parameter in the first call to the `panjang` parameter in the function definition. The second arrow connects the `l` parameter in the first call to the `lebar` parameter in the function definition.

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.

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.

ATURAN PENULISAN PARAMETER

```
int main() {  
    int p, l, luasPersegi;  
    luasPersegi = hitungLuas (p, l);  
}
```

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

Tiap parameter aktual harus bertipe data sama dengan tipe parameter formal yang bersesuaian

Pada contoh program tersebut, kedua parameter baik parameter formal maupun aktual bertipe data sama yaitu bertipe data integer

Jumlah parameter aktual dan parameter formal harus sama

Parameter aktual pada contoh program berjumlah dua yaitu p dan l. Serta parameter formal pada fungsi berjumlah dua yaitu panjang dan lebar.