

PARAMETER

Pada fungsi maupun kita dapat meneruskan suatu nilai atau informasi, informasi dapat diteruskan ke prosedur maupun fungsi sebagai **parameter**. Parameter bertindak sebagai variabel di dalam fungsi. Dengan parameter suatu prosedur maupun fungsi dapat melakukan tugas yang sama dengan beberapa nilai berbeda seperti pada contoh. Parameter menjadikan suatu prosedur maupun fungsi menjadi lebih dinamis.

1. PARAMETER FORMAL

Parameter dituliskan setelah nama prosedur, di dalam tanda kurung “()”. Parameter yang dideklarasikan pada saat pendefinisian prosedur maupun fungsi dinamakan parameter formal

2. PARAMETER AKTUAL

Lalu pada saat pemanggilan fungsi, program utama meneruskan atau mengirimkan informasi berupa parameter ke dalam prosedur. Parameter yang dideklarasikan pada saat pemanggilan dinamakan sebagai **parameter aktual**.

Parameter aktual dan formal melakukan pertukaran informasi, sehingga terdapat beberapa aturan penulisan sebagai berikut:

1. **Jumlah parameter aktual dan formal harus sama**
2. **Tipe data antara parameter formal dan aktual harus sama**
3. **Urutan penulisan parameter formal dan aktual harus sesuai**

Contoh:

Fungsi hitungLuas kini memiliki parameter berupa **variabel panjang dan lebar bertipe data integer**. Saat fungsi dipanggil, program utama **meneruskan nilai parameter aktual berupa variabel p dan l bertipe data integer** **(karena yang dipentingkan adalah nilainya, nama parameter formal dan aktual boleh berbeda asal bertipe data sama)**.

Selain itu, nilai parameter aktual dapat langsung diisikan pada saat pemanggilan seperti pada contoh berikut:

```
#include <iostream>
using namespace std;
// pendefinisian fungsi dengan parameter
int hitungLuas (int panjang, int lebar){
    int luas;
    luas = panjang * lebar;
    return luas; // nilai kembalian fungsi
}
int main(){
    int p, l, luasPersegi;
    p = 10; l = 5;
    luasPersegi = hitungLuas (p, l); // pemanggilan
    fungsi dengan parameter
    cout << "Luas Persegi = " << luasPersegi (8, 5); //
    pemanggilan      fungsi dengan parameter
    return 0;
}
```

Prosedur dengan Parameter

Prosedur yang bertugas mencetak kata Hello dengan menambahkan parameter nama

```
void cetak(string nama) {
    cout << "Hello" << nama;
}
int main() {
    string pnama;
    pnama = "Gita"
    cetak(pnama); // pemanggilan prosedur
    cetak("Tia"); // pemanggilan prosedur
    cetak("Dini"); // pemanggilan prosedur
    return 0;
}

// Output Hello Gita
// Output Hello Tia
// Output Hello Dini
```