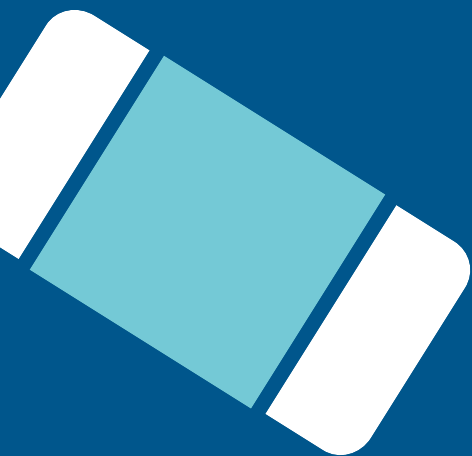
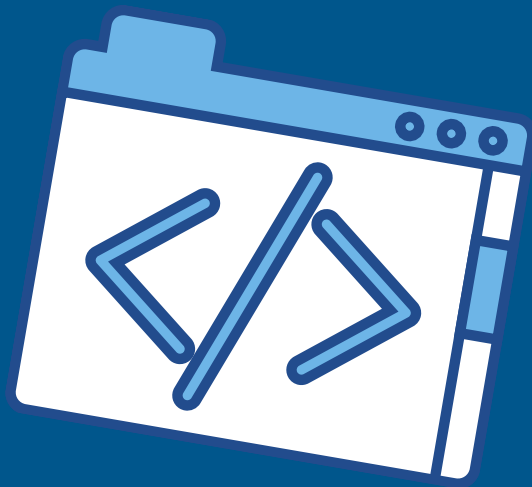
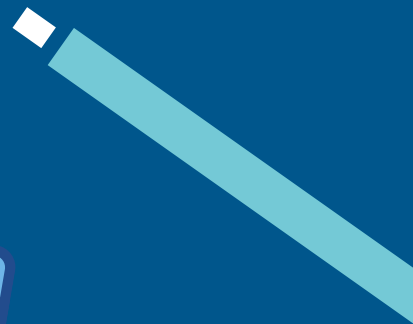
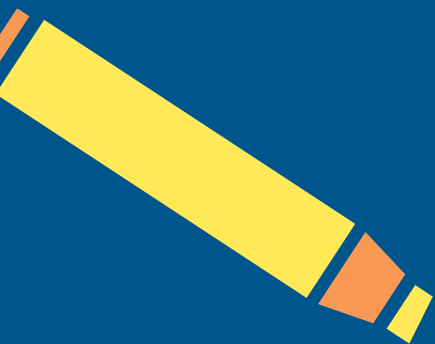


FUNGSI



FUNGSI

Pada program selain terdapat prosedur yaitu sub program yang tidak mengembalikan nilai, juga terdapat suatu fungsi atau sub program yang mengembalikan nilai atau dikenal dengan fungsi atau function.

Sama halnya seperti prosedur fungsi merupakan sub program yang mengerjakan tugas tertentu dan merupakan bagian dari program utama. Yang membedakan fungsi dengan prosedur adalah **fungsi mengembalikan suatu nilai**.



Ilustrasi tersebut menggambarkan bahwa prosedur hanya melakukan serangkaian instruksi tertentu, sedangkan fungsi melakukan serangkaian instruksi dan memberikan nilai keluaran.

Fungsi merupakan modul atau sub program yang merupakan bagian dari program utama dan bertugas mengerjakan tugas tertentu, serta **mengembalikan suatu nilai**

Sama halnya seperti pada prosedur, untuk dapat menggunakan fungsi pada program maka perlu dilakukan **pendefinisian** dan **pemanggilan** fungsi.



PENDEFINISIAN FUNGSI

Sebagaimana halnya pada prosedur, untuk dapat menggunakan fungsi pada program perlu dilakukan pendefinisian fungsi. Adapun sintaks pendefinisian fungsi dalam bahasa pemrograman C++ adalah sebagai berikut:

```
tipe-data-hasil namaFungsi (deklarasi parameter,  
jika ada){  
    DEKLARASI  
    {semua nama yang dipakai dalam fungsi,  
    variabel maupun konstanta}  
    ALGORITMA  
    {instruksi-instruksi yang dikerjakan oleh  
    fungsi}  
    return hasil  
    {pengembalian nilai yang dihasilkan fungsi}  
}
```

Fungsi biasanya akan mengembalikan sebuah nilai dari hasil prosesnya. Karena itu, kita harus menentukan **tipe data** untuk nilai yang akan dikembalikan. **Tipe data hasil** menspesifikasikan tipe nilai yang diberikan oleh fungsi. Nilai yang diberikan oleh fungsi dapat bertipe dasar maupun bertipe bentukan.

Dalam tubuh fungsi, pernyataan yang digunakan untuk memberikan nilai keluaran fungsi dituliskan dengan keyword **return**. Hasil yang dikembalikan dapat berupa konstanta, sebuah variabel, atau sebuah rumus.

CONTOH

Membuat sebuah program yang bertugas untuk menghitung **luas** dan **keliling** persegi panjang.

Pendefinsian fungsi hitung luas

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

Pada program tersebut didefinisikan fungsi bernama **hitungLuas**. Fungsi mengembalikan nilai variabel luas yang bertipe data **integer**, sehingga tipe data hasil yang dituliskan sebelum nama fungsi disesuaikan dengan tipe data kembalian fungsi yaitu **int**.

Di dalam badan fungsi, nilai yang dihasilkan oleh fungsi dikembalikan ke program pemanggil pada pernyataan

```
return luas
```

yang di dalam hal ini fungsi mengembalikan hasil dari variabel luas

Pendefinsian fungsi hitung keliling

```
int hitungKeliling(){  
    int panjang, lebar, keliling;  
    keliling = 2 * (panjang + lebar);  
    return keliling;  
}
```

Pada program tersebut didefinisikan fungsi bernama **hitungKeliling**. Fungsi mengembalikan nilai variabel keliling yang bertipe data **integer**, sehingga tipe data hasil yang dituliskan sebelum nama fungsi disesuaikan dengan tipe data kembalian fungsi yaitu **int**.

Di dalam badan fungsi, nilai yang dihasilkan oleh fungsi dikembalikan ke program pemanggil pada pernyataan

```
return keliling
```

yang di dalam hal ini fungsi mengembalikan hasil dari variabel keliling



PEMANGGILAN FUNGSI

Fungsi bukan merupakan suatu program yang dapat berdiri sendiri jadi ia tidak dapat dieksekusi secara langsung. Ini berarti, instruksi-instruksi di dalam fungsi baru dapat dilaksanakan ketika fungsi tersebut diakses.

Fungsi dapat diakses dengan melakukan pemanggilan fungsi dari program pemanggil (dapat dari program utama atau dari sub program lainnya). Karena mengembalikan suatu nilai maka pemanggilan suatu fungsi dapat dilakukan dilakukan dengan cara berikut:

Cara 1:

Nilai kembalian fungsi ditampung di dalam sebuah variabel yang bertipe data sama dengan tipe data keluaran fungsi.

```
variabel = namaFungsi (daftar parameter, jika ada);
```

Contoh:

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

Nilai kembalian
fungsi berupa
variabel **luas** bertipe data integer

```
int main() {  
    int luasPersegi;  
    luasPersegi = hitungLuas();  
}
```

Nilai keluaran fungsi
ditampung dalam
variabel **luasPersegi** bertipe data integer

Pada contoh tersebut fungsi hitungLuas dipanggil pada program utama. Pada saat pemanggilan fungsi, nilai keluaran fungsi ditampung dalam sebuah variabel yang **bertipe data sama** dengan nilai keluaran fungsi

Cara 2:

Nilai yang dikembalikan oleh fungsi dapat langsung dimanipulasi hasilnya

Contoh:

```
int hitungKeliling(){  
    int panjang, lebar, luas;  
    keliling = 2 * (panjang + lebar);  
    return keliling;  
}
```

Pendefinisian
Fungsi



```
int main(){  
    cout << "Keliling:" << hitungKeliling();  
}
```

Pemanggilan
Fungsi



Pada contoh tersebut pemanggilan fungsi hitungKeliling dipanggil pada pernyataan:

```
cout << "Keliling:" << hitungKeliling();
```

yang dalam hal ini nilai dalam fungsi langsung dicetak hasilnya.

Berbeda dengan prosedur yang tidak dapat ditampilkan hasilnya hasil secara langsung, pada fungsi karena fungsi mengembalikan suatu nilai maka nilai hasil keluaran fungsi dapat langsung ditampilkan hasilnya.

Pada satu program dapat didefinisikan satu atau lebih fungsi

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

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

int hitungKeliling(){
    int panjang, lebar, keliling;
    keliling = 2 * (panjang + lebar);
    return keliling;
}

int main(){
    int luasPersegi, kelilingPersegi;
    hitungLuas = hitungLuas();
    hitungLuas = hitungLuas();
    cout << "Keliling:"<< hitungKeliling();
}
```

Pada contoh tersebut, program mendefinisikan fungsi hitung luas dan hitung keliling. Dimana setiap fungsi memiliki tugas masing-masing.

Sama halnya seperti prosedur, fungsi **dapat dipanggil secara berulang kali sesuai dengan kebutuhan.**

Perbedaan Program dengan Fungsi dan Prosedur

FUNGSI

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

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

int main(){
    int luasPersegi;
    luasPersegi = hitungLuas();
    cout << "Luas: "<<hitungLuas();
}
```

PROSEDUR

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

void hitungLuas(){
    int panjang, lebar, luas;
    luas = panjang * lebar;
    cout << "Luas: "<<hitungLuas();
}

int main(){
    int luasPersegi;
    hitungLuas();
}
```

Pada contoh tersebut, terdapat sub program yang bertugas untuk menghitung luas persegi panjang dan didefinisikan ke dalam bentuk fungsi dan prosedur.

Pada program dengan fungsi pada saat pendefinisian diawali dengan **tipe data kembalian fungsi**, sedangkan pada prosedur karena tidak mengembalikan suatu nilai maka tipe data hasil ditulis dengan *keyword* **void**.

Pada badan fungsi terdapat *keyword* **return** hasil untuk mengembalikan nilai fungsi, sedangkan pada prosedur karena tidak mengembalikan suatu nilai maka tidak perlu menuliskan *return* hasil pada badan prosedur

Pada saat pemanggilan fungsi, nilai kembalian dapat ditampung dalam sebuah variabel yang bertipe data sama dengan tipe data nilai kembalian fungsi, atau dapat pula ditampilkan langsung hasilnya.

Sedangkan pada prosedur pemanggilan dilakukan dengan menuliskan nama prosedur saja. Selain itu karena prosedur tidak mengembalikan suatu nilai, hasil tidak dapat langsung ditampilkan pada bagian program lain diluar prosedur.

Program dengan Fungsi atau Prosedur?

Suatu sub program dapat didefinisikan dengan fungsi atau prosedur tergantung pada **kebutuhan program**. Fungsi digunakan untuk sub program yang mengembalikan suatu nilai, sementara prosedur digunakan untuk melakukan sekumpulan instruksi tanpa memiliki nilai kembalian.