

FUNGSI

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

Pada dasarnya fungsi maupun prosedur adalah sama. Fungsi maupun prosedur sama-sama merupakan subprogram. Selain itu, fungsi dan prosedur dapat digunakan atau dipanggil berkali-kali atau berulang-ulang dalam sebuah program. Yang membedakan fungsi dengan prosedur adalah **fungsi mengembalikan suatu nilai**.

Perhatikan ilustrasi berikut untuk mendapatkan pemahaman mengenai perbedaan fungsi dan prosedur.



Ilustrasi tersebut menggambarkan proses pengukuran sebuah koper, dalam kolom prosedur hanya dilakukan pengukuran koper tanpa adanya nilai keluaran atau hasil dari pengukuran koper tersebut, sedangkan dengan fungsi terdapat nilai keluaran yang ditampilkan yang menunjukkan hasil pengukuran. 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

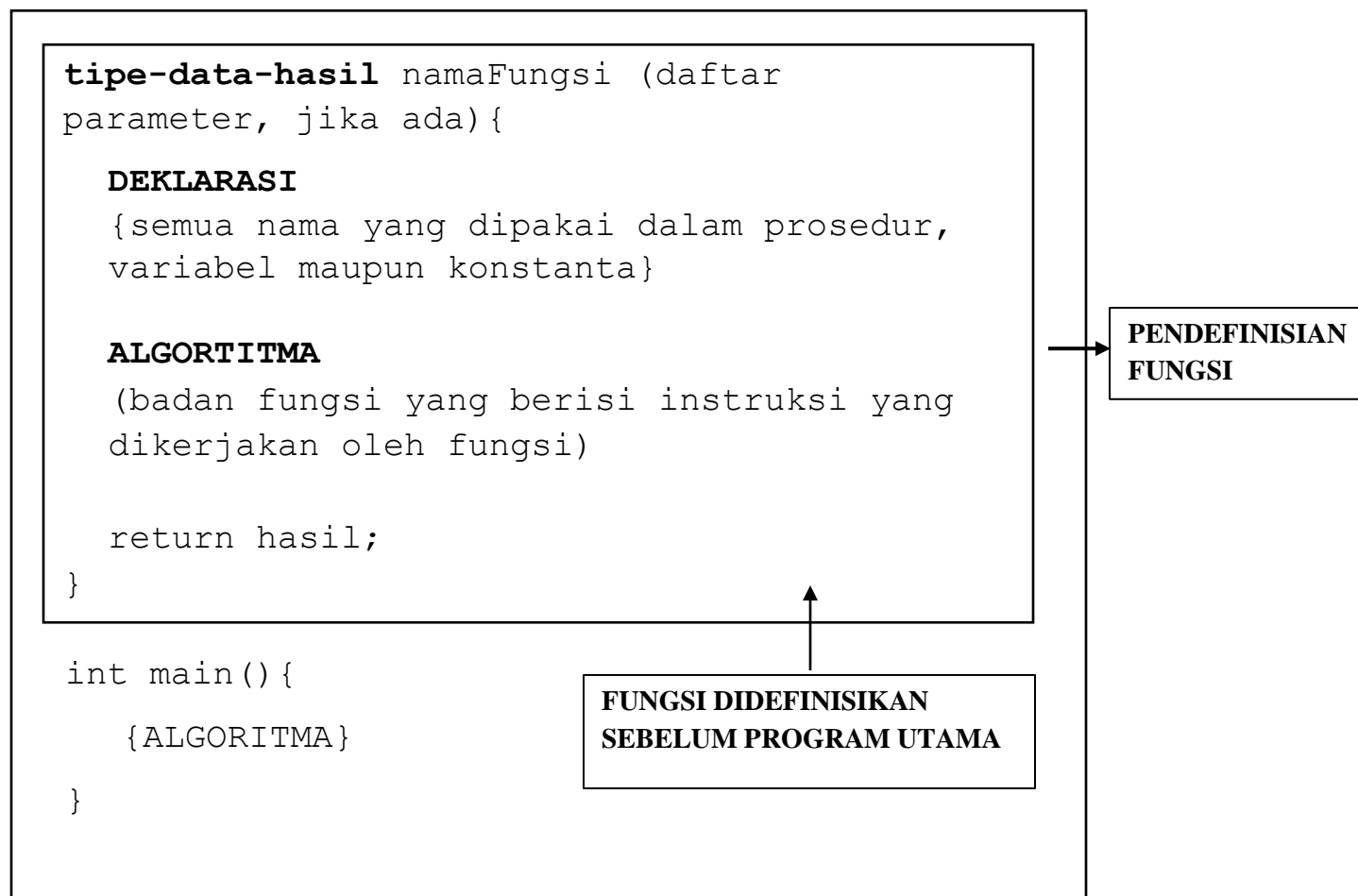


PENDEFINISIAN FUNGSI

Agar fungsi dikenali oleh program maka perlu dilakukan **pendefinisian fungsi**. Fungsi didefinisikan pada bagian atas program sebelum program utama. Dalam pembuatan sebuah fungsi, pemrogram harus mendefinisikan:

- Nama fungsi
- Tipe data yang dihasilkan oleh fungsi
- Satu atau lebih instruksi pada badan fungsi
- Keyword *return* untuk menyatakan nilai kembalian fungsi

Adapun sintaks pendefinisian fungsi adalah sebagai berikut:



Keterangan:

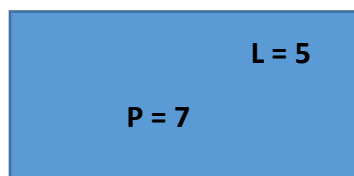
- **tipe-data-hasil**, karena fungsi mengembalikan suatu nilai maka tipe menspesifikasikan tipe data nilai yang dikembalikan oleh fungsi. Dapat berupa tipe salah satu tipe data misalnya char, int, float dll.
- **nama fungsi**, pemberian nama fungsi secara bebas. Aturan penulisan nama fungsi mengikuti aturan penulisan variabel.
- **daftar paramter**, diletakan diantara tanda kurung "()" yang terletak dibelakang nama fungsi. Parameter pada fungsi boleh ada atau tidak.
- **return**, pernyataan return hasil di dalam badan fungsi bertujuan untuk mengembalikan nilai yang dihasilkan oleh fungsi. Hasil yang dikembalikan dapat berupa konstanta, sebuah variabel, atau sebuah rumus.
- **badan fungsi**, berisi deklarasi variabel dan instruksi-instruksi yang dikerjakan fungsi. Badan fungsi ditulis di dalam tanda kurung kurawal " {} "

PARAMETER PADA FUNGSI

Biasanya fungsi membutuhkan parameter untuk melakukan pertukaran informasi dengan sub program lain atau program pemanggil untuk mendapatkan data. Parameter ditulis pada saat pendefinisian dan pemanggilan fungsi. (Penjelasan mengenai parameter dilihat di sub bab selanjutnya). Parameter adalah nama-nama variabel yang dideklarasikan pada bagian header pada saat pendefinisian fungsi. Tiap item data ditransfer oleh parameter pada program pemanggil dan akan diterima oleh fungsi sebagai nilai masukan.

CONTOH!

Membuat sebuah program yang bertugas untuk menghitung **luas** dari persegi panjang berikut!



Penyelesaian:

Fungsi membutuhkan input data berupa nilai panjang dan lebar untuk melakukan perhitungan. Sehingga nilai panjang dan lebar dapat dijadikan sebagai parameter.

1. Pendefinisian Fungsi `hitungLuas ()`

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

Pendefinisian
Fungsi

- Pada program tersebut didefinisikan fungsi bernama **hitungLuas**. Fungsi memiliki parameter berupa variabel **panjang** dan **lebar** yang akan digunakan sebagai perhitungan dalam badan fungsi.
- Pernyataan **int** pada badan awal pendefinisian fungsi menyatakan bahwa fungsi mengembalikan sebuah nilai dengan bertipe data integer. Dalam hal ini fungsi mengembalikan nilai variabel luas yang bertipe data **integer**.
- Di dalam badan fungsi, nilai yang dihasilkan oleh fungsi dikembalikan ke program pemanggil pada pernyataan

```
return luas;
```

yang dalam hal ini fungsi mengembalikan nilai hasil dari perhitungan variabel luas.



PEMANGGILAN FUNGSI

Fungsi dapat diakses dengan melakukan pemanggilan fungsi dari program pemanggil (dapat dari program utama atau dari sub program lainnya). Beberapa hal yang perlu diperhatikan pada saat pemanggilan fungsi antara lain adalah sebagai berikut:

- Penulisan nama pada saat pemanggilan dengan nama fungsi yang akan dipanggil harus sama.
- Fungsi hanya dapat dipakai sebagai bagian ekspresi bukan merupakan suatu instruksi yang dipanggil independen. Dalam ekspresi, fungsi hanya dapat diletakkan di ruas kanan
- Penulisan daftar parameter pada saat pemanggilan prosedur harus sama baik jumlah, urutan, dan typenya dengan daftar parameter pada pendefinisian prosedur
- Saat pemanggilan terjadi korespondensi antara parameter yang disertakan pada saat pemanggilan fungsi serta parameter yang dideklarasikan dalam fungsi
- Pada akhir dari eksekusi fungsi, nilai kembalian yang dihasilkan oleh fungsi dikirimkan ke program pemanggil

Pemanggilan fungsi pada program memiliki beberapa cara yaitu sebagai 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);
```

Cara 2:

Nilai kembalian fungsi dapat langsung dimanipulasi hasilnya.

Pemanggilan fungsi:

```
#include<iostream>
using namespace std;
//Pendefinisian Fungsi
int hitungLuas (int panjang, int lebar){
    int luas;
    luas = panjang * lebar;
    return luas;
}
int main(){
    int luasPersegi;
    //Pemanggilan Fungsi
    luasPersegi = hitungLuas (5, 7);
    cout<<"Luas: "<< hitungLuas (5, 7);

}
```

Tipe data keluaran fungsi dengan tipe data variabel penampung **harus sama**

Fungsi selalu dipanggil pada ruas kanan dari suatu ekspresi

Berdasarkan program tersebut fungsi `hitungLuas()` dipanggil pada pernyataan:

```
luasPersegi = hitungLuas (5, 7);
```

Pada contoh tersebut fungsi mengembalikan nilai variabel `luas` bertipe data **integer** dan pada saat pemanggilan ditampung dalam variabel `luasPersegi` yang juga bertipe data **integer**. Karena pada saat pemanggilan fungsi dengan cara ini, nilai keluaran fungsi ditampung dalam sebuah variabel yang **bertipe data sama** dengan nilai keluaran fungsi.

Selain itu pemanggilan fungsi pada pernyataan berikut dibenarkan

```
cout<<"Luas: "<< hitungLuas (5, 7);
```


dalam hal ini fungsi dipanggil dengan **langsung dimanipulasi hasilnya**. Yaitu fungsi dipanggil pada saat nilainya akan dicetak, karena fungsi mengembalikan suatu nilai sehingga fungsi hasil dapat ditampilkan secara langsung.

CARA KERJA FUNGSI

Ketika fungsi dipanggil, fungsi akan bekerja menurut mekanisme pemanggilan-pengembalian (*call-return mechanism*). Tahapan dalam mekanisme tersebut adalah:

1. Yang pertama kali dieksekusi adalah program utama, program membaca nilai variabel yang dideklarasikan oleh program utama.
2. Fungsi `hitungLuas()` dipanggil dengan mengirimkan nilai parameter 7 dan 5 yang nantinya akan diisikan ke parameter panjang dan lebar dan digunakan fungsi untuk melakukan operasi pada fungsi.

```
int main(){  
    int luasPersegi;  
    //Pemanggilan Fungsi  
    luasPersegi = hitungLuas (5, 7);  
}
```



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

3. Ketika dipanggil semua instruksi dalam badan fungsi dieksekusi, fungsi mengembalikan nilai dari hasil perhitungan variabel luas.
4. Hasil eksekusi dikembalikan ke program utama atau program pemanggil
5. Setelah semua instruksi selesai dilaksanakan, kendali program berpindah secara otomatis kembali ke instruksi sesudah pemanggilan fungsi.

Hasil Eksekusi:

Luas = 35

PERBEDAAN PROGRAM FUNGSI DENGAN 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. Berikut beberapa perbedaan antara program dengan fungsi dan prosedur:

1. Pada program dengan fungsi pada saat pendefinisian diawali dengan tipe data nilai kembalian fungsi. Sedangkan pada prosedur karena tidak mengembalikan suatu nilai maka diawali dengan keyword **void**
2. Karena fungsi mengembalikan suatu nilai maka pada badan fungsi terdapat keyword **return** untuk memberikan nilai kembalian. Sedangkan pada prosedur tidak.
3. Karena fungsi memiliki nilai kembalian, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya. Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenalan variabel yang lainnya.
4. Pada dasarnya tidak ada perbedaan yang signifikan antara fungsi dan prosedur pada Bahasa C++, hanya dibedakan dari return value. Jika suatu sub

program tersebut **memiliki nilai balik (return value)** maka sub program tersebut disebut **Fungsi (Function)** namun jika **tidak mempunyai nilai balik** maka sub program tersebut adalah **prosedur (Procedure)**.

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, sebagai contoh operasi perhitungan yang menghasilkan nilai. Sedangkan umumnya prosedur digunakan untuk melakukan proses-proses yang tidak menghasilkan nilai, seperti melakukan pengulangan, proses pengesetan nilai ataupun yang lainnya.