# Bab 12

# SUBPROGRAM

(FUNGSI)

Dosen : Sulistyowati, ST., M.Kom.

#### Pendahuluan

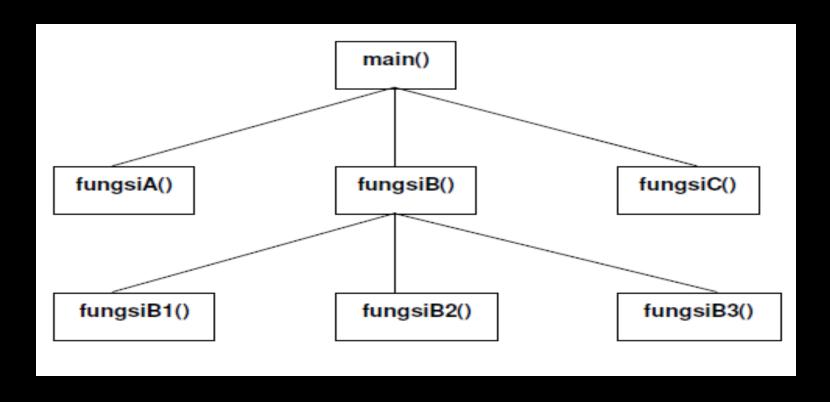
- Pemrograman yang baik harus bersifat modular agar suatu masalah program yang besar dan kompleks dapat dipecah-pecah menjadi bagian-bagian yang lebih kecil dan sederhana.
- Fungsi/function adalah bagian dari program yang memiliki nama tertentu, digunakan untuk mengerjakan suatu pekerjaan tertentu dan letaknya dipisahkan dari bagian program yang menggunakan fungsi tersebut serta di eksekusi ketika dipanggil dari bagian lain dalam suatu program.

Keuntungan menggunakan fungsi, yaitu:
☐ Menghemat ukuran program.
□ Dapat dikerjakan oleh beberapa orang sehingga koordinasi mudah.
□ Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu saja.
□ Modifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa mengganggu program keseluruhan.
☐ Mempermudah dokumentasi.
□ Reusability: Suatu fungsi dapat digunakan kembali oleh program atau fungsi lain.

- Fungsi merupakan bagian dari kode program yang terpisah dari program utama.
  - Bagain kode program ini di-execute jika diperlukan untuk melakukan tindakan khusus dalam program.
- Fungsi banyak dilibatkan dalam program dengan tujuan untuk mengurangi duplikasi pengkodean dan untuk mempermudah pemahaman.
- Salah satu fungsi yang sering dilibatkan dalam aplikasi berbasis console yaitu main().

```
Misalnya :
      Fungsi A()
      Fungsi B()
      Fungsi C()
   Didalam fungsi main(), terdapat pemanggilan
   ketiga fungsi tersebut sbb:
             main()
                  Fungsi A(); Fungsi B(); Fungsi C();
```

Konsep modularitas program:



- Jenis-jenis fungsi:
  - 🗖 Berdasarkan user yang mendefinisikan, dibedakan menjadi :
    - 1. *User-define function* > yaitu fungsi yang didefinisikan sendiri oleh pemrogram.
    - 2. Built-in function > yaitu fungsi yang sudah disediakan oleh compiler.
  - Berdasarkan data return value-nya, dibedakan menjadi :
    - 1. Return-Value Function (RVF) → yaitu fungsi yang memberikan nilai balik.
    - 2. Non Return-Value Function (NRVF) → yaitu fungsi yang tidak memberikan/mempunyai nilai balik.

### RVF - Prototipe Fungsi

- Sebelum fungsi digunakan, fungsi harus dideklarasikan terlebih dahulu
   -> yaitu berguna untuk memberitahukan compiler jumlah dan tipe
   parameter yang diterima dan nilai balik fungsi (bila ada).
   Deklarasi fungsi ini disebut sebagai prototipe fungsi.
- Prototipe fungsi ini harus dituliskan/diletakkan sebelum program utama dan sebelum pendefinisian fungsi.
- Manfaat dari prototipe fungsi adalah untuk menjamin tipe argumen yang dilewatkan pada pemanggilan fungsi benar-benar sesuai. Tanpa prototipe fungsi, sangatlah mudah bagi pemrogram untuk melakukan kesalahan tanpa sengaja dalam melewatkan argumen. Misalnya: melewatkan argumen bertipe float pada fungsi yang seharusnya meminta tipe data int.

# RVF - Prototipe Fungsi (con't)

Bentuk deklarasi fungsi/prototipe fungsi :
 Re\_type nama\_fungsi(argumen);

#### Dimana:

- \* Re\_type = tipe dari data yang akan dikembalikan/dihasilkan oleh function.
- \* nama\_fungsi = nama fungsi yang dibuat.
- \* argumen = daftar parameter yang akan diolah dalam fungsi tersebut. Argumen boleh ada, boleh tidak (dispesifikasikan sesuai kebutuhan).

# RVF - Prototipe Fungsi (con't)

- Parameter-parameter yang terdapat dalam argumen berfungsi sebagai masukan. Dan masukan tersebut akan diproses sehingga dihasilkan nilai tertentu sesuai dengan tipe datanya (*re\_type*).
- Untuk penulisan argumen :
   tipe\_data1 nama\_parameter1, tipe\_data2 nama\_parameter2, ....
- Jika suatu fungsi tidak mempunyai argumen, maka dituliskan :
   Re\_type nama\_fungsi ();

Bentuk deklarasi diatas biasanya digunakan pada saat mendeklarasikan sebuah fungsi yang tidak memiliki nilai balik / return value (akan dibahas tersendiri) atau untuk mendeklarasikan sebuah fungsi yangmana parameter-parameter didalam tubuh fungsi merupakan suatu variabel alobal.

# RVF - Definisi Fungsi

- Contoh penulisan prototipe fungsi :
  - ✓ double kuadrat (int x);
  - I float luas (float alas, float tinggi);
- Setelah prototipe fungsi dideklarasikan, selanjutnya adalah mendefinisikan fungsi/membuat fungsi secara detail. Suatu fungsi disebut juga subprogram, oleh karena itu strukturnya juga sama dengan struktur program utama.
- Bentuk definisi fungsi : Re\_type nama\_fungsi (argumen)
   { statement-statement; }

# RVF – Pemanggilan Fungsi

- Suatu fungsi yang sudah didefinisikan, bisa dipakai/dipanggil di bagian lain dari program, contohnya dipanggil di program utama atau di fungsi lain atau bahkan didalam fungsi itu sendiri (rekursif).
- Bentuk pemanggilan fungsi :

#### nama\_fungsi (argumen)

Parameter-parameter pada argumen, cukup dituliskan nama\_parameternya saja (tanpa menyertakan tipe\_data\_parameter).

Contoh:

cout« luas(alas,tinggi); info();

# RVF – Implementasi Fungsi

#### Contoh 1:

```
#include <iostream.h>
#include <comio.h>
float luas(float panjang,lebar); -> DEKLARASI FUNGSI
main()
    float p,1;
    cout << "HITUNG LUAS PERSEGI PANJANG" << endl;
                  Input Panjang : "; cin>>p;
    cout << "
    cout << " Input Lebar : "; cin>>1;
    cout << "
                  Luasnya
                                  : " << luas(p,1) << endl;
                                                PEMANGGILAN FUNGSI
float luas(float panjang, float lebar)
    float luas persegi;
                                              DEFINISI FUNGSI
    luas persegi = panjang * lebar;
    return luas persegi;
```

# RVF - Implementasi Fungsi (con't)

• Bentuk penulisan fungsi diatas, dapat diganti dengan :

```
float luas(float panjang, float lebar)
{
    return (panjang * lebar);
}
```

 Perintah return didalam fungsi digunakan untuk memberikan nilai balik fungsi, yaitu mengembalikan hasil operasi di sebelah kanannya ke perintah pemanggilan fungsi.

### Variabel Global dan Lokal

- Variabel Global, adalah variabel yang dideklarasikan oleh program utama (sehingga variabel ini dikenal oleh seluruh bagian program, baik itu program utama atau fungsi-fungsi yang lain).
- Variabel Lokal, adalah variabel yang dideklarasikan oleh fungsi (hanya dikenal didalam fungsi tersebut).

#### Catatan :

- 1. Jika variabel hanya digunakan dalam fungsi, maka sebaiknya dideklarasikan sebagai variabel lokal.
- 2. Usahakan memakai variabel global seminim mungkin.

#### Contoh penggunaan variabel global dan lokal:

```
#include <iostream.h>
#include <conio.h>
void cetak();
main()
    int a=10;
    cout<<"Nilai a = "<<a<<endl;</pre>
    cetak();
void cetak()
    a++;
    cout<<"Nilai a = "<< a << endl;</pre>
```

- Ketika program di atas dicompile, akan terdapat error yaitu variabel a dalam fungsi cetak() undefined. Artinya bahwa variabel a tidak dikenal dalam fungsi cetak().
   Variabel a hanya dikenal dalam program utama/fungsi main() saja.
- Maka dalam hal ini variabel a disebut variabel lokal (hanya dikenal dalam fungsi yang mendefinisikan a tersebut).

• Sehingga program diatas perlu dirubah :

```
#include <iostream.h>
#include <conio.h>
int a;
void cetak();
main()
    a=10;
    cout<<"Nilai a = "<<a<<end1;</pre>
     cetak();
void cetak()
    a++;
    cout<<"Nilai a = "<<a<<endl;</pre>
```

Apabila program di atas dijalankan maka akan tampil: Nilai a = 10Nilaia = 11Pada program di atas, variabel a disebut variabel global karena variabel tersebut dapat dikenali di setiap fungsi yang ada.

Bagaimana dengan yang ini ??????

```
#include <iostream.h>
#include <conio.h>
void cetak();
main()
       int a=10;
       cout << "Nilai a = " << a << endl;
       cetak();
void cetak()
       int a;
       cout << "Nilai a = " << a << endl;</pre>
```

#### Parameter

- Parameter digunakan untuk pertukaran informasi antara fungsi dengan titik dimana fungsi tersebut dipanggil.
- Parameter ada 2 jenis :
  - 1. Parameter formal, dideklarasikan dalam fungsi.
  - 2. Parameter aktual, disertakan pada waktu pemanggilan fungsi.
- Parameter formal dan aktual saling berpasangan, sehingga jumlah dan tipe data kedua parameter ini HARUS sama. Tapi untuk nama variabel parameter boleh tidak sama (dan dianjurkan untuk dibedakan).

Ketika dipanggil, parameter aktual menggantikan parameter formal.

#### Parameter (con't)

Contoh implementasi parameter formal & aktual :

```
#include <iostream.h>
#include <conio.h>
float luas(float alas, float tinggi);
main()
        int a = 10.5; int t = 11;
        cout<<"HITUNG LUAS SEGITIGA"<<endl;</pre>
        cout<<"Panjang alas : "<<a<<end1;</pre>
        cout<<"Tinggi : "<<t<endl;</pre>
        cout<<"Luasnya : "<<luas(a,t)<<endl;</pre>
float luas(float alas, float tinggi)
        float luas segitiga;
        luas segitiga = alas * tinggi * 0.5;
        return luas segitiga;
```

### Non Return-Value Function

• Adakalanya suatu fungsi tidak perlu memiliki nilai balik (return value). Misalnya fungsi yang hanya dimaksudkan untuk menampilkan suatu keterangan saja.

Pada fungsi seperti ini, tipe nilai balik fungsi yang diperlukan adalah void.

#### Contoh:

# RVF - Implementasi Fungsi (con't)

#### Contoh:

```
#include <iostream.h>
#include <conio.h>
void info();
main()
    info();
void info()
    cout<<"PEMROGRAMAN TERSTRUKTUR";</pre>
    //return;
```

#### Variabel Statis

- Variabel sstatis merupakan variabel yang menahan nilainya sementara pemanggilan fungsi berlangsung. Variabel ini dapat dideklarasikan di dalam fungsi tertentu maupun di luar fungsi.
- Yaitu dengan menambahkan kata kunci static di depan variabel.

```
Contoh:
     int a=10;
     main()
          coba(); coba(); coba();
          cout<<"A semula = "<<a;</pre>
     void coba()
          static int a=25;
          a++;
          cout<<"A = "<<a<<endl;
```

### Soal Latihan

Rancanglah flowchart dan program C++ untuk :

- 1. Membuat function yang akan memberikan nilai 1 jika nilai parameter yang dimasukkan huruf 'a', dan akan memberikan nilai 0 jika nilai parameter yang dimasukkan selain huruf 'a'.
- 2. Membuat function untuk mencari nilai fungsi

$$f(x) = 2x^2 - 3x + 1$$

Parameter functionnya adalah nilai x (tipe data float/double). Return valuenya adalah nilai f(x).

### Soal Latihan (con't)

3. Membuat function untuk membuat tampilan seperti di bawah :

\* \*

\* \* \*

\* \*

\*

4. Membuat fungsi untuk mencetak deret Fibonacci sebanyak x suku.

#### 5. Membuat 2 fungsi :

- fungsi 1 untuk mencari deret bilangan prima mulai dari x sampai y
- fungsi 2 untuk mencari hasil penjumlahan deret tersebut.

#### 6. Membuat menu:

- Menu 1 -> mencari deret bilangan non prima
- Menu 2 → mencari nilai kelipatan dari x dan y sebanyak 10 bilangan, mulai dari z.
- Menu 3 -> mencari KPK dari sebuah bilngan yang diinputkan.

- Menu 4 -> menentukan besar diskon dari sebuah total pembelian dengan ketentuan :
  - Total pembelian > 1 jt, maka diskon 30% + point = 30
  - Total pembelian antara > 500 rb 1 jt, maka diskon 20% + point = 20
  - Total pembelian antara > 300 rb 500 rb, maka diskon10% + point = 10

Kemudian tentukan hadiah untuk total point :

- jika point < 20, maka hadiah "voucher belanja 50 rb"
- jika point antara 20 30, maka hadiah "voucher belanja 75 rb"

Catatan : masing-masing menu dibuatkan fungsi sendiri-sendiri

- 7. Membuat fungsi untuk menentukan hasil penjumlahan 2 buah matrik
- 8. Membuat fungsi untuk menginputkan data mahasiswa (sebanyak n mahasiswa). Data mahasiswa yang disimpan adalah : nama, npm, alamat, umur, jkel,IPK

9. Membuat fungsi untuk mencari nilai rata-rata mahasiswa dari 4 buah nilai, yaitu : nilai UTS, UAS, Tugas dan Kuis. Sedangkan mahasiswanya sebanyak x mahasiswa.

#### PENTING:

Semua inputan dilakukan di fungsi utama (main()).