

MI2101

Praktikum Teknik Komputasi

Modul 3

Fadjar Fathurrahman

2018

1 Tujuan

- Mampu membuat program C++ sederhana dengan memanfaatkan subprogram
- Mampu membedakan antara *pass-by-value* dan *pass-by-pointer*
- Mampu membuat kurva parametrik sederhana (Lissajous)

2 Perangkat lunak yang diperlukan

- Linux OS
- CodeBlocks yang telah dikonfigurasi untuk kompiler GNU C/C++
- Terminal emulator dengan bash sebagai shell (baris perintah)
- Editor teks seperti gedit
- Perangkat lunak untuk membuat plot/grafik:
 - Spreadsheet program seperti Microsoft Excel atau Libre Office Calc
 - Gnuplot

3 Akar persamaan kuadrat dengan menggunakan subprogram (fungsi)

Pada bagian ini kita akan kembali meninjau program untuk menghitung akar persamaan kuadrat. Program ini akan kita ubah sedikit, yaitu dengan memanfaatkan subprogram (fungsi). Kita akan membagi langkah-langkah yang ada pada program terdahulu dengan menggunakan fungsi. Program lebih panjang daripada program sebelumnya (tanpa menggunakan fungsi) karena bertujuan untuk menguji pemahaman Anda tentang fungsi dalam bahasa C++.

```
#include <iostream>
#include <cmath>

using namespace std;

void selamat_datang();
float baca_input_a();
float baca_input_b();
float baca_input_c();
void tampilkan_abc(float a, float b, float c);
```

```

float hitung_diskriminan(float a, float b, float c);
void hitung_dan_tampilkan_akar(float a, float b, float c);

int main()
{
    selamat_datang();

    float a, b, c;
    a = baca_input_a();
    b = baca_input_b();
    c = baca_input_c();

    tampilkan_abc(a, b, c);

    hitung_dan_tampilkan_akar(a, b, c);

    return 0;
}

void selamat_datang()
{
    cout << "Selamat datang" << endl;
    cout << endl;
    cout << "Saya adalah program untuk menghitung akar-akar persamaan kuadrat" << endl;
    cout << endl;
    cout << "a*x^2 + b*x + c = 0" << endl;
    cout << endl;
    cout << "Copyright: Jaka Sembung (2018)" << endl;
    cout << endl;
}

float baca_input_a()
{
    float a;
    for(;;) {
        cout << "Masukkan nilai a: ";
        cin >> a;
        if( fabs(a) <= 0 ) {
            cout << "Nilai yang Anda masukkan tidak valid" << endl;
            cout << "Mohon masukkan nilai a yang valid (a != 0)" << endl;
        }
        else break;
    }
    return a;
}

// mirip dengan baca_input_a(), namun tanpa pengecekan nilai
float baca_input_b()
{...}

float baca_input_c()
{...}

void tampilkan_abc(float a, float b, float c)
{
    cout << endl;
    cout << "Berikut ini adalah nilai a, b, dan c yang telah saya baca" << endl;
    cout << endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    cout << "c = " << c << endl;
    cout << endl;
}

void hitung_dan_tampilkan_akar(float a, float b, float c)

```

```

{
    ...
    D = hitung_diskriminan(a,b,c);
    ...
    if( D >= 0.0 ) {
        ...
    }
    else {
        ...
    }
}

float hitung_diskriminan(float a, float b, float c)
{ ... }

```

3.1 Pertanyaan dan tugas

1. Jelaskan fungsi/subprogram yang digunakan pada program.
2. Sebutkan apa yang menjadi input dan output dari setiap fungsi tersebut.
3. Misalkan deklarasi fungsi hitung_diskriminan diubah menjadi

```
float hitung_diskriminan(float coef_a, float coef_b, float coef_c);
```

Bagian mana saja yang harus diubah ?

4. Perhatikan program berikut ini.

```

#include <iostream>

using namespace std;

void func1(int a, int b);
void func2(int& a, int& b);

int main()
{
    int a = 1, b = 3;

    cout << "a = " << a << endl;
    cout << "b = " << b << endl;

    func1(a, b);
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;

    func2(a, b);
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    return 0;
}

void func1(int a, int b)
{
    a = 33;
    b = 44;
}

void func2(int &a, int &b)
{
    a = 99;
    b = 88;
}

```

```
}
```

Tuliskan keluaran dari program di atas. Jelaskan apa perbedaan antara dua fungsi `func1` dan `func2`. Apa yang terjadi jika pada fungsi `main` urutan pemanggilan fungsi dibalik ? (`func2` kemudian baru `func1`). Catatan: fungsi `func1` disebut menggunakan *pass-by-value* dan fungsi `func2` disebut menggunakan *pass-by-reference*.

5. Tinjau kembali fungsi untuk mencari akar-akar persamaan kuadrat yang telah dibuat pada bagian awal modul ini. Kita akan mempersingkat fungsi input yang awalnya berjumlah tiga fungsi, yaitu `baca_input_a`, `baca_input_b`, dan `baca_input_c` menjadi satu fungsi yang dideklarasikan sebagai berikut.

```
void baca_input(float &a, float &b, float &c);
```

Dengan menggunakan *pass-by-reference* buatlah implementasi dari fungsi tersebut dan gunakan dalam program perhitungan akar-akar persamaan kuadrat.

6. Buatlah program seperti pada Modul 2, untuk membuat kurva Lissajous, yang didefinisikan oleh persamaan parametrik berikut.

$$x(t) = A \sin(at + \phi) \quad (1)$$

$$y(t) = B \cos(bt) \quad (2)$$

dengan A , B , a , b , dan ϕ adalah parameter. Bagian dari program Anda untuk menghitung x dan y dibuat dalam sebuah fungsi dengan deklarasi sebagai berikut.

```
void calc_lissajous(  
    double A, double B, double a, double b, double phi, // parameter  
    double t, // input t  
    double &x, double &y); // output x dan y
```

Hasilkan data x dan y pada selang $0 \leq t \leq 2\pi$ dan plot kurva yang dihasilkan dengan parameter-parameter berikut.

- $A = 1.0$, $B = 1.0$, $a = 1.0$, $b = 1.0$, $\phi = 0.0$
- $A = 1.0$, $B = 1.0$, $a = 1.0$, $b = 2.0$, $\phi = 0.0$
- $A = 1.0$, $B = 1.0$, $a = 1.0$, $b = 3.0$, $\phi = 0.0$
- $A = 2.0$, $B = 1.0$, $a = 4.0$, $b = 3.0$, $\phi = 0.0$
- $A = 2.0$, $B = 1.0$, $a = 4.0$, $b = 3.0$, $\phi = 0.3$