

Laporan Praktikum 3 - Praktikum Pemrograman 1

Ravif Gayuh Wicaksono

24/540583/PA/22953

September 10, 2024

1 Soal Pertama : Mencari Akar dari Persamaan Kuadrat

Diberikan sebuah masalah untuk menyelesaikan program yang mampu menghitung akar-akar dari persamaan kuadrat dengan bentuk:

$$ax^2 + bx + c$$

1.1 Source File

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int a, b, c;
6     float x1, x2;
7     float disc;
8
9     cout << "Input the value of a: ";
10    cin >> a;
11    cout << "Input the value of b: ";
12    cin >> b;
13    cout << "Input the value of c: ";
14    cin >> c;
15
16    disc = pow(b,2) - (4*a*c);
17    if(disc > 0){
18        x1 = ((-b) + sqrt(disc))/2*a;
19        x2 = ((-b) - sqrt(disc))/2*a;
20        cout << "x1 = " << setprecision(2) << fixed << x1 << endl;
21        cout << "x2 = " << setprecision(2) << fixed << x2 << endl;
22        return 0;
23    } else if (disc == 0){
24        x1 = (-b)/2*a;
25        cout << "x = " << setprecision(2) << fixed << x1 << endl;
26        return 0;
27    } else {
28        cout << "**There are no real roots (roots are complex)**";
29        return 0;
30    }
31 }
```

1.2 Deklarasi

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int a, b, c;
6     float x1, x2;
7     float disc;
```

Mendeklarasikan beberapa variabel dengan masing-masing tipe data:

- **a, b, dan c** (Tipe Data Integer): Koefisien dari $ax^2 + bx + c$
- **disc** (Tipe Data Float): Variabel untuk menyimpan diskriminan
- **x1 dan x2** (Tipe Data Float): Variabel untuk menyimpan akar dari persamaan kuadrat

1.2.1 Masukan

```
1 cout << "Input the value of a: ";
2 cin >> a;
3 cout << "Input the value of b: ";
4 cin >> b;
5 cout << "Input the value of c: ";
6 cin >> c;
```

User diminta untuk memasukan koefisien a, b, dan c dari persamaan kuadrat yang mereka miliki.

1.3 Prosesi Program

```
1 disc = pow(b,2) - (4*a*c);
```

Menghitung diskriminan dengan formula diskriminan yang berlaku yaitu:

$$\frac{-b^2}{4ac}$$

Selanjutnya, program menggunakan percabangan untuk menentukan formula yang akan digunakan dalam menentukan akar-akar dari persamaan kuadrat, yaitu:

- **Percabangan 1:** Jika diskriminan lebih besar dari 0 (positif)
- **Percabangan 2:** Jika diskriminan sama dengan 0
- **Percabangan 3:** Jika diskriminan lebih kecil dari 0 (negatif)

```
1 if(disc > 0){
2     ...
3 }else if(disc == 0){
4     ...
5 }else{
6     ...
7 }
```

Lalu masukan operasi yang diperlukan pada tiap-tiap percabangan:

- **Percabangan 1:**

$$\frac{-b \pm \sqrt{\Delta}}{2a}$$

- **Percabangan 2:**

$$\frac{-b}{2a}$$

- **Percabangan 3:** Tidak ada operasi yang perlu dilakukan

```
1  if (disc > 0){
2      x1 = ((-b) + sqrt(disc))/2*a;
3      x2 = ((-b) - sqrt(disc))/2*a;
4      ...
5  } else if (disc == 0){
6      x1 = (-b)/2*a;
7      ...
8  } else {
9      ...
10 }
```

1.4 Keluaran

Pada tiap percabangan, beri keluaran:

- **Percabangan 1:** keluarkan variabel x1 dan x2 bersamaan
- **Percabangan 2:** keluarkan nilai x satu-satunya
- **Percabangan 3:** keluarkan string yang menunjukan bahwa tak ada akar yang berupa bilangan ril

```
1  if (disc > 0){
2      x1 = ((-b) + sqrt(disc))/2*a;
3      x2 = ((-b) - sqrt(disc))/2*a;
4      cout << "x1 = " << setprecision(2) << fixed << x1 << endl;
5      cout << "x2 = " << setprecision(2) << fixed << x2 << endl;
6      return 0;
7  } else if (disc == 0){
8      x1 = (-b)/2*a;
9      cout << "x = " << setprecision(2) << fixed << x1 << endl;
10     return 0;
11 } else {
12     cout << "**There are no real roots (roots are complex)**";
13     return 0;
14 }
```

1.5 Test Code

1.5.1 Tes Pertama ($\Delta > 0$)

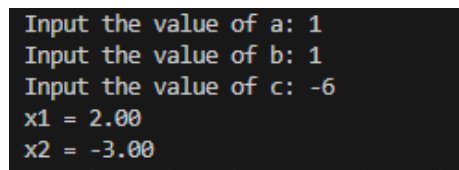
Input

- A: 1
- B: 1
- C: -6

Output Diharapkan

- X1: 2,00
- X2: 3,00

Output Riil



```
Input the value of a: 1
Input the value of b: 1
Input the value of c: -6
x1 = 2.00
x2 = -3.00
```

Figure 1: Output Riil Tes Pertama

1.5.2 Tes Kedua ($\Delta = 0$)

Input

- A: 1
- B: -2
- C: 1

Output Diharapkan

- X: 1,00

Output Riil

```
Input the value of a: 1
Input the value of b: -2
Input the value of c: 1
x = 1.00
```

Figure 2: Output Riil Tes Kedua

1.5.3 Tes Ketiga ($\Delta < 0$)

Input

- A: 5
- B: 4
- C: 10

Output Diharapkan

- There are no real roots (roots are complex)

Output Riil

```
Input the value of a: 5
Input the value of b: 4
Input the value of c: 10
**There are no real roots (roots are complex)**
```

Figure 3: Output Riil Tes Ketiga

1.5.4 Kesimpulan

Dari tiga tes yang dilakukan dengan koefisien yang berbeda, dengan diskriminan yang berbeda. Tes pertama dengan diskriminan > 0 , tes kedua dengan diskriminan $= 0$, dan tes ketiga dengan diskriminan < 0 . Didapatkan keluaran yang sama seperti keluaran yang diharapkan pada tiga test code tersebut, maka bisa dikatakan bahwa program ini tidak ada kendala.

2 Soal Kedua : Mencari Bilangan Terbesar

Diberikan sebuah masalah untuk menemukan bilangan terbesar dari tiga bilangan yang dimasukkan oleh pengguna.

2.1 Source File

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int a, b, c, maks;
6     cout << "Masukkan bilangan pertama: ";
7     cin >> a;
8     cout << "Masukkan bilangan kedua: ";
9     cin >> b;
10    cout << "Masukkan bilangan ketiga: ";
11    cin >> c;
12
13    maks = a;
14
15    if(b > maks){
16        maks = b;
17    }
18    if(c > maks){
19        maks = c;
20    }
21
22    cout << "Bilangan terbesar adalah: " << maks;
23    return 0;
24
25 }
```

2.2 Deklarasi

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int a, b, c, maks;
```

Mendeklarasikan beberapa variabel dengan masing-masing tipe data:

- **a, b, dan c** (Tipe Data Integer): Variabel bilangan bulat yang akan di prosesi oleh program
- **maks** (Tipe Data Integer): Variabel bilangan bulat untuk menyimpan nilai maksimum variabel.

2.3 Masukan

```
1 cout << "Masukkan bilangan pertama: ";
2 cin >> a;
3 cout << "Masukkan bilangan kedua: ";
4 cin >> b;
5 cout << "Masukkan bilangan ketiga: ";
6 cin >> c;
```

User diminta untuk memasukkan variabel bilangan bulat a, b dan c sesuai dengan kehendak mereka.

2.4 Prosesi Program

```
1 maks = a;
```

Memasukkan nilai a pada variabel maksimum.

```
1 if (b > maks){
2     maks = b;
3 }
4 if (c > maks){
5     maks = c;
6 }
```

Program melakukan percabangan:

- **Percabangan 1** : Jika nilai b lebih besar dari nilai variabel maksimum, maka variabel b menjadi variabel maksimum.
- **Percabangan 2** : Jika nilai c lebih besar dari nilai variabel maksimum, maka variabel c menjadi variabel maksimum.

2.5 Keluaran

```
1 cout << "Bilangan terbesar adalah: " << maks;
2 return 0;
```

Program mengeluarkan nilai maksimum yang telah disimpan.

2.6 Test Code

2.6.1 Tes Pertama ($\Delta > 0$)

Input

- A: -20
- B: 4
- C: -6

Output Diharapkan

- Maks: 4

```
Masukkan bilangan pertama: -20
Masukkan bilangan kedua: 4
Masukkan bilangan ketiga: -6
Bilangan terbesar adalah: 4
```

Figure 4: Output Rill Pertama

3 Pertanyaan

3.1 Soal 1 : Apa saja instruksi percabangan dalam C++

Pada dasarnya, instruksi percabangan dalam C++ terbagi dalam enam bagian, yaitu:

- Percabangan If
- Percabangan If/Else
- Percabangan If/Else/If
- Percabangan Switch/Case
- Percabangan Ternary Operator
- Percabangan Nested If

3.2 Soal 2 : Jelaskan instruksi diatas

- **Percabangan If**
Percabangan if merupakan percabangan yang hanya memiliki satu blok pilihan saat kondisi bernilai benar.
- **Percabangan If/Else**
Percabangan if/else merupakan percabangan yang memiliki dua blok pilihan. Blok pilihan pertama untuk kondisi benar, dan pilihan kedua untuk kondisi salah (else).
- **Percabangan If/Else/If**
Percabangan if/else/if merupakan percabangan yang memiliki lebih dari dua blok pilihan. Sehingga kita mampu untuk menambahkan sampai sangat banyak blok pilihan yang kita inginkan.
- **Percabangan Switch/Case**
Sebenarnya percabangan switch/case tak terlalu berbeda dengan if/else/if, tetapi switch/case lebih fungsional jika digunakan untuk program yang membutuhkan pilihan dengan tepat satu angka.

- **Percabangan Ternary Operator**

Percabangan menggunakan ternary operator merupakan bentuk lain dari percabangan if/else. Bisa dibilang ternary operator Bentuk singkatnya dari if/else.

- **Percabangan Nested If**

Percabangan Nested If merupakan percabangan yang memiliki percabangan di dalamnya, jadi dalam percabangan if, diberikan percabangan baru lagi, hingga menjadikannya percabangan bersarang.