1.3 PRAKTIK

1.3.1 Akar Kuadrat

Pada bagian ini, akan dipelajari cara untuk mengolah proses aritmatika menggunakan operator pada Bahasa pemrograman java, dan penggunaan kondisional menggunakan Bahasa pemrograman java.

1.3.1.1 Soal

Buatlah sebuah program yang dapat menghitung akar kuadrat dari sebuah persamaan kuadrat (PK)

$$PK = ax^2 + bx + c$$

Dengan variabel a, b dan c diketahui nilainya. Pencarian x_1 dan x_2 dapat menggunakan rumus abc:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Selain mencari kedua akar persamaan kuadrat, tentukan apakah $x_1 > x_2$ atau $x_1 < x_2$ atau $x_1 = x_2$.

1.3.1.2 Langkah Penyelesaian

Buat outline dari sebuah program java yang siap run:

```
public class AkarKuadrat {
    public static void main(String[] args) {

    }
}
```

Deklarasikan variabel a, b dan c. Dalam hal ini a diisi nilai 1, b diisi nilai -3 dan c diisi nilai 2.

```
public class AkarKuadrat {
    public static void main(String[] args) {
        double a=1, b=-3, c = 2;
    }
}
```

Double digunakan sebagai tipe data dari a,b dan c karena akan digunakan dalam operasi yang akan menghasilkan koma (akar dan pembagian). Alternatif selain menggunakan double, dapat juga menggunakan tipe data float. Pembagian integer yang dimasukkan ke dalam variabel bertipe double tetap akan menghasilkan nilai bilangan bulat. Setelah mendeklarasikan nilai a-c, deklarasikan x1 dan x2 sebagai Hitung akar-akar persamaan dengan rumus abc:

```
public class AkarKuadrat {
    public static void main(String[] args) {
        double a=1, b=-3, c = 2;
        double x1 = 0,x2 = 0;
        x1 = (-b+(Math.sqrt(b*b-4*a*c)))/2*a;
        x2 = (-b-(Math.sqrt(b*b-4*a*c)))/2*a;
        System.out.println("Akar-akar persamaan adalah x1: "+x1+" dan x2:"+x2);
    }
```

```
}
```

Setelah mendapatkan nilai x1 dan x2, tambahkan pengkondisian untuk menentukan apakah nilai x1 dan x2 lebih besar, kecil atau sama. Pengkondisian dituliskan sebagai berikut:

```
        if(x1 < x2){
            System.out.println("x1 lebih kecil dari x2");
        }else if (x1 > x2){
            System.out.println("x1 lebih besar dari x2");
        } else{
            System.out.println("x1 bernilai sama dengan x2");
        }
```

Simpan file di atas sesuai dengan nama kelas (AkarKuadrat.java), lalu compile pada command prompt dengan menuliskan "javac AkarKuadrat.java" dan jalankan dengan "java AkarKuadrat".

1.3.1.3 Solusi Lengkap

```
public class AkarKuadrat {
    public static void main(String[] args) {
        double a=1, b=-3, c = 2;
        double x1 = 0,x2 = 0;
        x1 = (-b+(Math.sqrt(b*b-4*a*c)))/2*a;
        x2 = (-b-(Math.sqrt(b*b-4*a*c)))/2*a;
        System.out.println("Akar-akar persamaan adalah x1: "+x1+" dan x2:"+x2);
        if(x1 < x2){
            System.out.println("x1 lebih kecil dari x2");
        }else if (x1 > x2){
            System.out.println("x1 lebih besar dari x2");
        } else{
            System.out.println("x1 bernilai sama dengan x2");
        }
    }
}
```

1.3.1.4 Pengamatan

1. Apa yang terjadi jika pada saat menghitung nilai x1, syntax yang diberikan adalah sebagai berikut:

```
x1 = -b+(Math.sqrt(b*b-4*a*c))/2*a;
x2 = -b-(Math.sqrt(b*b-4*a*c))/2*a;
```

2. Ubahlah program di atas, sehingga saat menampilkan keterangan, hanya ada 2 keterangan, yaitu "x2 lebih kecil sama dengan x1" atau "x2 lebih besar dari x1".

1.3.2 Palindrom

Pada bagian ini, akan dipelajari cara untuk menggunakan kondisional dan perulangan pada Bahasa pemrograman java.

1.3.2.1 Soal

Tentukanlah suatu kata bersifat palindrom atau bukan dengan menggunakan perulangan dan kondisional. Suatu kata bersifat palindrom jika kata tersebut mempunyai kata yang sama jika dibaca dari depan ke belakang dan dari belakang ke depan.

Contoh kata palindrom yaitu: “NABABAN”, “KATAK”, “ORO”. Penentuan palindrom pada penyelesaian masalah memperhatikan huruf besar dan huruf kecil. Jadi “Katak” bukanlah palindrom, tapi “katak” adalah kata palindrom

1.3.2.2 Langkah Penyelesaian

Deklarasikan sebuah variabel bertipe string untuk menampung kata yang akan dicek.

```
public class Palindrom {
    public static void main(String[] args) {
        String kata = "Katak";
    }
}
```

Lakukan perulangan dari belakang kata ke depan, ambil setiap huruf per-perulangan dan simpan ke dalam sebuah variabel bernama “temp”.

```
        String temp="";
        for (int i = kata.length()-1; i >= 0 ; i--) {
            temp = temp + kata.charAt(i);
        }
        System.out.println("Kata Balikan: "+temp);
```

Perhatikan bahwa looping merupakan decrement (penurunan hitungan). Gunakan fungsi length() dari string untuk mengetahui panjang kata. Dan charAt() untuk mengambil karakter ke-i. Fungsi length() menghasilkan nilai 5 untuk kata “Katak”, dengan posisi karakter ke-i adalah sebagai berikut:

K	a	t	a	k
i=0	i=1	i=2	i=3	i=4

Setelah mendapatkan kata balikan, berikan kondisional untuk menentukan apakah kata tersebut palindrom atau tidak.

```
        if(kata.equals(temp)){
            System.out.println("Kata Palindrom");
        }else{
            System.out.println("Bukan Palindrom");
        }
```

Gunakan fungsi equals atau equalsIgnoreCase untuk membandingkan string. Jangan gunakan “==”.