

Percobaan 2

Pemrograman OOP Dengan Java

Dosen : Ir. Nanang Syahroni M.Kom

Pokok Bahasan

- Konsep pemrograman bahasa Java
- Konsep Object Oriented Programming (OOP)
- Deklarasi pemrograman OOP dengan bahasa Java
- Penggunaan input menggunakan JOptionPane

Tujuan Belajar

- Mengenalkan tentang konsep paket pada bahasa pemrograman java
- Mengenalkan tentang konsep pemrograman berorientasi obyek
- Dengan praktikum ini mahasiswa diharapkan dapat:
 - Mengenalkan tentang konsep paket, class, dan konstruktor pada bahasa pemrograman java
 - Mengenalkan tentang konsep pemrograman berorientasi obyek dengan cara mempraktekkan secara langsung mulai dari membuat program utama, membuat kelas serta menjalankan program dan memeriksa terjadinya kesalahan sintak.
 - Mampu mendeklarasikan suatu class, atribut (variable), metode
 - Mengenalkan tentang konsep bagaimana cara mengakses anggota suatu obyek

Dasar Teori

- Deklarasi class dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> class <nama_class> {  
    [deklarasi_atribut]  
    [deklarasi_konstruktor]  
    [deklarasi_metode]  
}
```

Contoh:

```
public class Siswa {  
    ...  
}
```

- Deklarasi atribut dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <tipe> <nama_atribut> ;
```

Contoh:

```
public class Siswa {
    public int nrp;
    public String nama;
}
```

- Deklarasi metode dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <return_type> <nama_metode> ([daftar_argumen])
{
    [<statement>]
}
```

Contoh:

```
public class Siswa {
    public int nrp;
    public String nama;
    public void info() {
        System.out.println("Ini siswa PENS");
    }
}
```

- Untuk dapat mengakses anggota-anggota dari suatu obyek, maka harus dibuat instance dari class tersebut terlebih dahulu. Berikut ini adalah contoh pengaksesan anggota-anggota dari class Siswa:

```
public class TesSiswa {
    public static void main(String args[]) {
        Siswa it=new Siswa();
        it.nrp=5;
        it.nama="Andi";
        it.info();
    }
}
```

Praktek 1:

Kerjakan program berikut, kemudian lakukan pengamatan, lalu temukan mengapa nilai *parameter*, *Child* dan *Parent* berbeda.

```
package coba;

class Parent {
    public int x = 5;
}

class Child extends Parent {
    public int x = 10;
    public void Info(int x) {
        System.out.println("Nilai x sebagai parameter = " + x);
        System.out.println("Nilai x di class Child = " + this.x);
        System.out.println("Nilai x di class Parent = " + super.x);
    }
}

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

        Child coba = new Child();
        coba.Info(30);

    }
}
```

Praktek 2: Bilangan Fibonacci

Dalam matematika, deret bilangan Fibonacci adalah yang sering disebut, deret ini adalah barisan yang didefinisikan secara rekursif, berawal dari 1 kemudian angka berikutnya didapat dengan cara menambahkan dua buah bilangan yang berurutan sebelumnya.

Dengan aturan ini, maka deret bilangan Fibonacci adalah sebagai berikut:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, dst.....

Misalnya angka 8, diperoleh dari penjumlahan 2 angka didepannya yaitu 3+5.

Selanjutnya segera kerjakan program dibawah ini, dan setelah dikerjakan program ini maka lakukan pengamatan, method *jumlahkan()* dapat dipanggil oleh class *deretkan{}*.

Pada laporan resmi ada tugas tambahan yaitu gambarkan flowchart untuk kedua program pada Praktek 2 dan Praktek 3.

```

package fibonacci;
import javax.swing.JOptionPane;

class deret {
    public int i,aw,bts,a[]=new int[30];
    public void jumlahkan (int c, int d) {
        aw=c;
        bts=d;
        a[1]=aw;
        System.out.print(a[1]+" ");
        for (i=2;i<=20;i++){
            a[i]=a[i-2]+a[i-1];
            if (a[i]<=bts)
                System.out.print(a[i]+" ");
        }
        System.out.print("\n");
    }
}

public class Fibonacci {
    public static void main(String[] args) {
        int awal,batas;
        String nawal=JOptionPane.showInputDialog("Nilai
Awal : ");
        String nbatas=JOptionPane.showInputDialog("Batas :
");
        awal=Integer.parseInt(nawal);
        batas=Integer.parseInt(nbatas);
        deret deretkan=new deret();
        deretkan.jumlahkan(awal, batas);
    }
}

```

Praktek 3 :

Buat program dengan OOP untuk menampilkan tabel perkalian seperti dibawah ini dengan input adalah batas bilangan yang dikalikan.

Contoh:

Batas Perkalian : 5					
X	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25