

LAPORAN RESMI

Praktikum 2 Pengenalan Pemrograman Berbasis Obyek

Mata Kuliah: Praktek Pemrograman Berbasis Objek



Disusun oleh:

Bayu Hadi Leksana (3122500046)

2 D3 Teknik Informatika B

Dosen Pengampu: Yanuar Risah Prayogi S.Kom., M.Kom.

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

2023/2024

A. TUGAS PENDAHULUAN

1. Apakah yang dimaksud dengan kelas, method, atribut dan obyek?

Jawab:

Kelas: Merupakan template untuk membuat obyek. Merupakan prototipe / blue prints yang mendefinisikan variable-variabel dan method-method secara umum.

Atribut: adalah data yang membedakan antara obyek satu dengan yang lain.

Obyek: merupakan hasil instansiasi dari suatu kelas. Proses pembentukan obyek dari suatu class disebut dengan instantiation. Obyek disebut juga instances. Setiap obyek memiliki state sebagai status (atribut). Setiap obyek memiliki tingkah laku (method).

2. Buatlah contoh suatu kelas dan definisikan atribut dan methodnya!

Jawab:

Nama Kelas: Siswa

Atribut: nama, jenis kelamin, umur

Method: berjalan, berlari, berbicara

3. Buatlah kode program soal no. 2 diatas!

Jawab:

```
public class Orang {  
    public String nama;  
    public String jenisKelamin;  
    public int umur;  
    public void berjalan() {  
        System.out.println("Sedang berjalan");  
    }  
    public void berlari() {  
        System.out.println("Sedang berlari");  
    }  
    public void berbicara() {  
        System.out.println("Sedang berbicara");  
    }  
}
```

4. Buatlah kelas yang berisi main method yang membuat obyek dari kelas yang telah dibuat di soal no. 3. Selanjutnya obyek tersebut mengakses atribut dan methodnya.

Jawab:

```
public class IsiData {  
    public static void main(String args[] ) {  
        Siswa bayu= new Orang();  
        bayu.nama = 'Bayu Hadi Leksana';  
        bayu.jenisKelamin = 'Laki-laki';  
        bayu.umur = 20;  
        bayu.berjalan();  
        bayu.berlari();  
        bayu.berlari();  
    }  
}
```

B. PERCOBAAN

Percobaan 1

Source Code:

```
public class Siswa {  
    int nrp;  
    public void setNrp(int i) {  
        nrp=i;  
    }  
}
```

```
public class Test {  
    public static void main(String args[]) {  
        Siswa anak=new Siswa();  
        anak.setNrp(5);  
        System.out.println(anak.nrp);  
    }  
}
```

Output:

```
PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\percobaan\percobaan1\src> java Test  
5
```

Percobaan 2

Source Code:

```
public class Siswa {  
    int nrp;  
    String nama;  
    public void setNrp(int i) {  
        nrp=i;  
    }  
    public void setNama(String i) {  
        nama=i;  
    }  
}
```

Percobaan 3

Source Code:

```
public class Siswa {  
    public int nrp;  
    public Siswa(int i) {  
        nrp=i;  
    }  
    public void setNrp(int i) {  
        nrp=i;  
    }  
    public int getNrp() {  
        return nrp;  
    }  
}
```

C. LATIHAN

Latihan 1: Mengimplementasikan UML class diagram dalam program untuk class Tabungan.

Source Code:

```
public class Tabungan {
    public int saldo;

    public Tabungan(int amount) {
        saldo = amount;
    }

    public int ambilUang(int amount) {
        saldo -= amount;
        return amount;
    }
}

public class TesLatihan {
    public static void main(String args[]){
        Tabungan tabungan = new Tabungan(5000);
        System.out.println("Saldo awal : " + tabungan.saldo);
        tabungan.ambilUang(2300);
        System.out.println("Jumlah uang yang diambil : 2300");
        System.out.println("Saldo sekarang : " + tabungan.saldo);
    }
}
```

Output:

```
PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\latihan\latihan1\src> java TesLatihan
Saldo awal : 5000
Jumlah uang yang diambil : 2300
Saldo sekarang : 2700
```

Latihan 2 : Mengimplementasikan UML class diagram dalam program untuk class Mahasiswa

Source Code:

```
public class Mahasiswa {
    public int nrp;
    public String nama;
    public Mahasiswa(int nrpValue, String namaValue) {
        nrp = nrpValue;
        nama = namaValue;
    }
    public int getNrp() {
        return nrp;
    }
    public String getNama() {
        return nama;
    }
}

public class TesLatihan {
    public static void main(String args[]){
        Mahasiswa mhs = new Mahasiswa(12345, "Jono");
        System.out.println("NRP : " + mhs.getNrp());
        System.out.println("Nama : " + mhs.getNama());
    }
}
```

```
}
```

Output:

```
PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\latihan\latihan2\src> java TesLatihan
NRP : 12345
Nama : Jono
```

Latihan 3 : Mengimplementasikan UML class diagram dalam program untuk class Truk

Source Code:

```
public class Truk {
    double muatan = 0;
    double muatanMaks;
    public Truk(double beratMaks) {
        muatanMaks = beratMaks;
    }
    public double getMuatan() {
        return muatan;
    }
    public double getMuatanMaks() {
        return muatanMaks;
    }
    public void tambahMuatan(double berat) {
        if (muatan + berat <= 1000)
            muatan+=berat;
    }
}
```

```
public class TesLatihan3 {
    public static void main(String args[]){
        Truk truk = new Truk(1000);
        System.out.println("Muatan maksimal = "+truk.getMuatanMaks());
        truk.tambahMuatan(500.0);
        System.out.println("Tambah muatan : 500 ");
        truk.tambahMuatan(350.0);
        System.out.println("Tambah muatan : 350 ");
        truk.tambahMuatan(100.0);
        System.out.println("Tambah muatan : 100 ");
        truk.tambahMuatan(150.0);
        System.out.println("Tambah muatan : 150 ");
        System.out.println("Muatan sekarang = " + truk.getMuatan());
    }
}
```

Output:

```
PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\latihan\latihan3\src> java TesLatihan3
Muatan maksimal = 1000.0
Tambah muatan : 500
Tambah muatan : 350
Tambah muatan : 100
Tambah muatan : 150
Muatan sekarang = 950.0
```

D. TUGAS

Tugas 1 : Mengimplementasikan UML class diagram dalam program untuk class Tabungan

Source Code:

```
public class Tabungan {
    public int saldo;
    public Tabungan(int initSaldo) {
        saldo = initSaldo;
    }
    public int getSaldo() {
        return saldo;
    }
    public void simpanUang(int jumlah) {
        saldo+= jumlah;
    }
    public boolean ambilUang(int jumlah) {
        if (saldo < jumlah) {
            return false;
        } else {
            saldo-=jumlah;
            return true;
        }
    }
}

public class TesTugas1 {
    public static void main (String srt[]){
        boolean status;
        Tabungan tabungan = new Tabungan(5000);
        System.out.println("Saldo awal :"+tabungan.getSaldo());
        tabungan.simpanUang(3000);
        System.out.println("Jumlah uang yang disimpan : 3000");
        status = tabungan.ambilUang(6000);
        System.out.println("Jumlah uang yang diambil : 6000");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
        tabungan.simpanUang(3500);
        System.out.println("Jumlah uang yang disimpan :3500");
        status = tabungan.ambilUang(4000);
        System.out.println("Jumlah uang yang diambil : 4000");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
        status = tabungan.ambilUang(1600);
        System.out.println("Jumlah uang yang diambil : 1600");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
        tabungan.simpanUang(2000);
        System.out.println("Jumlah uang yang disimpan :2000");
        System.out.println("Saldo sekarang = " + tabungan.getSaldo());
    }
}
```

Output:

```

PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\tugas\tugas1\src> java TesTugas1
Saldo awal :5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000
Ok
Jumlah uang yang disimpan :3500
Jumlah uang yang diambil : 4000
Ok
Jumlah uang yang diambil : 1600
Gagal
Jumlah uang yang disimpan :2000
Saldo sekarang = 3500

```

Analisa:

Program di atas adalah implementasi OOP dengan membuat class Tabungan yang kemudian digunakan untuk membuat objek pada class TesTugas1.

1. Pertama saya membuat class Tabungan terlebih dahulu.
2. Kemudian pada class Tabungan dideklarasikan atribut saldo bertipe data integer dengan aksesibilitas public.
3. Lalu dibuat sebuah konstruktor yang menerima parameter initSaldo bertipe data integer. Konstruktor ini berfungsi untuk mengeset nilai awal dari objek tabungan yang dibuat yang mana atribut saldo diisi dengan initSaldo.
4. Selanjutnya dibuat fungsi/method getSaldo() yang berfungsi untuk mendapatkan nilai dari atribut saldo.
5. Dibuat juga method simpanUang() yang menerima parameter jumlah dengan tipe integer yang kemudian nilai atribut saldo saat ini akan ditambah dengan jumlah. Fungsi ini digunakan untuk menyimpan uang atau menambahkan saldo.
6. Ada juga method ambilUang() yang menerima parameter jumlah dengan tipe integer yang digunakan untuk mengambil uang dari saldo yang ada. Fungsi ini melakukan pengecekan jika saldo kurang dari jumlah yang ingin diambil maka fungsi akan mengembalikan nilai false yang artinya gagal mengambil uang dan saldo tidak akan berkurang. Namun jika saldo lebih besar sama dengan jumlah yang ingin diambil maka saldo saat ini akan dikurangi dengan jumlah dan fungsi akan mengembalikan nilai true.
7. Kemudian pada class TesTugas1 digunakan untuk membuat objek tabungan dan mengakses atribut dan method-method dari class Tabungan.

Tugas 2: Menganalisa, membuat UML class diagram dan implementasi program

UML:



Source Code:

```
public class Mobil {
    public String merek;
    public String warna;
    public String jenis;
    public int cc;
    public int seat;
    public Mobil(String merekMobil, String warnaMobil, String jenisMobil,
int ccMobil, int seatMobil) {
        merek = merekMobil;
        warna = warnaMobil;
        jenis = jenisMobil;
        cc = ccMobil;
        seat = seatMobil;
    }
    public void setMerek(String merekMobil) {
        merek = merekMobil;
    }
    public void setWarna(String warnaMobil) {
        warna = warnaMobil;
    }
    public void setJenis(String jenisMobil) {
```



```

        jenis = jenisMobil;
    }
    public void setCc(int ccMobil) {
        cc = ccMobil;
    }
    public void setSeat(int seatMobil) {
        seat = seatMobil;
    }
    public String getMerek() {
        return merek;
    }
    public String getWarna() {
        return warna;
    }
    public String getJenis() {
        return jenis;
    }
    public int getCc() {
        return cc;
    }
    public int getSeat() {
        return seat;
    }
    public void infoMobil() {
        System.out.println("Merek: " + merek);
        System.out.println("Warna: " + warna);
        System.out.println("Jenis: " + jenis);
        System.out.println("cc: " + cc);
        System.out.println("Jumlah Seat: " + seat);
    }
}

```

```

public class TesMobil {
    public static void main(String[] args) {
        Mobil mobil1 = new Mobil("Toyota", "Biru", "minibus", 2000, 7);
        Mobil mobil2 = new Mobil("Daihatsu", "Hitam", "pick up", 2000, 2);
        Mobil mobil3 = new Mobil("Suzuki", "Silver", "suv", 2000, 5);
        Mobil mobil4 = new Mobil("Honda", "Merah", "sedan", 2000, 5);
        System.out.println("--mobil1--");
        mobil1.infoMobil();
        System.out.println("--mobil2--");
        mobil2.infoMobil();
        System.out.println("--mobil3--");
        mobil3.infoMobil();
        System.out.println("--mobil4--");
        mobil4.infoMobil();
    }
}

```

Output:

```

PS C:\Users\bayuh\IdeaProjects\pbo\praktikum2\tugas\tugas2\src> java TesMobil
--mobil1--
Merek: Toyota
Warna: Biru
Jenis: minibus
cc: 2000
Jumlah Seat: 7
--mobil2--
Merek: Daihatsu
Warna: Hitam
Jenis: pick up
cc: 2000
Jumlah Seat: 2
--mobil3--
Merek: Suzuki
Warna: Silver
Jenis: suv
cc: 2000
Jumlah Seat: 5
--mobil4--
Merek: Honda
Warna: Merah
Jenis: sedan
cc: 2000
Jumlah Seat: 5

```

Analisa:

Kode program di atas adalah implementasi OOP dengan membuat class Mobil dan menggunakannya untuk membuat objek mobil1, mobil2, mobil3, mobil4.

1. Pertama dibuat sebuah class Mobil.
2. Kemudian dideklarasikan atribut merek, warna dan jenis yang masing-masing bertipe data string. Lalu atribut cc dan seat yang masing-masing bertipe data integer. Masing-masing atribut itu menggambarkan beberapa atribut umum yang ada pada mobil.
3. Lalu dibuat sebuah konstruktor yang menerima parameter merekMobil (string), warnaMobil(string), jenisMobil (string), ccMobil (int) dan seatMobil (int). Konstruktor ini digunakan untuk mengeset nilai awal atribut-atribut objek yang dibuat dengan merek diisi dengan merekMobil, warna diisi dengan warnaMobil, warna diisi dengan warnaMobil, jenis diisi dengan jenisMobil, cc diisi dengan ccMobil, seat diisi dengan seatMobil.
4. Selanjutnya, dibuat fungsi setter untuk masing-masing atribut, yaitu setMerek(), setWarna(), setJenis(), setCc(), dan setSeat() yang masing-masing menerima parameter atributnya lalu masing-masing atribut class Mobil itu diisi / diassignment dengan nilai atribut dari parameter yang diberikan.
5. Selain itu, dibuat juga fungsi getter untuk masing-masing atribut yang digunakan untuk mendapatkan nilai dari masing-masing atribut.
6. Lalu, dibuat method infoMobil() yang melakukan print masing-masing atribut.
7. Pada class TesMobil digunakan untuk membuat objek baru dari class Mobil, yakni mobil1, mobil2, mobil3 dan mobil4. Kemudian mengakses method infoMobil() dari masing-masing objek. Sehingga outputnya nanti akan menampilkan atribut-atribut dari masing-masing objek.

E. KESIMPULAN

Dalam Object Oriented Programming terdapat istilah class dan object. Class adalah sebuah blueprint yang digunakan untuk membuat objek. Objek merupakan hasil instansiasi dari suatu kelas. Dalam membuat class, kita perlu membuat atribut, konstruktor, dan method. Atribut adalah data yang membedakan antara obyek satu dengan yang lain. Konstruktor adalah fungsi yang pertama kali dijalankan saat objek dibuat yang biasanya digunakan untuk set nilai awal atribut-atribut objek. Lalu method adalah fungsi yang dimiliki suatu class yang menjalankan tugas tertentu. Class ini digunakan untuk membuat banyak objek berbeda namun secara umum memiliki ciri-ciri sesuai class yang dibuat.