

LAPORAN RESMI
MODUL I
OBJECT DAN CLASS
PEMROGRAMAN BERBASIS OBJEK



NAMA	: ANISYAFAAH
N.R.P	: 220441100105
DOSEN	: FIRMANSYAH ADIPUTRA, ST., M.Cs.
ASISTEN	: KUKUH COKRO WIBOWO
TGL PRAKTIKUM: 17 MARET 2023	

Disetujui : 23 Maret 2023 Asisten

<u>KUKUH COKRO WIBOWO</u> 21.04.411.00102



LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era yang serba menggunakan teknologi seperti saat ini, salah satu skill atau keterampilan yang harus dikuasai adalah ilmu komputer. Ilmu komputer mencakup topik-topik yang konkret, misalnya tentang software. Software digunakan untuk mengoperasikan komputer dan menjalankan tugas-tugas tertentu. Hal ini sangat berkaitan dengan pemrograman. Pemrograman sangat penting untuk menunjang era digital dan teknologi yang akan terus digunakan hingga masa yang akan datang.

Dalam pemrograman, salah satu hal yang harus diperhatikan yaitu pemilihan bahasa pemrograman. Terdapat beberapa jenis bahasa pemrograman untuk membuat sebuah program komputer, salah satunya yaitu Java. Java dikenal sebagai bahasa pemrograman yang kompleks untuk diterapkan. Selain itu, Java dapat dijalankan di berbagai sistem operasi atau platform. Berikut ini adalah contoh pembuatan program menggunakan bahasa pemrograman Java dan penggunaan Object serta class.

1.2 Tujuan

- ▶ Mahasiswa mampu memahami konsep Class dan Object dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.
- ▶ Mahasiswa mampu memahami atribut dan method dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.

BAB II

DASAR TEORI

2.1 Class dan Objek

Class diumpamakan seperti cetakan yang berguna untuk mencetak suatu object, class juga merupakan grup suatu object dengan kemiripan atribut, behaviour dan relasi ke object lain. Contoh dari sebuah class tersebut misalnya cetakan dari sebuah mobil sedan. Mobil sedan tentunya mempunyai konsep, dan konsep inilah dipakai untuk membentuk/mencetak hingga menjadi mobil sedan. Oleh karena itu, konsep bisa diartikan sebagai class.

Object adalah instance dari class. jika class secara umum mempresentasikan sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri. Contoh : Dari class Kendaraan, maka kita dapat membuat object sepeda, motor, mobil, becak, dll.

Pada dasarnya ada dua karakteristik yang utama pada sebuah Object yaitu :

1. Setiap object memiliki atribut sebagai status yang akan disebut state
2. Setiap object memiliki tingkah laku yang kemudian akan disebut dengan method (behaviour)

State	Behaviour
Pedal	Kecepatannya menaik
Koda	Kecepatannya menurun
Jeruji	Perpindahan gigi sepeda

Object Sepeda

Dalam pengembangan perangkat lunak berorientasi object, object dalam perangkat lunak akan menyimpan state-nya dalam variable dan menyimpan informasi tingkah laku (behaviour) dalam method-method atau fungsi-fungsi.

Untuk membuat object, kita menggunakan perintah new dengan sebuah nama class yang akan dibuat sebagai instance dari class.

Contoh :

1. Tahapan membuat suatu class

Untuk membuat suatu class ada beberapa tahap yang harus dilakukan seperti yang ada pada Gambar 1 dengan rincian tahapan sebagai berikut :

2. Buka aplikasi BlueJ, klik project pilih "New Project"
3. Pada form new project isilah nama project yang akan kita buat
 - a. Tampilan form setelah nama project diisi
 - b. Klik "New Class" untuk membuat class
 - c. Akan ada form baru yang digunakan untuk mengisi nama Class sesuai dengan yang kita inginkan pada kolom "Class Name", lalu tekan "OK"
 - d. Tampilan form setelah kita membuat class, dan kita bisa mengedit code di dalam class, pada contoh diatas terdapat pada class kendaraan.
 - e. Tampilan class yang didalamnya bisa kita edit sesuai dengan yang kita inginkan.

4. Contoh membuat object dari suatu class

```
/**
```

```
* Write a description of class Kendaraan here.
```

```
*
```

```
* @author (your name)
```

```
* @version (a version number or a date)
```

```
*/
```

```
Public class Kendaraan
```

```
{
```

```
    Public Kendaraan (String name
```

```
    // konstruktor ini memiliki satu parameter yaitu : name  
    System.out.println("Motorku Bernama : " + name);
```

```
}
```



```

public static void main (String[] args) {
    // untuk membuat object kendaraan
    Kendaraan Motor = new Kendaraan ("maniss");
}
}

```

Membuat object pada class Kendaraan

Pada Gambar 2 merupakan contoh dari suatu class Kendaraan yang didalamnya terdapat suatu object. Berdasarkan program diatas, telah dibuat object Motor dari class kendaraan, dan kita juga bisa menggunakan konstruktor yang akan dijalankan secara otomatis pada saat object dibuat yakni ketika perintah "new" dijalankan.

Contoh Konstruktor seperti berikut :

```

public Kendaraan (String name) {
    // Konstruktor ini memiliki satu parameter yaitu : nama
    System.out.println ("Motorka bernama : " + name);
}

```

Source code diatas disebut dengan konstruktor yang nantinya akan dipanggil secara otomatis pada saat object dibuat, tanpa membuat method terlebih dahulu. Konstruktor harus menggunakan nama class untuk penamaannya, dan dalam suatu class hanya boleh ada satu konstruktor, jika tidak maka akan terjadi SyntaxError.

Contoh object seperti berikut :

```

Kendaraan Motor = new Kendaraan ("maniss");

```

Kode program diatas artinya membuat suatu Object Motor dari class Kendaraan, dengan menggunakan keyword new. Karena menggunakan konstruktor maka tidak perlu menggunakan method untuk menginisialisasi object.

Output dari class Kendaraan :

Motorku Bernama : maniss

2.2 Method

Method merupakan kumpulan dari fungsi yang kita buat dalam suatu class, yang kemudian fungsi-fungsi itu akan dipanggil kembali. Dengan method kita bisa mempersingkat coding yang kita buat. Method didefinisikan pada class tetapi akan dipanggil melalui object, method menentukan perilaku object, yakni apa yang akan terjadi saat object itu dibuat serta berbagai operasi yang dapat dilakukan object.

Struktur dari method diantaranya sebagai berikut :

1. Method terdiri dari statement public, private, protected, yang menandakan hak akses method tersebut.
2. Nama method terdiri dari alfabeth saja
3. Parameter, dalam penulisannya harus diawali dengan tipe data baru nama parameternya.
4. Isi method

Contoh :

```
public class Kendaraan — Class
{
    public Kendaraan (String name) { — Konstruktor
        // Konstruktor ini memiliki satu parameter yaitu : nama
        System.out.println ("Motorku bernama : " + name);
    }

    public static void Warna (String warna) { — Method
        // Method Void (tanpa nilai kembali)
        System.out.println ("Warnanya : " + warna);
    }

    public static void Berjalan (String jalannya) {
        // Method Void (tanpa nilai kembali)
        System.out.println ("jalannya : " + jalannya);
    }
}
```

```

public static void main (String[] args) {
    // untuk membuat object kendaraan
    Kendaraan Motor = new Kendaraan ("Manis");
    Motor.Warna ("Coklat");
    Motor.Bergalan ("ngueng.... ngueng.... ngueng...");
}

```

Membuat Object pada class kendaraan

3

Pengelasan Gambar 1 :

- Pada class Kendaraan terdapat Konstruktor, Method, dan Object
- Konstruktor merupakan suatu method yang akan memberikan nilai awal pada suatu object dibuat. jadi apabila object diatas telah dibuat lalu dijalankan maka konstruktor akan bekerja secara otomatis. Biasanya nama konstruktor sama dengan nama Class yang dibuat
- Pada method diatas terdapat ("public static") yang merupakan modifier yaitu jenis akses method yang bersifat public. Pada gambar diatas ada dua method yang nantinya akan dipanggil saat object dijalankan yaitu method Warna dan Method Bergalan.
- Kode program dibawah artinya : membuat sebuah object Motor dari class Kendaraan dengan menggunakan keyword new.

Kendaraan Motor = new Kendaraan ("Manis");

- Kode program dibawah artinya : menginisialisasi object Motor menggunakan method Warna dan method Bergalan yang dimiliki class kendaraan yang nantinya akan dipanggil oleh Object.

Motor.Warna ("Coklat");

Motor.Bergalan ("ngueng.... ngueng.... ngueng...");

2.3 Atribut

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat memiliki hak akses `private`, `public` maupun `protected`. Sebuah atribut yang dinyatakan sebagai `private` hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.

Sebuah atribut yang dinyatakan sebagai `public` dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang dinyatakan sebagai `protected` tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

Karakteristik Atribut

Dalam class, atribut disebut sebagai variabel. Atribut dapat membedakan antara satu object dengan object yang lain. Sebagai contoh, pada class : mahasiswa, terdapat object mahasiswa si A, dan object mahasiswa si B. Yang membedakan antara object si A dan si B adalah NPM-nya (Nomor Pokok Mahasiswa) yang merupakan atribut dari object tersebut. Pada atribut, terdapat pula dua istilah variabel, yaitu Instance Variabel dan Class Variable.

Contoh penerapan atribut

Membuat class dengan nama sepeda dan semua atribut yang dimiliki oleh kelas sepeda (Gambar 1).

```
class Sepeda {
```

```
    int kecepatan = 0;
```

```
    int gir = 0;
```

```
    // method yang ada didalam class
```

```
    void ubahGir (int penambahanGir) {
```

```
        gir = gir + penambahanGir;
```

```
        System.out.println ("Gir : " + gir);
```

```
    }
```



```

    void TambahkanKecepatan (int pertambahanKecepatan){
        kecepatan = kecepatan + PertambahanKecepatan;
        System.out.println("Kecepatan : "+ kecepatan);
    }
}

```

Membuat public class yang memanfaatkan / memanggil atribut dalam kelas lainnya (Gambar 2).

```

public class SepedaBeraksi {

    public static void main (String[] args) {
        // TODO code application logic here
        Sepeda sepedaku = new Sepeda();
        // Memanggil atribut yang diatasnya
        sepedaku.kecepatan = 10;
        sepedaku.gir = 2;
        // Memanggil metode dan isi nilai
        sepedaku.tambahkanKecepatan (30);
        sepedaku.ubahGir (3);
    }
}

```

Hasil program setelah dijalankan (Gambar 3).

Gir : 5

BUILD SUCCESSFUL (total time: 0 seconds)

BAB III

TUGAS PENDAHULUAN

3.1 Soal

Jelaskan perbedaan class dan object dengan menggunakan bahasamu sendiri!

3.2 jawab

- Class

Class adalah sebuah template atau cetakan yang digunakan untuk membuat objek atau kerangka dasar. Class bertugas untuk mengumpulkan fungsi atau prosedur dan sebuah variabel dalam satu tempat. Class mempresentasikan sebuah object didalamnya, sehingga penamaannya harus diseraikan dengan object yang akan dibuat.

- Object

Object adalah sebuah perwujudan dari class. Object merupakan isi atau variabel dari class yang telah dibuat sehingga instance mempresentasikan class secara nyata. Object terdiri dari atribut dan method. Untuk membuat sebuah object, perlu menggunakan perintah new dan nama class.

BAB IV IMPLEMENTASI

4.1 Soal

1. Buatlah class manusia dengan atribut nama, umur, dan alamat serta memiliki method berjalan dan berlari. Buatlah beberapa objek dari kelas tersebut.
2. Buatlah program java untuk menampilkan nama, nim, jurusan/prodi, dan alamat mahasiswa menggunakan konsep OOP. Nama, nim, jurusan/prodi, dan alamat bersifat dinamis, dengan kata lain menggunakan input dari pengguna.

4.2 Jawaban

1. a) Hasil

----- OBJECT 1 -----

Namaku adalah : Anisa

Umurku : 18

Alamatku di : Bangkalan

Suka Berjalan

----- OBJECT 2 -----

Temaku bernama : Putri

Umurnya : 19

Alamatnya di : Gresik

Suka Berlari

BUILD SUCCESSFUL (total time : 0 seconds)

b) Source Code

```
package Project1;  
class Manusia {
```

```
    String nama;
```

```
    int umur;
```

```
    String alamat;
```



```
Void berjalan() {  
    System.out.println ("Suka Berjalan");  
}
```

```
Void berlari() {  
    System.out.println ("Suka Berlari");  
}
```

```
Public static void main (String[] args) {  
    Manusia orang1 = new Manusia ();  
    Manusia orang2 = new Manusia ();
```

```
    orang1.nama = "Anisa";  
    orang1.umur = 18;  
    orang1.alamat = "Bangkalan";
```

```
    orang2.nama = "Putri";  
    orang2.umur = 19;  
    orang2.alamat = "Gresik";
```

```
    System.out.println ("----- OBJECT 1 -----");  
    System.out.println ("Namaku adalah : " + orang1.nama);  
    System.out.println ("Umorku          : " + orang1.umur);  
    System.out.println ("Alamatku di       : " + orang1.alamat);  
    orang1.berjalan();
```

```
    System.out.println ("----- OBJECT 2 -----");  
    System.out.println ("Temanku bernama : " + orang2.nama);  
    System.out.println ("Umurnya         : " + orang2.umur);  
    System.out.println ("Alamatnya di    : " + orang2.alamat);  
    orang2.berlari();
```

```
}
```

```
}
```

c) Penjelasan

Untuk membuat sebuah program dengan class Manusia, maka perlu membuat packagenya terlebih dahulu. Disini saya meletakkan program ini di project prakt dan nama packagenya yaitu Project1. Kemudian saya membuat class dengan nama Manusia yang didalamnya saya isi dengan atribut nama (string), umur (integer), dan alamat (string). Setelah itu, membuat method atau langkah laku, maka perlu menggunakan perintah void diikuti kata langkah laku dan tanda (). Di dalam perintah ini saya isi dengan syntax pencetak untuk mencetak langkah laku.

Beralih ke pembuatan object, perlu menggunakan perintah public static void main (String[] args) yang didalamnya saya isi dengan dua object, yaitu orang1 dan orang2. Kemudian isi atribut sesuai dengan nama dan tipe data atribut. Setelah itu, cetak semua atribut dengan menambahkan nama object.nama atribut. Untuk memanggil method, hanya perlu memasukkan nama object.nama method dan tanda ().

2. a) Hasil

Silahkan Masukkan Nama Anda :

Anisyafaah

Silahkan Masukkan NIM Anda :

220441100105

Silahkan Masukkan Jurusan Anda :

Sistem Informasi

Silahkan Masukkan Alamat Anda :

Bangkalan

Nama : Anisyafaah

NIM : 220441100105

Jurusan : Sistem Informasi

Alamat : Bangkalan

BUILD SUCCESSFUL (total time : 32 seconds)

b) Source Code

```
package Project1;
```

```
import java.util.Scanner;
```

```
public class Mahasiswa {
```

```
    String nama;
```

```
    String nim;
```

```
    String jurusan;
```

```
    String alamat;
```

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

```
        Scanner input = new Scanner (System.in);
```

```
        Mahasiswa data = new Mahasiswa ();
```

```
        System.out.println ("Silahkan Masukkan Nama Anda : ");
```

```
        data.nama = input.nextLine();
```

```
        System.out.println ("Silahkan Masukkan Nim Anda : ");
```

```
        data.nim = input.nextLine();
```

```
        System.out.println ("Silahkan Masukkan jurusan Anda : ");
```

```
        data.jurusan = input.nextLine();
```

```
        System.out.println ("Silahkan Masukkan Alamat Anda : ");
```

```
        data.alamat = input.nextLine();
```

```
        System.out.println ("-----")
```

```
        System.out.println ("Nama : " + data.nama);
```

```
        System.out.println ("Nim : " + data.nim);
```



```
System.out.println("Jurusan : " + data.jurusan);  
System.out.println("Alamat : " + data.alamat);  
}  
}
```

c) Pengelaran

Untuk soal nomor dua, saya letakkan di package yang sama dengan sebelumnya, yaitu Project1. Nah karena programnya bersifat dinamis, maka perlu perintah Scanner, yaitu perintah yang digunakan agar pengguna bisa menginputkan sebuah data. Class Scanner ini berada di java.util sehingga untuk menggunakannya perlu dilakukan import. Kemudian pada class Mahasiswa saya isi dengan nama, nim, jurusan, dan alamat. Kemudian membuat objek Scanner dari class Scanner dan kelas Mahasiswa membuat objek data. Setelah itu, panggil variabel dan cetak semua dengan menggunakan perintah input.nextLine agar bisa menginput dan menyimpan data. Selanjutnya cetak semua data yang telah diinputkan dengan menggunakan syntax pencetak dan panggil objek dan atributnya.

BAB V PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa dalam membuat sebuah object dan class dalam bahasa pemrograman Java, maka dibutuhkan sebuah file dengan dokumen java. Dokumen ini berisi beberapa perintah dalam membuat sebuah class, seperti perintah class, void, dan public static void main (String[] args) {}.

Perintah class digunakan untuk membuat class, cara penggunaannya yaitu class diikuti nama class dan tanda {} dan diisi dengan beberapa atribut, method, dan object. Perintah void digunakan untuk membuat method, cara penggunaannya yaitu void diikuti nama method dan tanda (). Perintah public static void main (String[] args) {} digunakan untuk membuat object, tentunya harus disesuaikan dengan nama class. Selain itu, perintah ini dapat digunakan untuk mengisi atribut dan menjalankan program. Semua statement yang akan dieksekusi harus berada di dalam perintah ini.

5.2 Kesimpulan

1. Class adalah cetakan yang berguna untuk mencetak suatu object yang memiliki kumpulan atribut, behaviour, dan relasi ke object lain. Sedangkan Object adalah instance yang merepresentasikan class itu sendiri secara nyata.
2. Method adalah sebuah perilaku atau fungsi yang akan dipanggil melalui object. Sedangkan Atribut adalah karakteristik atau ciri khas dari suatu Object.
3. Program yang telah diimplementasikan diatas menggunakan perintah class, void, dan public static void main (String[] args) {}, serta statement pencetak Program yaitu System.out.println("").