

LAPORAN RESMI
MODUL I
OBJECT DAN CLASS
PEMROGRAMAN BERBASIS OBJECT



NAMA	: YULIATIN AGUSTINA
N.R.P	: 230114400013
DOSEN	: FIRMANSYAH ADIPUTRA, S.T., M.Cs.
ASISTEN	: PUTRI NIHAYATUL HUSNA
TGL PRAKTIKUM	: 19 MARET 2024

Disetujui : MARET 2024
Asisten

PUTRI NIHAYATUL HUSNA
21.04.411.00009



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Latar belakang dari konsep class dan object ini berasal dari paradigma pemrograman berorientasi objek (OOP), di mana pemrograman difokuskan pada objek-objek yang memiliki atribut dan perilaku yang terkait, dan interaksi antara objek-objek ini memungkinkan untuk memodelkan dan menyelesaikan masalah dengan cara yang lebih terstruktur dan modular. OOP membantu dalam pengorganisasian kode, pengurangan redundansi, dan meningkatkan keterbacaan dan pemeliharaan kode.

kelas dan objek dapat ditemukan dalam paradigma pemrograman berorientasi objek, yang memungkinkan pengembang untuk memodelkan dunia nyata atau konsep-konsep yang kompleks.. Ini memungkinkan untuk pembuatan kode yang lebih mudah dimengerti, dikelola, dan diperbarui. OOP memungkinkan juga untuk penulisan kode yang lebih efisien dan reusable, karena kelas dan objek dapat digunakan kembali dalam berbagai konteks dalam program.

Dengan menggunakan kelas dan objek, pemrogram dapat membuat struktur program yang lebih modular, reusable, dan mudah dimengerti. OOP memfasilitasi pengembangan perangkat lunak yang lebih fleksibel dan skalabel.

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 Pengertian class dan Object

Class diumpamakan seperti cetakan yang berguna untuk mencetak suatuobject, class juga merupakan grup suatu object dengan kemiripan attribut, brhaviour 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 merepresentasikan 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 attribut sebagai status yang akan disebut state
- 2. Setiap object memiliki tingkah laku yang kemudian akan disebut denganmethod (behaviour

state	Behavior
roda	Kecepatan turun
jeruji	Perpindahan gigi sepeda
pedal	Perpindahan naik

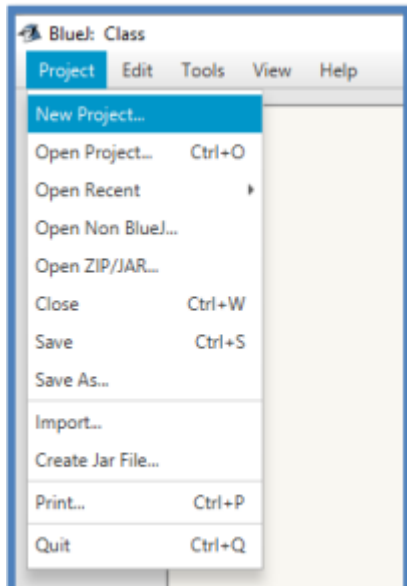
Object sepeda

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 sebuahnama class yang akan dibuat sebagai instance dari class.

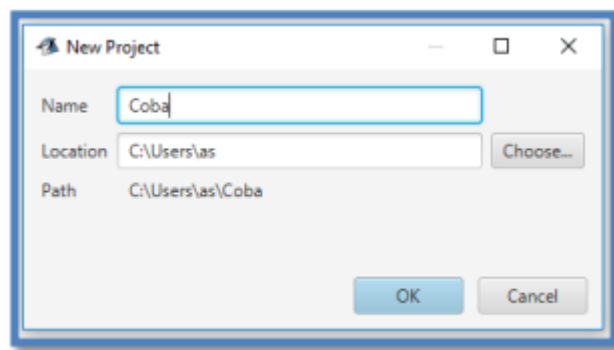
Tahapan membuat suatu class

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

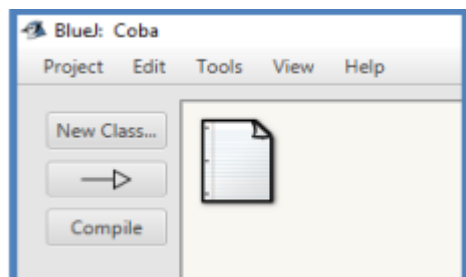
- 2. Buka aplikasi BlueJ, klick project pilih “New Project”.



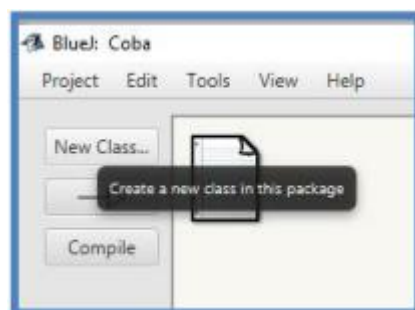
3. Pada form new project isilah nama project yang akan kita buat.



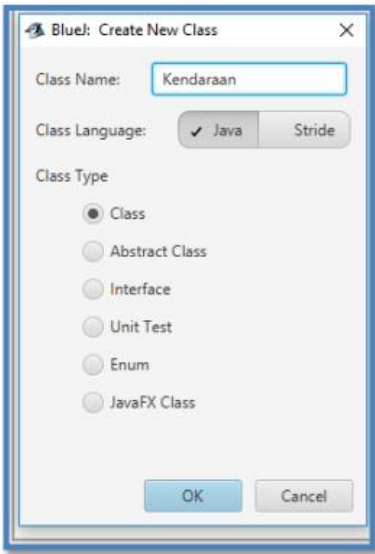
1. Tampilan form setelah nama project diisi.



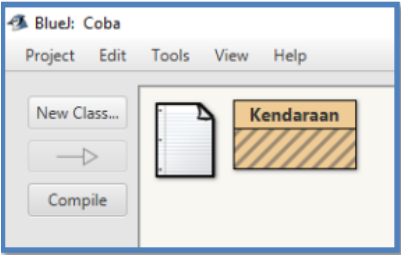
2. Klik “New Class”, untuk membuat class



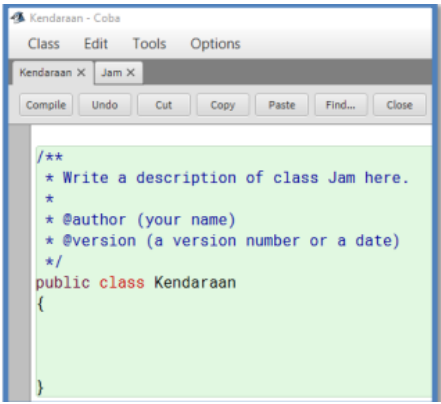
3. Akan ada form baru yang digunakan untuk mengisi nama Class sesuai dengan yang kita inginkan pada kolom “Class Name”, lalutekan “OK”.



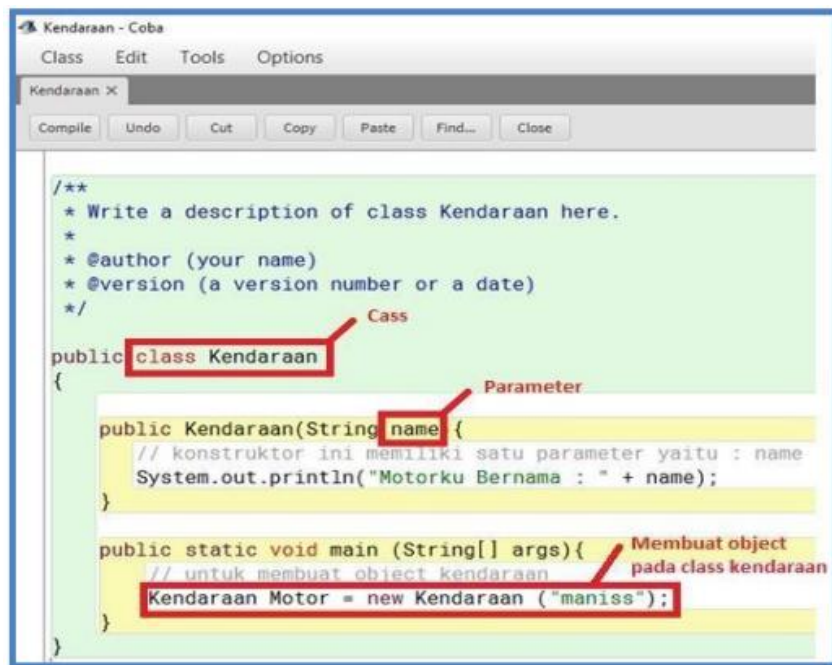
4. Tampilan form setelah kita membuat class, dan kita bisa mengedit code di dalam class, pada contoh diatas terdapat pada classkendaraan.



5. Tampilan class yang didalamnya bisa kita edit sesuai dengan yang kita inginkan.



4. Contoh membuat object dari suatu class



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 ("Motorku 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, dandalam 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.

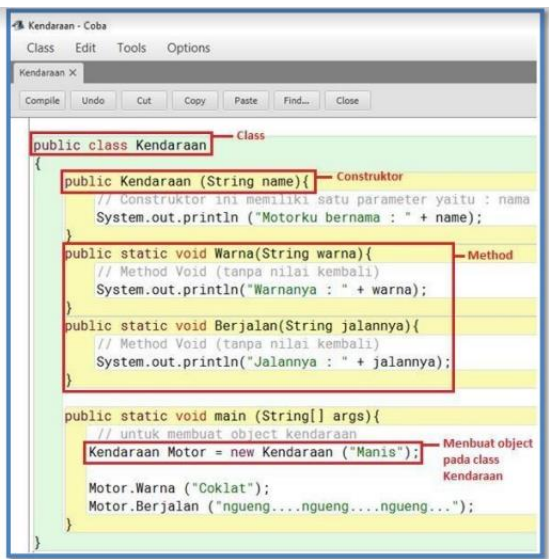
2.2 Method

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 tetapakan 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 namaparameternya.
4. Isi method

Contoh



1. Pada class Kendaraan terdapat Konstruktor, Method, dan Object
2. Konstruktor merupakan suatu method yang akan memberikan nilai awal pada saat 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.
3. 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 Berjalan.
4. Kode program dibawah artinya : membuat sebuah object Motor dari class Kendaraan dengan menggunakan keyword new.

```
Kendaraan Motor = new Kendaraan ("Manis");
```

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

```
Motor.Warna ("Coklat");  
Motor.Berjalan ("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. 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 variable, yaitu instance variable dan Class Variable.

BAB III

3.1. Soal

1. Jelaskan perbedaan class dan object dengan menggunakan bahasamu sendiri

3.2. Jawaban

1. class merupakan pendefinisian dari suatu benda yang berguna untuk membuat atau sebelum dibuatkannya suatu object. Jikalau class tidak atribut dan method suatu object didefinisikan sedangkan object merupakan bentuk nyata dari sebuah class yang memiliki data dan method didalamnya.

BAB IV

IMPLEMENTASI

4.1 Tugas Praktikum

- 1, Buatlah class manusia dengan dengan attribut nama, umur dan alamat serta memiliki method berjalan dan berlari. Buatlah beberapa objek dari class 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 Source code

1. Source code soal 1

```
package manusia;

class perilaku {
    String nama;
    int umur;
    String alamat;

    void berjalan () {
        System.out.println(nama+ " sedang berjalan");
    }

    void berlari () {
        System.out.println(nama+ " kadang juga berlari");
    }
}

public class Manusia {
    public static void main(String[] args) {
        perilaku objek = new perilaku();
        objek.nama = "Yuliatin Agustina";
        objek.umur = 19;
        objek.alamat = "Bangkalan";

        System.out.println("Nama Saya " + objek.nama);
        System.out.println("Umur saya " + objek.umur);
        System.out.println("Alamat Saya " + objek.alamat);
    }
}
```

```

        objek.berjalan();
        objek.berlari();
    }
}

```

2. Source code soal 2

```

package mahasiswa;

import java.util.Scanner;

/**
 *
 * @author natas
 */
public class Mahasiswa {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        String nama,nim,prodi,alamat;


        System.out.print("Masukan Nama : ");
        nama= sc.nextLine();
        System.out.print("Masukan Nim : ");
        nim= sc.nextLine();
        System.out.print("Masukan Prodi : ");
        prodi= sc.nextLine();
        System.out.print("Masukan Alamat : ");
        alamat= sc.nextLine();


        System.out.println("");
        System.out.println("=====");
        System.out.println("  Data Mahasiswa  ");
        System.out.println("=====");
        System.out.println("Nama    : " + nama);
        System.out.println("NIM     : " + nim);
        System.out.println("Prodi   : " + prodi);
    }
}

```

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

4.3 Hasil

1. Soal 1

Nama Saya Yuliatin Agustina

Umur saya 19

Alamat Saya Bangkalan

Yuliatin Agustina sedang berjalan

Yuliatin Agustina kadang juga berlari

2. Soal 2

=====

Data Mahasiswa

=====

Nama : Yuliatin Agustina

NIM : 230441100013

Prodi : sistem informasi

Alamat : Bangkalan

4.4 Penjelasan

Pada soal nomer 1 saya membuat program oop di mana langkah pertama yaitu membuat class manusia dengn atribut berisi nama, umur, alamat dengan menggunakan string.

Pada soal nomer 2 saya membuat program oop di mana langkah pertama yaitu membuat class datamahasiswa dengn atribut berisi nama, nim, prodi, dan alamat dengan menggunakan static void.

BAB V

PENUTUP

5.1 Analisa

Dari hasil pembelajarn pada praktikum modul ini menambah pengetahuan kita tentang class dan object dalam pemograman khususnya OOP, di mana pemrograman difokuskan pada objek-objek yang memiliki atribut dan perilaku yang terkait, dan interaksi antara objek-objek ini memungkinkan untuk memodelkan dan menyelesaikan masalah dengan cara yang lebih terstruktur

Pembacaan yang lebih mudah OOP memfasilitasi penulisan kode yang lebih mudah dimengerti dan lebih intuitif, karena konsepnya mencerminkan cara pikir manusia tentang objek dalam dunia nyata. Skalabilitas Dengan menggunakan kelas dan objek, program dapat diperluas dan disesuaikan dengan lebih mudah karena strukturnya yang terorganisir dan modular.

Dengan demikian, analisis tentang kelas dan objek menunjukkan bahwa mereka adalah elemen penting dalam pengembangan perangkat lunak yang modern, memungkinkan untuk pembuatan program yang lebih terstruktur, mudah dimengerti, dan mudah dikelola.

5.2 Kesimpulan

Dari materi dan Latihan soal yang ada pada laporan di atas. Kita dapat mengetahui manfaat dari penggunaan Bahasa pemrograman Java dan bisa mengimplementasikan dalam kehidupan sehari-hari. Contohnya seperti dapat menerapkan Class dan Object dalam sebuah program sederhana dengan menggunakan Bahasa pemrograman Java.

kelas dan objek merupakan elemen utama dalam pemrograman berorientasi objek yang memungkinkan pembuatan program yang lebih terstruktur, modular, reusable, dan mudah dimengerti. Dengan menggunakan kelas dan objek, pengembang dapat menciptakan kode yang lebih efisien, dapat dipelihara, dan dapat diperluas dengan lebih mudah. Oleh karena itu, pemahaman yang baik tentang konsep kelas dan objek sangat penting bagi pengembang perangkat lunak dalam membangun aplikasi yang kompleks dan skalabel. Selain kelas dan objek kita perlu juga memahami yang Namanya method dimana method ialah merupakan kumpulan dari fungsi yang kita buat dalam suatu class.