

LAPORAN RESMI
MODUL I
OBJEK DAN CLASS
PEMROGRAMAN BERBASIS OBJEK



NAMA	: MIFTAHUZZAIN ABRORI
N.R.P	: 230441100038
DOSEN	: ACHMAD ZAIN NUR, S.Kom., M.T
ASISTEN	: MUHAMMAD AKMAL NABIL HIBRIZI
TGL PRAKTIKUM	: 19 MARET 2024

Disetujui : 23 Maret 2023
Asisten

MUHAMMAD AKMAL NABIL HIBRIZI
220441100068



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Class adalah blueprint atau template untuk membuat suatu objek. Class mendefinisikan atribut dan metode yang akan dimiliki oleh objek. Atribut adalah properti atau karakteristik objek, sementara metode adalah tindakan yang dapat dilakukan oleh objek. Dengan kata lain class adalah suatu entitas yang merupakan dasar dari suatu objek dan digunakan untuk mendefinisikan suatu objeknya. Class berisikan variabel dan fungsi yang digunakan oleh objek tersebut. Dalam pemrograman kita bisa mendefinisikan class sekali dan kemudian membuat banyak objek dari class tersebut. Setiap objek ini, yang juga dikenal sebagai instance, akan memiliki atributnya tersendiri.

Objek adalah instance dari class dengan kata lain, objek adalah relasi konkrit dari class. Misalnya, jika kita memiliki class "mobil", maka objeknya bisa berupa "Mobil Toyota, Mobil Honda dan lain sebagainya". Setiap objek memiliki atribut dan metode yang ditentukan oleh class-nya, tetapi nilai atribut dapat berbeda antara satu objek dengan objek lainnya. Misalnya, kita bisa membuat class "Mobil" dengan atribut "Merk" dan warna dan "metode" drive, kemudian kita bisa membuat objek "Mobil1" dengan "merk" Toyota dan "warna" merah, dan objek "Mobil2" dengan "Merk" Honda dan "warna" biru. Kedua objek ini bisa melakukan metode drive, tetapi atribut dari mereka berbeda.

1.2 Tujuan

- Mampu memahami konsep Class dan Objek dalam pemrograman yang berorientasi pada objek serta mampu mengimplementasikannya.
- Mampu memahami atribut dan method dalam pemrograman berorientasi pada objek serta mampu mengimplementasikannya.

BAB II

DASAR TEORI

2.1 Class dan Objek

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

Objek adalah instance dari class. Jika class secara umum menggambarkan suatu objek, sebuah instance adalah representasi nyata dari class itu sendiri. contoh: dari class kendaraan, maka kita dapat membuat objek sepeda, motor, mobil, becak, dan lain sebagainya. Pada dasarnya ada dua karakteristik yang utama pada sebuah objek yaitu :

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

State	Behavior
Pedal	Kecepatannya menaik
Roda	Kecepatannya menurun
Jeruji	Perpindahan gigi sepeda

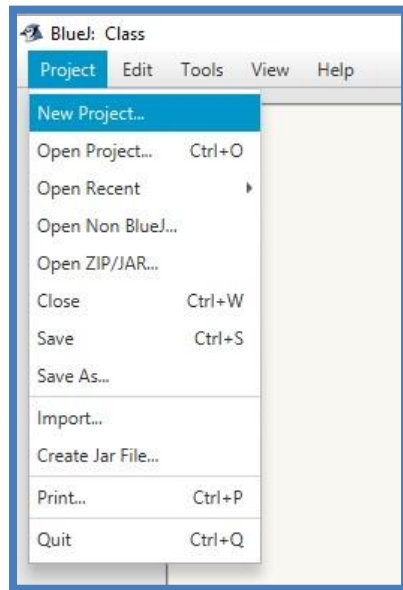
Dalam pengembangan perangkat lunak berorientasi objek, objek dalam perangkat lunak akan menyimpan state-nya dalam variabel dan menyimpan informasi tingkah laku (behaviour) dalam method-method atau fungsi-fungsi. Untuk membuat objek, kita menggunakan perintah New dengan sebuah nama class yang akan dibuat sebagai instance dari class.

Contoh:

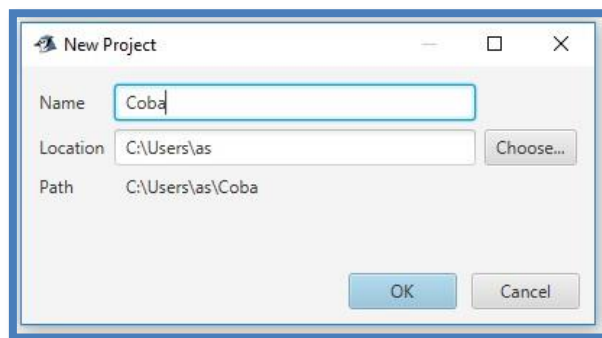
1. Tahapan membuat suatu class

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

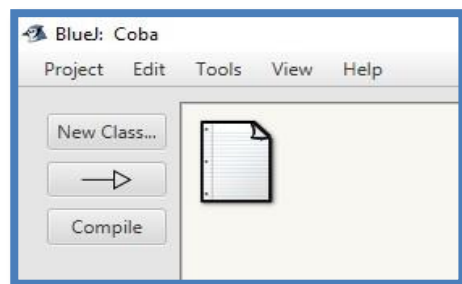
2. Buka aplikasi BlueJ, klik project pilih “New Project”.



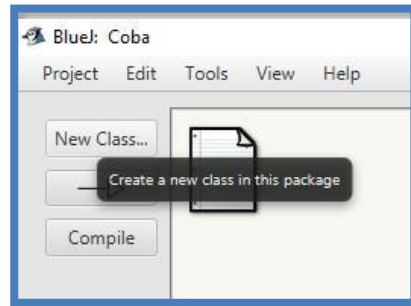
3. Pada form new project isilah nama project yang akan kita buat.



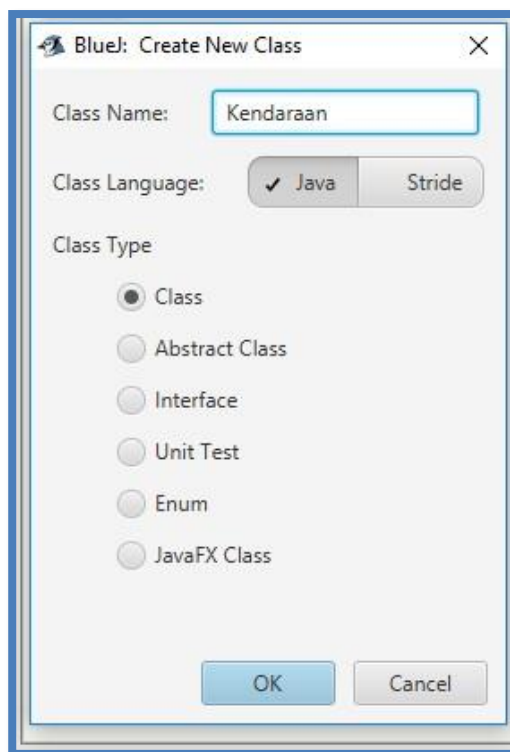
a. Tampilan form setelah nama project diisi.



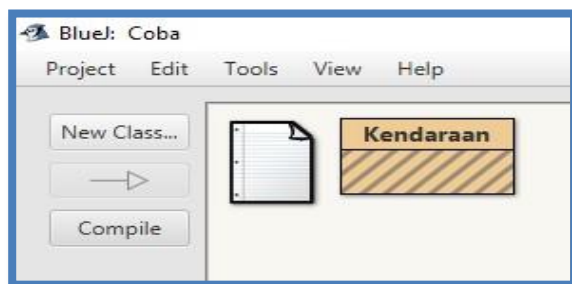
- b. Klik “**New Class**”, untuk membuat *class*



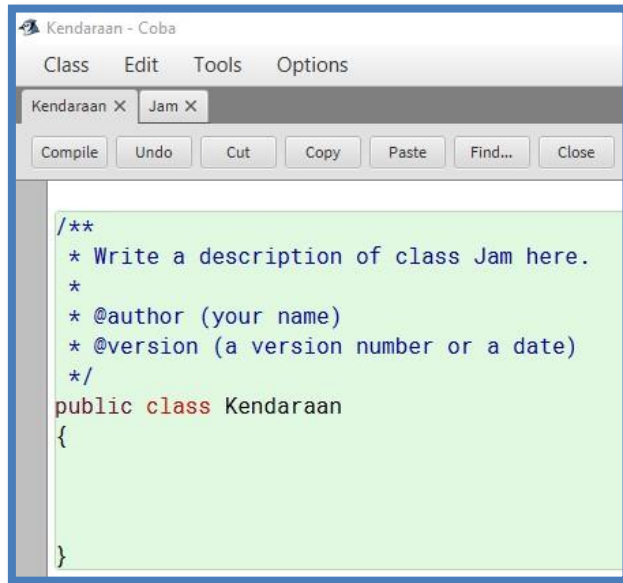
- c. Akan ada form baru yang digunakan untuk mengisi nama *Class* sesuai dengan yang kita inginkan pada kolom “**Class Name**”, lalu tekan “OK”.



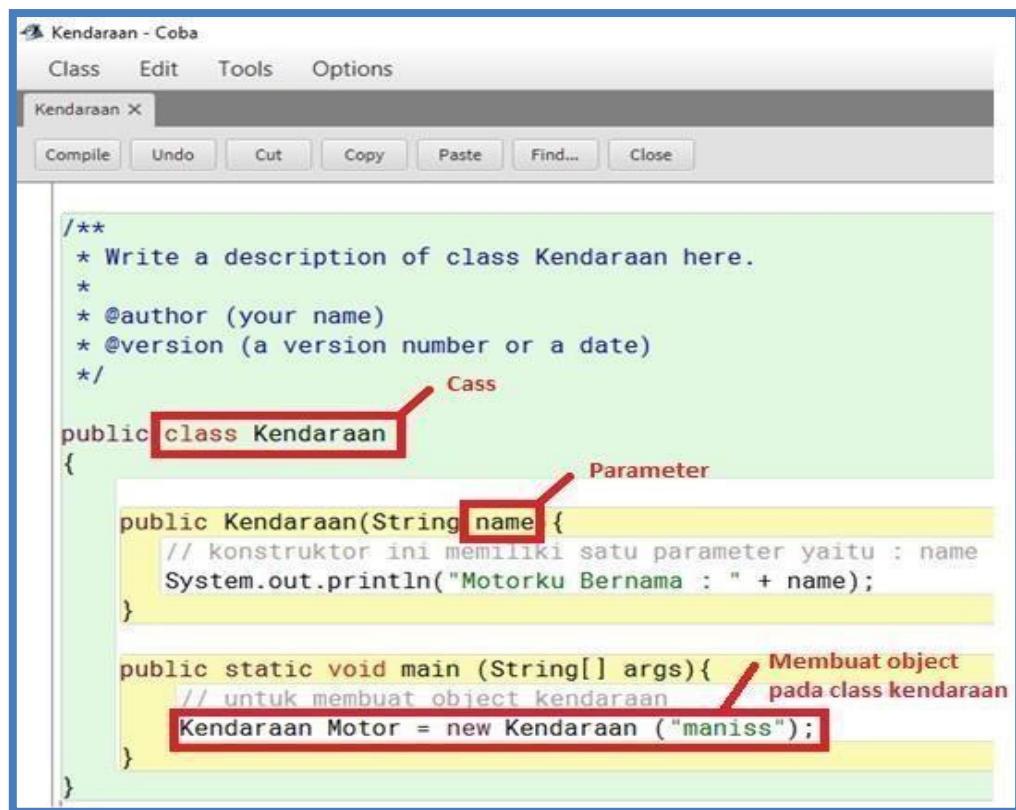
- d. Tampilan form setelah kita membuat *class*, dan kita bisa mengedit code di dalam *class*, pada contoh diatas terdapat pada *class* kendaraan.



- e. Tampilan *class* yang didalamnya bisa kita edit sesuai dengan yang kita inginkan.



4. Contoh membuat *object* dari suatu *class*



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 :

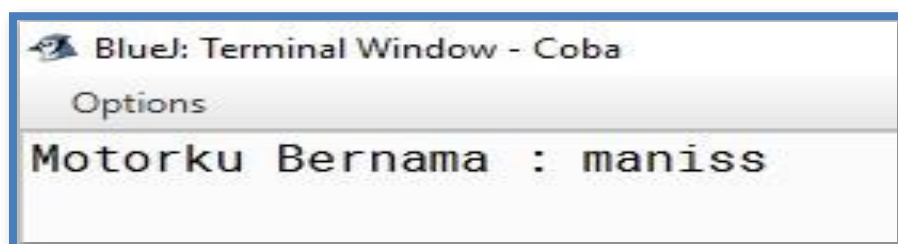
```
public Kendaraan (String name){  
    // Konstruktor ini memiliki satu parameter yaitu : nama  
    System.out.println ("Motorku bernama : " + name);  
}
```

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, 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*.



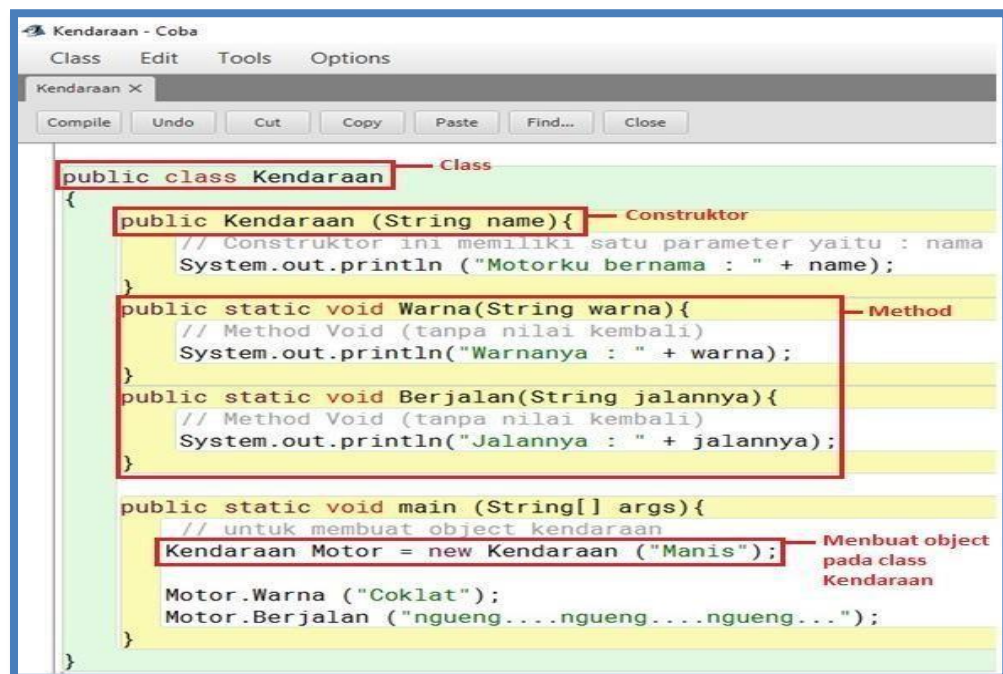
2.2 Method

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 :



Gambar 4

- a. Pada *class* Kendaraan terdapat Konstruktor, *Method*, dan *Object*
- b. 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.
- c. 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.
- d. Kode program dibawah artinya : membuat sebuah *object* Motor dari *class* Kendaraan dengan menggunakan keyword new.

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

- e. 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 oleh kelas lain di

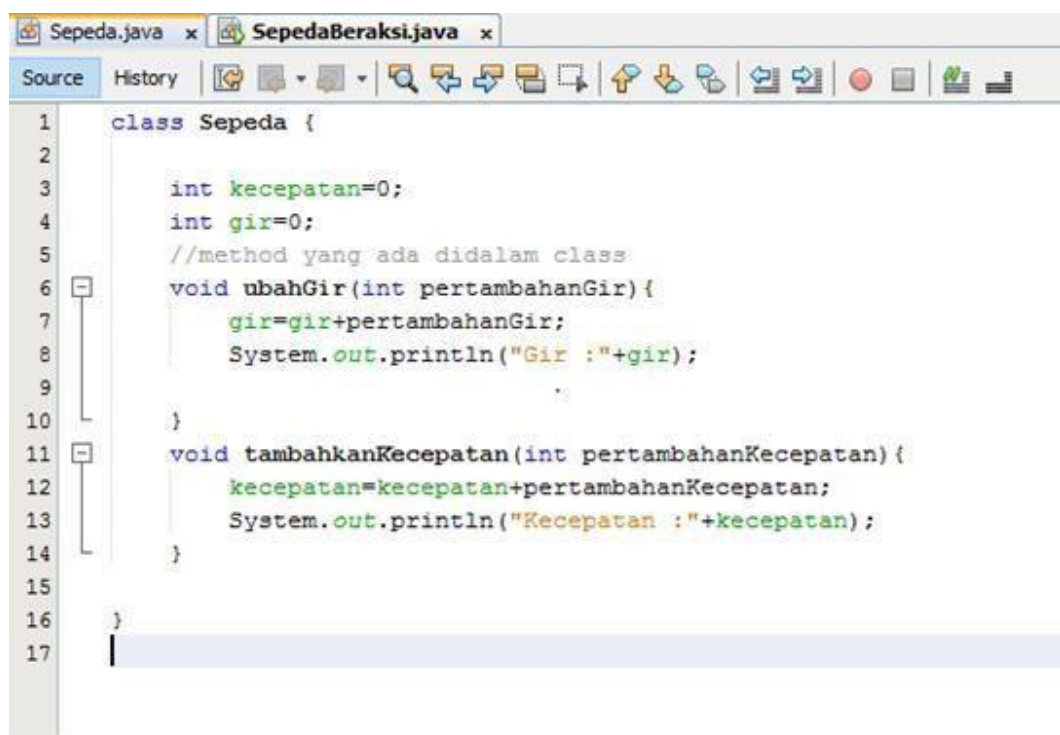
luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

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.

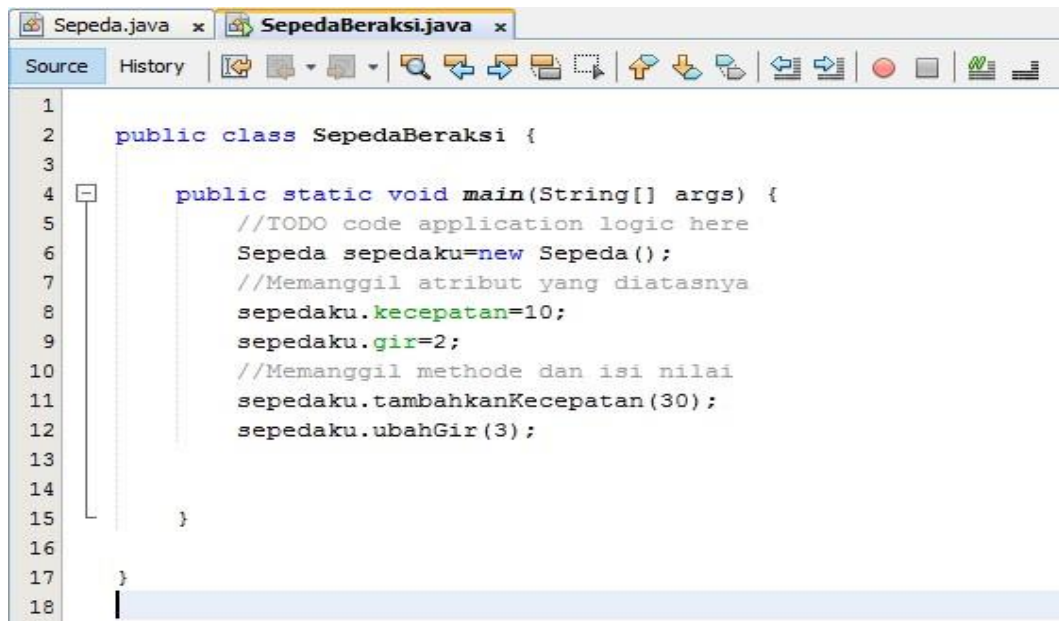
Contoh Penerapan Atribut

Membuat *class* dengan nama sepeda dan semua atribut yang dimiliki oleh kelas sepeda :



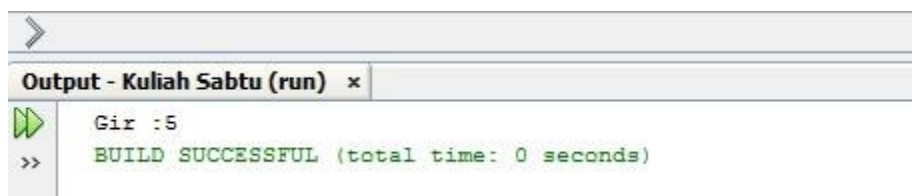
```
1 class Sepeda {
2
3     int kecepatan=0;
4     int gir=0;
5     //method yang ada didalam class
6     void ubahGir(int pertambahanGir){
7         gir=gir+pertambahanGir;
8         System.out.println("Gir :"+gir);
9     }
10
11     void tambahkanKecepatan(int pertambahanKecepatan){
12         kecepatan=kecepatan+pertambahanKecepatan;
13         System.out.println("Kecepatan :"+kecepatan);
14     }
15
16 }
17
```

Membuat *public class* yang memanfaatkan / memanggil atribut dalam kelas lainnya



```
1
2 public class SepedaBeraksi {
3
4     public static void main(String[] args) {
5         //TODO code application logic here
6         Sepeda sepedaku=new Sepeda();
7         //Memanggil atribut yang diatasnya
8         sepedaku.kecepatan=10;
9         sepedaku.gir=2;
10        //Memanggil methode dan isi nilai
11        sepedaku.tambahkanKecepatan(30);
12        sepedaku.ubahGir(3);
13
14    }
15
16
17 }
18
```

Hasil program setelah dijalankan



```
>
Output - Kuliah Sabtu (run) x
>> Gir :5
BUILD SUCCESSFUL (total time: 0 seconds)
```

BAB III

TUGAS PENDAHULUAN

3.1 Soal

- 1) Jelaskan perbedaan class dan object dengan menggunakan bahasamu sendiri !

3.2 Jawaban

- 1) Class adalah cetakan biru yang mendefinisikan sebuah atribut (variabel) dan perilaku (metode) yang dimiliki oleh objek. Class adalah template dasar untuk menciptakan objek.

Object adalah instance atau perwujudan nyata dari suatu class. Object memiliki atribut dan perilaku yang telah ditentukan oleh class. Object juga bisa memanggil metode yang sudah didefinisikan didalam class.

BAB IV

IMPLEMENTASI

4.1 Soal

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 Jawaban

1. A Source code

```
package manusia;

public class Manusia {
    // atribut
    String nama;
    int umur;
    String alamat;

    void berjalan(){
        System.out.println(nama + " SEDANG BERJALAN");
    }
    void berlari(){
        System.out.println(nama + " SEDANG BERLARI");
    }

    public static void main(String[] args) {

        Manusia atribut2 = new Manusia();
```

```

    atribut2.nama="MIFTAHUZZAIN ABRORI";
    atribut2.umur =18;
    atribut2.alamat="KAB SUMENEP";

    System.out.println("Nama : "+atribut2.nama);
    System.out.println("Umur : "+atribut2.umur);
    System.out.println("Alamat : "+atribut2.alamat);
    atribut2.berjalan();
    atribut2.berlari();

}

}

```

B. Output

run:

```

Nama : MIFTAHUZZAIN ABRORI
Umur : 18
Alamat : KAB SUMENEP
MIFTAHUZZAIN ABRORI SEDANG BERJALAN
MIFTAHUZZAIN ABRORI SEDANG BERLARI
BUILD SUCCESSFUL (total time: 1 second)

```

C. Penjelasan

- # Publik Class untuk diisi beberapa atribut dari objek
- # Void untuk diisi beberapa method dari objeknya
- # System.out.println untuk perintah print pada bahasa java
- # New untuk memperbarui nama class dan memanggil atribut dan mengisinya

2. A. Source code

```
package oop;
import java.util.Scanner;

public class OOP {
    String nama ;
    int nim;
    String prodi;
    String alamat;

    OOP (String Nama,int Nim, String Prodi,String Alamat){
        this.nama = Nama;
        this.nim = Nim;
        this.prodi = Prodi;
        this.alamat = Alamat;
    }

    void display(){
        System.out.println("NAMA :"+nama);
        System.out.println("NIM :"+nim);
        System.out.println("PRODI :"+prodi);
        System.out.println("ALAMAT :"+alamat);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Masukkan Nama : ");
        String Nama = scanner.nextLine();

        System.out.println("Masukkan Prodi : ");
        String Prodi = scanner.nextLine();
```

```

        System.out.println("Masukkan Alamat : ");
        String Alamat = scanner.nextLine();

        System.out.println("Masukkan Nim : ");
        int Nim = scanner.nextInt();

        OOP mhs = new OOP>Nama, Nim, Prodi, Alamat);
        mhs.display();
    }
}

```

B. Output

```

run:
Masukkan Nama :
MIFTAHUZZAIN ABRORI
Masukkan Prodi :
SISTEM INFORMASI
Masukkan Alamat :
KAB SUMENEP
Masukkan Nim :
230038
NAMA :MIFTAHUZZAIN ABRORI
NIM :230038
PRODI :SISTEM INFORMASI
ALAMAT :KAB SUMENEP
BUILD SUCCESSFUL (total time: 42 seconds)

```

C. Penjelasan

- # Scanner untuk membaca input dari user
- # Display digunakan agar user dapat meng-inputkan nilai variabel
- # This digunakan untuk membedakan antara variabel instance dan variabel lokal

nexLine untuk membaca sebuah string yang diinputkan user

nextInt untuk membaca sebuah bilangan bulat yang diinputkan user

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa Dalam pemrograman berorientasi objek, kelas adalah cetak biru untuk membuat objek. Kelas menentukan apa yang dimiliki sebuah objek (atribut) dan apa yang dapat dilakukan objek (metode). Umumnya kelas mewakili sebuah kata benda yang dapat merupakan seseorang, hewan, buah, ataupun benda mati.

Sedangkan objek adalah benda yang memiliki data dan perilaku tertentu, misalnya orang atau mobil. Kelas: blueprint atau template untuk membuat objek, yang mendefinisikan properti dan tindakan yang akan dimiliki objek dari kelas tersebut. Sedangkan menurut KBBI, objek merupakan suatu benda, hal, dan sebagainya yang dapat menjadi sasaran untuk diteliti hingga diperhatikan. Selain itu, Objek juga dapat diartikan sebagai nomina yang melengkapi verba transitif dalam klausa. Objek adalah instance dari class yang terdiri dari data yang spesifik untuk objek tersebut. Anda dapat menggunakan objek atau instance class secara bergantian.

5.2 Kesimpulan

Class merupakan template yang menggambarkan “tindakan” dan “sifat” suatu objek. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program. Objek adalah hasil instansiasi dari sebuah class dan setiap Setiap object itu unik. Meskipun dua object mungkin memiliki atribut yang sama, nilai atau data yang dipegang oleh atribut tersebut dapat bervariasi antar object.