

LAPORAN RESMI
MODUL I
(OBJECT DAN CLASS)

PEMROGRAMAN BERBASIS OBJEK



NAMA : DONY EKA OCTAVIAN PUTRA
N.R.P : 230441100041
DOSEN : NAMA DOSEN
ASISTEN : PUTRI NIHAYATUL HUSNA
TGL PRAKTIKUM : 18 MARET 2024

Disetujui : 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

Pemrograman berbasis objek adalah paradigma pemrograman yang menggunakan konsep "objek" sebagai unit utama dalam merancang dan mengembangkan aplikasi. Objek dalam OOP merupakan representasi dari data bersama dengan operasi-operasi yang dapat dilakukan pada data tersebut. Konsep utama dalam pemrograman berbasis objek meliputi: Encapsulation (Pembungkusan), Inheritance (Pewarisan), Polymorphism (Polimorfisme), dan Abstraction (Abstraksi). Dalam pemrograman berbasis objek memiliki bagian – bagian tersendiri salah satunya yaitu Object dan Class.

Dalam pemrograman berbasis objek, "objek" adalah instansi konkret dari sebuah "kelas". Objek memiliki atribut (variabel yang menyimpan data) dan metode (fungsi yang bekerja pada data tersebut). Sebagai contoh, dalam program simulasi perpustakaan, sebuah objek bisa menjadi "buku" dengan atribut seperti judul, pengarang, dan nomor panggil, serta metode seperti meminjam dan mengembalikan.

Sementara itu, "kelas" adalah blueprint atau template untuk membuat objek. Kelas mendefinisikan atribut dan metode yang akan dimiliki oleh objek yang dibuat dari kelas tersebut. Dengan kata lain, kelas adalah struktur data yang mendefinisikan karakteristik suatu objek. Misalnya, dalam contoh perpustakaan, kelas "Buku" akan mendefinisikan atribut seperti judul dan pengarang, serta metode seperti meminjam dan mengembalikan yang dimiliki oleh semua objek buku dalam program.

1.2 Tujuan

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

BAB II

DASAR TEORI

2.1 Class dan Object

Class diumpamakan seperti cetakan yang berguna untuk mencetak suatu object, class juga merupakan grup suatu object dengan kemiripan attribut, 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 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 dengan method (behaviour)

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

Object Sepeda

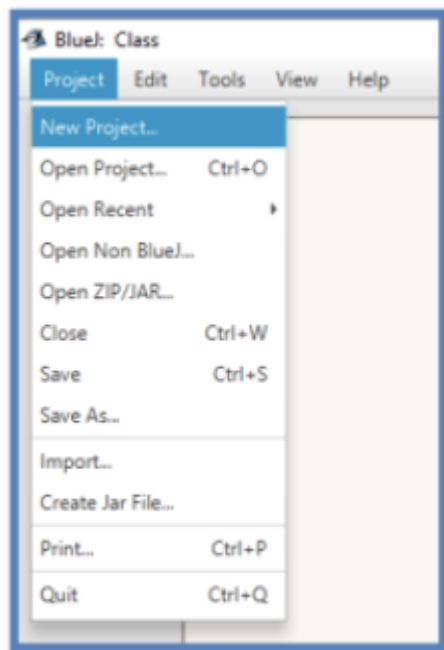
Dalam pengembangan perangkat lunak berorientasi object, object dalam perangkat lunak akan menyimpan state-nya dalam variable dan menyimpan informasi tingkah laku (behaviour) dalam method-method atau fungsi-fungsi. Untuk membuat object, kita menggunakan perintah new dengan sebuah nama class yang akan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

Contoh :

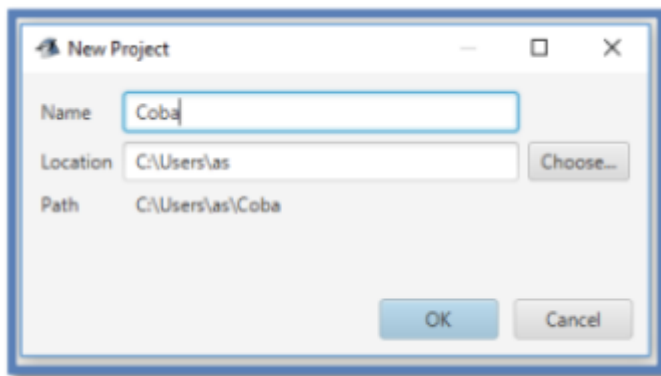
1. Tahapan membuat suatu class

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

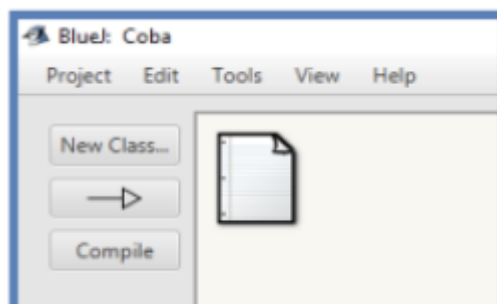
2. Buka aplikasi BlueJ, klik project pilih “New Project”.



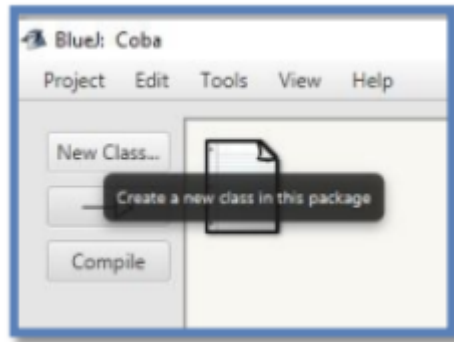
3. Pada form new project isilah nama project yang akan kita buat.



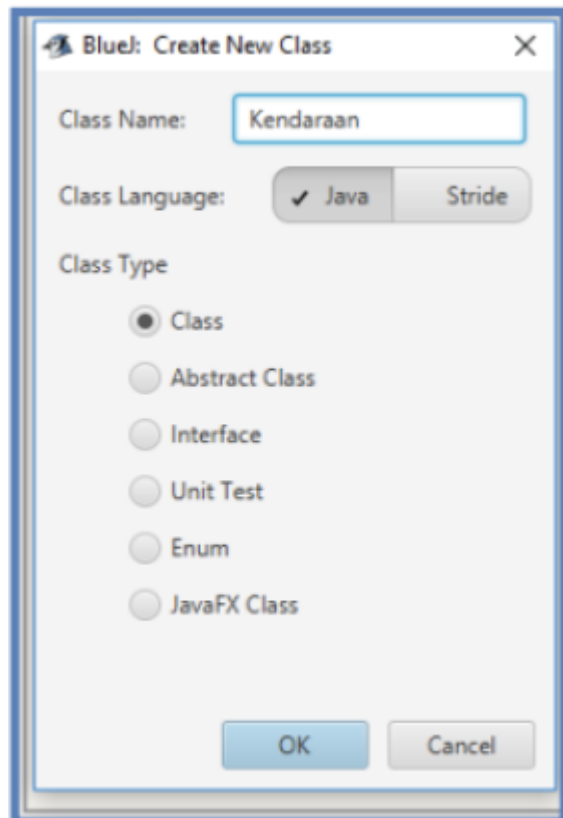
a. Tampilan form setelah nama project diisi.



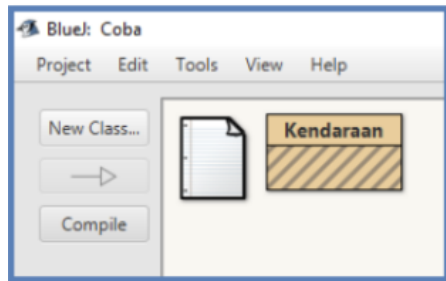
- b. Klik “New Class”, untuk membuat class



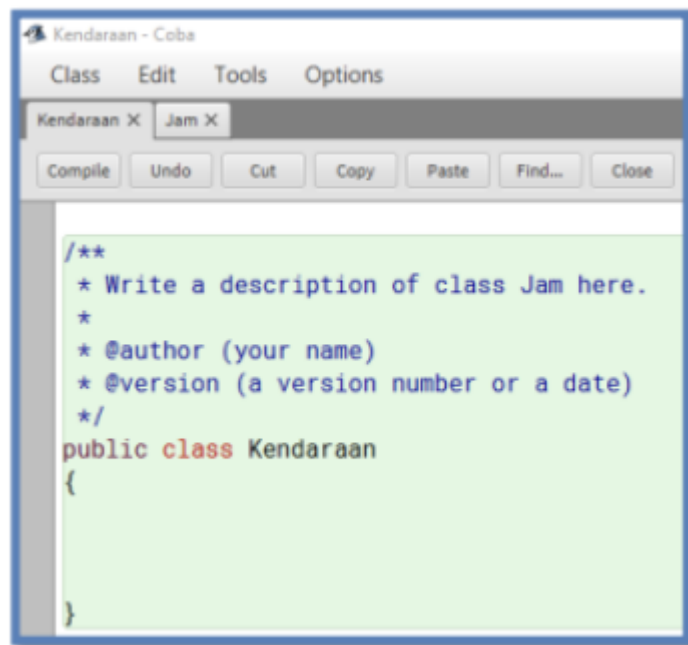
- c. Akan ada form baru yang digunakan untuk mengisi nama Class sesuai dengan yang kita inginkan pada kolom “Class Name”, lalu tekan “OK”.



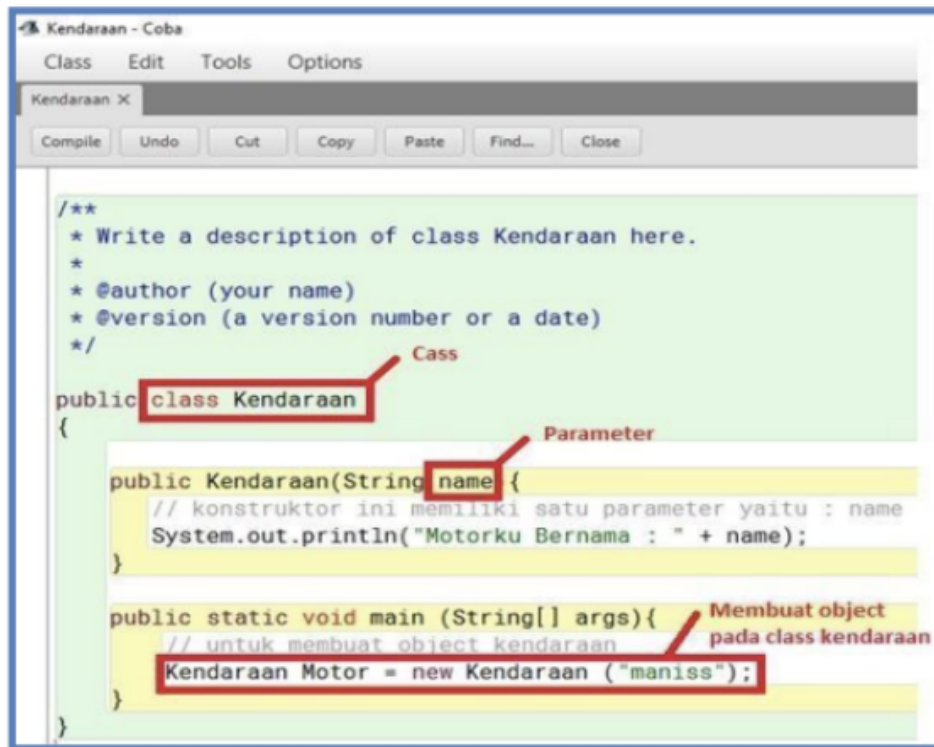
- d. Akan ada form baru yang digunakan untuk mengisi nama Class sesuai dengan yang kita inginkan pada kolom “Class Name”, lalu tekan “OK”.



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



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



Gambar 2. Source code

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 yakni ketika perintah “new” dijalankan.

Contoh Konstruktor seperti berikut :

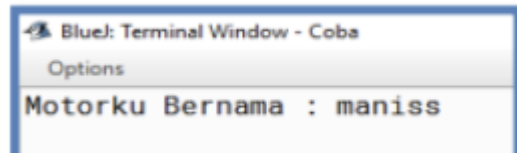
```
public Kendaraan (String name){
    // Konstruktor ini memiliki satu parameter yaitu : nama
    System.out.println ("Motorku bernama : " + name);
}
```

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.



Gambar 3. Output dari Class Kendaraan

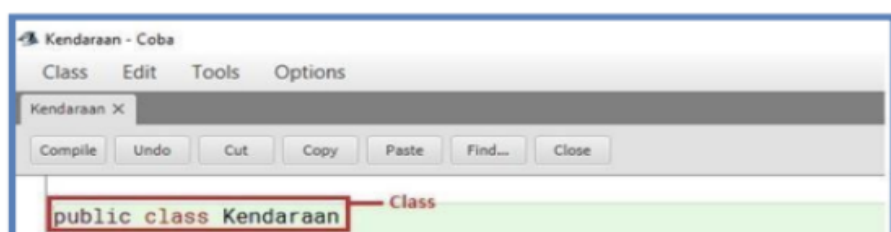
2,2 Method

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 namaparameternya.
4. Isi method.

Contoh :



Gambar 4

Penjelasan Gambar.1 :

- 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 membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.

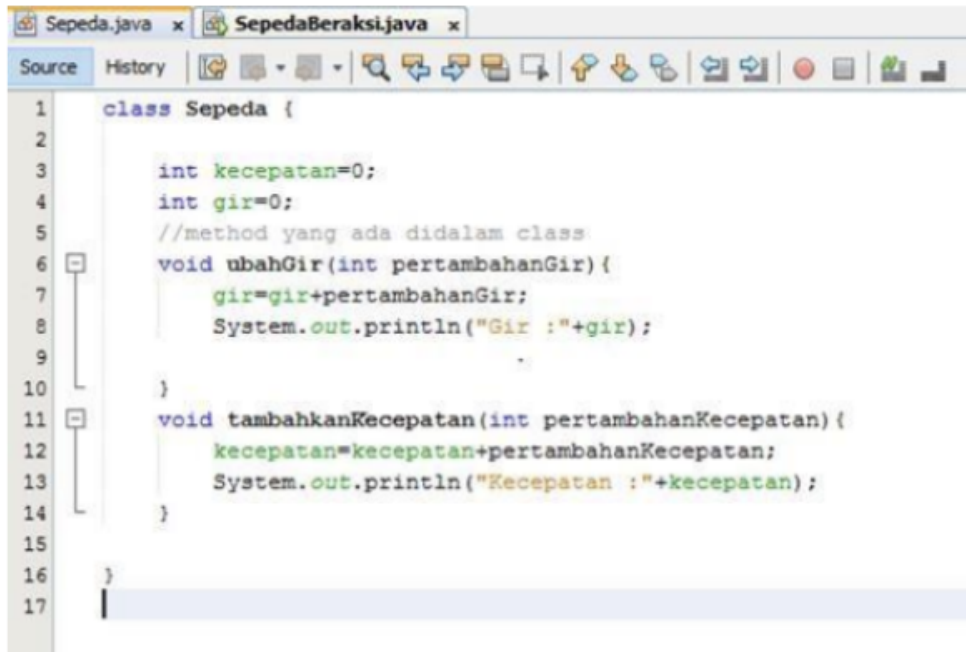
Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

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.

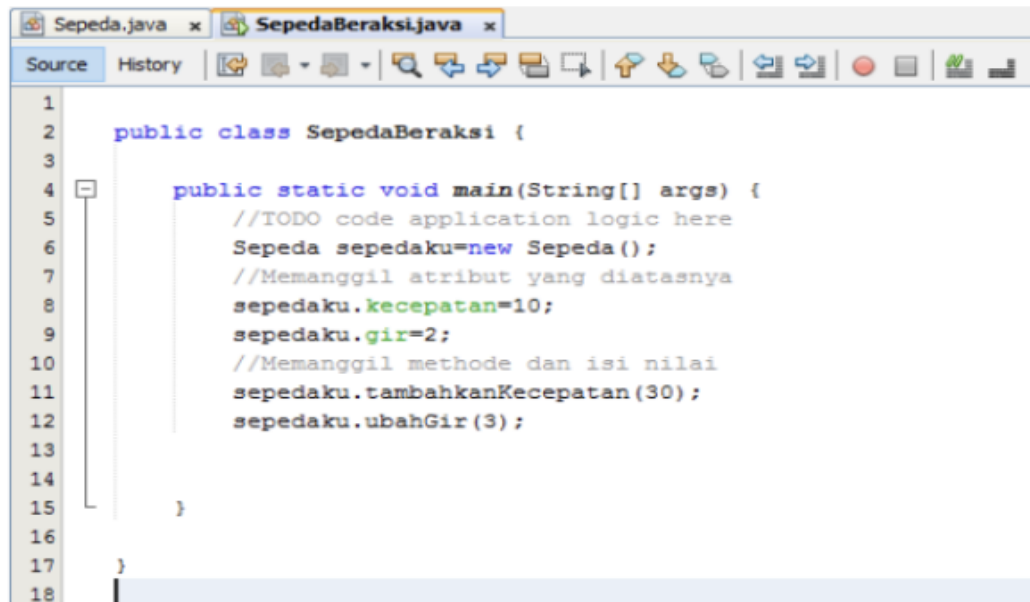
Contoh Penerapan Atribut

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



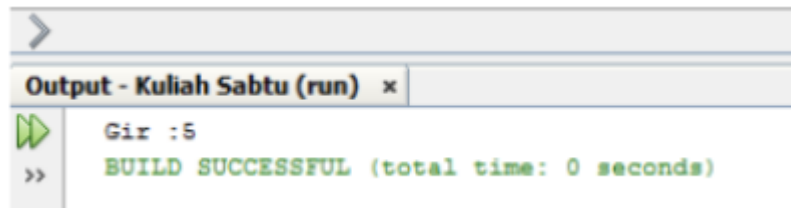
```
1 class Sepeda {
2
3     int kecepatan=0;
4     int gir=0;
5     //method yang ada didalam class
6     void ubahGir(int pertambahanGir){
7         gir=gir+pertambahanGir;
8         System.out.println("Gir :"+gir);
9     }
10
11     void tambahkanKecepatan(int pertambahanKecepatan){
12         kecepatan=kecepatan+pertambahanKecepatan;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

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



```
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 (Gambar 3).



BAB III

TUGAS PENDAHULUAN

3.1 soal

1. Jelaskan perbedaan class dan object dengan bahasamu sendiri!

3.2 Jawaban

Class merupakan kerangka kerja yang digunakan untuk membuat object. Ini menetapkan karakteristik (attribut) dan perilaku (metode) yang dimiliki oleh objek yang akan dibuat. Sebagai contoh, jika kita memiliki class "Mobil", maka class tersebut akan menerapkan atribut seperti "merk", "warna", dan metode seperti "maju", "mundur" dan "berhenti".

Object adalah instance konkret dari class yang sesungguhnya ada di dalam program. Ketika kita membuat objek dari class, kita menciptakan suatu entitas yang memiliki karakteristik dan perilaku yang telah ditentukan dalam class, misalnya jika kita membuat objek "mobil" dari class "mobil" maka "mobil" akan menjadi suatu objek yang memiliki merk, warna dan perilaku yang telah ditentukan dalam class "mobil". Jadi class adalah kerangka kerja.

BAB IV

IMPLEMENTASI

4.1 Source Code

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.

Jawaban

NO 1.

```
public class Manusia {  
    String Nama,Alamat;  
    int Umur,Berjalan,Berlari;  
  
    public void berjalan(){  
        System.out.println("Object " + this.Nama + " berjalan Sejauh "+  
this.Berjalan + "KM");  
    }  
    public void berlari (){  
        System.out.println("Object " + this.Nama + " Sedang berlari " +  
this.Berlari + "KM");  
    }  
  
    /**  
    * @param args the command line arguments  
    */  
    public static void main(String[] args) {  
        System.out.println("-----");  
        System.out.println("CONTOH OBJECT ORANG KE 1 ");  
        System.out.println("-----");  
    }  
}
```

```

Manusia Orang1 = new Manusia();
Orang1>Nama = "Dony Eka Octavian Putra";
Orang1.Umur = 19;
Orang1.Alat = "sidoarjo";
Orang1.Berjalan = 4;
Orang1.Berlari = 15;
System.out.println("Perkenalkan Nama saya "+Orang1>Nama);
System.out.println("Tahun ini saya Berumur "+Orang1.Umur +
"Tahun");
System.out.println("Saya Berasal dari Kota "+Orang1.Alat);
Orang1.berjalan();
Orang1.berlari();
System.out.println("-----");
System.out.println("CONTOH OBJECT ORANG KE 2 ");
System.out.println("-----");
Manusia Orang2 = new Manusia();
Orang2>Nama = "Donzz";
Orang2.Umur = 19;
Orang2.Alat = "Bojonegoro";
Orang2.Berjalan = 10;
Orang2.Berlari = 20;
System.out.println("Perkenalkan Nama saya "+Orang2>Nama);
System.out.println("Tahun ini saya Berumur "+Orang2.Umur + "
Tahun");
System.out.println("Saya Berasal dari Kota "+Orang2.Alat);
Orang2.berjalan();
Orang2.berlari();
}
}

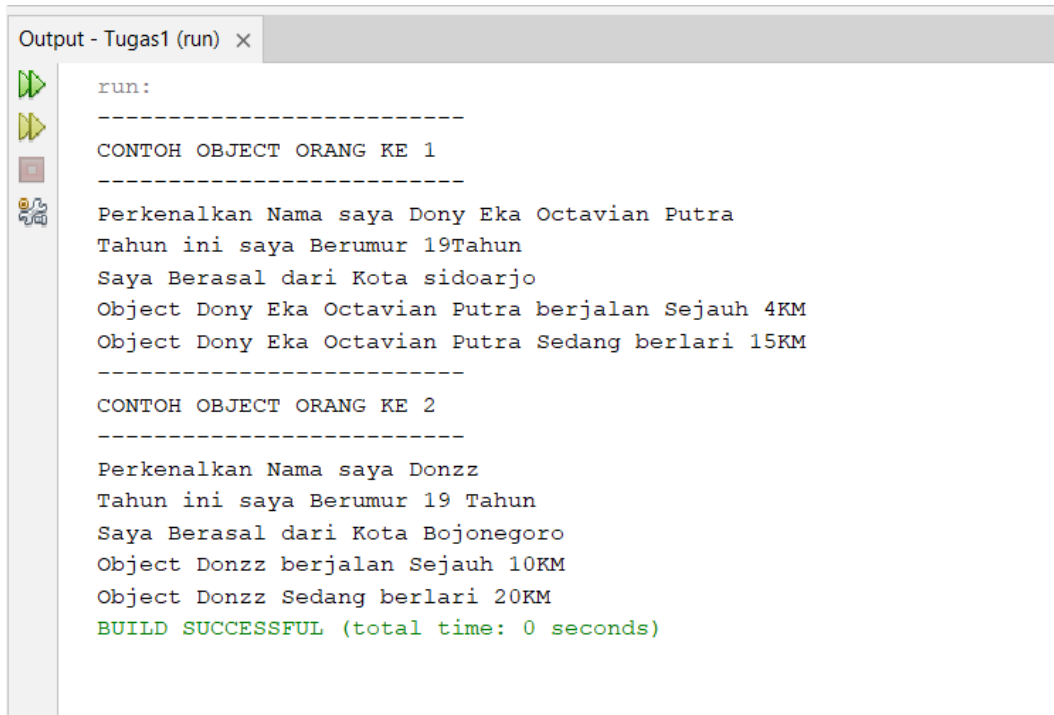
```

2.

```
public class Mahasiswa {  
    public static void main(String[] args) {  
        Scanner input = new Scanner (System.in);  
        String Nama, Jurusan, Alamat, Nim;  
  
        System.out.print("Masukkan Nama Anda Disini = ");  
        Nama=input.nextLine();  
  
        System.out.print("Masukkan Nim Anda Disini = ");  
        Nim=input.nextLine();  
  
        System.out.print("Masukkan Jurusan Anda Disini = ");  
        Jurusan=input.nextLine();  
        System.out.print("Masukkan Alamat Anda Disini = ");  
        Alamat=input.nextLine();  
        System.out.println("-----");  
        System.out.println("Nama kamu adalah " + Nama);  
        System.out.println("Nim kamu adalah " + Nim);  
        System.out.println("Jurusan kamu adalah " + Jurusan);  
        System.out.println("Alamat kamu adalah " + Alamat);  
        System.out.println("-----");  
    }  
}
```

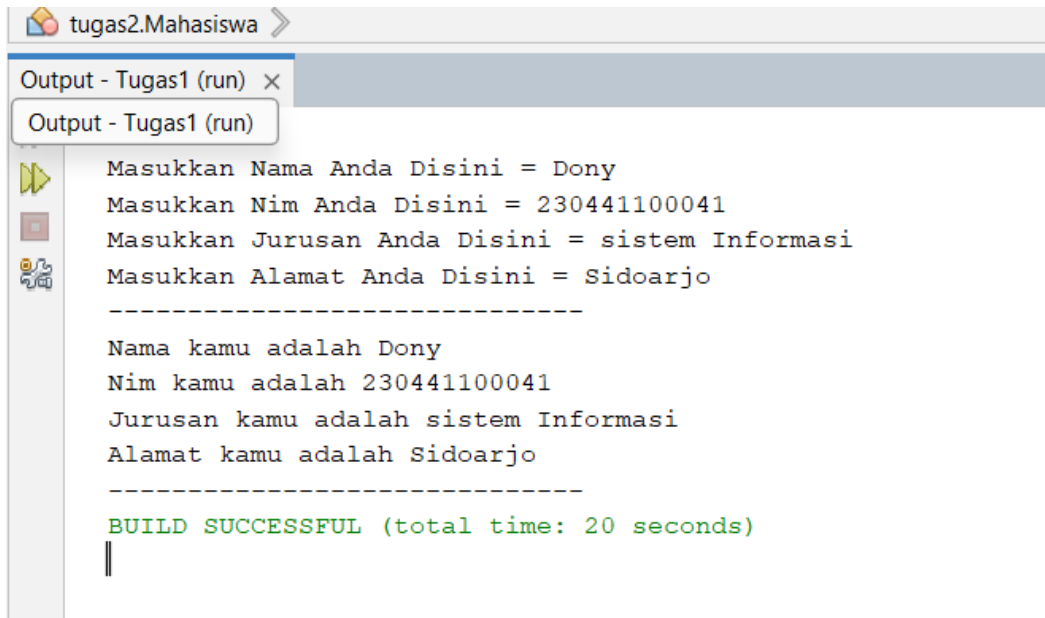

4.2 Hasil

1.



```
run:
-----
CONTOH OBJECT ORANG KE 1
-----
Perkenalkan Nama saya Dony Eka Octavian Putra
Tahun ini saya Berumur 19Tahun
Saya Berasal dari Kota sidoarjo
Object Dony Eka Octavian Putra berjalan Sejauh 4KM
Object Dony Eka Octavian Putra Sedang berlari 15KM
-----
CONTOH OBJECT ORANG KE 2
-----
Perkenalkan Nama saya Donzz
Tahun ini saya Berumur 19 Tahun
Saya Berasal dari Kota Bojonegoro
Object Donzz berjalan Sejauh 10KM
Object Donzz Sedang berlari 20KM
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.



```
tugas2.Mahasiswa >
Output - Tugas1 (run) x
Output - Tugas1 (run)
Masukkan Nama Anda Disini = Dony
Masukkan Nim Anda Disini = 230441100041
Masukkan Jurusan Anda Disini = sistem Informasi
Masukkan Alamat Anda Disini = Sidoarjo
-----
Nama kamu adalah Dony
Nim kamu adalah 230441100041
Jurusan kamu adalah sistem Informasi
Alamat kamu adalah Sidoarjo
-----
BUILD SUCCESSFUL (total time: 20 seconds)
||
```

4.3 Penjelasan

1.

Pertama, saya membuat sebuah class bernama Manusia yang memiliki atribut nama, umur, alamat, serta method Berjalan dan Berlari. Dalam method tersebut, saya menggunakan kata kunci `this` untuk memanggil objek yang bersangkutan. Kemudian, saya membuat method baru yaitu Berlari yang juga menggunakan `this` untuk memanggil objek.

Setelah itu, saya membuat dua objek, yaitu `orang1` dan `orang2`, yang merupakan instance dari class Manusia. Kemudian, saya memasukkan nilai ke dalam objek tersebut, seperti nama, umur, dan alamat. Terakhir, saya mencetak hasilnya menggunakan `System.out.println`.

2.

Pertama-tama, saya mengimport kelas Scanner untuk memungkinkan pengguna untuk memberikan input. Kemudian, saya membuat objek baru dari kelas Scanner untuk mengambil input dari pengguna. Setelah itu, saya membuat atribut-atribut seperti nama, nim, jurusan, dan alamat untuk menyimpan informasi yang diinputkan oleh pengguna. Selanjutnya, saya mencetak pertanyaan-pertanyaan kepada pengguna untuk mengisi informasi yang diperlukan. Terakhir, saya menggunakan metode `nextLine` dari objek Scanner untuk mengambil inputan dari pengguna dan menyimpannya ke dalam atribut yang sesuai.

BAB V

PENUTUP

5.1 Analisa

Class dan objek merupakan materi pertama dari modul pemrograman Berbasis Objek Yang di mana class merupakan cara awal untuk membuat program sedangkan objek merupakan atribut di dalam class, Class berfungsi untuk menamai sebuah program yang nantinya akan di tambah oleh objek di dalamnya sebagai contoh class bernama manusia yang di dalamnya ada objek nama, tinggi, jenis_kelamin, umur, dan lain sebagainya.

5.2 Kesimpulan

Di dalam Java, class adalah sebuah blueprint untuk menciptakan objek. Objek sendiri adalah instance dari class, yang memiliki atribut dan perilaku yang telah didefinisikan dalam class tersebut. Dengan menggunakan class dan object, kita bisa membuat kode yang lebih terstruktur dan modular, mempermudah pengelolaan dan pengembangan aplikasi.

Materi class dan object dalam Java memungkinkan pembuatan blueprint (class) untuk objek, yang merupakan instansi konkret dari class tersebut. Dengan konsep ini, kode dapat diorganisir secara terstruktur, modular, dan mudah dimengerti, me