

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 3
“KELAS DAN OBJEK ”**



Disusun oleh
Bunga Laelatul Muna
NIM : 21102010

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

BAB 1 – TUJUAN PRAKTIKUM

Tujuan dari praktikum instalasi Java (IntelliJ IDEA) adalah untuk memberikan pengetahuan dan keterampilan kepada mahasiswa tentang kelas dan objek yang ada di Java

1. Mahasiswa dapat memahami konsep dari class dan object pada pemrograman berorientasi objek
2. Mahasiswa mengetahui susunan yang ada pada class dan object
3. Mahasiswa dapat membuat suatu class dan object pada pemrograman Java

BAB II- DASAR TEORI

- **Kelas Dan Objek**

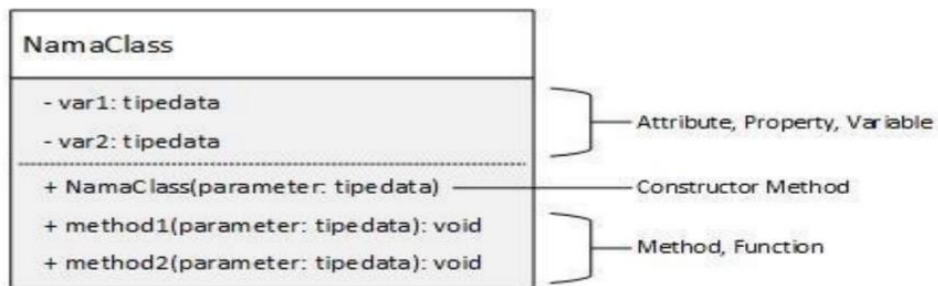
Objek adalah kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu sistem atau konsep yang memiliki sifat karakteristik dan fungsi. Dalam kehidupan sehari-hari kita sering menjumpai banyak objek dengan jenis yang sama, contohnya sepeda yang Anda miliki adalah salah satu jenis dari sepeda yang ada di dunia. Sepeda Anda adalah instance dari kelas yang disebut kelas sepeda.

Kelas adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (method) yang dimiliki oleh objek tersebut. Kelas dapat dianalogikan sebagai struktur data dari objek. Perbedaan kelas pada pemrograman berorientasi objek dengan struktur data pada pemrograman terstruktur adalah bahwa kelas pada pemrograman berorientasi objek tidak hanya berisi data saja tetapi juga fungsi-fungsi yang mengaksesnya, sehingga data dan fungsi harus dirancang secara bersamaan (kelas = struktur data + fungsi).

- **Diagram UML**

Dalam mengerjakan sebuah proyek, biasanya ada lebih dari satu orang yang terlibat dalam pengembangan proyek tersebut. Untuk mempermudah komunikasi dan perencanaan antar programmer dibuatlah yang namanya Class Diagram. Class Diagram adalah sebuah diagram yang didesain sebagai pedoman seorang atau sekelompok programmer dalam mendesain dan mengimplementasikan sebuah

Class diprogram atau proyek yang sedang dikerjakan. Perhatikan contoh Class Diagram berikut:



1. Attribute atau Property adalah "Apa saja yang dimiliki oleh Object yang dihasilkan dari Class tersebut".
2. Pada dasarnya Attribute atau Property adalah sebuah variable. Anda boleh menggunakan tipe data primitif maupun tipe data kompleks. Contoh tipe data kompleks adalah tipe data atau Class yang Anda buat sendiri.
3. Method pada dasarnya adalah Function. Di sebuah Class, method mewakili "Apa saja yang bisa dilakukan oleh Object yang dihasilkan dari Class tersebut". Setiap Class wajib memiliki sebuah Constructor method.
4. Constructor method adalah method atau function yang kita panggil saat kita ingin membuat sebuah Object dari Class tersebut.

BAB III – GUIDED

Guided 1 – Class animal

- Source Code (animal.Java)

```
//Bunga Laelatul Muna
// 21102010
package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.guided;

public class animal {
    //===== Atribut =====
    String name, color;
    int age;
    //===== Konstruktor =====
    public animal() {}
    //===== Konstruktor Berparameter =====
    public animal(String name, int age, String color) {
        this.name = name;
        this.age = age;
        this.color = color;
    }
    //===== Method =====
    public void tampildata() {
        System.out.println("Name : " + name);
        System.out.println("Age : " + age + " Tahun");
        System.out.println("Color : " + color);
        System.out.println();
    }
}
```

- Source Code (main.java)

```
package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.guided;

public class main {
    public static void main(String[] args) {
        animal cat = new animal();
        animal ant = new animal();
        animal spider = new animal();
        // ===== Data Cat =====
        cat.name = "Kitty";
        cat.age = 3;
        // ===== Data Ant =====
        ant.name = "Scott";
        ant.age = 1;
        // ===== Data Spider =====
        spider.name = "Maguire";
        spider.age = 5;

        //===== Tampil Data =====
        cat.tampildata();
        ant.tampildata();
        spider.tampildata();
    }
}
```

```
//===== ISI Data Dengan Parameter =====
animal dog = new animal("Puppy", 4, "Black");

//===== Tampil Data Berparameter =====
dog.tampildata();
    }
}
```

- **SS Output**

```
Name : Kitty
Age : 3 Tahun
Color : null

Name : Scott
Age : 1 Tahun
Color : null

Name : Maguire
Age : 5 Tahun
Color : null

Name : Puppy
Age : 4 Tahun
Color : Black
```

- **Penjelasan**

Ini adalah kelas Java bernama "animal" yang mendefinisikan atribut dan metode untuk objek hewan. Kelas ini memiliki tiga atribut: nama, warna, dan usia, yang direpresentasikan sebagai tipe data String dan int. Ada dua konstruktor yang didefinisikan untuk kelas tersebut. Konstruktor pertama adalah konstruktor default tanpa parameter, sedangkan konstruktor kedua adalah konstruktor berparameter yang menerima tiga parameter: nama, umur, dan warna.

Metode tampildata() adalah metode void yang mencetak nilai atribut objek ke konsol. Secara keseluruhan, kelas ini mendefinisikan cetak biru untuk membuat objek hewan dengan atribut tertentu. Setelah itu kita buat class Bernama "main" yang mendemonstrasikan penggunaan kelas "animal".

- Metode main() adalah titik masuk program. Kode tersebut membuat tiga objek hewan: kucing, semut, dan laba-laba, menggunakan konstruktor default dari kelas "animal".
- Atribut dari setiap objek hewan kemudian diatur menggunakan notasi titik.
- Metode tampildata() dipanggil pada setiap objek hewan untuk menampilkan atribut mereka ke konsol.
- Objek binatang lain, anjing, dibuat menggunakan konstruktor berparameter dari kelas "binatang", dan atributnya diatur dalam pemanggilan konstruktor.

Guided 2 – Class animal with Set and Get

- Source Code (animal.Java)

```
package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.guided;

public class animal {

    //===== Atribut =====
    String name, color;
    int age;

    // ===== Atribut Get and Set Public =====
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name = name;
    }
    public int getAge(){
        return age;
    }
    public void setAge(int age){
        this.age = age;
    }
    public String getColor(){
        return color;
    }
    public void setColor(String color){
        this.color = color;
    }

    //===== Konstruktor =====
    public animal(){}
    //===== Konstruktor Berparameter =====
    public animal(String name, int age,String color){
        this.name = name;
        this.age = age;
        this.color = color;
    }

    //===== Method =====
```

```

    public void tampildata() {
        System.out.println("Name : " + name);
        System.out.println("Age : " + age + " Tahun");
        System.out.println("Color : " + color);
        System.out.println();
    }
}

```

- **Source Code (main.java)**

```

package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.guided;
public class main {
    public static void main(String[] args) {
        //===== ISI Data Dengan Parameter =====
        animal dog = new animal("Puppy", 4, "Black");

        //===== Tampil Data Berparameter =====
        dog.tampildata();

        //===== Uji Coba Get dan Set =====
        System.out.println("Dog's Name (Before) : " +
dog.getName());
        dog.setName("Chedar");//ini kita ganti namanya
        System.out.println("Dog's Name (After) : " +
dog.getName());
    }
}

```

- **Output**

```

Name : Puppy
Age : 4 Tahun
Color : Black

Dog's Name (Before) : Puppy
Dog's Name (After) : Chedar

Process finished with exit code 0

```

Penjelasan :

Disini kita membuat class yang sama seperti guided 1. Hanya saja kita buat get untuk mengambil data dan set untuk merubah data. Setelah di animal.java dibuat konstruktornya kita atur di file main nya. Kita buat pemanggilan kelas dengan parameter dan kita buat set merubah data sebelumnya.

BAB IV – UNGUIDED

UnGuided 1 – Konversi Waktu

- Source Code

```
//Bunga Laelatul Muna
// 21102010
package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.tugas;

public class Profile {
    //===== attribute =====
    private String name;
    private int age;
    private String Nim;

    // ===== Atribut Get and Set Public =====
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name = name;
    }
    public int getAge(){
        return age;
    }
    public void setAge(int age){
        this.age = age;
    }
    public String getNim(){
        return Nim ;
    }
    public void setNim(String Nim){
        this.Nim = Nim;
    }

    //===== Konstruktor =====
    public Profile(){}
    //===== Konstruktor Berparameter =====
    public Profile(String name, int age,String Nim){
        this.name = name;
        this.age = age;
        this.Nim = Nim;
    }

    //===== Method =====
    public void tampildata(){
        System.out.println("Name : " + name);
        System.out.println("Age : " + age + " Tahun");
        System.out.println("Nim : " + Nim);
        System.out.println();
    }
}
```

- Source Code (main.java)

```
package com.Bunga_Laelatul_Muna.PBO.Pertemuan3.tugas;

import
```



```

com.Bunga_Laelatul_Muna.PBO.Pertemuan3.guided.animal;

public class Main {
    public static void main(String[] args) {
        //===== ISI Data Dengan Parameter =====
        Profile bunga_21102010 = new Profile("Bunga", 19,
"21102010");

        //===== Tampil Data Berparameter =====
        bunga_21102010.tampildata();

        //===== Uji Coba Get dan Set =====
        System.out.println("Profile's Name (Before) : " +
bunga_21102010.getName());
        bunga_21102010.setName("Bunga Laelatul
Muna");//ini kita ganti namanya
        System.out.println("Profile's Name (After) : " +
bunga_21102010.getName());
    }
}

```

- **SS Output**

```

Name : Bunga
Age : 19 Tahun
Nim : 21102010

Profile's Name (Before) : Bunga
Profile's Name (After) : Bunga Laelatul Muna

Process finished with exit code 0

```

- **Penjelasan**

Ini adalah kelas Java bernama "Profile" dengan tiga atribut: "nama", "umur", dan "Nim" (yang mungkin merupakan singkatan dari "Nomor Induk Mahasiswa" atau nomor induk mahasiswa dalam bahasa Indonesia). Kelas ini memiliki metode public getter dan setter untuk setiap atribut, dan dua konstruktor - satu tanpa parameter dan satu lagi dengan tiga parameter (nama, umur, dan Nim).

Kelas ini juga memiliki metode yang disebut "tampildata" (yang mungkin berarti "tampilkan data" dalam bahasa Indonesia), yang mencetak nilai atribut ke konsol dengan cara yang terformat.

Kemudian di code main akan dibuatkan class pemanggil untuk tiap-tiap atribut dan melakukan perubahan nama dengan set.

A. REFERENSI

1. PetaniKode
2. Java Documentation
3. Modul 3