

**LAPORAN PRAKTIKUM
PERTEMUAN 3 MODUL II**

Kelas dan Objek



Alif Alpian Sahrul Muharom (20102007)

Dosen :

Agus Priyanto S.kom, M.kom

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2022**

I. TUJUAN

1. Mahasiswa diharapkan mampu memahami mengenai objek dan kelas
2. Mahasiswa diharapkan mampu menerjemahkan objek dan kelas dalam Bahasa pemrograman

II. TOOL

1. CodeBlocks / Borland C++ / Visual C++
1. NetBeans IDE 13
2. Java SE Development Kit 18

III. DASAR TEORI

1. Kelas Dan Objek

Objek adalah kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu system 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. Objek adalah suatu abstraksi dari suatu problem. Sedangkan kelas adalah kumpulan objek yang memiliki atribut dan perilaku yang sama (karakteristik sama). Setiap objek memiliki nilai atribut/state yang unik yang membedakannya dengan objek lain dari kelas yang sama, dan objek memiliki perilaku/behaviour untuk mengakses atribut/state yang dimilikinya.

Contoh : kelas sepeda memiliki beberapa atribut/state (posisi gir, kecepatan, model, warna, dll) dan perilaku/behavior yang sama (mengubah gir, mengerem, dll). Objek sepeda Anda memiliki nilai atribut yang berbeda bila dibandingkan dengan objek sepeda yang lain, tetapi sepeda Anda dan sepeda-sepeda yang lain memiliki perilaku/behavior yang sama. 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). Praktikum Pemrograman Berorientasi Objek II – 2.

2. Mendefinisikan Kelas

Elemen-elemen dasar dalam mendefinisikan kelas :

A. Fields/variable

Field atau variable adalah implementasi dari atribut suatu objek. Field atau variable digunakan untuk menyimpan data dari objek.

Jenis Field atau variable :

- Instance variable
Setiap objek memiliki salinan sendiri dan salinan tersebut memiliki nilai masing-masing.
- Class variable
Suatu kelas hanya memiliki satu variable jenis ini dan digunakan bersama oleh semua objek dari kelas tersebut.

B. Constructor / Konstruktor

Konstruktor adalah method yang berfungsi untuk menginisialisasi variabel-variabel instance yang akan dimiliki oleh objek. Method konstruktor harus memiliki nama yang sama dengan nama kelas. Konstruktor ini dipanggil pada saat proses instansiasi kelas menjadi objek.

Kegunaan konstruktor :

- Mengalokasikan ruang bagi sebuah objek.
- Memberikan nilai awal terhadap anggota data suatu objek.
- Membentuk tugas-tugas umum lainnya.

Perlu diketahui :

- Konstruktor tidak mempunyai nilai balik(bahkan tanpa void).
- Konstruktor harus diletakkan pada bagian public.

Karakteristik constructor :

- Tidak pernah memiliki nilai balikan.
- Memiliki nama sama dengan nama kelas.

C. Methods

Method merupakan fungsi-fungsi implementasi perilaku objek untuk mengakses atribut - atributnya. Sebagai ilustrasi, kita ingin membuat kelas Buku, yang memiliki objek diantaranya novel dan fiksi. Karena kelas merupakan abstraksi dari objek, maka pemilihan atribut haruslah yang dapat merepresentasikan objek secara umum. Beberapa atribut/property yang digunakan, yaitu judul dan pengarang yang bertipe char/string, kemudian jumlah (untuk mengetahui berapa banyaknya buku maka bertipe integer). Selain data yang telah didefinisikan sebelumnya,kita juga dapat menentukan method yang dimiliki oleh kelas tersebut diantaranya fungsi untuk mengisikan data dan menampilkan data.

IV. GUIDED

Project Buku 1

A. Main

```
*/  
package com.Alpiann.pertemuan3.Guided.projectbuku1;  
  
/*  
 * @author Lenovo  
 */  
public class Main {  
    public static void main(String[] args) {  
        Buku naruto = new Buku();  
        naruto.setNilai("Luffy", "Mashashi Kishimoto", 1024);  
        //naruto.judul = "Naruto and Luffy";  
        //naruto.pengarang = "luffy";  
        //naruto.halaman = "1024"  
        //System.out.println("judul : " + this.judul);  
        //System.out.println("pengarang : " +  
this.pengarang);  
        //System.out.println("halaman : " + this.halaman);  
        naruto.cetakKeLayar();  
    }  
}
```

B. Buku

```
*/  
package com.Alpiann.pertemuan3.Guided.projectbukul;  
/**  
 *  
 * @author Lenovo  
 */  
public class Buku {  
    private String judul;  
    private String pengarang;  
    private int halaman;  
  
    public void setNilai(String judul,  
        String pengarang, int halaman){  
        this.judul = judul;  
        this.pengarang = pengarang;  
        this.halaman = halaman;  
    }  
    public void cetakKeLayar(){  
        System.out.println("judul : " + this.judul);  
        System.out.println("pengarang : " + this.pengarang);  
        System.out.println("halaman : " + this.halaman);  
    }  
}
```

Deskripsi Program

Pada guided yang pertama merupakan program yang memuat dua class yaitu class Buku untuk mendeklarasikan atribut konstruktor yang nantinya akan dieksekusi dengan dimasukan elemen yang nantinya memperoleh output berisikan atribut – atribut yang sudah dideklarasikan sebelumnya.

Project Buku 2

A. Main

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
to edit this template
 */
package com.Alpiann.pertemuan3.Guided.projectbuku2;

/**
 *
 * @author Lenovo
 */
public class Main {
    public static void main(String[] args) {
        Buku naruto = new Buku("Naruto", "Oda", 89);
        Buku Onepiece = new Buku("luffy", "hema", 1024);
        Buku Hilman = new Buku("hilman", "Frank", 53);
        naruto.cetakKeLayar();
        Onepiece.cetakKeLayar();
        Hilman.cetakKeLayar();
    }
}
```

B. Buku

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.Alpiann.pertemuan3.Guided.projectbuku2;

/**
 *
 * @author Lenovo
 */
public class Buku {

    private String judul;
    private String pengarang;
    private int halaman;

    public Buku(String judul,
                String pengarang,
                int halaman){
        this.judul = judul;
        this.pengarang = pengarang;
        this.halaman = halaman;
    }

    public Buku(String judul){
        this.judul = judul;
        this.pengarang = "tidak diketahui";
        this.halaman = 0;
    }

    public Buku(){
```

```
        this.judul = "tidak diketahui";  
        this.pengarang = "tidak diketahui";  
        this.halaman = 0;  
    }  
    public void cetakKeLayar(){  
        System.out.println("judul : " + this.judul);  
        System.out.println("pengarang : " + this.pengarang);  
        System.out.println("halaman : " + this.halaman);  
    }  
}
```

Deskripsi Program

Sama seperti guided sebelumnya dimana program memiliki dua class yaitu class Buku dan Main. Dimana pada class Buku mendeklarasikan tiga buah konstruktor yang akan dieksekusi pada class Main. Pada class Main yaitu mengisi elemen yang akan disikan pada konstruktor yang telah dibuat sebelumnya, lalu dirun akan keluar seperti gambar diatas.

V. UNGUIDED

A. UNGUIDED 1

1. Bank

```
/*
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
 to edit this template
 */
package com.Alpiann.pertemuan3.Unguided;

/**
 *
 * @author Lenovo
 */
public class Bank {
    private int saldo,ambil,simpan;

    public Bank(){
        this.saldo = 100000;
    }
    public void simpanUang(int simpan){
        this.simpan = simpan;
    }
    public void ambilUang(int ambil){
        this.ambil = ambil;
    }
    public int getSaldo(){
        return saldo;
    }

    public void cetakKelayar(){
        System.out.println("Selamat datang di Bank ABC");
        System.out.println("Saldo saat ini : Rp." + saldo);
        System.out.println("Simpan uang : Rp." + simpan);
        System.out.println("Saldo saat ini : Rp." + (simpan +
saldo));
        System.out.println("Ambil uang : Rp." + ambil);
    }
}
```

```
        System.out.println("Saldo saat ini : Rp." + (simpan +  
saldo - ambil));  
    }  
}
```

2. BankBeraksi

```
/*  
 * Click  
nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-  
default.txt to change this license  
 * Click  
nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java  
to edit this template  
 */  
package com.Alpiann.pertemuan3.Unguided;  
  
/**  
 *  
 * @author Lenovo  
 */  
public class BankBeraksi {  
    public static void main(String[] args) {  
        Bank BankBeraksi = new Bank();  
        BankBeraksi.getSaldo();  
        BankBeraksi.simpanUang(500000);  
        BankBeraksi.ambilUang(150000);  
        BankBeraksi.cetakKelayar();  
    }  
}
```

Deskripsi Program

Pada unguided yang pertama ialah membuat dua class yaitu class Bank dan BankBereaksi. Dimana isi dari class Bank mendeklarasikan konstruktor yang nantinya akan dieksekusi. Pada class BankBereaksi yaitu berisikan prosedur yang berisikan elemen – elemen untuk dimasukan nilainya kedalam konstruktor pada class sebelumnya.

B. UNGUIDED 2

1. Main

```
/*
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
 to edit this template
 */
package com.Alpiann.pertemuan3.Unguided;

import java.util.Scanner;

/**
 *
 * @author Lenovo
 */
public class Main {
    public static void main(String[] args) {
        int pilih;
        int a,b,c,d,e;
        Scanner inputan;

        inputan = new Scanner(System.in);

        Titik titik = new Titik();

        do{
            titik.tampilMenu();
            System.out.println("Pilih menu : ");
            pilih = inputan.nextInt();

            switch(pilih){
            case 1 -> {
                System.out.println("Masukkan Nilai x: ");
                a = inputan.nextInt();
                System.out.println("Masukkan Nilai y: ");
```

```

        b = inputan.nextInt();
        titik.inisialisasiTitik(a, b);
    }

    case 2 -> titik.tampilTitik();
    case 3 -> {
        System.out.println("Masukkan Nilai Skalar ; " );
        c = inputan.nextInt();
        titik.perkalianSkalar(c);
    }
    case 4 -> titik.pencerminanSumbuX();
    case 5 -> titik.pencerminanSumbuY();
    case 6 -> {
        System.out.println("Masukkan Nilai TITik x Kedua :
");
        d = inputan.nextInt();
        System.out.println("Masukkan Nilai Titik y Kedua :
");
        e = inputan.nextInt();
        System.out.println("Jarak Antar Titik = " +
titik.Jarak(d, e));
    }
        default -> System.out.println("Inputan yang
dimasukkan Salah!");
    }
    }while(pilih != 0);
}
}

```

2. Titik

```
/*
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
 to edit this template
 */
package com.Alpiann.pertemuan3.Unguided;

/**
 *
 * @author Lenovo
 */
public class Titik {
    private int x,y,x1,y1;
    private int skalar;

    public Titik(){
        this.x = 1;
        this.y = 2;
    }

    public void inisialisasiTitik(int x, int y){
        this.x = x;
        this.y = y;
    }

    public void tampilTitik(){
        System.out.println("Titik x : " + this.x);
        System.out.println("Titik y : " + this.y);
    }

    public void perkalianSkalar(int skalar){
        this.skalar = skalar * this.x * this.y;
        System.out.println("Hasil perkalian skalar = " +
this.skalar);
    }
}
```

```

void pencerminanSumbuX() {
    System.out.println("Titik x : " + this.x);
    System.out.println("Titik y : -" + this.y);
}

void pencerminanSumbuY() {
    System.out.println("Titik x : -" + this.x);
    System.out.println("Titik y : " + this.y);
}

double Jarak(int x1, int y1){
    this.x1 = x1;
    this.y1 = y1;
    return Math.sqrt((this.x1 - this.x) * (this.x1 - this.x) *
        (this.y1 - this.y) * (this.y1 - this.y));
}

    void tampilMenu(){
        System.out.println("\n \n...MENU OPERASI TITIK... ");
        System.out.println("1. Inisialisasi Titik");
        System.out.println("2. Tampil Titik");
        System.out.println("3. Perkalian Skalar Titik");
        System.out.println("4. Pencerminan Terhadap Sumbu
X");
        System.out.println("5. Pencerminan Terhadap Sumbu
Y");
        System.out.println("6. Jarak Antara Dua Titik");
    }
}

```

Deskripsi Program

Pada unguided yang kedua yaitu pada class Titik membuat sebuah menu menggunakan switch case dan juga mendeklarasikan konstruktor. Lalu pada class Main yaitu memanggil konstruktor yang telah dibuat di class sebelumnya, setelah dirun user akan diminta untuk memasukan nilai yang diminta kedalam prosedur. Lalu akan muncul menu-menu dan output hasilnya seperti gambar diatas.

VI. KESIMPULAN

Class merupakan suatu blueprint atau cetakan untuk menciptakan suatu instant dari object. Class juga merupakan grup suatu object dengan kemiripan attributes/properties, behaviour dan relasi ke object lain.

Method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. Metode menentukan perilaku objek, yakni apa yang terjadi ketika objek itu dibuat serta berbagai operasi yang dapat dilakukan objek sepanjang hidupnya. Ada 4 (Empat) bagian dasar yang dimiliki metode antara lain:

- Nama metode
- Tipe Objek atau tipe primitive yang dikembalikan metode.
- Daftar parameter.
- Badan atau isi metode.

Construktor adalah Constructor merupakan suatu method yang akan memberikan nilai awal pada saat suatu objek dibuat. Pada saat program dijalankan, constructor akan bekerja dengan constructor, hal mendasar yang perlu diperhatikan, yaitu :

- Nama Constructor sama dengan nama Class.
- Tidak ada return type yang diberikan kedalam Constructor Signature.
- Tidak ada return statement, didalam tubuh constructor.