

LAPORAN PRAKTIKUM
PERTEMUAN 2
KELAS DAN OBJEK



Nama :

Aufa Salsabila Nahrowi(20102040)

Dosen :

Agus Priyanto, S.Kom., M.Kom

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

I. TUJUAN

- a. Mahasiswa diharapkan mampu memahami mengenai objek dan kelas
- b. Mahasiswa diharapkan mampu menerjemahkan objek dan kelas dalam bahasa pemrograman

II. TOOL

Software Visual Studio Code

III. DASAR TEORI

a. 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).

b. Elemen-elemen Kelas

1. 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.

2. 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 adalah :

- 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.

3. Methods

Method merupakan fungsi-fungsi implementasi perilaku objek untuk mengakses atributatributnya.

IV. GUIDED I

```
public class Buku {  
    private String judul;  
    private String pengarang;  
    private int jumlah;  
  
    public void setNilai(String judul, String pengarang, int  
jumlah){  
        this.judul = judul;  
        this.pengarang = pengarang;  
        this.jumlah = jumlah;  
    }  
    void cetakKelayar(){
```

```

        System.out.println("Judul: " + judul);
        System.out.println("Pengarang: " + pengarang);
        System.out.println("Jumlah: " + jumlah);
    }
}

```

```

public class App {
    public static void main(String[] args){
        Buku a = new Buku();
        a.setNilai(" Jurassic Park ", " Michael Crichton ", 21);
        a.cetakKelayar();
    }
}

```

V. GUIDED II

```

public class Buku2 {
    private String judul;
    private String pengarang;

    public Buku2(){
        judul = "Tidak diketahui";
        pengarang = "Tidak diketahui";
    }

    public Buku2(String judul, String pengarang){
        System.out.println("konstruktor buku sedang
berjalan...");
        this.judul = judul;
        this.pengarang = pengarang;
    }

    void cetakKelayar(){
        if(judul==null && pengarang==null) //null = vRIABLE YANG
TAK TERBCA
        return;
        System.out.println("Judul: " + judul);
        System.out.println("Pengarang: " + pengarang);
    }
}

```

```

public class App {

```

```

    public static void main(String[] args) throws Exception {
        Buku2 a,b = new Buku2();
        a = new Buku2(" Siaga Merah ", " Alistair Maclean");
        b = new Buku2();
        a.cetakKelayar();
        b.cetakKelayar();
    }
}

```

VI. UNGUIDED I

```

public class Bank {
    private int saldo;
    private int simpanUang;
    private int ambilUang;

    public Bank(int saldo){
        this.saldo=saldo;
        System.out.println("Selamat Datang di Bank ABC");
        System.out.println("Saldo saat ini: Rp. " + saldo);
    }
    public void simpanUang(int simpanUang){
        saldo += simpanUang;
        System.out.println("\nSimpan Uang : Rp. "+ simpanUang);
        System.out.println("Saldo saat ini : Rp. " +saldo);
    }
    public void ambilUang(int ambilUang){
        saldo -= ambilUang;
        System.out.println("\nAmbil Uang : Rp. "+ ambilUang);
        System.out.println("Saldo saat ini : Rp. " +saldo);
    }

    public int getSaldo(){
        return saldo;
    }
}

```

```

public class BankBeraksi {
    public static void main(String[] args) throws Exception {
        Bank BankBeraksi = new Bank (100000);
        BankBeraksi.simpanUang(50000);
        BankBeraksi.ambilUang(15000);
    }
}

```

VII. UNGUIDED II

```
public class Titik {
    private double x;
    private double y;

    public Titik(){
        x=0;
        y=0;
    }
    public void inisialisasiTitik(int x, int y){
        this.x = x;
        this.y = y;
        System.out.println("Sukses");
    }
    public void tampilTitik(){
        System.out.println(x);
        System.out.println(y);
        System.out.println("Titik X dan Y adalah: (" + x + " , "
+y+"))");
    }
    public void perkalianSkalar(int skalar){
        System.out.println("Perkalian skalar dengan X : " +
(x*skalar));
        System.out.println("Perkalian skalar dengan Y : " +
(y*skalar));
    }
    public void pencerminanSumbuX(){
        System.out.println("Sebelum dicermin: (" + x + " , " + y
+"))");
        System.out.println("Setelah dicermin: " + x + " , " + (y
*= -1));
    }
    public void pencerminanSumbuY(){
        System.out.println("Sebelum dicermin: (" + x + " , " + y
+"))");
        System.out.println("Setelah dicermin: " + (x *= -1) +
" , " + y);
    }
    public static double jarak(Titik t1, Titik t2){
        double jarakX = t1.x - t2.x;
        double jarakY = t1.y - t2.y;
        return Math.sqrt((jarakX * jarakX) + (jarakY * jarakY));
    }
}

import java.util.Scanner;
```

```

public class App {
    public static void main(String[] args) throws Exception {

        Titik baru = new Titik();

        Scanner input = new Scanner(System.in);

        System.out.println("MENU OPERASI TITIK");
        System.out.println("1. INISIALISASI TITIK");
        System.out.println("2. TAMPIL TITIK");
        System.out.println("3. PERKALIHAN 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");
        System.out.println("7. KELUAR");

        while(true) {
            System.out.println("MASUKAN PILIHAN: ");
            int pil = input.nextInt();
            switch(pil) {
                case 1:
                    System.out.print("Masukkan nilai X : ");
                    int nilaiX = input.nextInt();
                    System.out.print("Masukkan nilai Y : ");
                    int nilaiY = input.nextInt();
                    baru.inisialisasiTitik(nilaiX, nilaiY);
                    break;
                case 2 :
                    baru.tampilTitik();
                    break;
                case 3 :
                    System.out.print("Masukkan nilai skalar : ");
                    int skalar = input.nextInt();
                    baru.perkalianSkalar(skalar);
                    break;
                case 4 :
                    baru.pencerminanSumbuX();
                    break;
                case 5:
                    baru.pencerminanSumbuY();
                    break;
                case 6:
                    Titik t1 = new Titik();
                    System.out.print("Masukkan nilai X titik 1 : ");
                    int x1 = input.nextInt();

```

```

        System.out.print("Masukkan nilai Y titik 1 : ");
        int y1 = input.nextInt();

        Titik t2 = new Titik();
        System.out.print("Masukkan nilai X titik 2 : ");
        int x2 = input.nextInt();
        System.out.print("Masukkan nilai Y titik 2 : ");
        int y2 = input.nextInt();

        System.out.println("Jarak kedua titik adalah : "
+ baru.jarak(t1,t2));
        break;
    case 7:
        break;
    default:
        break;
    }
}
}
}

```

VI. KESIMPULAN

Lebih mengenal objek, kelas, method setelah melakukan implementasian kode ke dalam program. Dan menjadi lebih memahaminya. Kelas bisa dipanggil sebanyak-banyaknya di main program. Dan nama kelas menyesuaikan penamaan filenya.