

## **Laporan Praktikum**

### **MODUL 2**

#### **Kelas Dan Objek**



# **Institut Teknologi Telkom Purwokerto**

**Nama :**

**Nathaya Elang Mariantaka (2211103128)**

**Nama Dosen :**

**Nicolaus Euclides Wahyu Nugroho, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SISTEM INFORMASI  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2023**

## BAB I. TUJUAN

### I. Tujuan

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

### II. Tool

- Code Blocks
- IntelliJ 2023

### III. Dasar Teori

#### a. Kelas Dan Objek

**Objek** dalam Praktikum Pemrograman Berorientasi Objek (PBO) adalah instance (contoh konkret) dari sebuah class. Pemrograman Berorientasi Objek adalah paradigma pemrograman yang berfokus pada pengorganisasian kode dalam bentuk objek-objek yang merepresentasikan entitas-entitas dalam dunia nyata. Setiap objek memiliki atribut (data) dan metode (fungsi) yang dapat digunakan untuk berinteraksi dengan objek tersebut.

Contoh sederhana dari PBO adalah jika Anda memiliki class "Mobil," maka objek-objek yang dapat kita buat dari class tersebut mungkin termasuk "Mobil1" dan "Mobil2," yang memiliki atribut seperti warna, kecepatan, dan metode seperti "Maju" atau "Mundur."

**Kelas** dalam Pemrograman Berorientasi Objek (PBO) adalah blueprint atau cetak biru yang digunakan untuk membuat objek. Kelas adalah struktur dasar yang digunakan untuk mendefinisikan atribut (data) dan metode (fungsi) yang akan dimiliki oleh objek-objek yang akan dibuat berdasarkan kelas tersebut. Dalam PBO, kelas merupakan abstraksi dari entitas atau objek yang ada dalam dunia nyata.

Sebagai contoh, jika kita ingin membuat sebuah program untuk mengelola data pelanggan, kita dapat membuat kelas "Pelanggan." Kelas "Pelanggan" ini akan mendefinisikan atribut-atribut seperti nama, alamat, nomor telepon, dan metode-metode seperti menambahkan pelanggan baru, menghapus pelanggan, atau memperbarui informasi pelanggan.

## **b. Mendefinisikan Kelas**

Elemen-elemen dasar dalam mendefinisikan 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 :**

- 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 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

### a. Hubungan antara kelas, objek, atribut, dan method

```
#include <iostream>
#include <cstring>

class Buku
{
private:
    char judul[35];
    char pengarang[25];
    int jumlah;

public:
    void inisialisasiBuku(const char *jdl, const char *pngarang, int jmlh)
    {
        strcpy(judul, jdl);
        strcpy(pengarang, pngarang);
        jumlah = jmlh;
    }

    void infoBuku()
    {
        std::cout << " Judul : " << judul << std::endl;
        std::cout << " Pengarang : " << pengarang << std::endl;
        std::cout << " Jumlah buku : " << jumlah << std::endl;
    }
};

int main()
{
    Buku novel, fiksi;
    novel.inisialisasiBuku("Meriam Benteng Navarone", "Alistair Maclean", 12);
    fiksi.inisialisasiBuku("Jurassic Park", "Michael Crichton", 3);

    novel.infoBuku();
    fiksi.infoBuku();

    return 0;
}
```

CPP

```
Judul      : Meriam Benteng Navarone
Pengarang   : Alistair Maclean
Jumlah buku : 12
Judul      : Jurassic Park
Pengarang   : Michael Crichton
Jumlah buku : 3
```

**Buku.java**

```
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 );
    }
}
```

ProjectBuku1

**Main.java**

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

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program
Judul      :  Jurassic Park
Pengarang   :  Michael Crichton
Jumlah      :  21

Process finished with exit code 0
```

**b. Konstruktor**

```
#include <iostream>
```

CPP Komplex

```
class Kompleks
```

```
{
```

```
private:
```

```
    double re, im;
```

```
public:
```

```
    Kompleks()
```

```
{
```

```
    std::cout << "Konstruktor Kompleks dijalankan...." << std::endl;
```

```
    re = 5.2;
```

```
    im = 3.6;
```

```
}
```

```
    Kompleks(double real, double imajiner)
```

```
{
```

```
    std::cout << "Konstruktor Kompleks dijalankan...." << std::endl;
```

```
    re = real;
```

```
    im = imajiner;
```

```
}
```

```
    void info()
```

```
{
```

```
    std::cout << "\nBilangan kompleks : " << std::endl;
```

```
    std::cout << "real = " << re << std::endl;
```

```
    std::cout << "imajiner = " << im << std::endl;
```

```
    std::cout << std::endl;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    Kompleks a;
```

```
    a.info();
```

```
    Kompleks b(3.2, 4.1);
```

```
    b.info();
```

```
    return 0;
```

```
}
```

Konstruktor Kompleks dijalankan....

Bilangan kompleks :

real = 5.2

imajiner = 3.6

Konstruktor Kompleks dijalankan....

Bilangan kompleks :

real = 3.2

imajiner = 4.1

Buku.java

```
import org.w3c.dom.ls.LSOutput;
```

ProjectBuku2

```
class Buku {  
    private String judul;  
    private String pengarang;  
  
    public Buku() {  
        judul = "Semua Karena Cinta";  
        pengarang = "Ahmad Bahlul";  
    }  
    public Buku(String judul, String pengarang) {  
        System.out.println("Konstruktor buku sedang dijalankan...");  
        this.judul = judul;  
        this.pengarang = pengarang;  
    }  
  
    void cetakKeLayar() {  
        if (judul == null && pengarang == null)  
            return;  
        System.out.println("Judul      : " + judul);  
        System.out.println("Pengarang  : " + pengarang);  
        System.out.println();  
    }  
}
```

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Buku a, b;  
        a = new Buku("Siaga Merah", "Alistair Maclean");  
        b = new Buku();  
        a.cetakKeLayar();  
        b.cetakKeLayar();  
    }  
}
```

Konstruktor buku sedang dijalankan...

Judul : Siaga Merah

Pengarang : Alistair Maclean

Judul : Semua Karena Cinta

Pengarang : Ahmad Bahlul



## V. UNGUIDED

1. Buatlah program dalam bahasa Java dimana anda diminta untuk mengimplementasikan kasus operasi terhadap suatu Bank dalam konsep PBO.

a. Buat class Bank

- Buat konstruktor class Bank dengan parameter: saldo (bobot 5 %)

```
public class Bank {
    private double saldo;

    public Bank(double saldoAwal) {
        saldo = saldoAwal;
    }
}
```

- Buat method: simpanUang, ambilUang, dan getSaldo (bobot 30 %)

```
public void simpanUang(double jumlah) {
    if (jumlah > 0) {
        saldo += jumlah;
        System.out.println("Berhasil menyimpan uang sebesar " +
formatRupiah(jumlah));
    } else {
        System.out.println("Jumlah yang disimpan harus lebih dari 0");
    }
}

public void ambilUang(double jumlah) {
    if (jumlah > 0 && saldo >= jumlah) {
        saldo -= jumlah;
        System.out.println("Berhasil mengambil uang sebesar " +
formatRupiah(jumlah));
    } else if (jumlah <= 0) {
        System.out.println("Jumlah yang diambil harus lebih dari 0");
    } else {
        System.out.println("Saldo tidak mencukupi untuk pengambilan uang sebesar "
+ formatRupiah(jumlah));
    }
}

public double getSaldo() {
    return saldo;
}
```

c. Buat class BankBeraksi, tetapkan saldo awal lewat konstruktor Rp. 100000, jalankan 3 method di atas, dan tampilkan proses sebagai berikut : (bobot 15 %)

Selamat Datang di Bank ABC

Saldo saat ini: Rp. 100000

Simpan uang: Rp. 500000

Saldo saat ini: Rp. 600000

Ambil uang: Rp. 150000

Saldo saat ini: Rp. 450000

```
public class BankBeraksi {
    public static void main(String[] args) {
        // Membuat objek Bank dengan saldo awal Rp. 100.000
        Bank bankSaya = new Bank(100000.0);

        // Menampilkan pesan selamat datang
        System.out.println("Selamat Datang di Bank ABC");

        // Menampilkan saldo awal
        System.out.println("Saldo saat ini : " +
            bankSaya.formatRupiah(bankSaya.getSaldo()));

        // Simpan uang Rp. 500.000
        double jumlahSimpan = 500000.0;
        bankSaya.simpanUang(jumlahSimpan);
        System.out.println("Simpan uang : " +
            bankSaya.formatRupiah(jumlahSimpan));

        // Menampilkan saldo setelah penyimpanan
        System.out.println("Saldo saat ini : " +
            bankSaya.formatRupiah(bankSaya.getSaldo()));

        // Ambil uang Rp. 150.000
        double jumlahAmbil = 150000.0;
        bankSaya.ambilUang(jumlahAmbil);
        System.out.println("Ambil uang : " + bankSaya.formatRupiah(jumlahAmbil));

        // Menampilkan saldo setelah penarikan
        System.out.println("Saldo saat ini : " +
            bankSaya.formatRupiah(bankSaya.getSaldo()));
    }
}
```

```
Selamat Datang di Bank ABC
Saldo saat ini : Rp100.000,00
Berhasil menyimpan uang sebesar Rp500.000,00
Simpan uang : Rp500.000,00
Saldo saat ini : Rp600.000,00
Berhasil mengambil uang sebesar Rp150.000,00
Ambil uang : Rp150.000,00
Saldo saat ini : Rp450.000,00
```

2. Buatlah program dalam bahasa Java dimana anda diminta untuk mengimplementasikan kasus operasi terhadap suatu Titik dalam konsep PBO. Buatlah kelas Titik dengan ketentuan sebagai berikut :

a. Memiliki atribut x dan y (bobot 5 %)

```
public class Titik {
    private int x; // Koordinat x
    private int y; // Koordinat y
```

b. Memiliki default konstruktor (bobot 10 %)

```
public Titik() {
}
```

c. Memiliki method inisialisasiTitik (int x, int y), untuk mengeset nilai Titik sesuai masukan dari user (bobot 10 %)

```
public Titik(int x, int y) {
    this.x = x;
    this.y = y;
}
```

d. Memiliki method tampilTitik(), untuk menampilkan nilai Titik (bobot 5 %)

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

e. Memiliki method perkalianSkalar(int skalar), untuk mengalikan nilai Titik dengan suatu nilai skalar sesuai masukan dari user (x dan y dikalikan dengan nilai skalar ) (bobot 10%)

```
public void perkalianSkalar(int skalar) {
    this.x *= skalar;
    this.y *= skalar;
}
```

f. Memiliki metod pencerminanSumbuX(), untuk melakukan pencerminan Titik terhadap sumbu X (nilai y menjadi negatif) (bobot 10%)

```
public void pencerminanSumbuX() {
    this.y = -this.y;
}
```

g. Memiliki metod pencerminanSumbuY(), untuk melakukan pencerminan Titik terhadap sumbu Y (nilai x menjadi negatif) (bobot 10%)

```
public void pencerminanSumbuY() {
    this.x = -this.x;
}
```

h. Memiliki method `int jarak(Titik t2)`, untuk mencari jarak antara 2 buah titik, dengan rumus :  $\text{jarak} = (-) + (-)$  (bobot 10 %)

```
public double jarak(Titik t2) {
    int deltaX = t2.x - this.x;
    int deltaY = t2.y - this.y;
    return Math.sqrt(deltaX * deltaX + deltaY * deltaY);
}
```

i. Pada Menu Utama, buatlah menu untuk mengimplementasikan fungsi-fungsi dari kelas Titik diatas.

(bobot 30 %)

MENU OPERASI TITIK

1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIHAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR

MASUKAN PILIHAN :

Pada Menu JARAK ANTARA DUA TITIK, masukan dahulu koordinat Titik kedua

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    Titik titik = new Titik();
    boolean selesai = false;

    while (!selesai) {
        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("0. KELUAR");
        System.out.print("MASUKAN PILIHAN: ");

        int pilihan = input.nextInt();

        switch (pilihan) {
            case 1:
                System.out.print("Masukkan nilai x: ");
                int x = input.nextInt();
                System.out.print("Masukkan nilai y: ");
                int y = input.nextInt();
                titik.inisialisasiTitik(x, y); // Inisialisasi Titik
                System.out.println("Titik berhasil diinisialisasi.");
            // ... other cases ...
        }
    }
}
```

```

        break;
    case 2:
        titik.tampilTitik();
        break;
    case 3:
        System.out.print("Masukkan skalar: ");
        int skalar = input.nextInt();
        titik.perkalianSkalar(skalar);
        System.out.println("Titik berhasil dikalikan dengan skalar.");
        break;
    case 4:
        titik.pencerminanSumbuX();
        System.out.println("Titik berhasil dicerminkan terhadap sumbu X.");
        break;
    case 5:
        titik.pencerminanSumbuY();
        System.out.println("Titik berhasil dicerminkan terhadap sumbu Y.");
        break;
    case 6:
        Titik titik2 = new Titik();
        System.out.print("Masukkan koordinat Titik kedua (x): ");
        int x2 = input.nextInt();
        System.out.print("Masukkan koordinat Titik kedua (y): ");
        int y2 = input.nextInt();
        titik2.inisialisasiTitik(x2, y2); // Inisialisasi Titik kedua
        double jarak = titik.jarak(titik2);
        System.out.println("Jarak antara dua Titik: " + jarak);
        break;
    case 0:
        selesai = true;
        System.out.println("Terima kasih!");
        break;
    default:
        System.out.println("Pilihan tidak valid. Silakan coba lagi.");
    }
}

input.close();
}
}

```

## HASIL TITIK :

```
import java.util.Scanner;

public class Titik {
    private int x; // Koordinat x
    private int y; // Koordinat y

    // Konstruktor default
    public Titik() {
        // Tidak melakukan apa-apa, nilai x dan y tetap 0
    }

    // Konstruktor dengan parameter
    public Titik(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Method untuk mengeset nilai Titik sesuai masukan dari user
    public void inisialisasiTitik(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Method untuk menampilkan nilai Titik
    public void tampilTitik() {
        System.out.println("Koordinat Titik: (" + x + ", " + y + ")");
    }

    // Method untuk mengalikan nilai Titik dengan suatu nilai skalar
    public void perkalianSkalar(int skalar) {
        this.x *= skalar;
        this.y *= skalar;
    }

    // Method untuk melakukan pencerminan Titik terhadap sumbu X (nilai y menjadi negatif)
    public void pencerminanSumbuX() {
        this.y = -this.y;
    }

    // Method untuk melakukan pencerminan Titik terhadap sumbu Y (nilai x menjadi negatif)
    public void pencerminanSumbuY() {
        this.x = -this.x;
    }
}
```

```

// Method untuk mencari jarak antara 2 buah titik
public double jarak(Titik t2) {
    int deltaX = t2.x - this.x;
    int deltaY = t2.y - this.y;
    return Math.sqrt(deltaX * deltaX + deltaY * deltaY);
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    Titik titik = new Titik();
    boolean selesai = false;

    while (!selesai) {
        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("0. KELUAR");
        System.out.print("MASUKAN PILIHAN: ");

        int pilihan = input.nextInt();

        switch (pilihan) {
            case 1:
                System.out.print("Masukkan nilai x: ");
                int x = input.nextInt();
                System.out.print("Masukkan nilai y: ");
                int y = input.nextInt();
                titik.inisialisasiTitik(x, y); // Inisialisasi Titik
                System.out.println("Titik berhasil diinisialisasi.");
                break;
            case 2:
                titik.tampilTitik();
                break;
            case 3:
                System.out.print("Masukkan skalar: ");
                int skalar = input.nextInt();
                titik.perkalianSkalar(skalar);
                System.out.println("Titik berhasil dikalikan dengan skalar.");
                break;
            case 4:
                titik.pencerminkanSumbuX();
                System.out.println("Titik berhasil dicerminkan terhadap sumbu X.");
                break;
        }
    }
}

```

```

        case 5:
            titik.pencerminkanSumbuY();
            System.out.println("Titik berhasil dicerminkan terhadap sumbu Y.");
            break;
        case 6:
            Titik titik2 = new Titik();
            System.out.print("Masukkan koordinat Titik kedua (x): ");
            int x2 = input.nextInt();
            System.out.print("Masukkan koordinat Titik kedua (y): ");
            int y2 = input.nextInt();
            titik2.inisialisasiTitik(x2, y2); // Inisialisasi Titik kedua
            double jarak = titik.jarak(titik2);
            System.out.println("Jarak antara dua Titik: " + jarak);
            break;
        case 0:
            selesai = true;
            System.out.println("Terima kasih!");
            break;
        default:
            System.out.println("Pilihan tidak valid. Silakan coba lagi.");
    }
}

input.close();
}
}

```

```

MENU OPERASI TITIK
1. INISIALISASI TITIK
2. TAMPIL TITIK
3. PERKALIHAN SKALAR TITIK
4. PENCERMINAN TERHADAP SUMBU X
5. PENCERMINAN TERHADAP SUMBU Y
6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN:

```

```

6. JARAK ANTARA DUA TITIK
0. KELUAR
MASUKAN PILIHAN: 6
Masukkan koordinat Titik kedua (x): 7
Masukkan koordinat Titik kedua (y): 9
Jarak antara dua Titik: 11.40175425099138

```



## **LINK REPOSITORY GITHUB :**

<https://github.com/NathayaElang/Laprak-PBO-Modul2>