

LAPORAN RENCANA PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



NAMA : FERRY WAHYUDI
NIM : 223030503194
KELAS : DATA SCIENCE
MODUL : III (MEMULAI JAVA OOP DENGAN ECLIPSE)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2023

MODUL III

MEMULAI JAVA OOP DENGAN ECLIPSE

1.1. Tujuan

- 1.1.1.** Mahasiswa mampu membuat Program Java serta menerapkan konsep OOP menggunakan aplikasi Eclipse.
- 1.1.2.** Mahasiswa mampu membedakan *Class Driver* dan *Class Object*.

1.2. Landasan Teori

Java adalah bahasa pemrograman yang dibuat oleh James Gosling. Sintaks Java mirip seperti C/C++, karena Java banyak terinspirasi dari kedua bahasa ini. Java dirancang untuk tujuan umum (general-purpose) dan sepenuhnya menganut paradigma OOP (*Object Oriented Programming*). Java merupakan bahasa pemrograman yang digunakan secara luas untuk pengodean aplikasi web. Bahasa ini telah menjadi pilihan populer di antara developer selama lebih dari dua dekade, dengan jutaan aplikasi Java yang digunakan saat ini. Java merupakan bahasa multiplatform yang berorientasi pada objek dan berpusat pada jaringan yang dapat digunakan sebagai platform di dalamnya. Java merupakan bahasa pemrograman yang cepat, aman, dan andal untuk mengodekan segala sesuatu mulai dari aplikasi seluler dan perangkat lunak korporasi hingga aplikasi big data dan teknologi sisi server.

Eclipse adalah salah satu lingkungan pengembangan paling populer untuk Java. Program ini menyediakan segala hal yang Anda butuhkan untuk membuat proyek Java dari nol. Sebelum mulai mengerjakan proyek baru, Anda harus membuat proyek terlebih dahulu. Proses pembuatan proyek baru di Eclipse sebenarnya cukup mudah. Namun, jika Anda memasang Eclipse untuk membuat program dengan bahasa lain, Anda mungkin akan merasa bingung.

1.2.1. Membuat Project pada Eclipse

1. Pilih File → New → Java Project*.
2. Masukkan nama project (CamelCase) dan klik finish.



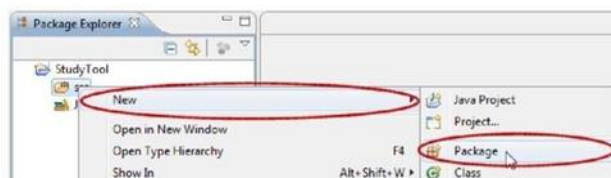
Gambar 1.1 Membuat Java Project

3. Dengan klik finish value project ter-set default.

1.2.2. Membuat Package

Package merupakan sebuah folder yang berisi sekumpulan program Java. Deklarasi package biasanya dilakukan saat membuat program atau aplikasi besar. Biasanya nama package mengikuti nama program yang sudah kita buat ataupun bisa kita buat sesuai dengan kebutuhan kita.

1. Select src folder pada package view.
2. Klik kanan src folder dan pilih New → Package.



Gambar 1.2 Membuat Package

3. Berikan nama package sama seperti nama project namun menggunakan gaya penulisan lowerCamelCase.

4. Klik finish.

Terdapat dua cara dalam menggunakan package, yaitu:

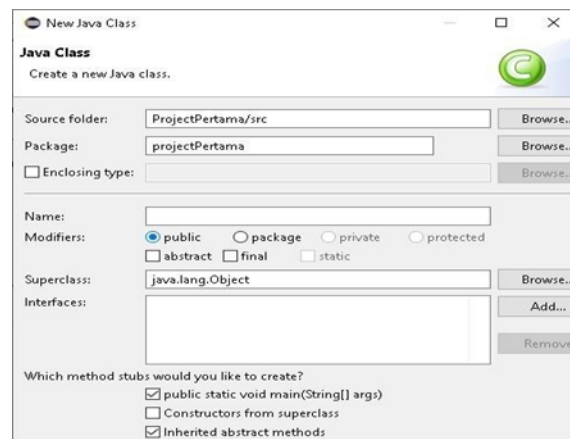
- a. Apabila class yang menggunakan berada dalam package yang sama dengan class yang digunakan, maka tidak perlu mengimpor package.
- b. Apabila class yang menggunakan berada dalam package yang berbeda dengan class yang digunakan, maka pada awal source code harus dicantumkan perintah :

```
import namaPackage.namaClass; //atau import
namaPackage.*;
```

1.2.3. Membuat Class

Java merupakan bahasa pemrograman yang menggunakan paradigma OOP (*Object Oriented Programming*). Setiap program harus dibungkus di dalam *class* agar nanti bisa dibuat menjadi objek.

1. Buat class pada folder src dengan cara klik kanan kemudian pilih New → Class.
2. Masukkan nama class kemudian pilih untuk membuat main method didalam class.



Gambar 1.3 Membuat Class

1.2.4. Acces Modifiers

Access Modifier adalah sebuah “hak akses” yang diberikan kepada sebuah variabel/method/class dengan tujuan untuk menjaga integritas dari data tersebut ketika ingin diakses object lain. Hak akses sendiri diberikan oleh pembuat program sebagai salah satu bentuk implementasi dari OOP itu sendiri. Dengan adanya Access Modifier, kita dapat membatasi resource-resource mana saja yang dapat diakses oleh object tertentu, turunannya, ataupun oleh method tertentu.

1. Public

Access modifier public mempunyai hak akses paling luas dibanding yang lainnya. Karena aksesnya sangat luas, maka access modifier ini biasanya digunakan untuk method setter getter sesuai konsep OOP.

2. Protected

Access modifier protected biasanya digunakan untuk mewariskan variabel yang ada di super class terhadap child class.

3. Private

Access modifier private bersifat tertutup. Sesuai dengan konsep OOP Encapsulation, maka setiap variabel wajib untuk dilindungi hak aksesnya secara langsung dari luar. Oleh karena itu, variabel diberikan hak akses private dan untuk melakukan pengaksesan/perubahan data digunakan setter getter.

4. Default

Default adalah hak akses yang tidak perlu dituliskan di method/variabelnya. Dengan hak akses ini, variabel/method dapat diakses dari class lain asalkan masih dalam satu package yang sama.

Ada 4 hak akses yang tersedia, yaitu default, public, protected, private. Berikut ini tabel rinciannya.

Tabel 1.1 Acces Modifers

Aksesabilitas	Public	Protected	Private	Default
Class	Ya	Ya	Ya	Ya
Package	Ya	Ya	Tidak	Ya
Subclass	Ya	Ya	Tidak	Tidak
Global	Ya	Tidak	Tidak	Tidak

1.2.5. Konstruktor

Konstruktor adalah method unik di java, karena metode ini membuat instance kelas(objek). Konstruktor selalu memiliki nama yang sama dengan kelas dan tidak mendeklarasikan tipe kembalian. Konstruktor dipanggil menggunakan kata kunci baru.

1. Konstruktor tanpa parameter

Jika Anda membuat konstruktor tanpa parameter (tanda kurung kosong), Anda dapat mengosongkan konten konstruktor (antara {dan}). Ini disebut konstruktor default, dan sama dengan konstruktor yang disediakan Java jika Anda tidak mendeklarasikannya.

2. Konstruktor dengan parameter

Jika Anda membuat konstruktor dengan parameter (tanda kurung TIDAK kosong), Anda juga akan menginisialisasi variabel antara {dan}. Konstruktor ini akan menginisialisasi variabel kelas dengan nilai yang dikirim dari kelas driver utama.

1.2.6. Main Method

Untuk menjalankan program Java, harus menentukan metode utama dalam Driver Class. Metode utama secara otomatis dipanggil saat kelas dipanggil. Ingatlah untuk memberi nama file tersebut sama dengan kelasnya. Main method merupakan blok program yang akan dieksekusi pertama kali. Ini adalah inti dari program. Main method wajib kita buat. Kalau tidak, maka programnya tidak akan bisa dieksekusi atau dijalankan.

1.2.7. Accessor dan Mutator Methods

Accessor and Mutator Methods merupakan hal yang umum untuk membuat sekumpulan metode yang memanipulasi dan mengambil nilai data kelas.

1. Accessors (getter): Metode yang mengembalikan (mendapatkan) nilai dari setiap variabel kelas.
2. Mutators (setter): Metode yang mengubah (set) nilai setiap variabel kelas.

```
public int getStudentId() {  
    return studentId;  
}  
  
public void setStudentId(int studentId)  
{ this.studentId = studentId; }
```

1.3. Praktikum

1.3.1. Students.java

```
package com.example.domain;  
  
public class Students {  
    private int studentId;  
    private String name;  
    private String ssn;  
    private double gpa;  
    public final int SCHCODE = 34958;  
    public Students() {  
  
    }  
  
    public Students(int studentId, String name, String ssn,  
double  
gpa){  
        this.studentId=studentId;  
        this.name=name;  
        this.ssn=ssn;  
        this.gpa=gpa;  
    }  
  
    public int getStudentId() {
```

```

return studentId;
    }

    public String getName() {
        return name;
    }

    public String getSsn() {
        return ssn;
    }

    public double getGpa() {
        return gpa;
    }

    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }

    public void setStudentName(String name) {
        this.name=name;
    }

    public void setSSN(String ssn) {
        this.ssn=ssn;
    }

    public void setGPA(double gpa) {
        this.gpa=gpa;
    }

    public String toString()
    {
        String s1 = "";
        s1 = "Student Id: " + getStudentId() +
            "Student Name: " +
            getName() +"Student SSN: "
            + getSsn() +
            "Student GPA: " + getGpa();
        return s1;
    }
}

```


1.3.2. StudentTester.java

```
package com.example.domain;

public class StudentTester {

    public static void main(String[] args) {
        Students s1 = new Students(123, "Mary Smith",
        "999-99-9999", 3.4);
        System.out.println(s1);
        Students s2 = new Students();
        s2.setStudentId(124);
        s2.setStudentName("John
        Jacoby");
        s2.setSSN("123-45-
        6789");s2.setGPA(4.0);
        System.out.println(s2);
    }
}
```

1.4. Tugas

Membuat proyek inventori:

1. Untuk bagian pertama proyek, Anda diminta untuk memikirkan tentang apa yang akan disimpan oleh sistem inventaris Anda.
 - a. Pikirkan produk tertentu yang dapat disimpan dalam inventaris (misalnya, produk di rumah, sekolah, atau tempat kerja Anda. Tulis daftar setidaknya 6 produk yang ingin Anda simpan di sistem Anda, proyek ini dapat digunakan untuk menyimpan berbagai macam produk.
 - b. Untuk setiap produk yang Anda identifikasi, buat dalam tabel seperti berikut ini:

Atribut	Sample Data
Nama produk (nilai yang akan mengidentifikasi produk di sistem Anda).	Rayzen 5 5600X
Harga (nilai ini memegang harga setiap barang yang akan dijual).	5000000
Jumlah unit yang tersedia (nilai ini akan menyimpan berapa banyak dari setiap item produk yang saat ini tersedia).	5

Nomor item (digunakan untuk mengidentifikasi produk secara unik di sistem Anda).	1
--	---

Tabel ini memberi Anda pemahaman tentang jenis data yang ingin Anda simpan untuk atribut setiap produk. Hal ini berguna sehingga Anda memiliki pemahaman yang jelas tentang data yang akan Anda tangani!

- Langkah selanjutnya adalah memikirkan tipe data yang akan Anda gunakan untuk menyimpan nilai di sistem Anda.

Atribut	Sample Data	Tipe Data
Nama produk	Rayzen 5 5600X	String
Harga	5000000	double
Jumlah	5	int
Nomor Item	1	int

- Buka Eclipse dan buat proyek bernama inventori
- Buat kelas objek bernama Product
- Tambahkan variabel dengan menggunakan tipe data yang sudah Anda identifikasi di tugas 2:
 - nomor barang
 - nama produk
 - jumlah unit yang tersedia
 - harga setiap unit
- Buat 2 Konstruktor
 - Sebuah konstruktor default tanpa parameter yang memungkinkan compiler untuk menginisialisasi field ke nilai defaultnya. Tambahkan komentar di atas konstruktor Anda yang menjelaskan tujuan kode.
 - Konstruktor dengan parameter untuk keempat variabel kelas sehingga dapat diinisialisasi dengan nilai dari kelas driver.

7. Buat metode getter / accessor dan setter / mutator untuk masing-masing dari empat variabel instance. Tambahkan komentar di atas mereka untuk menjelaskan tujuan mereka.
8. Buat sebuah metode untuk menampilkan nilai tiap variabel.

Format :

Nomor : 1
Nama : Ryzen 5 5600x
Stok : 5
Harga : 5000000

9. Buat kelas utama Java yang disebut ProductTester.
10. Membuat dan menginisialisasi enam objek Produk berdasarkan daftar yang Anda buat di tugas 1.
 - a. Dua dari Produk harus dibuat menggunakan konstruktor default.
 - b. Empat lainnya harus dibuat dengan memberikan nilai untuk argumen yang cocok dengan parameter konstruktor.
11. Menggunakan kelas ProductTester, tampilkan detail setiap produk ke konsol.
12. Simpan proyek Anda.