

Nama : Muhammad Izzuddin Almansyur

NPM : 51422084

Kelas : 4IA07

Praktikum : Rekayasa Perangkat Lunak 2, ACT2

---

1. Jelaskan anatomi dasar dari sebuah class dalam Java. Apa saja komponen utama yang ada dalam class dan bagaimana fungsinya? Berikan contoh sederhana dari sebuah class dengan atribut dan method di dalamnya.

Dalam Java, sebuah *class* merupakan cetak biru atau blueprint yang digunakan untuk membuat objek. Class menyatukan data dan perilaku dalam satu unit, dan terdiri dari beberapa komponen utama. Komponen pertama adalah **atribut** atau **field**, yaitu variabel yang menyimpan data atau keadaan (state) dari objek. Komponen kedua adalah **method**, yaitu fungsi yang menentukan perilaku atau aksi yang dapat dilakukan oleh objek tersebut. Komponen ketiga adalah **constructor**, yaitu method khusus yang digunakan untuk menginisialisasi objek saat pertama kali dibuat. Selain itu, class juga dapat memiliki **access modifiers** seperti `public`, `private`, atau `protected` untuk mengatur tingkat akses terhadap atribut dan method, serta keyword seperti `static` atau `final` untuk menentukan sifat khusus dari komponen tersebut. Dengan memanfaatkan struktur ini, Java memungkinkan pengembangan perangkat lunak yang modular, terstruktur, dan mudah dipelihara.

Contoh:

```
public class Mahasiswa {  
    // Atribut  
    private String nama;  
    private int umur;  
  
    // Constructor  
    public Mahasiswa(String nama, int umur) {  
        this.nama = nama;  
        this.umur = umur;  
    }  
  
    // Method  
    public void tampilkanInfo() {  
        System.out.println("Nama: " + nama);  
        System.out.println("Umur: " + umur + " tahun");  
    }  
}
```

2. Screenshot kode, output, dan beri penjelasan singkat dari program yang telah dibuat pada video

## - User

```
1 1 /*
2 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 4  */
5 5 package com.mycompany.rpl2_pert2;
6 6
7 7 /**
8 8  *
9 9  * @author Izza
10 10 */
11 11 public class User {
12 12     String nama;
13 13     String email;
14 14     String npm;
15 15
16 16     public void login(){
17 17         System.out.println("User melakukan Login");
18 18     }
19 19
20 20     public void logout(){
21 21         System.out.println("User melakukan Logout");
22 22     }
23 23 }
24 24
```

## - Seller

```
1 1 /*
2 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 4  */
5 5 package com.mycompany.rpl2_pert2;
6 6
7 7 /**
8 8  *
9 9  * @author Izza
10 10 */
11 11 public class Seller extends User{
12 12     private String product;
13 13
14 14     public void addProduct(String productName){
15 15         System.out.println("Seller menggunakan produk, " + productName);
16 16         this.product = productName;
17 17     }
18 18 }
19 19
```

## - Buyer

```
1 1 /*
2 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 4  */
5 5 package com.mycompany.rpl2_pert2;
6 6
7 7 /**
8 8  *
9 9  * @author Izza
10 10 */
11 11 public class Buyer extends User {
12 12     @Override
13 13     public void login() {
14 14         System.out.println("Buyer melakukan login");
15 15     }
16 16
17 17     @Override
18 18     public void logout() {
19 19         System.out.println("Buyer melakukan logout");
20 20     }
21 21 }
```

## - Admin

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.rpl2_pert2;
6
7  /**
8   *
9   * @author Izza
10  */
11  public class Admin extends User {
12      public void login() {
13          System.out.println("Admin melakukan Login");
14      }
15
16      public void logout() {
17          System.out.println("Admin melakukan Logout");
18      }
19
20      public void manageUser() {
21          System.out.println("Admin mengelola user!");
22      }
23  }
```

## - Rpl2\_pert2

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  */
4
5  package com.mycompany.rpl2_pert2;
6
7  /**
8   *
9   * @author Izza
10  */
11  public class Rpl2_pert2 {
12
13      public static void main(String[] args) {
14          User user = new User();
15          user.login();
16          user.logout();
17
18          Seller seller = new Seller();
19          seller.addProduct("Nasi Goreng");
20
21          Buyer buyer = new Buyer();
22          buyer.login();
23          buyer.logout();
24
25          Admin admin = new Admin();
26          admin.login();
27          admin.manageUser();
28          admin.logout();
29      }
30  }
```

## - Output

```
Output - Run (rpl2_pert2)
--- resources:3.3.1:resources (default-resources) @ rpl2_pert2 ---
skip non existing resourceDirectory D:\Personal\Akademik\University\Gunadarma Informatik S1\Semester 7\Rekayasa Perangkat Lunak 2\Praktikum\Pertemuan2_rpl2\rpl2_pert2\src\main\resources
--- compiler:3.13.0:compile (default-compile) @ rpl2_pert2 ---
Recompiling the module because of changed source code.
Compiling 5 source files with javac [debug release 24] to target\classes
--- exec:3.1.0:exec (default-cli) @ rpl2_pert2 ---
User melakukan Login
User melakukan Logout
Seller menggunakan produk, Nasi Goreng
Buyer melakukan login
Buyer melakukan logout
Admin melakukan Login
Admin mengelola user!
Admin melakukan Logout

BUILD SUCCESS
-----
Total time: 1.053 s
Finished at: 2025-10-25T12:59:57+07:00
-----
```

## - Penjelasan Kode

Kode ini merupakan implementasi prinsip dasar pemrograman berorientasi objek (OOP) dalam Java, khususnya konsep inheritance dan polymorphism.

Class User berfungsi sebagai superclass yang memiliki tiga atribut: nama, email, dan npm. Dua method login() dan logout() digunakan untuk menampilkan pesan saat pengguna melakukan proses masuk dan keluar dari sistem. Class ini menjadi dasar bagi class turunan lainnya.

Class Seller merupakan subclass dari User. Selain mewarisi atribut dan method dari User, class ini menambahkan method addProduct(String productName) yang berfungsi untuk menampilkan pesan saat penjual menambahkan produk. Atribut product disimpan secara privat untuk menjaga enkapsulasi.

Class Buyer juga merupakan subclass dari User, namun melakukan overriding terhadap method login() dan logout(). Tujuannya adalah untuk menyesuaikan pesan yang ditampilkan agar lebih relevan dengan peran pembeli.

Class Admin adalah subclass ketiga dari User. Sama seperti Buyer, class ini melakukan overriding terhadap method login() dan logout(). Selain itu, class ini menambahkan method manageUser() yang berfungsi untuk menampilkan pesan bahwa admin memiliki kemampuan untuk mengelola pengguna.

Class Rpl2\_pert2 berisi method main() sebagai titik awal eksekusi program. Di dalamnya dibuat objek dari masing-masing class (User, Seller, Buyer, dan Admin) dan dipanggil method-method yang relevan. Pemanggilan ini menunjukkan bagaimana objek yang berasal dari superclass yang sama (User) dapat memiliki perilaku yang berbeda tergantung pada subclass-nya. Ini merupakan contoh penerapan polymorphism, di mana method yang sama (login() dan logout()) menghasilkan output yang berbeda tergantung pada jenis objek yang memanggilnya.