

ACTIVITY PERTEMUAN 2

NAMA : Arya Wahyu Wijaya
NPM : 50421223
KELAS : 4IA14
MATERI : ANATOMY CLASS
MATA PRAKTIKUM : REKAYASA PERANGKAT LUNAK2

SOAL :

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

2. Screenshot Code,Output dan penjelasan!

JAWABAN :

1.

Dalam bahasa pemrograman Java, class adalah cetak biru atau template yang digunakan untuk membuat objek. Class mendefinisikan atribut (variabel) dan method (fungsi) yang dimiliki oleh objek-objek dari class tersebut. Berikut adalah anatomi dasar dari sebuah class dalam Java:

Komponen Utama dalam Sebuah Class:

1. Deklarasi Class: Bagian ini mendefinisikan nama class, kata kunci akses (seperti public), dan jika diperlukan, hubungan dengan class lain (seperti inheritance).
 - o Contoh: `public class NamaClass { ... }`
2. Atribut/Field: Atribut adalah variabel yang menyimpan data atau properti dari sebuah objek. Atribut dapat memiliki berbagai tipe data (misalnya, int, String, dll.), dan biasanya menggunakan modifier akses (seperti private) untuk kontrol aksesnya.
 - o Contoh: `private String nama;`
3. Constructor: Constructor adalah method khusus yang digunakan untuk menginisialisasi objek saat dibuat. Nama constructor harus sama dengan nama class, dan constructor tidak memiliki tipe pengembalian.
 - o Contoh: `public NamaClass(String nama) { this.nama = nama; }`
4. Method: Method adalah fungsi yang mendefinisikan perilaku dari objek. Method dapat mengambil argumen, memproses data, dan mengembalikan nilai (meskipun tidak harus).
 - o Contoh: `public void cetakNama() { System.out.println(nama); }`
5. Modifier Akses: Modifier akses (public, private, protected) menentukan visibilitas atribut atau method dari luar class. Modifier ini penting untuk enkapsulasi, yaitu membatasi akses langsung ke atribut dan method dari luar class.

6. Keyword this: this digunakan dalam sebuah class untuk merujuk pada objek saat ini. Ini sering digunakan untuk membedakan antara atribut class dan parameter dalam method atau constructor jika keduanya memiliki nama yang sama.

2.

CODE:

- Mahasiswa.java

```
Help Mahasiswa.java - pert2_50421223 - Visual Studio Code [Administrator]
Mahasiswa.java 1 x MahasiswaSarjana.java 1 pert2_50421223.java 1
Mahasiswa.java > Mahasiswa
1 package pert2_50421223;
2
3 public class Mahasiswa {
4
5     private String nama;
6     private String npm;
7     private int umur;
8
9     public Mahasiswa(String nama, String npm, int umur) {
10         this.nama = nama;
11         this.npm = npm;
12         this.umur = umur;
13     }
14
15     public void tampilkanData() {
16         System.out.println("Nama: " + nama);
17         System.out.println("NPM: " + npm);
18         System.out.println("Umur: " + umur);
19     }
20 }
21
```

- pert2_50421223.java

```
Help pert2_50421223.java - pert2_50421223 - Visual Studio Code [Administrator]
J Mahasiswa.java 1 J MahasiswaSarjana.java 1 J pert2_50421223.java 1 X

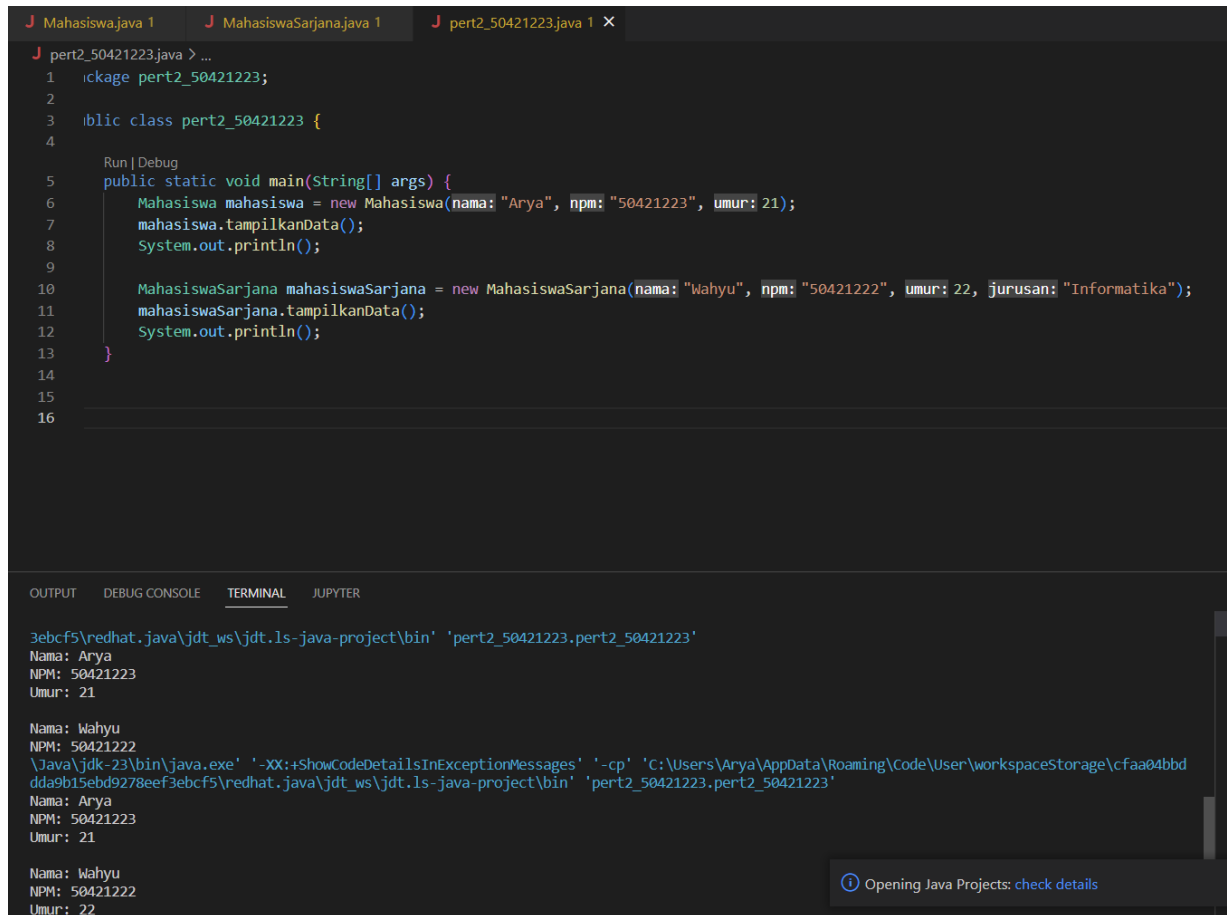
J pert2_50421223.java > ...
1 package pert2_50421223;
2
3 public class pert2_50421223 {
4
5     Run | Debug
6     public static void main(String[] args) {
7         Mahasiswa mahasiswa = new Mahasiswa(nama: "Arya", npm: "50421223", umur: 21);
8         mahasiswa.tampilkanData();
9         System.out.println();
10
11         MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana(nama: "Wahyu", npm: "50421222", umur: 22, jurusan: "Informatika");
12         mahasiswaSarjana.tampilkanData();
13         System.out.println();
14     }
15 }
16
```

- MahasiswaSarjana.java

```
Help MahasiswaSarjana.java - pert2_50421223 - Visual Studio Code [Administrator]
J Mahasiswa.java 1 J MahasiswaSarjana.java 1 X J pert2_50421223.java 1 ●

J MahasiswaSarjana.java > ...
1 package pert2_50421223;
2
3 public class MahasiswaSarjana extends Mahasiswa {
4
5     private String jurusan;
6
7     public MahasiswaSarjana(String nama, String npm, int umur, String jurusan) {
8         super(nama, npm, umur);
9         this.jurusan = jurusan;
10    }
11
12    public void tampilkanData() {
13        super.tampilkanData();
14        System.out.println("Jurusan: " + jurusan);
15    }
16 }
17
```

OUTPUT:

The screenshot shows an IDE with three tabs: 'Mahasiswa.java 1', 'MahasiswaSarjana.java 1', and 'pert2_50421223.java 1'. The active tab is 'pert2_50421223.java 1', which contains the following Java code:

```
1 package pert2_50421223;
2
3 public class pert2_50421223 {
4
5     public static void main(String[] args) {
6         Mahasiswa mahasiswa = new Mahasiswa(nama: "Arya", npm: "50421223", umur: 21);
7         mahasiswa.tampilkanData();
8         System.out.println();
9
10        MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana(nama: "Wahyu", npm: "50421222", umur: 22, jurusan: "Informatika");
11        mahasiswaSarjana.tampilkanData();
12        System.out.println();
13    }
14
15
16 }
```

The bottom panel shows the 'TERMINAL' output:

```
3ebcf5\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'pert2_50421223.pert2_50421223'
Nama: Arya
NPM: 50421223
Umur: 21

Nama: Wahyu
NPM: 50421222
\Java\jdk-23\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Arya\AppData\Roaming\Code\User\workspaceStorage\cfaa04bbd
dda9b15ebd9278eef3ebcf5\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'pert2_50421223.pert2_50421223'
Nama: Arya
NPM: 50421223
Umur: 21

Nama: Wahyu
NPM: 50421222
Umur: 22
```

A notification bubble at the bottom right says 'Opening Java Projects: check details'.

PENJELASAN

Berikut penjelasan yang lebih mendalam untuk masing-masing file:

1. File pert2_50421223.java:

- File ini berisi kelas utama bernama pert2_50421223, yang memiliki metode main sebagai titik awal program.
- Di dalam metode main, dua objek dibuat untuk mendemonstrasikan penggunaan konsep pewarisan:
 - Objek pertama adalah mahasiswa, yang merupakan instance dari kelas Mahasiswa. Objek ini merepresentasikan seorang mahasiswa dengan atribut seperti nama, NPM (Nomor Pokok Mahasiswa), dan umur. Setelah objek dibuat, program memanggil metode tampilkanData() untuk menampilkan informasi mahasiswa.
 - Objek kedua adalah mahasiswaSarjana, yang merupakan instance dari kelas MahasiswaSarjana, turunan dari kelas Mahasiswa. Objek ini mewakili mahasiswa yang memiliki atribut tambahan berupa jurusan. Program juga memanggil metode tampilkanData() untuk menampilkan informasi mahasiswa sarjana, termasuk jurusannya.
- Program ini menampilkan data dari kedua objek tersebut, dengan pemisahan data

menggunakan baris kosong (`System.out.println();`), sehingga output terlihat rapi.

2. File MahasiswaSarjana.java:

- File ini berisi kelas MahasiswaSarjana yang merupakan subclass dari kelas Mahasiswa. Konsep pewarisan (inheritance) digunakan di sini, di mana kelas MahasiswaSarjana mewarisi atribut dan metode dari kelas Mahasiswa.
- Kelas ini memperkenalkan atribut baru, yaitu jurusan, yang menyimpan informasi tentang program studi yang diambil oleh mahasiswa sarjana.
- Konstruktor dari kelas ini memanggil konstruktor kelas induknya (Mahasiswa) menggunakan `super()`, untuk menginisialisasi atribut nama, NPM, dan umur. Selain itu, atribut jurusan diinisialisasi secara langsung.
- Metode `tampilkanData()` di-overriding dari kelas induk untuk menambahkan informasi tentang jurusan mahasiswa. Metode ini pertama kali memanggil metode `tampilkanData()` dari kelas induk menggunakan `super.tampilkanData()`, kemudian menambahkan informasi jurusan di akhir.

3. File Mahasiswa.java:

- Kelas ini adalah kelas dasar yang mendefinisikan atribut dan perilaku umum dari seorang mahasiswa.
- Atribut kelas ini meliputi:
 - nama: untuk menyimpan nama mahasiswa,
 - npm: untuk menyimpan NPM mahasiswa,
 - umur: untuk menyimpan umur mahasiswa.
- Konstruktor kelas ini digunakan untuk menginisialisasi atribut-atribut tersebut ketika objek dari kelas ini dibuat.
- Metode `tampilkanData()` digunakan untuk menampilkan atribut-atribut dari objek yang dibuat. Saat metode ini dipanggil, nama, NPM, dan umur mahasiswa akan dicetak ke layar.