

Activity Pertemuan 2

Nama : Simon Praja H P Pasaribu

Kelas : 4IA14

NPM : 51421422

1. Anatomi dasar dari sebuah class dalam Java terdiri dari beberapa komponen utama, yaitu:
 1. Modifier Akses: Menentukan visibilitas dari class. Contohnya adalah public, private, dan protected.
 2. Kata Kunci class: Digunakan untuk mendeklarasikan sebuah class.
 3. Nama Class: Nama unik untuk class yang harus mengikuti konvensi penamaan Java (biasanya menggunakan huruf besar di awal setiap kata).
 4. Atribut (Fields): Variabel yang menyimpan data atau status dari objek yang dibuat berdasarkan class tersebut.
 5. Method: Fungsi atau prosedur yang mendefinisikan perilaku dari objek. Method dapat mengambil parameter dan mengembalikan nilai.
 6. Konstruktor: Sebuah method khusus yang dipanggil saat objek dari class dibuat. Konstruktor biasanya digunakan untuk menginisialisasi atribut objek.

Penjelasan Komponen

- Modifier Akses (public): Menunjukkan bahwa class Mobil dapat diakses dari mana saja.
- Atribut (merk, warna, tahun): Menyimpan informasi tentang mobil.
- Konstruktor: Digunakan untuk menginisialisasi objek Mobil dengan merk, warna, dan tahun.
- Method tampilkanInfo(): Menampilkan informasi tentang mobil ke konsol.
- Method gantiWarna(String warnaBaru): Mengubah atribut warna menjadi warna baru yang diberikan sebagai parameter.

Contoh Sederhana dari Sebuah Class

Berikut adalah contoh sederhana dari sebuah class bernama Mobil:


```
public class Mobil {  
    // Atribut  
    private String merk;  
    private String warna;  
    private int tahun;  
  
    // Konstruktor  
    public Mobil(String merk, String warna, int tahun) {  
        this.merk = merk;  
        this.warna = warna;  
        this.tahun = tahun;  
    }  
}
```

```
// Method untuk menampilkan informasi mobil
public void tampilkanInfo() {
    System.out.println("Merk: " + merk);
    System.out.println("Warna: " + warna);
    System.out.println("Tahun: " + tahun);
}

// Method untuk mengganti warna mobil
public void gantiWarna(String warnaBaru) {
    this.warna = warnaBaru;
}
}
```

2. Screenshot Code, Output, dan Penjelasan

1. `Pert2_51421422.java`



```
J pert2_51421422.java 1 x J Mahasiswajava 1 J MahasiswaSarjana.java 1
J pert2_51421422.java > pert2_51421422
1 package pert2_51421422;
2
3 public class pert2_51421422 {
4
5     Run | Debug
6     public static void main(String[] args) {
7         Mahasiswa mahasiswa = new Mahasiswa(nama:"Simon", npm:"51421422", umur:24);
8         mahasiswa.tampilkanData();
9         System.out.println();
10
11         MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana(nama:"Praja", npm:"51421421", umur:22, jurusan:"Informatika");
12         mahasiswaSarjana.tampilkanData();
13         System.out.println();
14     }
15 }
```

- **Package Declaration:**

`package pert2_51421422;`

Menyatakan bahwa kelas ini berada dalam package bernama `pert2_51421422`. Ini berguna untuk mengorganisir kode dan menghindari konflik nama dengan kelas lain.

- **Kelas `pert2_51421422`:**

`public class pert2_51421422 {`

Ini adalah kelas utama dari program yang berisi method `main`, titik awal eksekusi program.

- **Method `main`:**

`public static void main(String[] args) {`

Ini adalah method statis yang dijalankan saat program dimulai. Dalam method ini, objek dari kelas `Mahasiswa` dan `MahasiswaSarjana` dibuat.

- **Objek `Mahasiswa`:**

`Mahasiswa mahasiswa = new Mahasiswa("Simon", "51421422", 24);`
`mahasiswa.tampilkanData();`

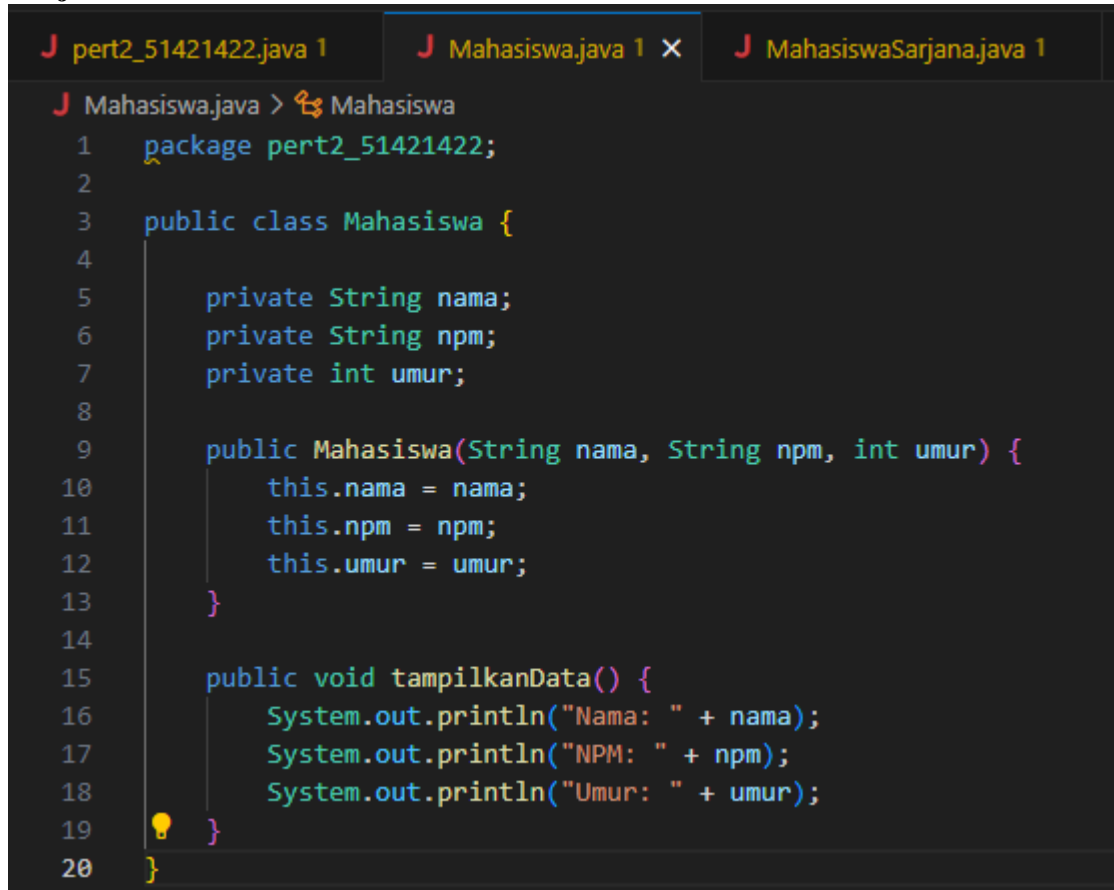
Di sini, sebuah objek `mahasiswa` dari kelas `Mahasiswa` diinisialisasi dengan nama, nomor induk mahasiswa, dan umur. Kemudian, method `tampilkanData()` dipanggil untuk menampilkan informasi mahasiswa.

- **Objek MahasiswaSarjana:**

```
MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana("Praja",
"51421421", 22, "Informatika");
mahasiswaSarjana.tampilkanData();
```

Selanjutnya, sebuah objek *mahasiswaSarjana* dari kelas *MahasiswaSarjana* diinisialisasi dengan nama, nomor induk mahasiswa, umur, dan jurusan. Method *tampilkanData()* juga dipanggil untuk menampilkan informasi mahasiswa sarjana.

2. Penjelasan Kelas Mahasiswa



```
J Mahasiswa.java > Mahasiswa
1 package pert2_51421422;
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 }
```

- **Deklarasi Kelas:**

```
public class Mahasiswa {
```

Kelas ini bernama *Mahasiswa* dan dideklarasikan dengan modifier *public*, sehingga dapat diakses dari kelas lain.

- **Atribut Kelas:**

```
private String nama;
```

```
private String npm;
```

```
private int umur;
```

Terdapat tiga atribut:

- *nama*: Menyimpan nama mahasiswa (tipe data *String*).
- *npm*: Menyimpan nomor induk mahasiswa (NPM) (tipe data *String*).
- *umur*: Menyimpan umur mahasiswa (tipe data *int*). Atribut-atribut ini dideklarasikan sebagai *private*, yang berarti hanya dapat diakses dari dalam kelas *Mahasiswa* itu sendiri.

- **Konstruktor:**

```
public Mahasiswa(String nama, String npm, int umur) {
    this.nama = nama;
    this.npm = npm;
    this.umur = umur;
}
```

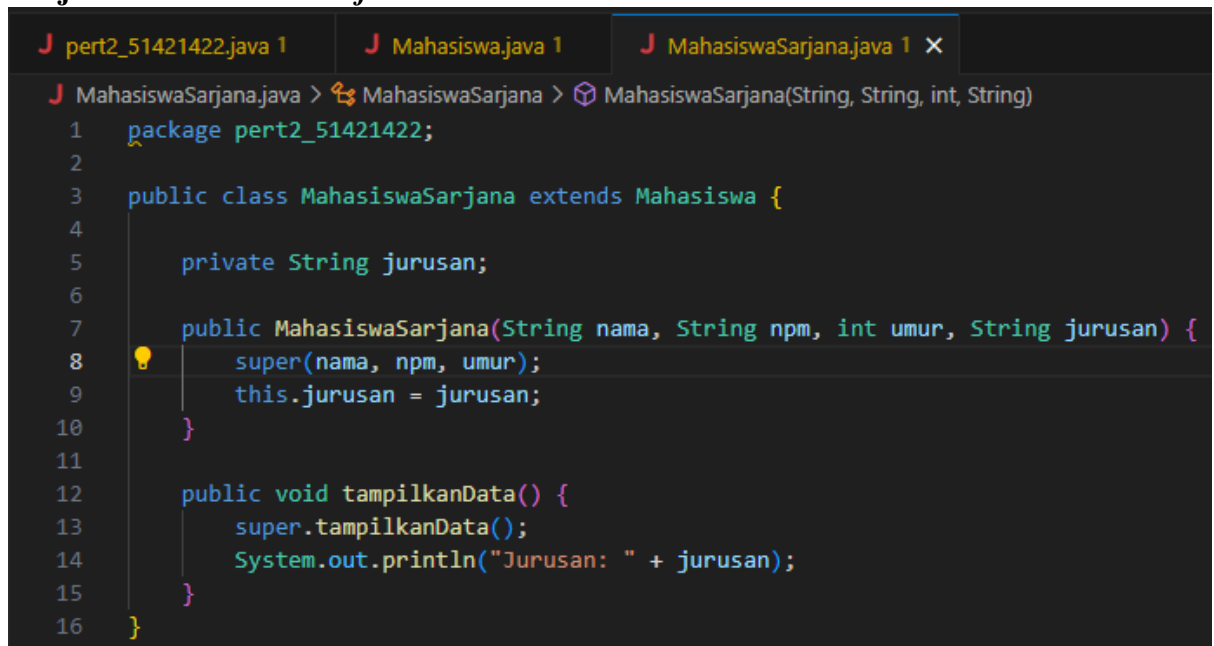
Konstruktor ini digunakan untuk menginisialisasi objek *Mahasiswa*. Parameter yang diterima (*nama*, *npm*, dan *umur*) digunakan untuk mengisi nilai atribut kelas. Kata kunci *this* digunakan untuk membedakan antara parameter dan atribut kelas yang memiliki nama yang sama.

- **Method *tampilkanData()*:**

```
public void tampilkanData() {
    System.out.println("Nama: " + nama);
    System.out.println("NPM: " + npm);
    System.out.println("Umur: " + umur);
}
```

Method ini digunakan untuk menampilkan informasi mengenai objek *Mahasiswa*. Ketika method ini dipanggil, program akan mencetak nama, NPM, dan umur mahasiswa ke konsol.

3. Penjelasan *MahasiswaSarjana*:



```
J pert2_51421422.java 1    J Mahasiswa.java 1    J MahasiswaSarjana.java 1 X
J MahasiswaSarjana.java >  MahasiswaSarjana > MahasiswaSarjana(String, String, int, String)
1  package pert2_51421422;
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 }
```

- **Deklarasi Kelas:**

```
public class MahasiswaSarjana extends Mahasiswa {
```

Kelas ini bernama *MahasiswaSarjana* dan menggunakan kata kunci *extends*, yang menunjukkan bahwa kelas ini mewarisi dari kelas *Mahasiswa*. Ini berarti *MahasiswaSarjana* dapat mengakses atribut dan method publik dari *Mahasiswa*.

- **Atribut Kelas:**

```
private String jurusan;
```

Kelas ini memiliki satu atribut tambahan:

- *jurusan*: Menyimpan informasi tentang jurusan yang diambil oleh mahasiswa sarjana (tipe data *String*). Atribut ini juga dideklarasikan sebagai *private*.

- **Konstruktor:**

```
public MahasiswaSarjana(String nama, String npm, int umur, String jurusan) {
    super(nama, npm, umur);
    this.jurusan = jurusan;
}
```

Konstruktor ini digunakan untuk menginisialisasi objek MahasiswaSarjana.

- *super(nama, npm, umur)*: Memanggil konstruktor dari kelas induk (*Mahasiswa*) untuk menginisialisasi atribut nama, npm, dan umur.
- *this.jurusan = jurusan*: Menginisialisasi atribut *jurusan* dengan parameter yang diterima.

- **Method *tampilkanData()*:**

```
public void tampilkanData() {
    super.tampilkanData();
    System.out.println("Jurusan: " + jurusan);
}
```

Method ini mengoverride method *tampilkanData()* dari kelas induk.

- *super.tampilkanData()*: Memanggil method *tampilkanData()* dari kelas *Mahasiswa* untuk mencetak nama, NPM, dan umur.
- Setelah itu, mencetak jurusan mahasiswa sarjana.

Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS M:\File Kuliah\Semester 7\Praktikum RPL 2\Pert 2\pert2_51421422>
PS M:\File Kuliah\Semester 7\Praktikum RPL 2\Pert 2\pert2_51421422> cd 'm:\File Kuliah\Semester 7\Praktikum RPL 2\Pert 2\pert2_51421422' & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Simon\AppData\Local\Programs\Code\User\workspaceStorage\2\3f536e2267818630726e81f48abe990\redhat_java\jdk_us\jdk-1s-java-project\bin' 'pert2_51421422.pert2_51421422'

Nama: Simon
NPM: 51421422
Umur: 24

Nama: Praja
NPM: 51421421
Umur: 22
Jurusan: Informatika

Nama: Simon
NPM: 51421422
Umur: 24

Nama: Praja
NPM: 51421421
Umur: 22
Jurusan: Informatika

```

1. Output Pertama (Dari Objek Mahasiswa):

- Nama: Simon
- NPM: 51421422
- Umur: 24

Output ini dihasilkan dari pemanggilan method *tampilkanData()* pada objek *mahasiswa*. Method ini mencetak nama, NPM, dan umur mahasiswa yang telah diinisialisasi.

2. Output Kedua (Dari Objek MahasiswaSarjana):

- Nama: Praja
- NPM: 51421421
- Umur: 22
- Jurusan: Informatika

Output ini dihasilkan dari pemanggilan method *tampilkanData()* pada objek *mahasiswaSarjana*.

- Pertama, method ini memanggil method *tampilkanData()* dari kelas induk *Mahasiswa*, yang mencetak nama, NPM, dan umur.
- Kemudian, method ini mencetak jurusan, yang merupakan atribut tambahan di kelas *MahasiswaSarjana*.