

ACTIVITY PERTEMUAN 2

NAMA : Yolanda Pangemanan

NPM : 51421536

KELAS : 4IA14

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

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

Jawab:

Anatomi Dasar Sebuah Kelas dalam Java

Sebuah kelas dalam Java adalah blueprint atau cetak biru untuk membuat objek. Kelas mendefinisikan sifat (atribut) dan perilaku (metode) yang akan dimiliki oleh objek-objek yang dibuat darinya.

Komponen Utama Kelas Java

1. Nama Kelas:

- Setiap kelas harus memiliki nama yang unik.
- Nama kelas biasanya dimulai dengan huruf kapital.
- Contoh: Mobil, Mahasiswa, AkunBank.

2. Atribut (Fields):

- Merupakan variabel yang mendefinisikan karakteristik atau sifat dari objek.
- Atribut menyimpan data yang terkait dengan objek.
- Contoh: warna, merk, tahunPembuatan untuk kelas Mobil.

3. Metode:

- Merupakan fungsi yang mendefinisikan perilaku atau tindakan yang dapat dilakukan oleh objek.
- Metode dapat menerima parameter dan mengembalikan nilai.
- Contoh: bergerak(), ngebrem(), isiBensin() untuk kelas Mobil.

4. Konstruktor:

- Metode khusus yang digunakan untuk menginisialisasi objek ketika dibuat.

- Nama konstruktor sama dengan nama kelas.
- Contoh: Mobil() untuk membuat objek baru dari kelas Mobil.

5. Blok Kode:

- Bagian kode yang dieksekusi pada saat-saat tertentu, seperti saat objek dibuat atau dihancurkan.
- Contoh: blok static, blok initializer.

Contoh Kelas Sederhana

Java

```
public class Mobil {
    // Atribut
    String warna;
    String merk;
    int tahunPembuatan;

    // Konstruktor
    public Mobil(String warna, String merk, int tahunPembuatan) {
        this.warna = warna;
        this.merk = merk;
        this.tahunPembuatan = tahunPembuatan;
    }

    // Metode
    public void bergerak() {
        System.out.println("Mobil sedang bergerak");
    }

    public void ngebrem() {
        System.out.println("Mobil sedang mengerem");
    }
}
```

Penjelasan Contoh:

- **Kelas Mobil:** Mendefinisikan blueprint untuk objek mobil.
- **Atribut:** warna, merk, dan tahunPembuatan menyimpan informasi tentang mobil.
- **Konstruktor:** Mobil(String warna, String merk, int tahunPembuatan) digunakan untuk membuat objek mobil baru dengan nilai atribut yang diberikan.
- **Metode:** bergerak() dan ngebrem() mendefinisikan tindakan yang dapat dilakukan oleh objek mobil.

Membuat Objek dari Kelas:

Java

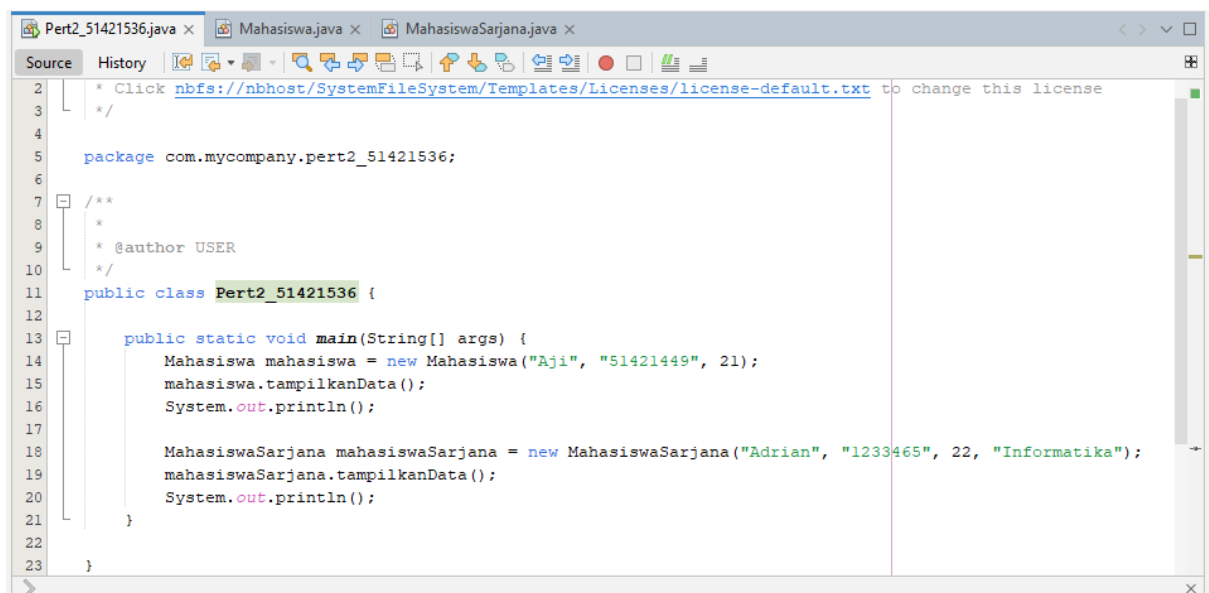
```

public class Main {
    public static void main(String[] args) {
        Mobil mobilku = new Mobil("Merah", "Toyota", 2023);
        mobilku.bergerak();
    }
}

```

Dalam contoh di atas, kita membuat objek mobilku dari kelas Mobil. Objek ini memiliki warna merah, merek Toyota, dan tahun pembuatan 2023. Kita kemudian memanggil metode bergerak() untuk membuat mobil tersebut bergerak.

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



```

Pert2_51421536.java x  Mahasiswa.java x  MahasiswaSarjana.java x
Source  History  [Icons]
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  */
4
5  package com.mycompany.pert2_51421536;
6
7  /**
8   *
9   * @author USER
10  */
11  public class Pert2_51421536 {
12
13      public static void main(String[] args) {
14          Mahasiswa mahasiswa = new Mahasiswa("Aji", "51421449", 21);
15          mahasiswa.tampilkanData();
16          System.out.println();
17
18          MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana("Adrian", "1233465", 22, "Informatika");
19          mahasiswaSarjana.tampilkanData();
20          System.out.println();
21      }
22
23  }

```

Kode ini mendefinisikan sebuah program Java sederhana yang memiliki satu kelas utama bernama `Pert2_51421536`. Di dalam kelas ini, terdapat metode `main` yang menjadi titik awal eksekusi program.

```
9  * @author USER
10 */
11 public class Mahasiswa {
12
13     private String nama;
14     private String npm;
15     private int umur;
16
17     public Mahasiswa(String nama, String npm, int umur) {
18         this.nama = nama;
19         this.npm = npm;
20         this.umur = umur;
21     }
22
23     public void tampilkanData() {
24         System.out.println("Nama: " + nama);
25         System.out.println("NPM: " + npm);
26         System.out.println("Umur: " + umur);
27     }
28 }
29
30 package com.mycompany.pert2_51421536;
31
32 class Mahasiswa {
```

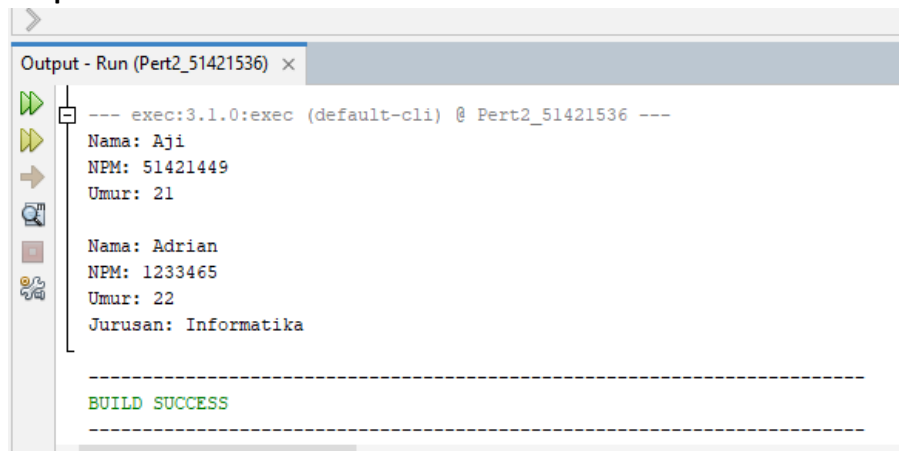
Kelas Mahasiswa ini merepresentasikan data tentang seorang mahasiswa. Ketika kita membuat objek dari kelas ini, kita akan membuat sebuah "instansi" dari mahasiswa dengan nilai nama, NPM, dan umur yang spesifik. Metode `tampilkanData()` digunakan untuk menampilkan informasi tentang mahasiswa tersebut.

```
5  package com.mycompany.pert2_51421536;
6
7  /**
8   *
9   * @author USER
10  */
11  public class MahasiswaSarjana extends Mahasiswa {
12
13      private String jurusan;
14
15      public MahasiswaSarjana(String nama, String npm, int umur, String jurusan) {
16          super(nama, npm, umur);
17          this.jurusan = jurusan;
18      }
19
20      @Override
21      public void tampilkanData() {
22          super.tampilkanData();
23          System.out.println("Jurusan: " + jurusan);
24      }
25  }
26
27  package com.mycompany.pert2_51421536;
28
29  class MahasiswaSarjana {
```

Kode ini memperkenalkan konsep **pewarisan** (inheritance) dalam pemrograman berorientasi objek. Kelas `MahasiswaSarjana` adalah spesialisasi dari kelas `Mahasiswa`. Ia mewarisi semua sifat dari mahasiswa umum, tetapi memiliki tambahan atribut `jurusan` yang spesifik untuk mahasiswa sarjana. Metode `tampilkanData()` pada `MahasiswaSarjana` tidak

hanya menampilkan informasi umum seperti nama, NPM, dan umur, tetapi juga menampilkan informasi spesifik tentang jurusan.

Output



```
--- exec:3.1.0:exec (default-cli) @ Pert2_51421536 ---
Nama: Aji
NPM: 51421449
Umur: 21

Nama: Adrian
NPM: 1233465
Umur: 22
Jurusan: Informatika

-----
BUILD SUCCESS
-----
```

Output ini menunjukkan bahwa program Java telah berjalan dengan sukses. Program ini berhasil membuat dua objek: satu objek dari kelas `Mahasiswa` dan satu objek dari kelas `MahasiswaSarjana`. Kedua objek ini kemudian menampilkan informasi tentang diri mereka masing-masing. Konsep pewarisan yang diterapkan pada kelas `MahasiswaSarjana` memungkinkan kita untuk membuat kelas yang lebih spesifik dengan menambahkan atribut dan metode tambahan.