

Nama : Benony Gabriel

NIM : 105222002

Dalam pemrograman berorientasi objek (OOP) di Java, kelas abstrak (abstract class) adalah kelas yang tidak dapat diinstansiasi secara langsung. Artinya, kita tidak bisa membuat objek langsung dari kelas abstrak. Kelas abstrak berfungsi sebagai cetak biru bagi kelas-kelas lain yang akan mengimplementasikan atau memperluasnya.

Karakteristik Abstract Class:

- Deklarasi Abstract Class: Dideklarasikan dengan kata kunci abstract.
- Abstract Method: Bisa memiliki metode abstrak (tanpa implementasi) dan metode non-abstrak (dengan implementasi).
- Tidak Dapat Diinstansiasi: Tidak bisa membuat objek dari kelas abstrak.
- Inheritance: Kelas abstrak biasanya digunakan untuk mendefinisikan metode umum yang harus diimplementasikan oleh kelas turunannya.
- Instance Variables: Dapat memiliki variabel instance dan konstruktor.

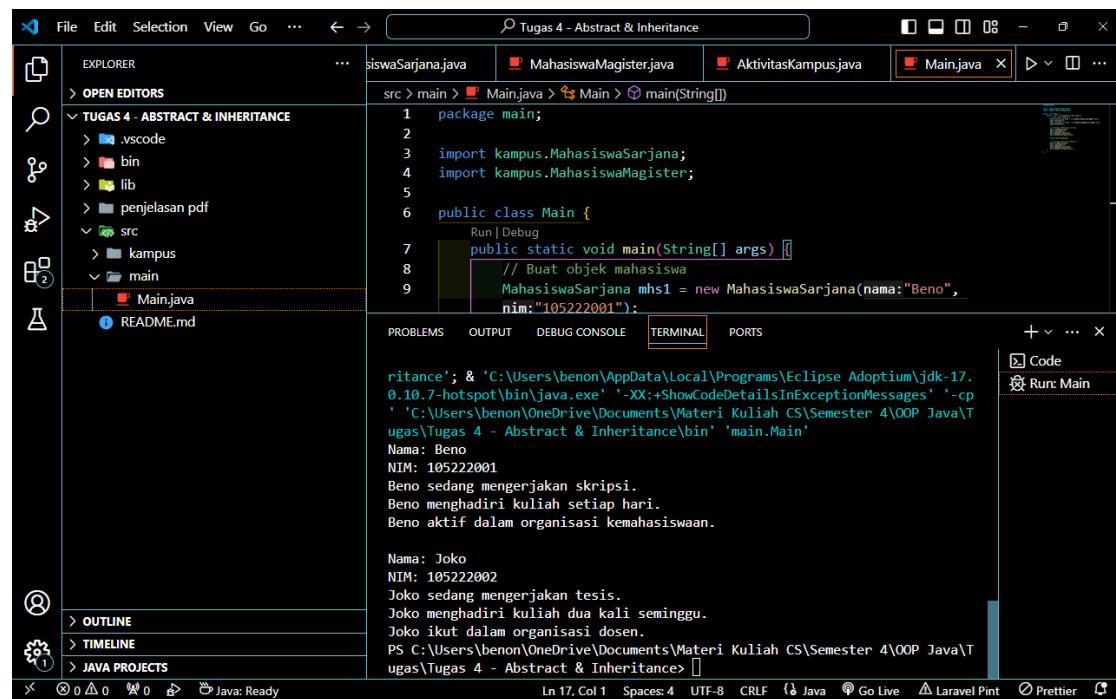
Interface

Interface di Java adalah blueprint dari sebuah kelas. Interface dapat berisi hanya deklarasi metode (tanpa implementasi) dan konstanta (variabel yang secara implisit adalah public, static, dan final). Interface digunakan untuk menentukan kontrak atau serangkaian metode yang harus diimplementasikan oleh kelas yang menerapkan interface tersebut.

Karakteristik Interface:

- Deklarasi Interface: Dideklarasikan dengan kata kunci interface.
- Abstract Methods: Semua metode dalam interface secara default adalah abstrak dan tidak memiliki implementasi (kecuali metode default dan statik di Java 8 ke atas).
- Multiple Inheritance: Sebuah kelas dapat mengimplementasikan lebih dari satu interface, memungkinkan pewarisan ganda.
- Implementasi: Kelas yang mengimplementasikan sebuah interface harus menyediakan implementasi untuk semua metode yang dideklarasikan dalam interface.
- Constant Variables: Dapat memiliki variabel yang secara default bersifat public, static, dan final.

Hasil Output dari main.java:



The screenshot shows an IDE window titled "Tugas 4 - Abstract & Inheritance". The Explorer panel on the left shows a project structure with folders like .vscode, bin, lib, penjelasan.pdf, src, kampus, main, and Main.java. The Editor panel displays the code for Main.java, which includes imports for kampus.MahasiswaSarjana and kampus.MahasiswaMagister, and a main method that creates a MahasiswaSarjana object named mhs1 with the name "Beno" and NIM "105222001". The Terminal panel at the bottom shows the output of the program, which prints the details of the student and their activities.

```
1 package main;
2
3 import kampus.MahasiswaSarjana;
4 import kampus.MahasiswaMagister;
5
6 public class Main {
7     public static void main(String[] args) {
8         // Buat objek mahasiswa
9         MahasiswaSarjana mhs1 = new MahasiswaSarjana(nama:"Beno",
10             nim:"105222001");
11     }
12 }
```

Output:

```
ritance'; & 'C:\Users\benon\AppData\Local\Programs\Eclipse Adoptium\jdk-17.
0.10.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp
' 'C:\Users\benon\OneDrive\Documents\Materi Kuliah CS\Semester 4\OOP Java\T
ugas\Tugas 4 - Abstract & Inheritance\bin' 'main.Main'
Nama: Beno
NIM: 105222001
Beno sedang mengerjakan skripsi.
Beno menghadiri kuliah setiap hari.
Beno aktif dalam organisasi kemahasiswaan.

Nama: Joko
NIM: 105222002
Joko sedang mengerjakan tesis.
Joko menghadiri kuliah dua kali seminggu.
Joko ikut dalam organisasi dosen.
PS C:\Users\benon\OneDrive\Documents\Materi Kuliah CS\Semester 4\OOP Java\T
ugas\Tugas 4 - Abstract & Inheritance>
```