

ACTIVITY PERTEMUAN 1

NAMA : Gerald Evan Ariapraja Loppies

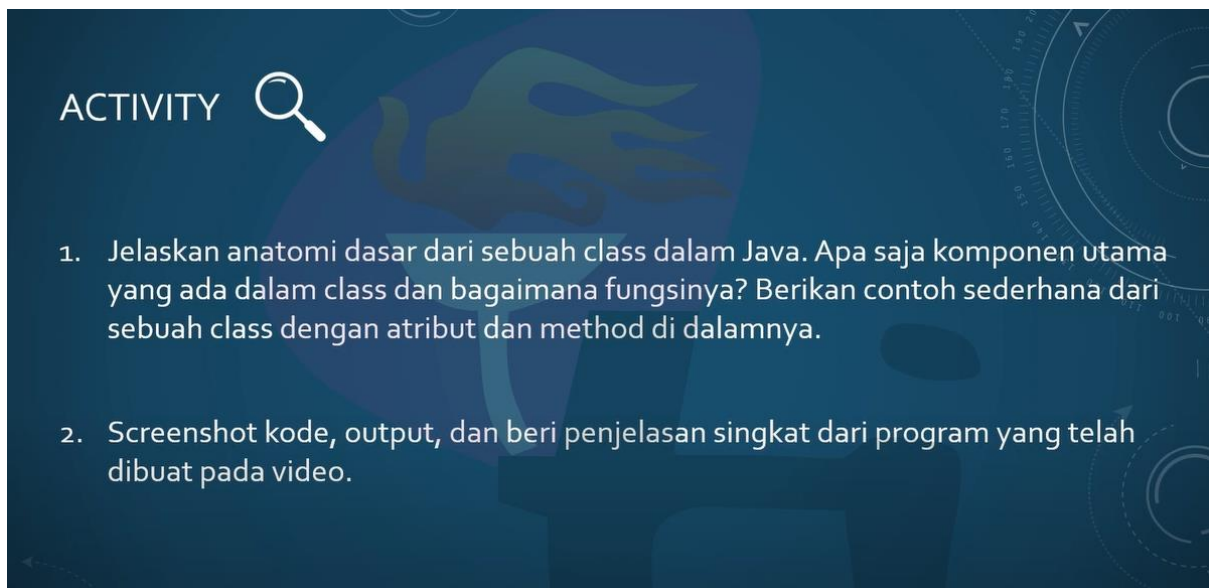
NPM : 50421548

KELAS : 4IA14

MATERI : Anatomi Class, Struktur Aplikasi Java, Object Oriented Programming (OPP) Dengan Java

MATA PRAKTIKUM : Praktikum RPL 2

(Screenshoot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)



ACTIVITY

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

Jawab :

Komponen Utama dari Sebuah Class

1. Nama Class:

- o Nama class adalah identifier yang digunakan untuk membuat objek. Dalam Java, konvensinya adalah menggunakan camel case, di mana huruf pertama dari setiap kata digabungkan, dan huruf pertama dari class harus huruf besar.
- o Contoh: Person, Car.

2. Atribut (Fields/Variables):

- o Atribut adalah variabel yang menyimpan keadaan atau data dari objek yang dihasilkan oleh class tersebut.
- o Atribut bisa bersifat **private** (hanya bisa diakses dalam class itu saja) atau **public** (dapat diakses dari luar class).

3. Constructor:

- Constructor adalah method khusus yang dipanggil saat sebuah objek dari class dibuat. Constructor digunakan untuk menginisialisasi atribut dari class.
- Nama constructor selalu sama dengan nama class dan tidak memiliki return type.

4. **Method:**

- Method adalah fungsi atau prosedur yang mendefinisikan perilaku atau aksi yang dapat dilakukan oleh objek. Method digunakan untuk memanipulasi data (atribut) dari class atau untuk menjalankan operasi tertentu.
- Method dapat menerima parameter dan dapat memiliki nilai return.

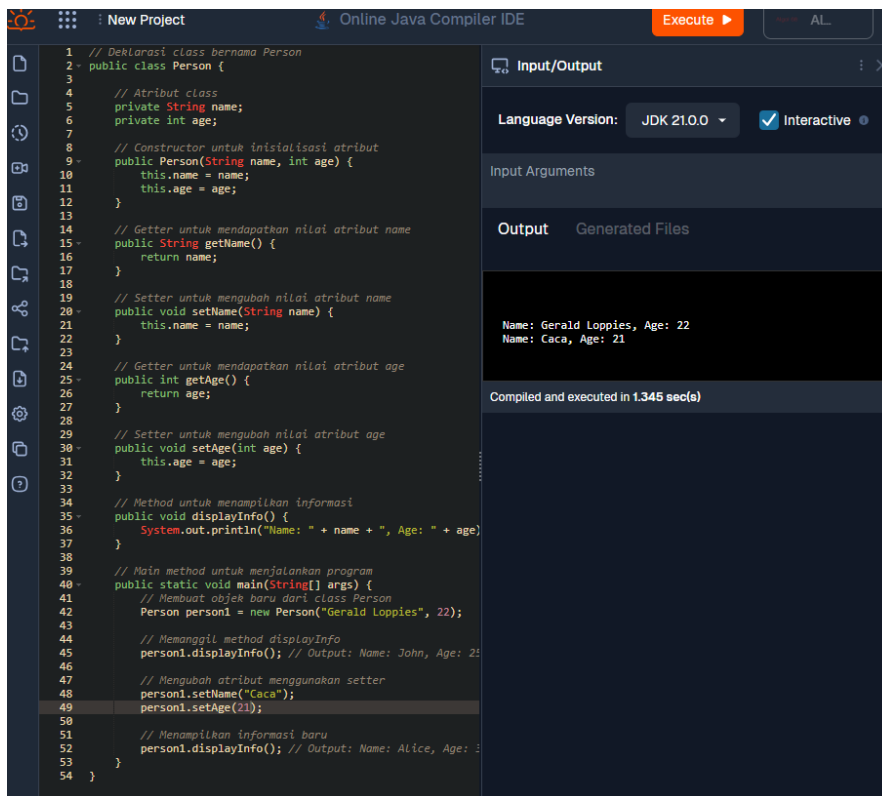
5. **Access Modifiers:**

- Access modifiers menentukan visibilitas dari class, atribut, dan method. Modifier yang umum digunakan adalah:
 - **public:** Dapat diakses dari mana saja.
 - **private:** Hanya dapat diakses dari dalam class tersebut.
 - **protected:** Dapat diakses oleh class dalam package yang sama atau subclass.

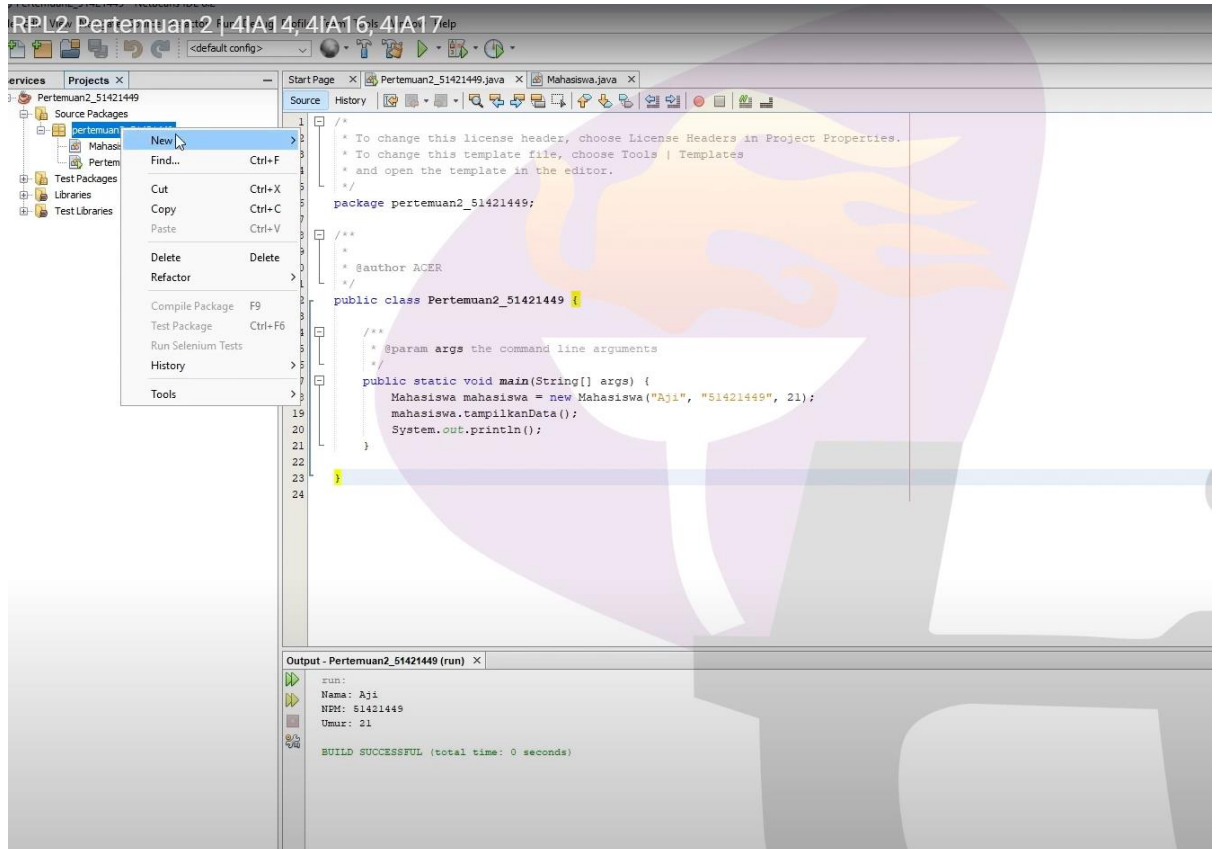
6. **Getter dan Setter:**

- Getter dan setter adalah method yang digunakan untuk mendapatkan (get) dan mengubah (set) nilai dari atribut private. Ini adalah bagian dari konsep **encapsulation** dalam OOP (Object-Oriented Programming).

Contoh Java



2.



□ **Package Declaration:**

- `package pert2_50421437;` menunjukkan bahwa class ini berada di dalam paket `pert2_50421437`.

□ **Deklarasi Class:**

- `public class Mahasiswa { ... }` mendeklarasikan class Mahasiswa. Class ini bersifat **public**, sehingga dapat diakses dari mana saja di dalam proyek.

□ **Atribut (Fields/Instance Variables):**

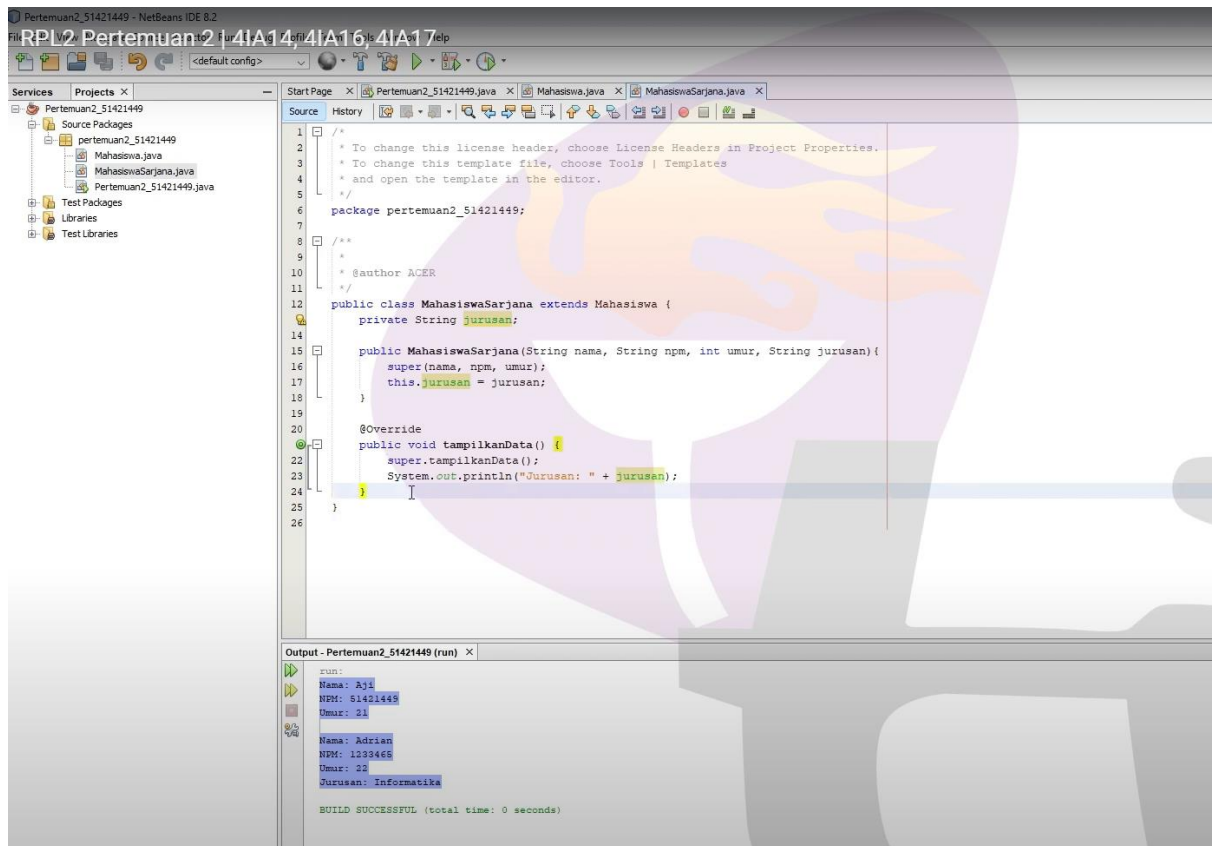
- Tiga atribut privat (`private`) dideklarasikan:
 - `String nama`: menyimpan nama mahasiswa.
 - `String npm`: menyimpan nomor pokok mahasiswa (NPM).
 - `int umur`: menyimpan umur mahasiswa.
- Atribut diberi akses **private** untuk menjaga enkapsulasi, sehingga hanya bisa diakses dari dalam class ini.

□ **Constructor:**

- `public Mahasiswa(String nama, String npm, int umur)` adalah **constructor** yang digunakan untuk menginisialisasi objek Mahasiswa. Constructor ini menerima tiga parameter:
 - `String nama`: untuk mengisi atribut nama.
 - `String npm`: untuk mengisi atribut npm.
 - `int umur`: untuk mengisi atribut umur.

□ **Method tampilkanData:**

- `public void tampilkanData()` adalah method yang digunakan untuk menampilkan informasi mahasiswa. Method ini tidak menerima parameter dan tidak mengembalikan nilai (`void`).
- Method ini mencetak nilai dari atribut nama, npm, dan umur ke **console** dengan menggunakan `System.out.println()`.



Kode Java di atas mendefinisikan sebuah class bernama MahasiswaSarjana, yang merupakan subclass dari class Mahasiswa. Berikut penjelasan singkat dari kode tersebut:

1. **Deklarasi Package:**

- package pert2_50421437; menunjukkan bahwa class ini berada di dalam package pert2_50421437.

2. **Inheritance (Pewarisan):**

- public class MahasiswaSarjana extends Mahasiswa { ... } mendeklarasikan class MahasiswaSarjana sebagai subclass dari class Mahasiswa. Dengan menggunakan keyword **extends**, class ini mewarisi semua atribut dan method yang ada di class Mahasiswa.

3. **Atribut (Field):**

- private String jurusan; adalah atribut tambahan milik MahasiswaSarjana yang menyimpan data jurusan mahasiswa.

4. **Constructor:**

- public MahasiswaSarjana(String nama, String npm, int umur, String jurusan) adalah constructor untuk MahasiswaSarjana. Constructor ini menerima parameter:
 - String nama: untuk mengisi atribut nama dari superclass Mahasiswa.
 - String npm: untuk mengisi atribut npm dari superclass Mahasiswa.
 - int umur: untuk mengisi atribut umur dari superclass Mahasiswa.
 - String jurusan: untuk mengisi atribut jurusan dari class ini.
- Constructor menggunakan keyword **super()** untuk memanggil constructor dari superclass (Mahasiswa) dan menginisialisasi atribut nama, npm, dan umur.

5. **Method tampilkanData():**

- **Override Method:** @Override menunjukkan bahwa method tampilkanData() di class MahasiswaSarjana **mengoverride** method dengan nama yang sama di superclass Mahasiswa.
- Dalam method ini:
 - super.tampilkanData(); memanggil method tampilkanData() dari superclass (Mahasiswa) untuk menampilkan data nama, npm, dan umur.
 - System.out.println("Jurusan: " + jurusan); menambahkan tampilan untuk atribut jurusan, sehingga jurusan juga ditampilkan setelah memanggil method dari superclass.