

Praktikum **Algoritma dan Pemrograman II**

# Pendahuluan OOP, Class and Object

5002221053 - Putri Ghaida Tsuroyya

5002221072 - Mohammad Febryan Khamim

Kelompok 5



# Outline Pembahasan

- Pengantar OOP
- Class and Object
- Unified Modeling Language (UML)

# Pengantar OOP

# Apa itu OOP?

**Object Oriented Programming (OOP)** atau Pemrograman Berorientasi Objek (PBO) adalah suatu paradigma pemrograman pada Java yang konsepnya selayaknya objek. Dalam konsep ini, program memiliki struktur selayaknya sebuah objek yang dapat berinteraksi satu sama lain untuk mencapai sebuah tujuan tertentu.

**Algoritma  
Pemrograman**

Sekumpulan Langkah



**Mencapai tujuan  
tertentu**

Dalam OOP, masing-masing langkah atau aksi yang harus dilakukan tersebut disusun terpisah seakan-akan menjadi 'objek' yang berbeda.

# Mengapa harus OOP?

Menurut W3Schools

- OOP lebih cepat dan mudah dieksekusi
- OOP menawarkan struktur yang jelas pada sebuah program
- OOP memudahkan seorang programmer melakukan maintenance dan perbaikan
- OOP memungkinkan untuk membuat suatu program dengan *syntax* yang lebih pendek

# 4 Pilar OOP

- **Encapsulation** : Membungkus data dan method dalam **class**.
- **Inheritance** : Pewarisan sifat dari **class** lain (Extends)
- **Polymorphism** : Satu nama, tetapi memiliki banyak bentuk (overloading atau overriding)
- **Abstraction** : Menyembunyikan kompleksitas (abstract class / interface)

# Class and Object

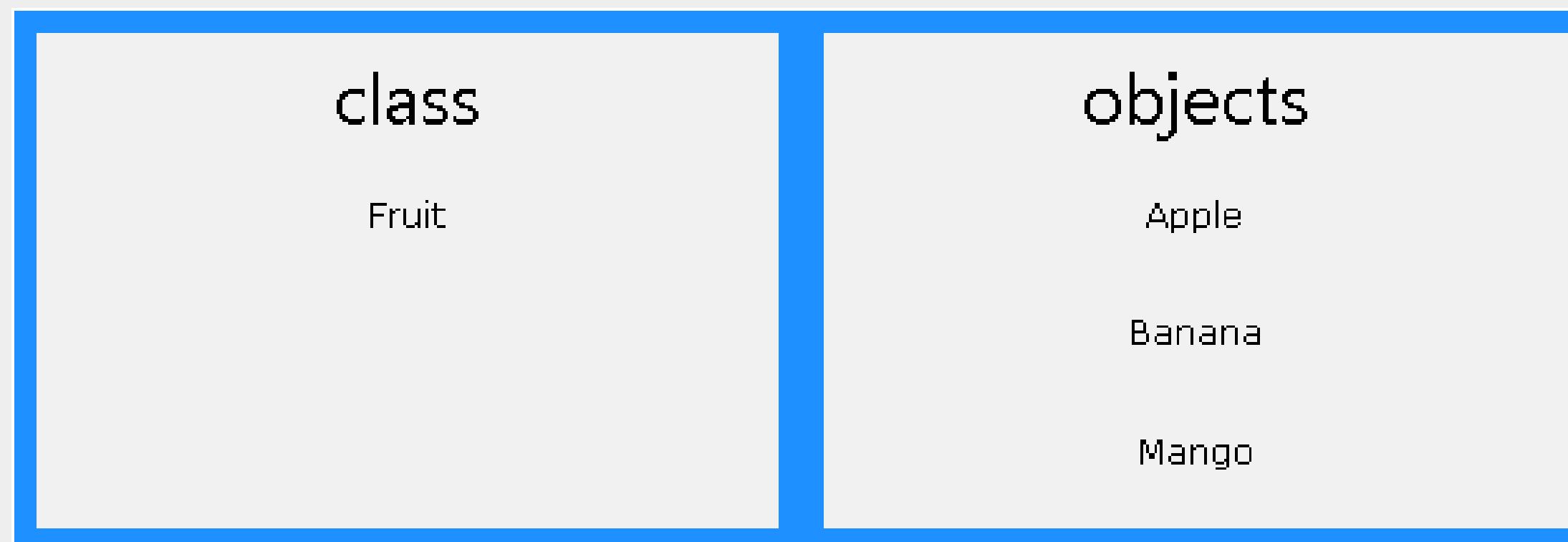
# Class and Object

**Class** adalah cetak biru atau prototipe yang ditentukan pengguna iuntuk membuat beberapa **object** sekaligus dengan ‘perilaku’ atau ‘karakter’ yang sama dengan **class**.

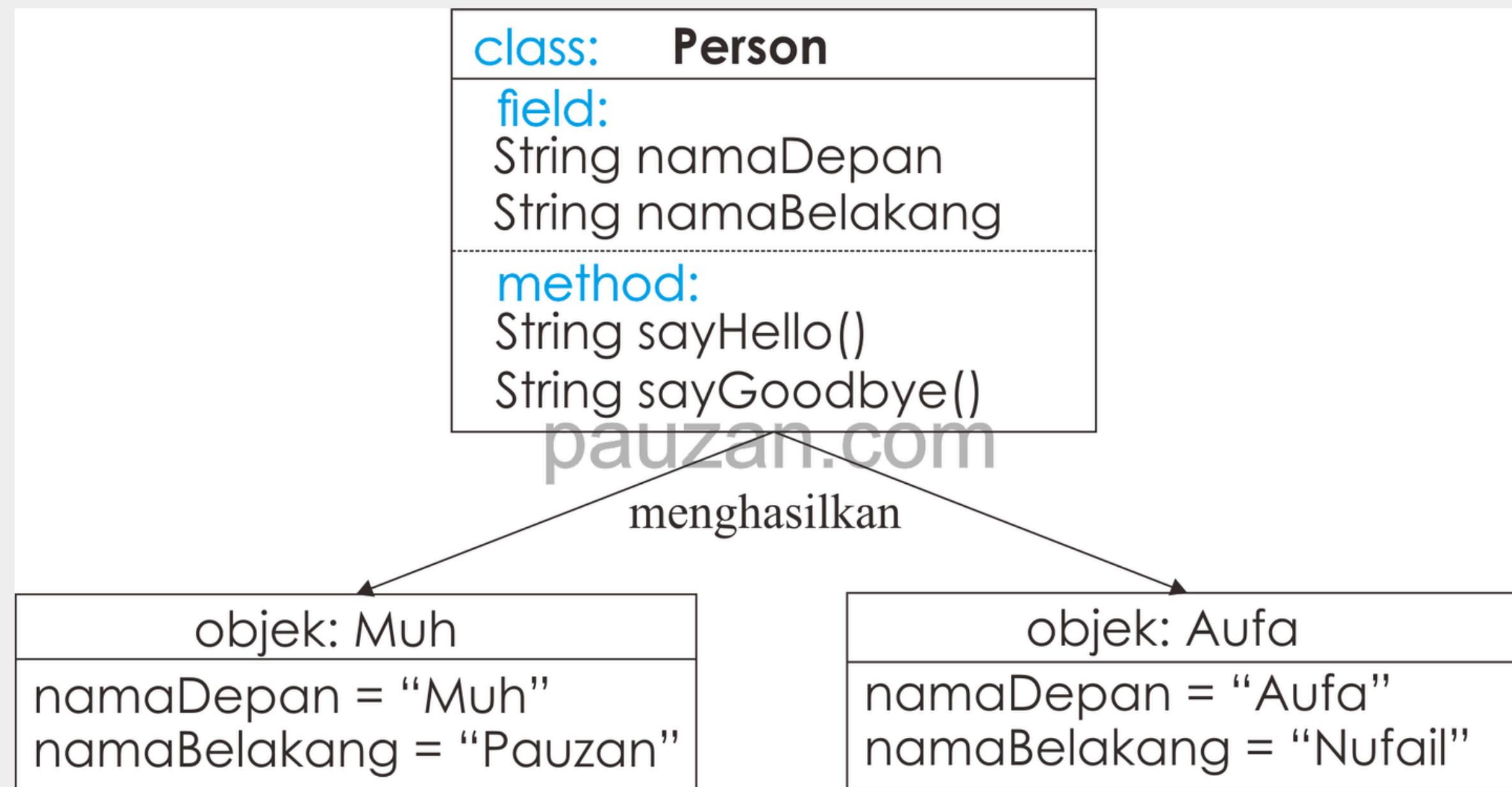
# Object

- **Object** merupakan perwujudan dari **Class** → setiap **object** memiliki Atribut dan Method yang dimiliki oleh **class**-nya
- Setiap *object* dapat berinteraksi dengan *object* lain, meskipun berasal dari *class* berbeda
- Memiliki **Attribute** (status *object*) dan **Method** (tingkah laku *object*).

# Example



# Example



# Atribut dan Method

- **Atribut** : Variabel dalam **class** (contoh: nama, umur)
- **Method** : Fungsi / aktivitas dalam **class** (contoh: belajar(), makan())

```
class Mahasiswa {  
    // Atribut (data)  
    String nama;  
    int umur;  
  
    // Method (perilaku)  
    void belajar() {  
        System.out.println(nama + " sedang belajar");  
    }  
}
```

# Mendefinisikan Class

```
class NamaKelas{  
    tipe data 1;  
    tipe data 2;  
    ...  
    tipe data n;  
  
    tipe method1 (daftar-parameter){  
        kode untuk method1  
    }  
  
    tipe method1 (daftar-parameter){  
        kode untuk method1  
    }  
}
```

# Langkah Membuat Class

## 1. Membuat Class

```
Public class Mahasiswa{  
    String NRP, Nama, Departemen;  
    int Umur, IPK;  
}
```

# Langkah Membuat Class

## 2. Instansiasi Objek

```
public static void main(String[] args) {
    // Membuat objek baru mahasiswa dengan nama Ole
    Mahasiswa Ole = new Mahasiswa();

    // Mengisi nilai ke dalam data-data objek Mahasiswa
    Ole.NRP = "5002221072";
    Ole>Nama = "Ole Romeny";
    Ole.Departemen = "Matematika";
    Ole.Umur = 20;
    Ole.IPK = 3.74;
}
```

# Langkah Membuat Class

## 3. Akses Data Objek

```
// Mencetak data-data
System.out.println("Data Mahasiswa");
System.out.println("NRP      : " + Ole.NRP);
System.out.println("Nama     : " + Ole>Nama);
System.out.println("Departemen : " + Ole.Departemen);
System.out.println("Umur     : " + Ole.Umur);
System.out.println("IPK      : " + Ole.IPK);
}
```

# UML (Unified Modeling Language)

# Unified Modeling Language (UML)

**UML (Unified Modeling Language)** merupakan diagram yang biasa digunakan untuk mengilustrasikan sebuah kelas dan sekaligus hubungan antar kelas. Didalam satu class diagram terdapat tiga row atau baris dimana masing-masing row memiliki kegunaan yang berbeda-beda diantaranya :

- Baris pertama menunjukkan nama class
- Baris kedua digunakan untuk menyatakan variabel yang terdapat pada class, dan
- Baris ketiga digunakan untuk menyatakan method-method yang ada pada suatu class

<b>Orang</b>	→ Nama class
- <b>nama</b> : String - <b>alamat</b> : String - <b>noIdentitas</b> : String	→ Variabel pada class
+ <b>setNama(n : String) : void</b> + <b>getNama() : String</b> + <b>setAlamat(a : String) : void</b> + <b>getAlamat() : String</b>	→ Method pada class

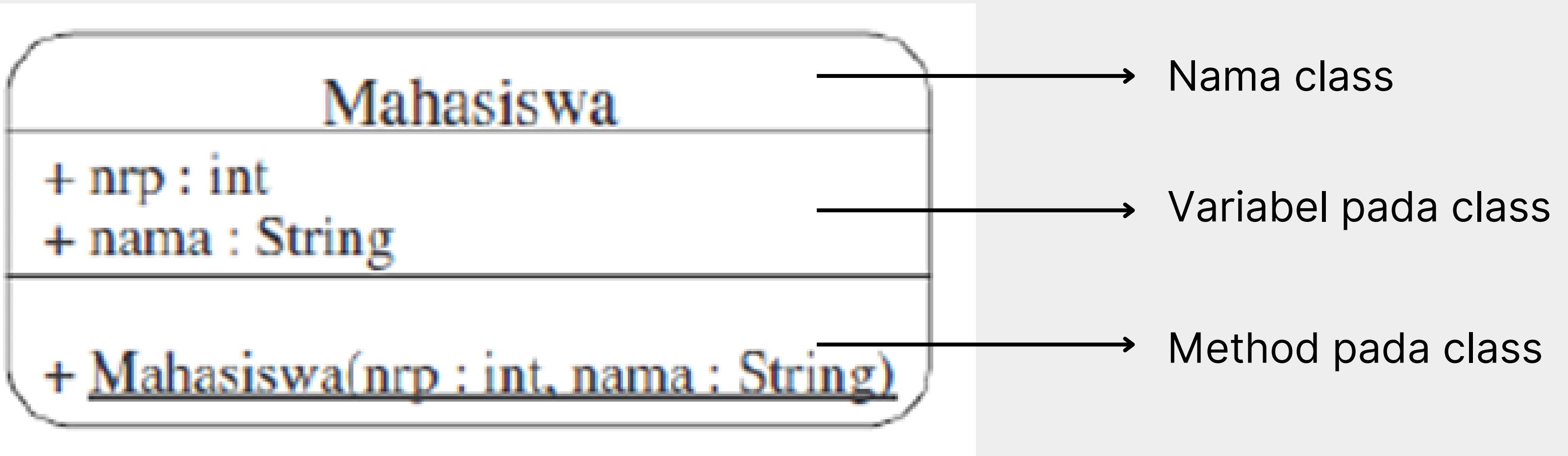
# Unified Modeling Language (UML)

Selain itu UML juga bisa digunakan untuk memberikan indikasi tentang akses modifier dari sebuah variabel atau method antara lain :

- + : Tanda ini menunjukkan bahwa suatu variabel atau method memiliki akses modifier public
- - : Tanda ini menunjukkan bahwa suatu variabel atau method memiliki akses modifier private
- # : Tanda ini menunjukkan bahwa suatu variabel atau method memiliki akses modifier protected

# Modifier

Modifier	Class dan Interface	Method dan Variabel
<b>Default (tak ada modifier ) Friendly</b>	Dikenali di paketnya	Diwarisi subclass di paket yang sama dengan superclassnya. Dapat diakses oleh method-method di class-class yang sepaket.
<b>Public</b>	Dikenali di manapun	Diwarisi oleh semua subclassnya. Dapat diakses dimanapun.
<b>Protected</b>	Tidak dapat diterapkan	Diwarisi oleh semua subclassnya. Dapat diakses oleh method-method di class-class yang sepaket.
<b>Private</b>	Tidak dapat diterapkan	Tidak diwarisi oleh subclassnya Tidak dapat diakses oleh class lain.



# Constructor

- **Constructor** adalah method khusus yang digunakan untuk menginisiasi objek.
- **Constructor** dipanggil ketika sebuah objek dari suatu **class** dibuat.
- **Constructor** dapat menjadi initial value untuk atribut objek tersebut.



# Mari Mencoba

Kelompok 5

Buat **Class Buku** dengan atribut:

- **Judul** (String)
- **Penulis** (String)
- **Tahun Terbit** (int)
- **Harga** (double)

Lalu, cetak informasi dari objek **buku1** !

Kelompok 5

### Buku

- + judul : String
- + pengarang : String

- + Buku(judul : String, pengarang : String)
- + baca() : void



# Penugasan Pertemuan 5

Kelompok 5

Kerjakan soal-soal pada sebuah tautan berikut:

[intip.in/Pertemuan5Alprokom2](http://intip.in/Pertemuan5Alprokom2)

Kelompok 5

# Thank You