

Pemrograman Berbasis Obyek

Pengenalan Konsep OOP

Oleh Politeknik Elektronika Negeri Surabaya

Yunia Ikawati



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Konten

- Dasar Class Diagram
- Java Naming Convention

Dasar Class Diagram

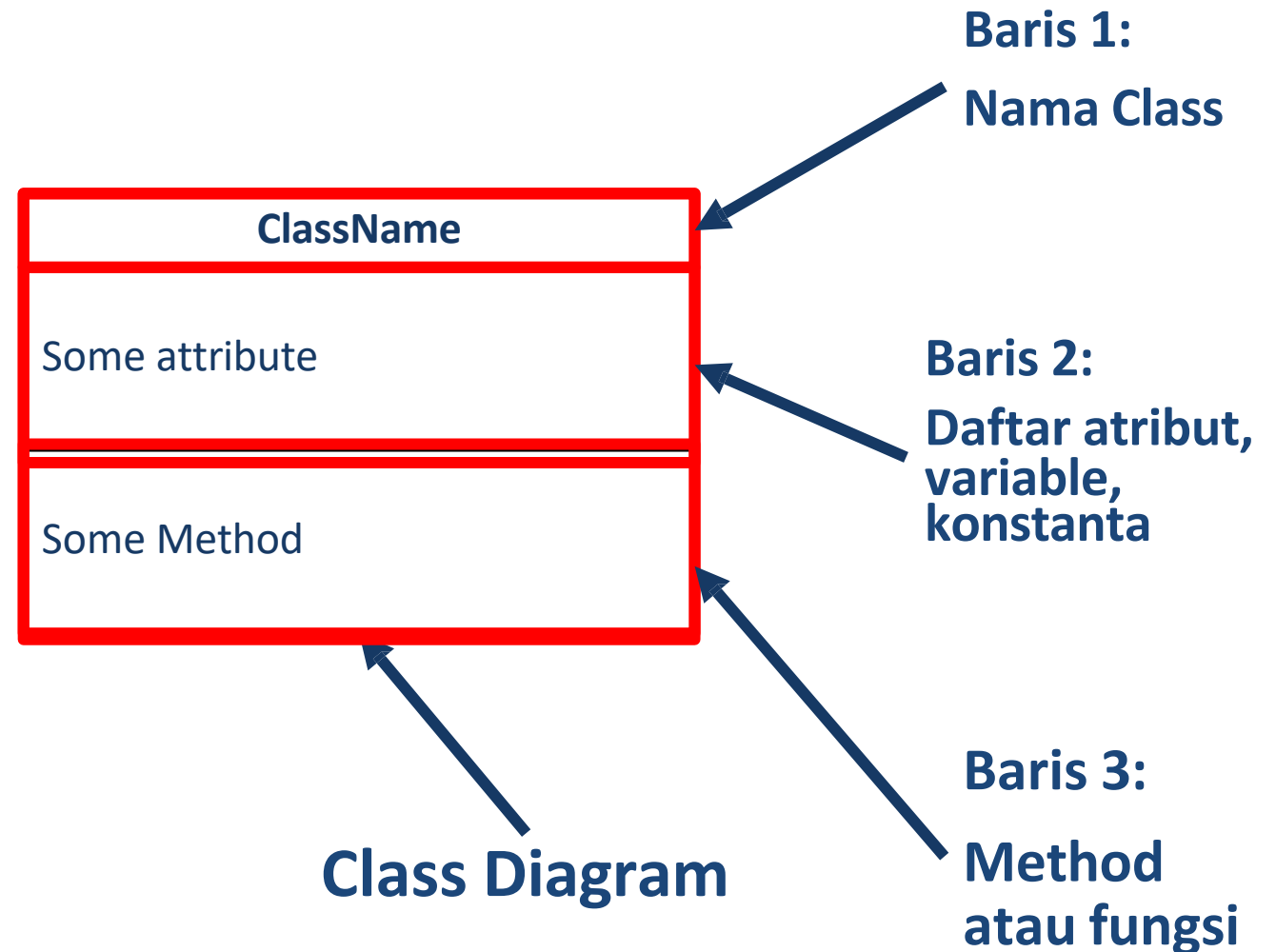
OOP

tidak mungkin terlepas dari
Class Diagram

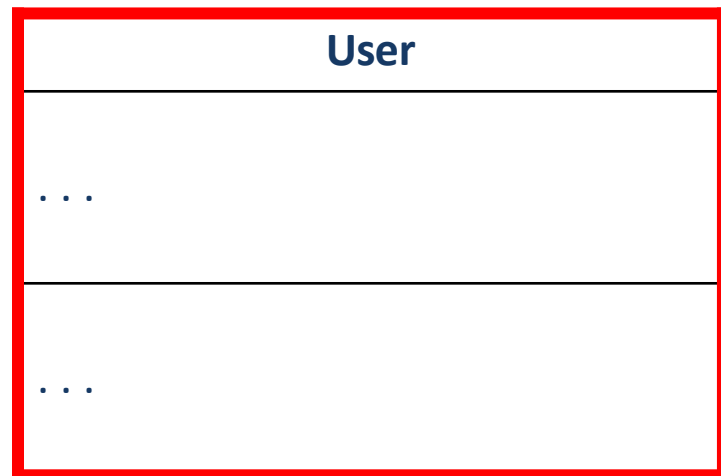
CLASS DIAGRAM

Apa itu **Class Diagram**?

Diagram yang menggambarkan **Class**



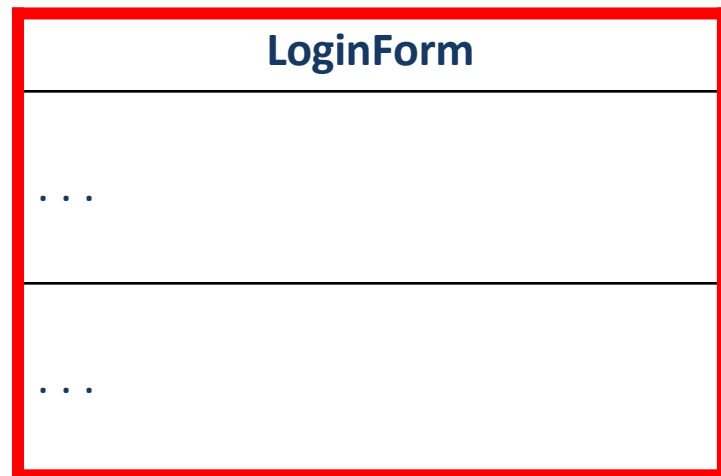
CLASS NAME



Jika di implementasikan
dalam Bahasa java:

```
public class User{  
    ...  
}
```

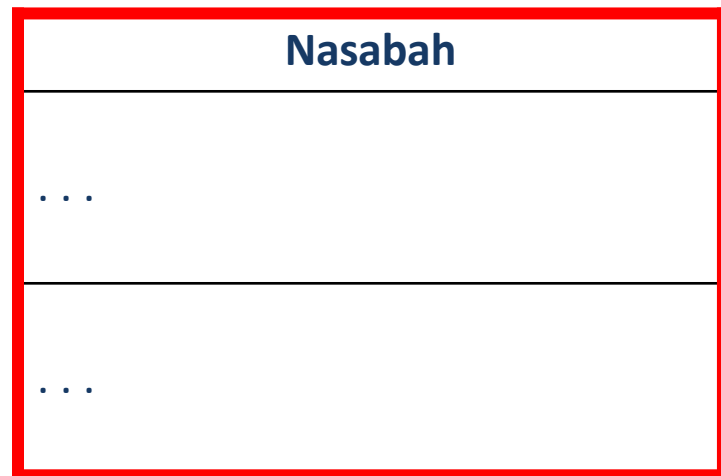
Ini adalah satu buah class
dengan nama class : **User**



Jika di implementasikan dalam Bahasa java:

```
public class LoginForm{
    ...
}
```

Ini adalah satu buah class dengan nama class : **LoginForm**

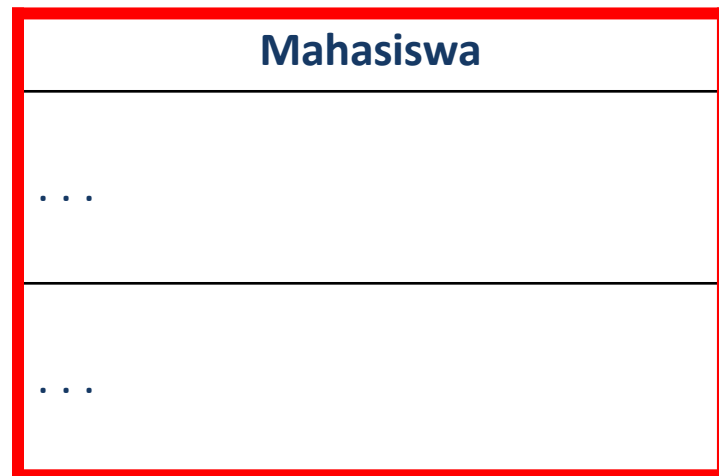


Jika di implementasikan dalam Bahasa java:

```

public class Nasabah{
    ...
}
    
```

Ini adalah satu buah class dengan nama class : **Nasabah**

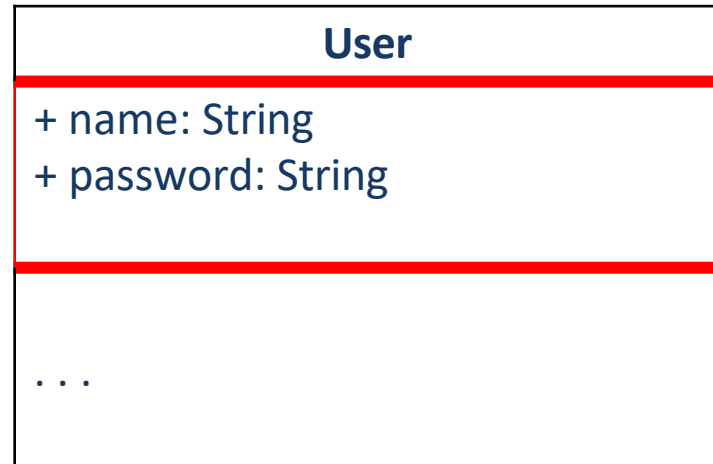


Jika di implementasikan
dalam Bahasa java:

... ?

Ini adalah satu buah class
dengan nama class : **Mahasiswa**

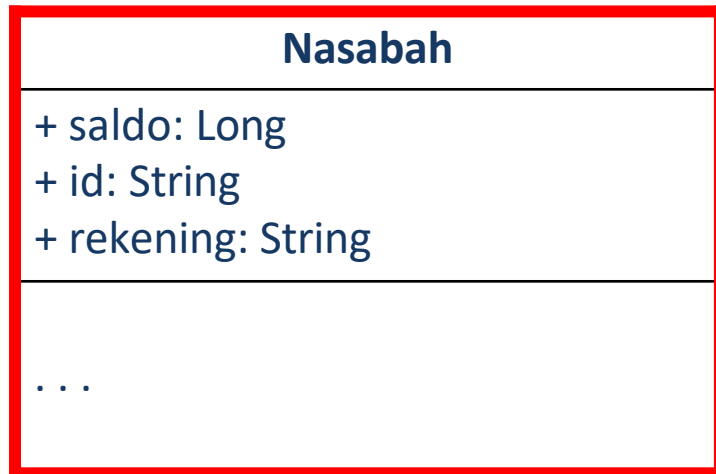
ATTRIBUTE



Jika di implementasikan dalam Bahasa java:

```
public class User{
    public String name;
    public String password;
}
```

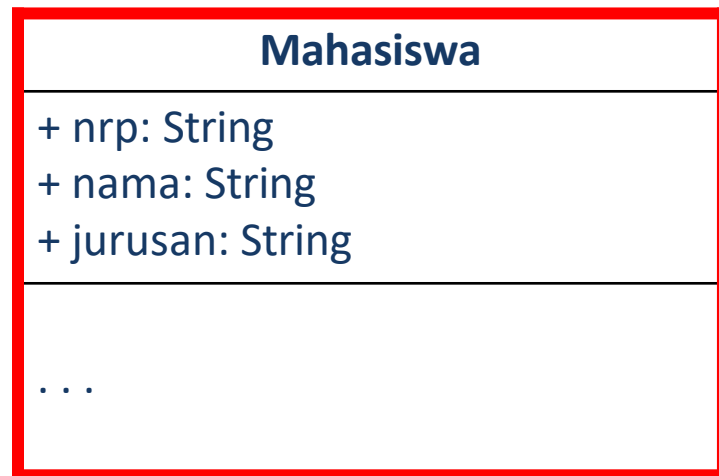
Dalam class User terdapat dua variable : name dan password



Jika di implementasikan dalam Bahasa java:

```
public class Nasabah{
    public Long saldo;
    public String id;
    public String rekening;
}
```

Dalam class Nasabah terdapat 3 buah variable: saldo, id, rekening

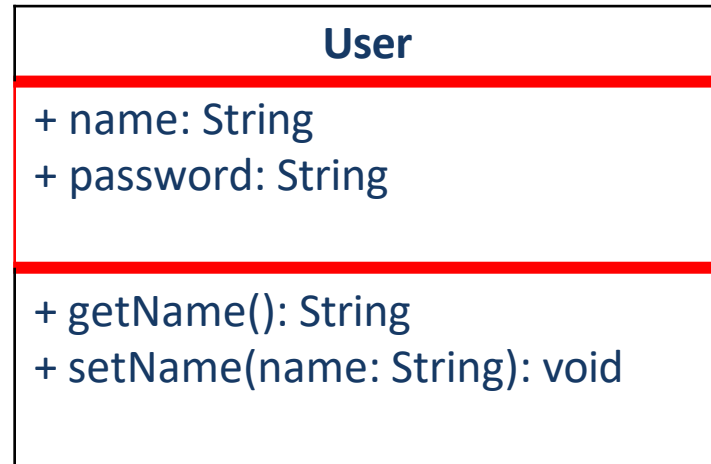


Jika di implementasikan
dalam Bahasa java:

... ?

Dalam class Mahasiswa ada 3
variable: nrp, nama, jurusan

METHOD

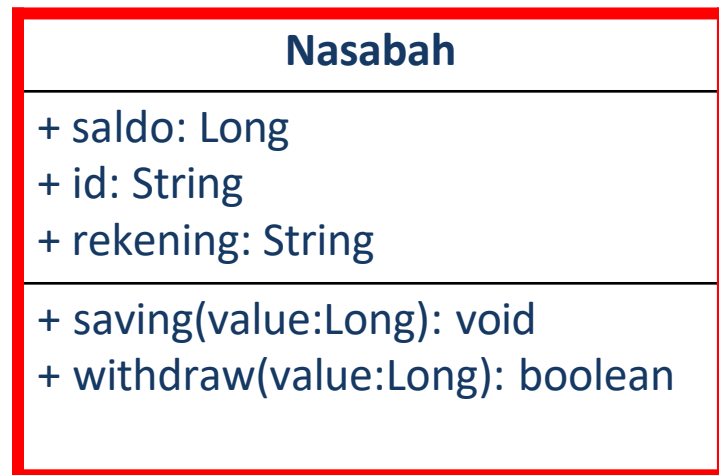


Dalam class User terdapat dua method : `getName()` dan `setName()`

Jika di implementasikan dalam Bahasa java:

```
public class User{
    public String name;
    public String password;

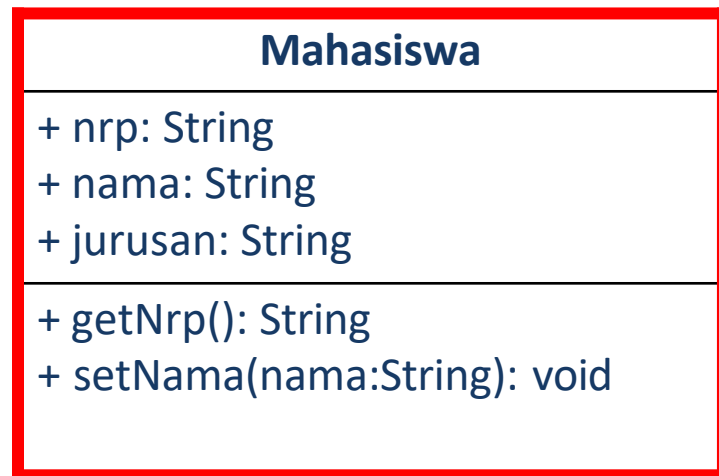
    public String getName(){...}
    public void setName(String
name){...}
}
```



Jika di implementasikan dalam Bahasa java:

```
public class Nasabah{  
    public Long saldo;  
    public String id;  
    public String rekening;  
  
    public void saving(Long value){...}  
    public Boolean withdraw(Long value){...}  
}
```

Dalam class Nasabah terdapat 2 buah method: saving() dan withdraw()



Jika di implementasikan dalam Bahasa java:

... ?

Dalam class Mahasiswa ada 2 method: getNrp() dan setName()

MODIFIER

+	adalah public
-	adalah private
#	adalah protected
~	adalah default



Mahasiswa
+ nrp: String - nama: String # jurusan: String ~ wali: String
...

Simbol	Access Modifier	Dapat Diakses dalam Class Sendiri	Dapat Diakses Class Lain	Dapat Diakses Subclass (Turunan)	Dapat Diakses dalam Package yang Sama	Keterangan
+	Public	✓ Ya	✓ Ya	✓ Ya	✓ Ya	Dapat diakses dari mana saja.
-	Private	✓ Ya	✗ Tidak	✗ Tidak	✗ Tidak	Hanya dapat diakses oleh class yang memilikinya.
#	Protected	✓ Ya	✗ Tidak	✓ Ya	✓ Ya	Dapat diakses oleh class sendiri dan class turunan.
~	Default (Package)	✓ Ya	✗ Tidak*	✗ Tidak*	✓ Ya	Hanya dapat diakses oleh class dalam package yang sama.

Java Naming Convention

Java Naming Convention

- Code Convention
 - Cara penulisan kode program yang telah disepakati sekelompok programmer
- Java Code Convention
 - <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>
 - 12 September 1997
 - Revisi: 20 April 1999
<https://www.oracle.com/java/technologies/javase/codeconventions-contents.html>
- Java Naming Convention
 - <https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html>



Naming Convention: Classes

```
class Raster;  
class ImageSprite;
```

- Class names should be **nouns**, in **mixed case** with the **first letter** of **each** internal **word** **capitalized**
- Try to keep your class names **simple** and **descriptive**
- Use **whole words**—avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML)

Naming Convention: Interfaces

```
interface RasterDelegate;  
interface Storing;
```

- Interface names should be capitalized like class names

Naming Convention: Methods

```
run();  
runFast();  
getBackground();
```

- Methods should be verbs
- It is written in mixed case with the first letter lowercase, with the first letter of each internal word capitalized

Naming Convention: Variables

```
int  
char  
float
```

```
i;  
*cp;  
myWidth;
```

- Variable names should be **short** yet **meaningful**
- The choice of a variable name should be **mnemonic**— that is, designed to indicate to the casual observer the intent of its use
- **One-character** variable names should be **avoided** except for temporary “throwaway” variables

Naming Convention: Constants

```
int MIN_WIDTH = 4;  
int MAX_WIDTH = 999;  
int GET_THE_CPU = 1;
```

- The names of variables declared class constants should be **all uppercase**

Tugas 1

```
1  public class mahasiswa_data {
2
3      public static final int maxnilai = 100;
4
5      private String NAMA_MHS;
6      private int a;
7
8      public void Print_Data() {
9          System.out.println(NAMA_MHS);
10     }
11
12     public void NILAI() {
13         System.out.println(a);
14     }
15 }
```

1. Identifikasi minimal 5 kesalahan naming convention pada kode di atas.
2. Jelaskan alasan mengapa penamaan tersebut tidak sesuai standar Java.
3. Perbaiki seluruh kesalahannya.



Tugas 2

Buat program **Sistem Peminjaman Laptop Kampus**.

Ketentuan:

- Interface: `Borrowable`
- Class: `Laptop`
- Variabel:
 - `laptopBrand`
 - `serialNumber`
 - `borrowerName`
- Konstanta:
 - `MAX_BORROW_DAYS`
 - `LAB_NAME`
- Method:
 - `displayLaptopInfo()`
 - `borrowLaptop()`
 - `returnLaptop()`

Tugas 3

```
+-----+
|                               |
|                               |
+-----+
| - nrp : String               |
| - nama : String              |
| - jurusan : String           |
+-----+
| + tampilData() : void        |
| + ubahJurusan(jurusanBaru:String) : void |
+-----+
```

Implementasikan diagram kelas di atas ke dalam bahasa Java dengan ketentuan:

1. Buat class bernama Mahasiswa.
2. Buat atribut:
 1. nrp
 2. nama
 3. jurusan
3. Buat constructor untuk mengisi data mahasiswa.
4. Buat method:
 1. tampilData()
 2. ubahJurusan(String jurusanBaru)
5. Gunakan Java Naming Convention yang benar.

1. Oracle Java Documentation, The Java™ Tutorials, <https://docs.oracle.com/javase/tutorial/>, Copyright © 1995, Oracle 2015.
2. Tita Karlita, Yuliana Setrowati, Rizky Yuniar Hakkun, Pemrograman Berorientasi Obyek, PENS-2012
3. Sun Java Programming, Sun Educational Services, Student Guide, Sun Microsystems, 2001.
bridge to the future
4. John R. Hubbard, Programming With Java, McGraw-Hill, ISBN: 0-07-142040-1, 2004.
<http://www.eepis-its.edu>
5. Patrick Niemeyer, Jonathan Knudsen, Learning Java, O'reilly, CA, ISBN: 1565927184, 2000.
6. Philip Heller, Simon Roberts, Complete Java 2 Certification Study Guide, Third Edition, Sybex, San Francisco, London, ISBN: 0-7821-4419-5, 2002.
7. Herbert Schildt, The Complete Reference, Java™ Seventh Edition, Mc Graw Hill, Osborne, ISBN: 978-0-07-163177-8, 2007