

Pemrograman Berorientasi Obyek

Class Diagram


Oleh Politeknik Elektronika Negeri Surabaya
2020



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

1

CLASS DIAGRAM



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika & Komputer

2

OOP

tidak mungkin terlepas dari
Class Diagram



CLASS DIAGRAM



Politeknik Elektronika Negeri Surabaya

Apa itu **Class Diagram**?

Diagram yang menggambarkan **Class**

Baris 1:
Nama Class

Baris 2:
Daftar atribut,
variable,
konstanta

Baris 3:
Method
atau fungsi

Class Diagram

Departemen Teknik Informatika & Komputer

5

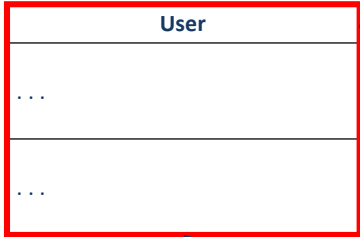
Politeknik Elektronika Negeri Surabaya

CLASS NAME

Departemen Teknik Informatika & Komputer

6

Politeknik Elektronika Negeri Surabaya




The diagram shows a class box for 'User'. The top section contains the name 'User'. Below it, there are two sections separated by horizontal lines, each containing three dots '...' to represent attributes and methods.

Jika di implementasikan dalam Bahasa java:

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

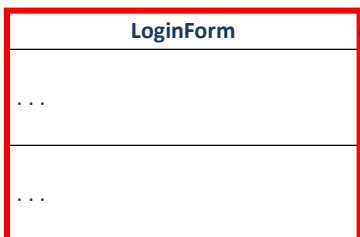
Ini adalah satu buah class dengan nama class : **User**



Departemen Teknik Informatika & Komputer

7

Politeknik Elektronika Negeri Surabaya




The diagram shows a class box for 'LoginForm'. The top section contains the name 'LoginForm'. Below it, there are two sections separated by horizontal lines, each containing three dots '...' to represent attributes and methods.

Jika di implementasikan dalam Bahasa java:

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

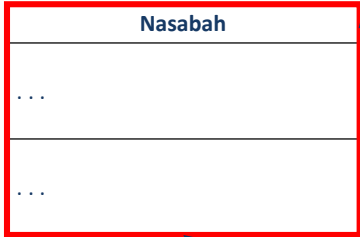
Ini adalah satu buah class dengan nama class : **LoginForm**



Departemen Teknik Informatika & Komputer

8

Politeknik Elektronika Negeri Surabaya




The diagram shows a class box for 'Nasabah' with a header section containing the name and two body sections containing three dots '...'.

Jika di implementasikan dalam Bahasa java:

```
public class Nasabah{  
    ....  
}
```

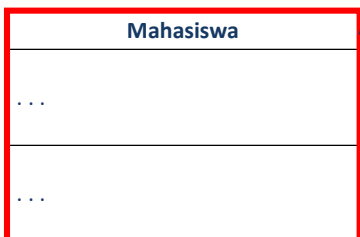
Ini adalah satu buah class dengan nama class : **Nasabah**



Departemen Teknik Informatika & Komputer

9

Politeknik Elektronika Negeri Surabaya




The diagram shows a class box for 'Mahasiswa' with a header section containing the name and two body sections containing three dots '...'.

Jika di implementasikan dalam Bahasa java:

```
...?
```

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



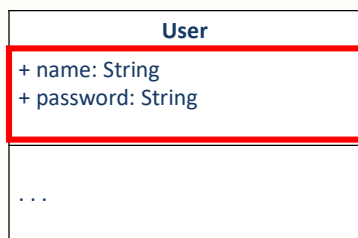
Departemen Teknik Informatika & Komputer

10

ATTRIBUTE



11



Jika di implementasikan dalam Bahasa java:

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

Dalam class User terdapat dua variable : name dan password



12


Politeknik Elektronika Negeri Surabaya

Nasabah
+ saldo: Long
+ id: String
+ rekening: String
...

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



Departemen Teknik Informatika & Komputer

13


Politeknik Elektronika Negeri Surabaya

Mahasiswa
+ nrp: String
+ nama: String
+ jurusan: String
...

Jika di implementasikan dalam Bahasa java:

...?

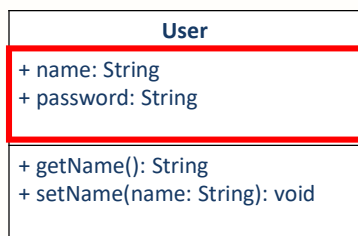
Dalam class Mahasiswa ada 3 variable: nrp, nama, jurusan



Departemen Teknik Informatika & Komputer

14

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){...}
}
  
```



Politeknik Elektronika Negeri Surabaya

Nasabah

+ saldo: Long
+ id: String
+ rekening: String


+ saving(value:Long): void
+ withdraw(value:Long): boolean

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()



Departemen Teknik Informatika & Komputer

17

Politeknik Elektronika Negeri Surabaya

Mahasiswa


+ nrp: String
+ nama: String
+ jurusan: String

+ getNrp(): String
+ setName(nama:String): void

Jika di implementasikan dalam Bahasa java:

...?

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



Departemen Teknik Informatika & Komputer

18

MODIFIER



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



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



CLASS DIAGRAM RELATIONSHIP

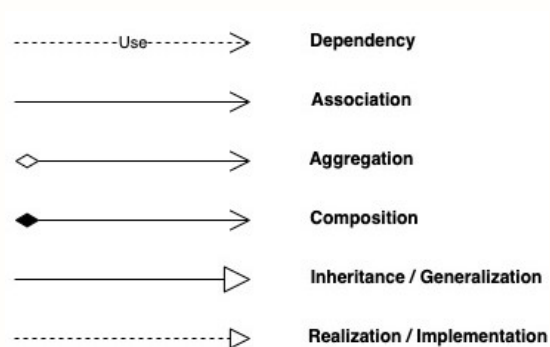


Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika & Komputer

21

Politeknik Elektronika Negeri Surabaya

CLASS DIAGRAM RELATIONSHIP



Departemen Teknik Informatika & Komputer

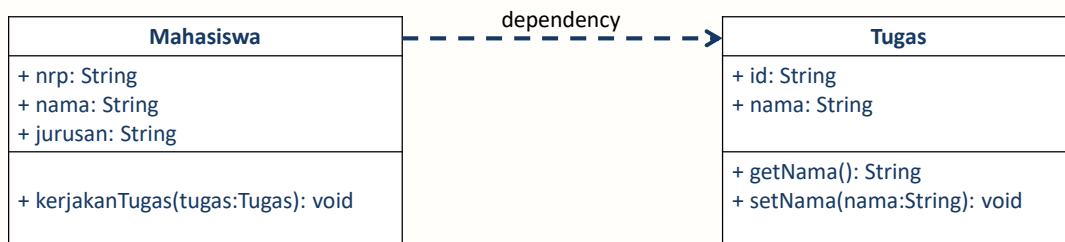
22

CLASS DIAGRAM RELATIONSHIP

Dependency



23



```

public class Mahasiswa{
    public String nrp;
    public String nama;
    public String jurusan;

    public void kerjakanTugas(Tugas tugas){
        ...
    }
}
  
```

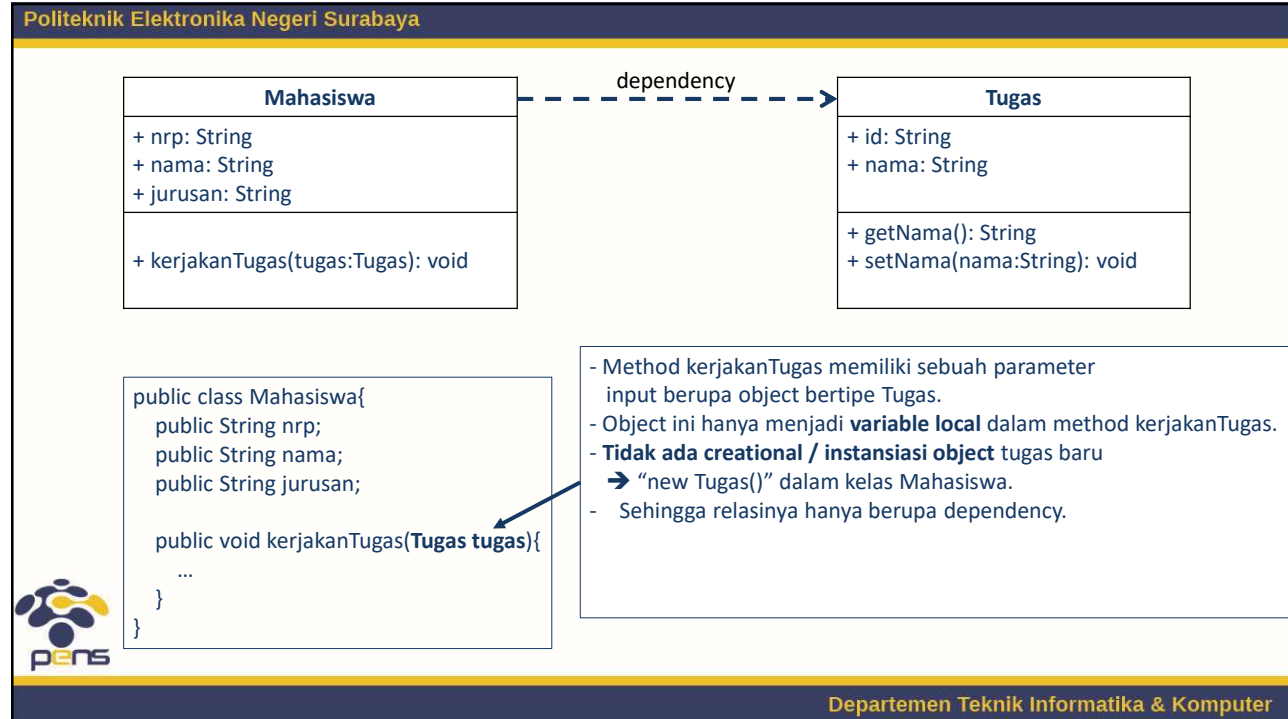
```

public class Tugas{
    public String id;
    public String nama;

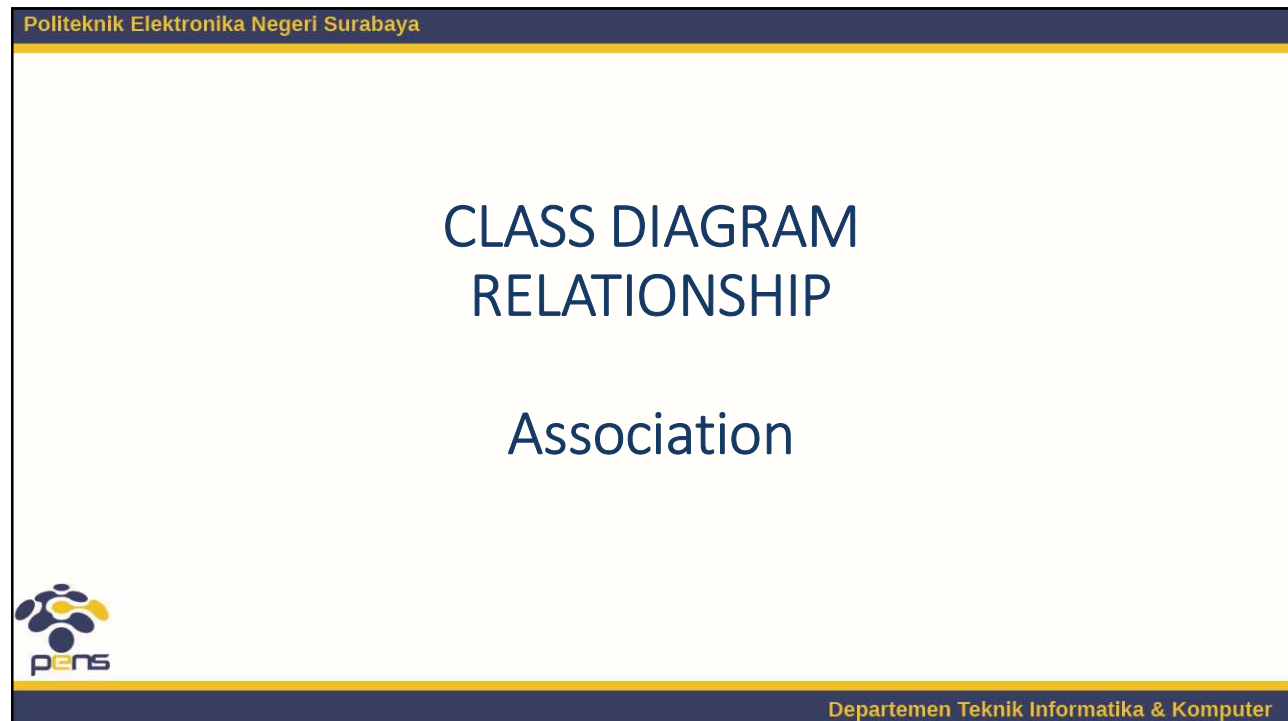
    public String getNama(){ ... }
    public void setNama(String nama){ ... }
}
  
```



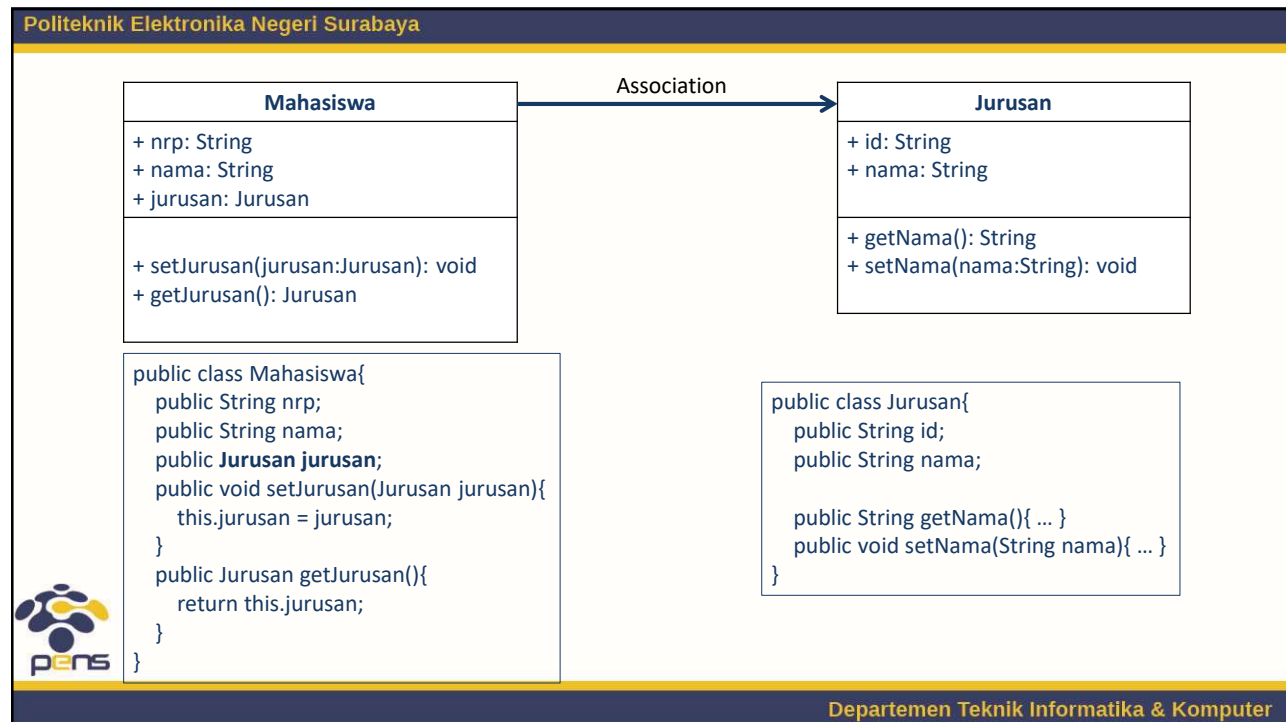
24



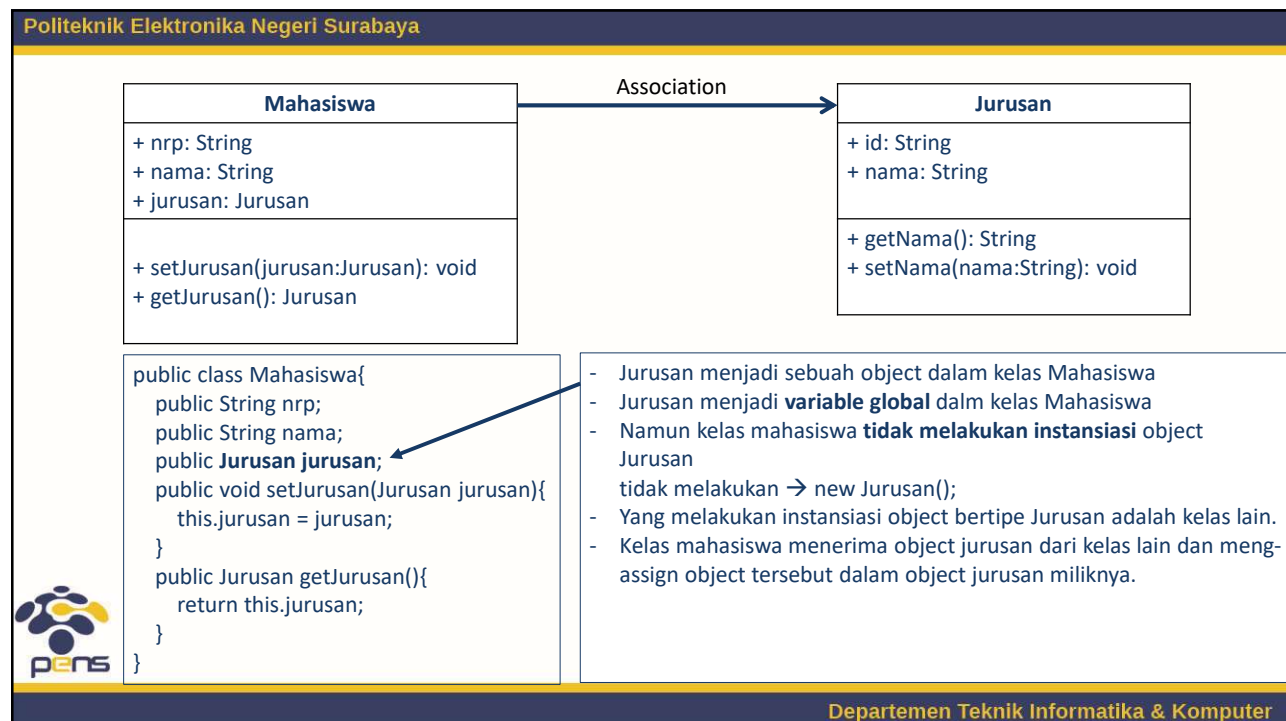
25



26



27



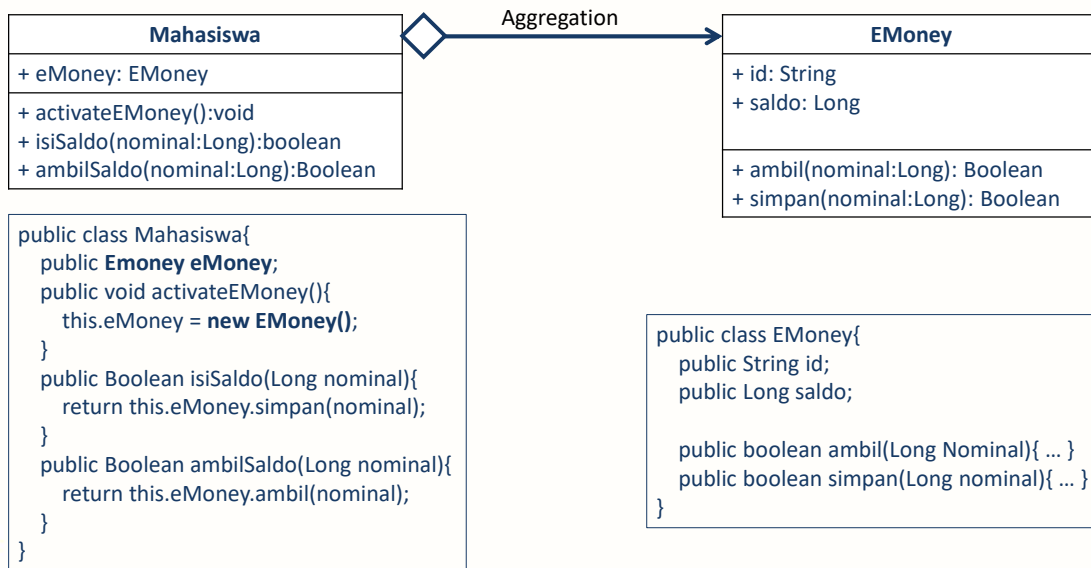
28

CLASS DIAGRAM RELATIONSHIP

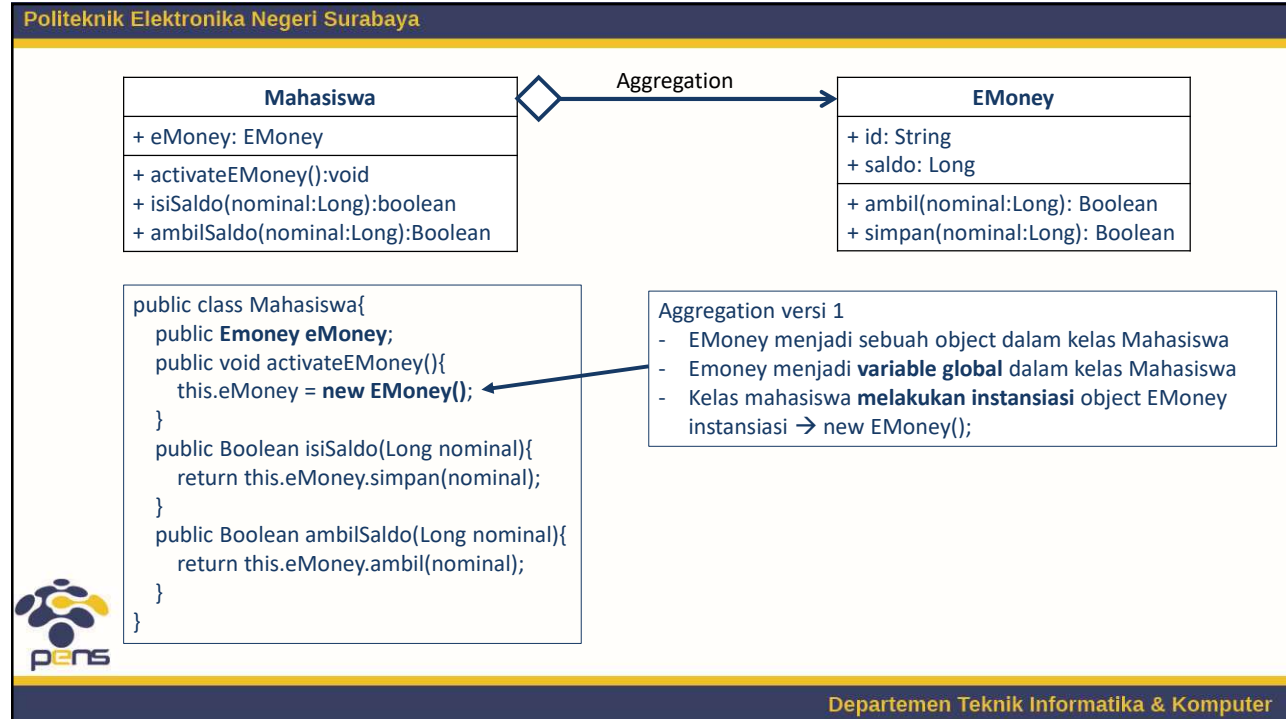
Aggregation



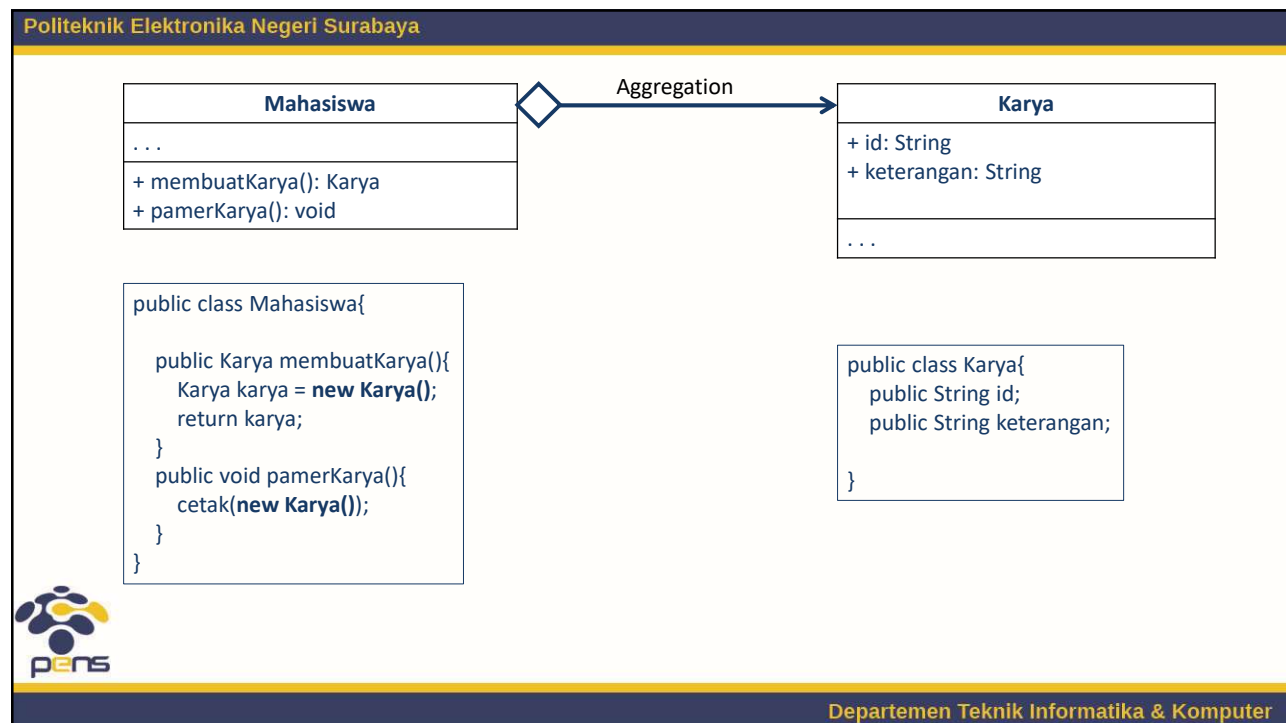
29



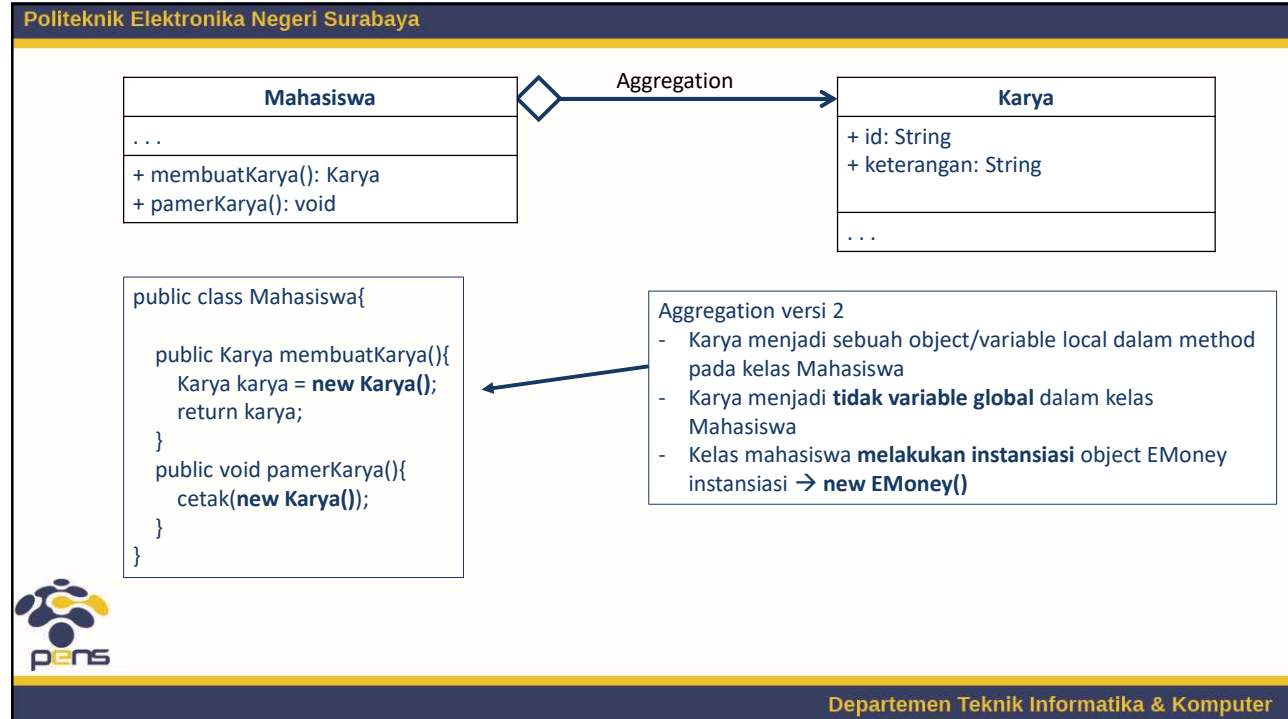
30



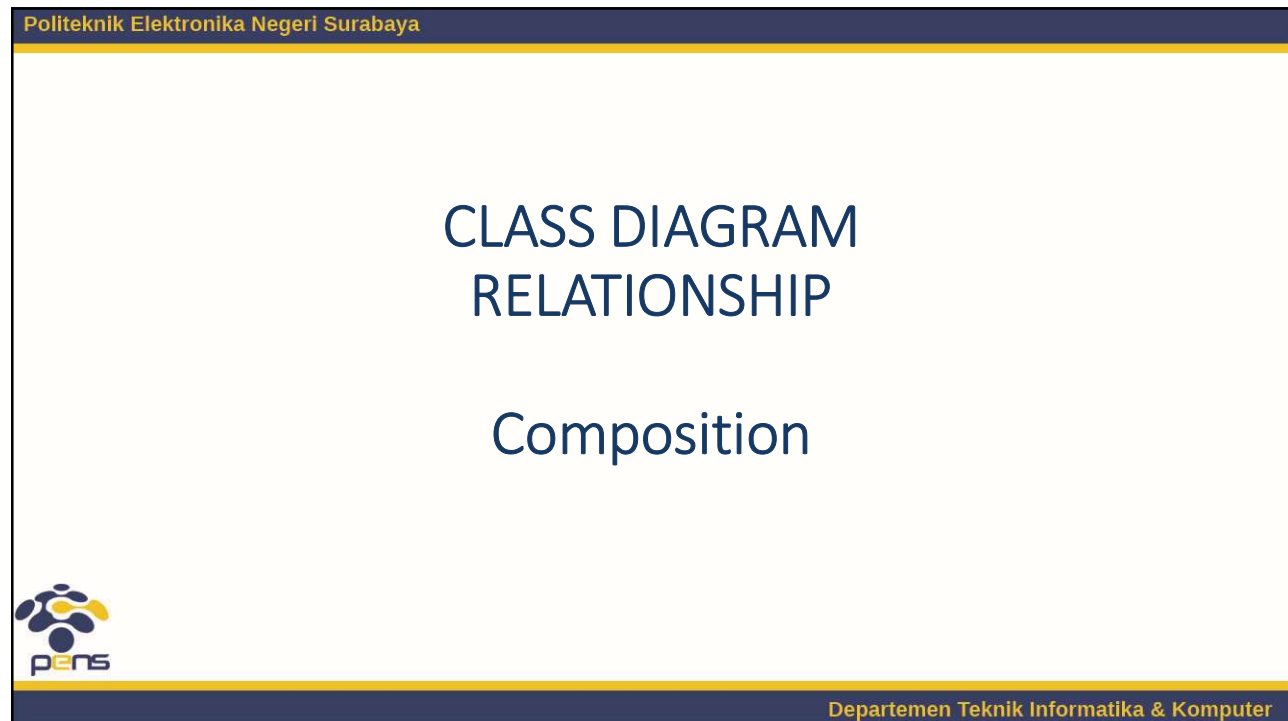
31



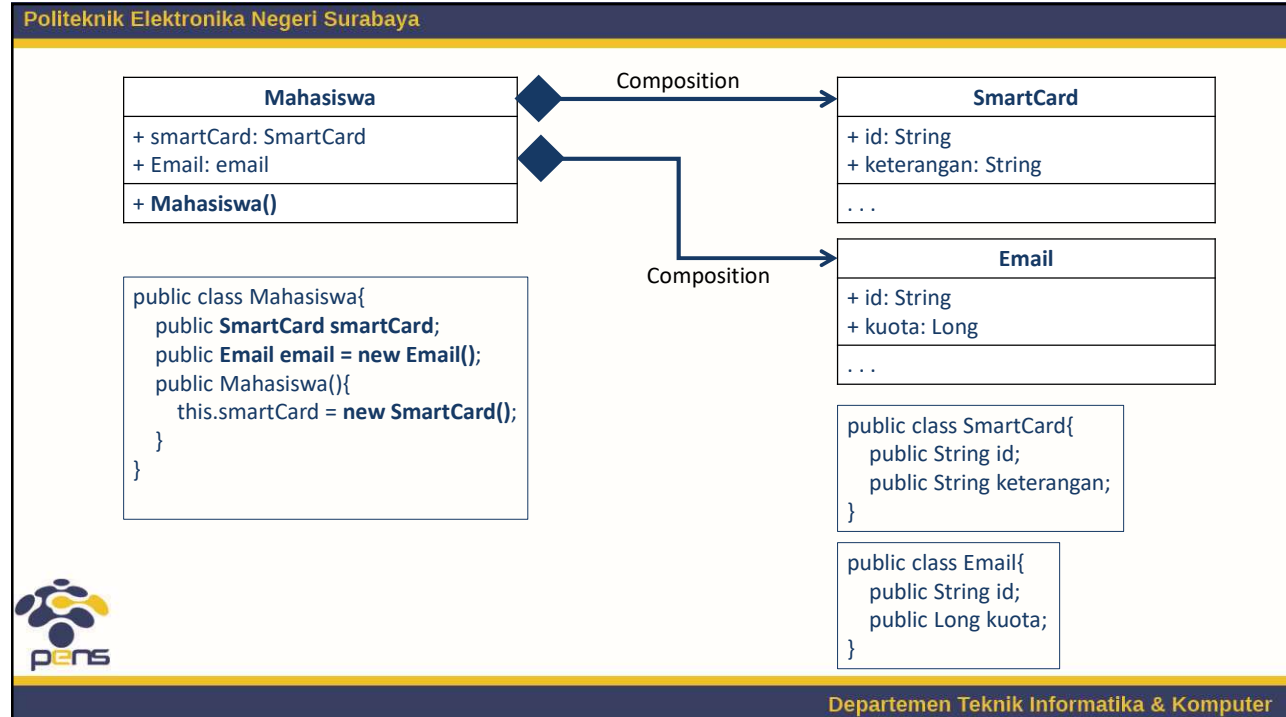
32



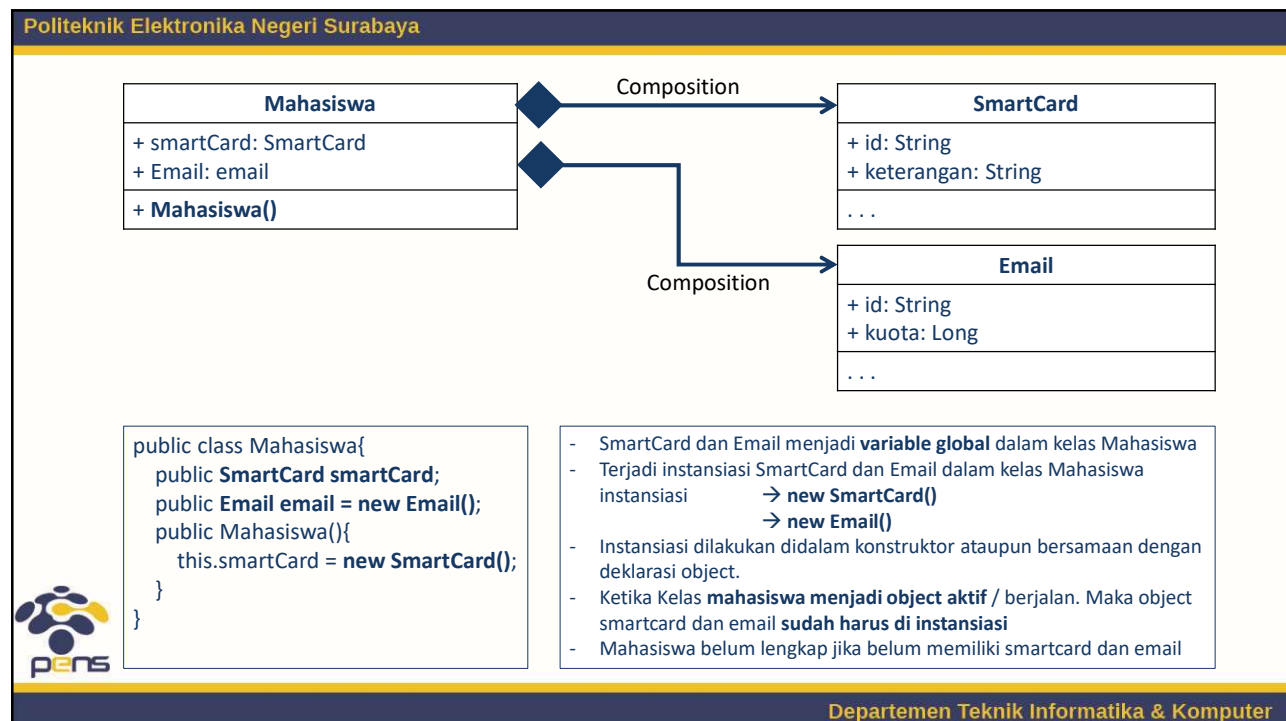
33



34



35



36

CLASS DIAGRAM RELATIONSHIP

Inheritance / Extends



37

Inheritance

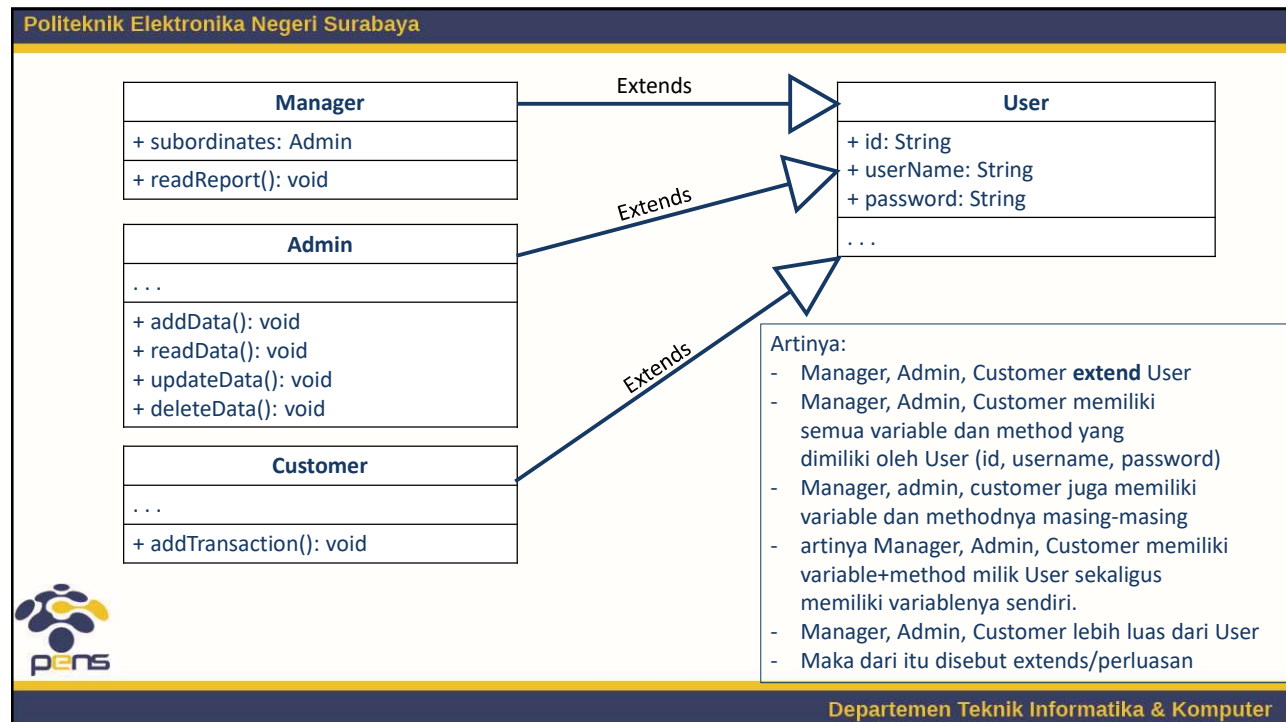
- **Inheritance** adalah pewarisan antar class parent terhadap child
- Disebut juga **Extends**
- Yang diwariskan adalah **semua variable dan method**
- Yang **tidak diwariskan** adalah **constructor**, variable dan method dengan modifier **private** atau **default(unspecified)**
- Dalam class diagram ditandai dengan garis relationship sebagai berikut:



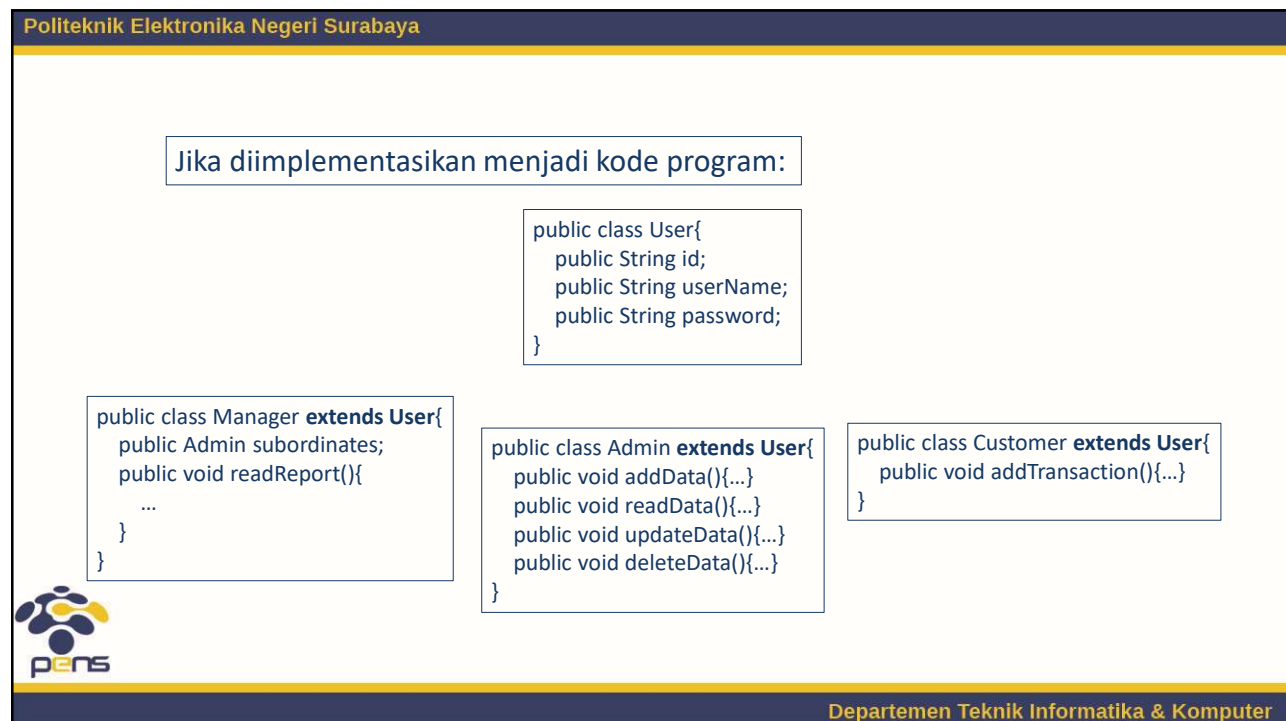
- Jika class Child inherit terhadap class Parent maka child memiliki semua variable dan method yang dimiliki parent kecuali yang private dan default(unspecified) serta constructor.



38



39



40

CLASS DIAGRAM RELATIONSHIP

Realization



Realization

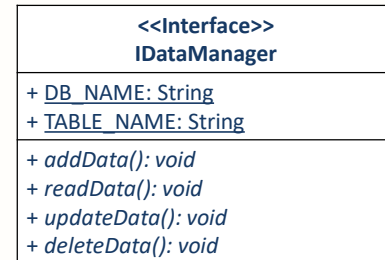
- Realization terjadi antara **Class** dengan **Interface**.
- Sama dengan class, **Interface dapat memiliki variables dan methods**.
- Namun Interface hanya bisa memiliki **method Abstract**.
- Method abstract adalah method kosong tanpa method body. Method ini hanya deklarasi saja. Contoh:


```
public int getData();
public void addData();
public Boolean edit(int data);
```
- Variable yang dideklarasikan dalam Interface harus **public, static & final**

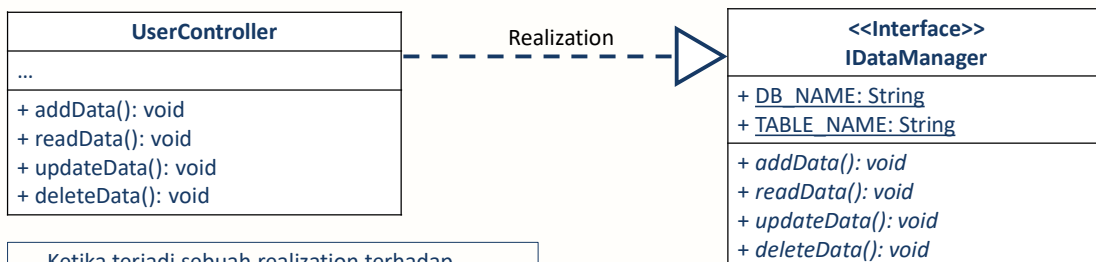


Realization

- Menggambarkan Interface dalam class diagram sedikit berbeda dengan class yaitu dengan menambahkan kata interface diatas nama interface
- variable static dituliskan dengan tanda underline/garis bawah
- Variable final dituliskan dengan huruf Capital semua. Dengan pemisah antar kata menggunakan underscore
- Method abstract ditandai dengan tulisan italic/miring



43

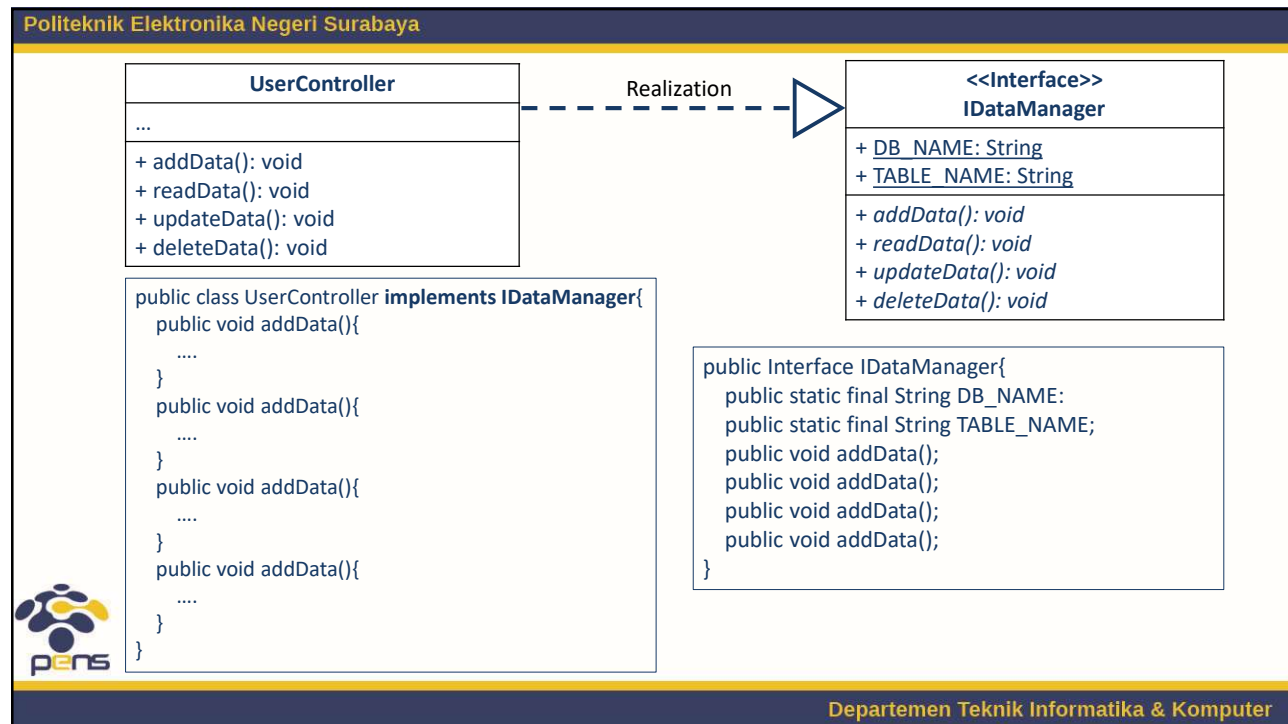


- Ketika terjadi sebuah realization terhadap sebuah interface, maka class yang merealisasi wajib mengimplementasikan semua method abstract yang dimiliki oleh Interface.
- UserController wajib merealisasikan method *addData()*, *readData()*, *updateData()*, *deleteData()*
- Method yang ditulis miring/italic menunjukkan bahwa itu adalah method abstract. Sedangkan method yang ditulis tegak/normal menunjukkan itu method concrete/nyata.

- Jika UserController tidak mengimplementasikan semua method abstract milik Interface IDataManager, maka UserController juga harus menjadi Interface atau Abstract Class.



44



45



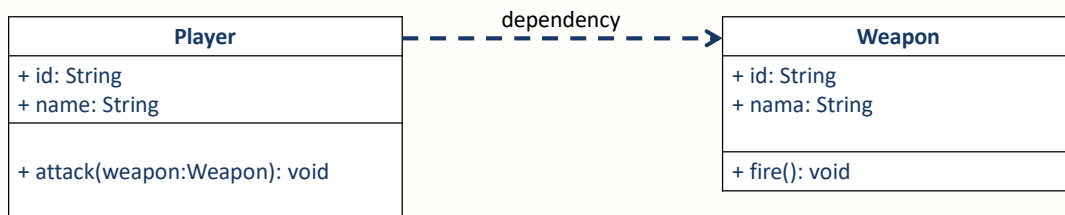
46

CLASS DIAGRAM

LATIHAN



47



Implementasikan class diagram tersebut menjadi code program



48

Gambarkan class diagram dari code program berikut

```
public class Car{
    public String id;
    public String type;

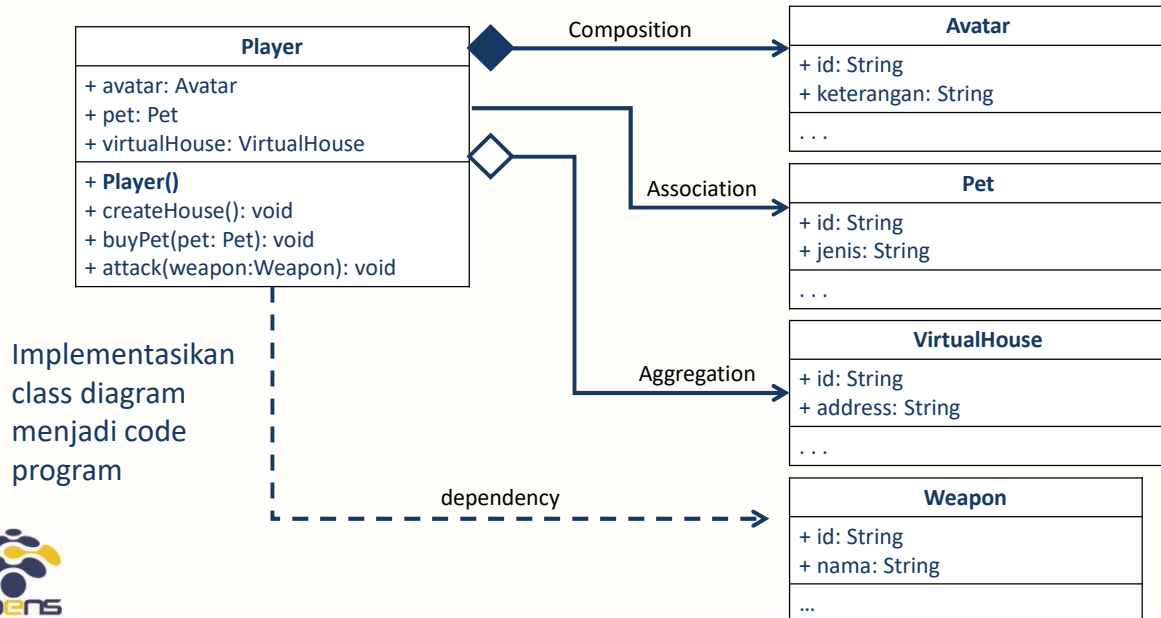
    public void drive(Gasoline gasoline){
        ...
    }
}
```

```
public class Gasoline{
    public Long octane;
    public Long volume;

    public String runEngine(){ ... }
}
```

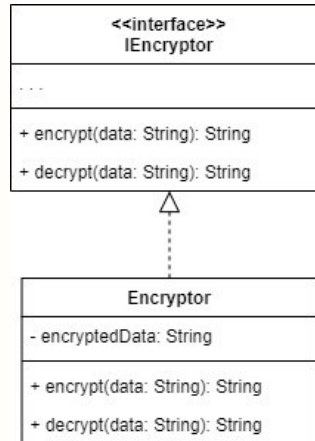


49



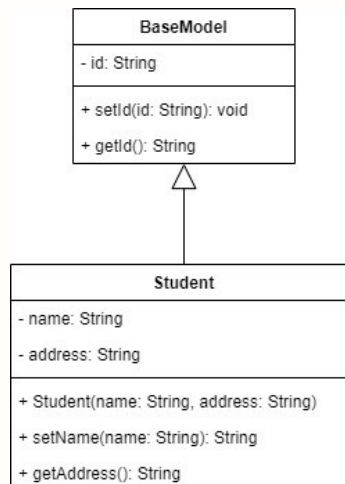
50

Implementasikan
menjadi kode program



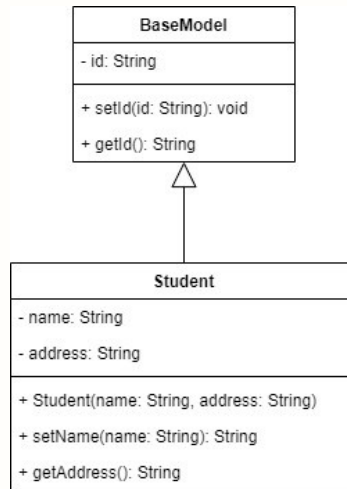
51

Implementasikan
menjadi kode program



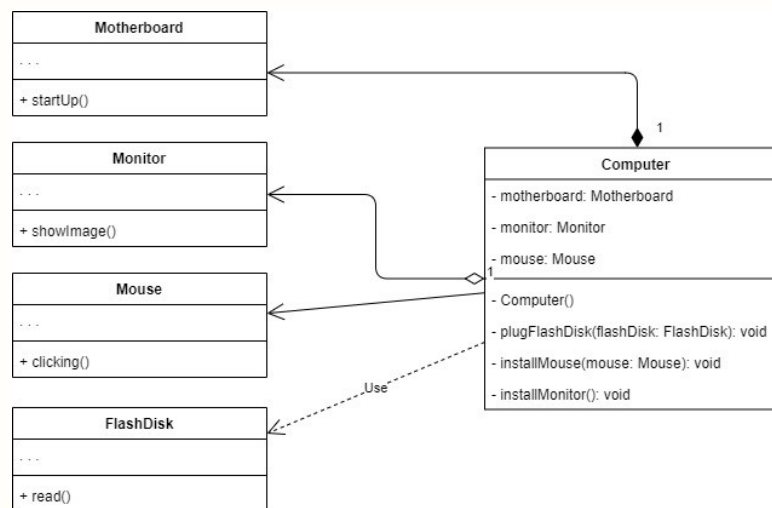
52

Implementasikan
menjadi kode program



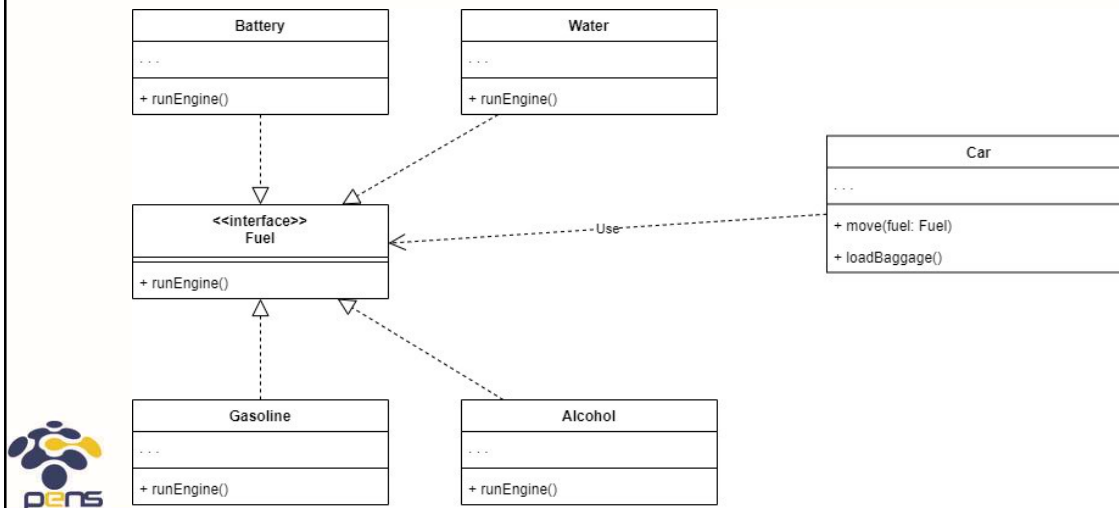
53

Implementasikan menjadi kode program



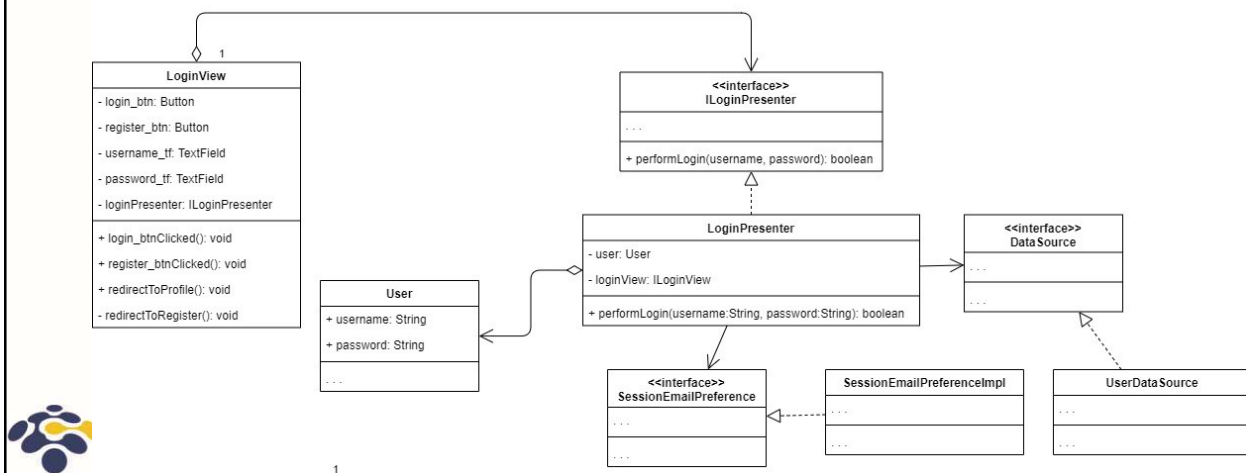
54

Implementasikan menjadi kode program



55

Implementasikan menjadi kode program



56

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.
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