

# BAB 5

## MENGELOLA CLASS

### A. POKOK BAHASAN

- ⊕ Package
- ⊕ Import class
- ⊕ Kata kunci *this*

### B. TUJUAN BELAJAR

Dengan praktikum ini mahasiswa diharapkan dapat:

- ✓ Memahami konsep package dan import
- ✓ Menggunakan kata kunci *this*

### C. DASAR TEORI

#### ➤ Package

Package adalah suatu cara untuk memenej class-class yang kita buat package akan sangat bermanfaat jika class-class yang kita buat sangat banyak sehingga perlu dikelompokan berdasarkan kategori tertentu. Yang harus diperhatikan adalah biasanya nama class utama harus sama dengan nama package. Berikut ini adalah sintak pernyataan package untuk meletakkan hasil kompilasi sebuah kelas ke dalam paket :

```
package nama-paket;  
<identifier> class <nama_class sama dengan package>  
Contoh : package Utama;  
public class Utama{ }
```

#### ➤ Import Class

Perintah import digunakan untuk memberitahukan kepada program untuk mengacu pada class-class yang terdapat dalam package tertentu, buka menjalankan class-class tersebut.

Deklarasi import class :

```
import <nama_package>.*;  
Contoh : import matematik.*;
```

Sedangkan untuk mengimpor class tertentu anda dapat menuliskan nama class setelah nama package. Deklarasi import :

```
import <nama_package>.<nama_class>;  
Contoh : import matematik.bilangan;
```

Contoh penggunaan import (dengan implementasinya):

```
import matematik.bilangan;  
class Utama {  
    public static void main(String[] args) {  
        Bilangan x = new Bilangan(); x.setDesimal(-  
        44); System.out.println("Bilangan biner  
        dari "+  
        x.getDesimal() + " adalah "+ x.biner()); } }
```

➤ **Kata kunci *this***

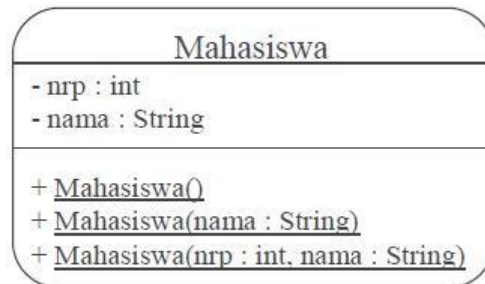
Kata kunci ini digunakan dalam sebuah kelas untuk menyatakan object sekarang. Kata kunci *this* sangat berguna untuk menunjukkan suatu member dalam class-nya sendiri. *this* dapat digunakan baik untuk data member maupun untuk function member, serta dapat juga digunakan untuk konstruktor. Adapun format penulisannya adalah:

<b>this.data_member</b>	merujuk pada data member
<b>this.function_member()</b>	merujuk pada function member
<b>this()</b>	merujuk pada konstruktor

## D. PERCOBAAN

### • Percobaan 1

Memakai kata kunci *this* pada overloading constructor

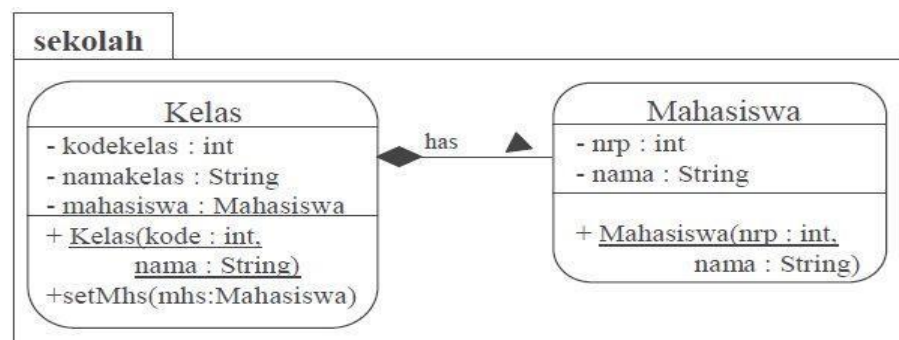


Dari class diagram tersebut, dapat diimplementasikan ke dalam program sebagai berikut :

```
public class Mahasiswa {
    private int nrp;
    private String nama;
    public Mahasiswa() {
        this(0, "");
    }
    public Mahasiswa(String nama) {
        this(0, nama);
    }
    public Mahasiswa(int nrp, String nama) {
        this.nrp=nrp;
        this.nama=nama; }
}
```

### • Percobaan 2

Menggunakan Package dan import



Dari class diagram tersebut, dapat diimplementasikan ke dalam program dibawah ini. Sebelum melakukan kompilasi, daftarkan direktori tempat package diatas disimpan.

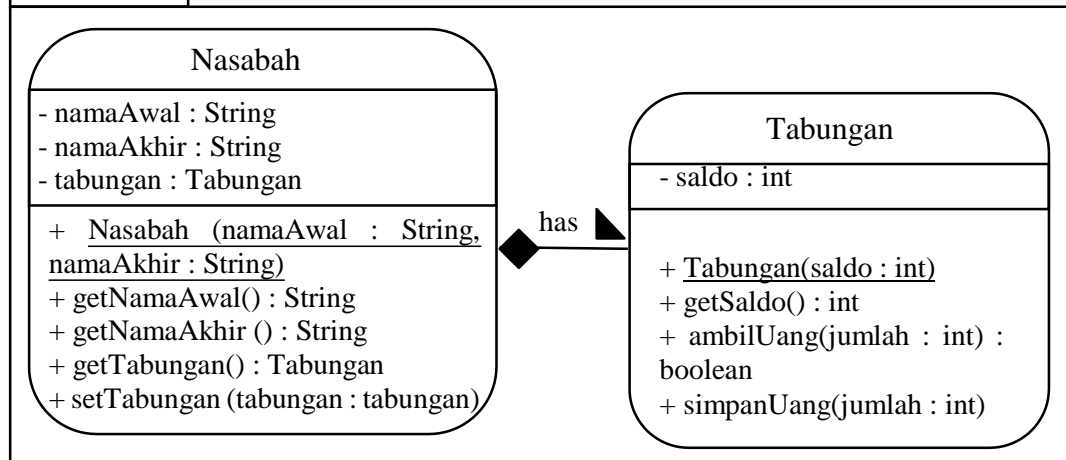
<pre>package sekolah; public class Kelas { private int kodekelas; private String namakelas; private Mahasiswa mahasiswa; public Kelas(int kode, String nama) { this.kodekelas=kode; this.namakelas=nama; } public void setMhs (Mahasiswa mhs) { this.mahasiswa=mhs;} }</pre>	<pre>package sekolah; public class Mahasiswa { private int nrp; private String nama; public Mahasiswa(int nrp, String nama) { this.nrp=nrp; this.nama=nama; } }</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

## E. LATIHAN

### Mengimplementasikan UML class diagram dalam program untuk package perbankan

Transformasikan class diagram diatas ke dalam bentuk program! Tulislah listing program berikut ini sebagai pengetesan.

## Perbankan



```

import perbankan.*;

private class TesLatihan {
    private static void main(String args[]) {
        int tmp;
        boolean status;
        Nasabah nasabah=new Nasabah("Agus","Daryanto");
        System.out.println("Nasabah atas nama : " +
        nasabah.getNamaAwal() + " " + nasabah.getNamaAkhir());
        nasabah.setTabungan(new Tabungan(5000));
        tmp=nasabah.getTabungan().getSaldo();
        System.out.println("Saldo awal : " + tmp);
        nasabah.getTabungan().simpanUang(3000);
        System.out.println("Jumlah uang yang disimpan : 3000");
        status=nasabah.getTabungan().ambilUang(6000);
        System.out.print("Jumlah uang yang diambil : 6000");
        if (status) System.out.println(" ok");
        else
        System.out.println(" gagal");
        nasabah.getTabungan().simpanUang(3500);
        System.out.println("Jumlah uang yang disimpan : 3500");
        status=nasabah.getTabungan().ambilUang(4000);
        System.out.print("Jumlah uang yang diambil : 4000");
        if (status) System.out.println(" ok");
        else
        System.out.println(" gagal");
        status=nasabah.getTabungan().ambilUang(1600);
        System.out.print("Jumlah uang yang diambil : 1600");
        if (status) System.out.println(" ok");
        else
        System.out.println(" gagal");
        nasabah.getTabungan().simpanUang(2000);
        System.out.println("Jumlah uang yang disimpan : 2000");
        tmp=nasabah.getTabungan().getSaldo();
        System.out.println("Saldo sekarang = " + tmp);} }
  
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Nasabah atas nama : Agus Daryanto
Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000 ok
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000 ok
Jumlah uang yang diambil : 1600 gagal
Jumlah uang yang disimpan : 2000
Saldo sekarang = 3500
```