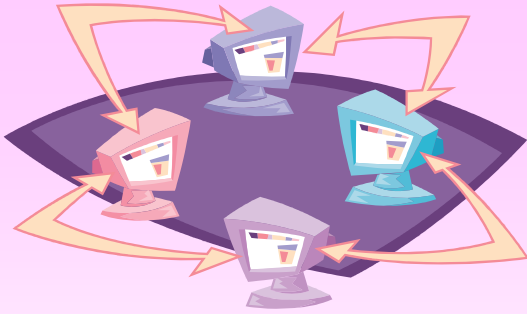


UML Class Diagram

Tessy Badriyah, SKom. MT.

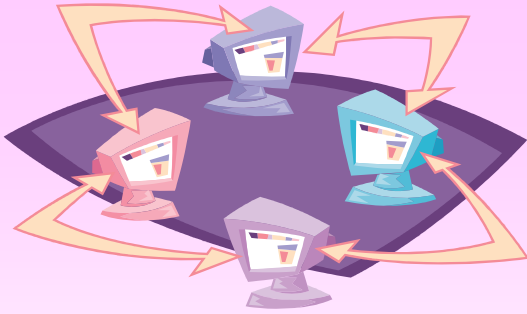
<http://lecturer.eepis-its.edu/~tessy>





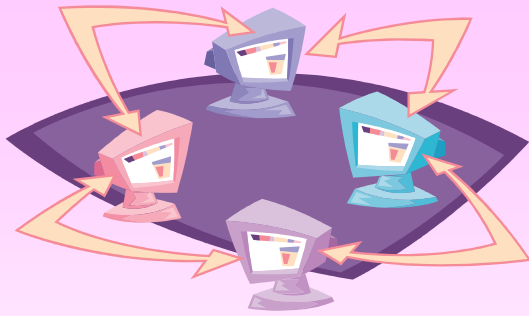
Tujuan Pembelajaran

- Memahami Definisi dari Unified Modelling Language (UML) Diagram
- Memahami tipe dari UML Diagram
- Memahami penggunaan salah satu UML Diagram yaitu Class Diagram untuk Pemrograman Berbasis Obyek



Apa itu UML ?

- The Unified Modeling Language (UML) adalah bahasa standart untuk melakukan spesifikasi, visualisasi, konstruksi, dan dokumentasi dari komponen-komponen perangkat lunak, dan digunakan untuk pemodelan bisnis.
- UML menggunakan notasi grafis untuk menyatakan suatu desain
- Pemodelan dengan UML berarti menggambarkan yang ada dalam dunia nyata ke dalam bentuk yang dapat dipahami dengan menggunakan notasi standart UML



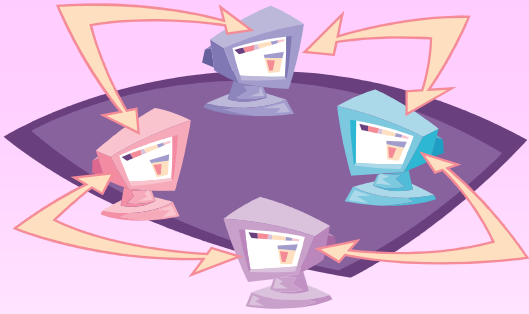
UML Diagram

- Pemodelan dengan UML terdiri dari 8 tipe diagram yang berbeda untuk memodelkan sistem perangkat lunak.
- Masing-masing diagram UML didesain untuk menunjukkan satu sisi dari bermacam-macam sudut pandang (perspektif) dan terdiri dari tingkat abstraksi yang berbeda



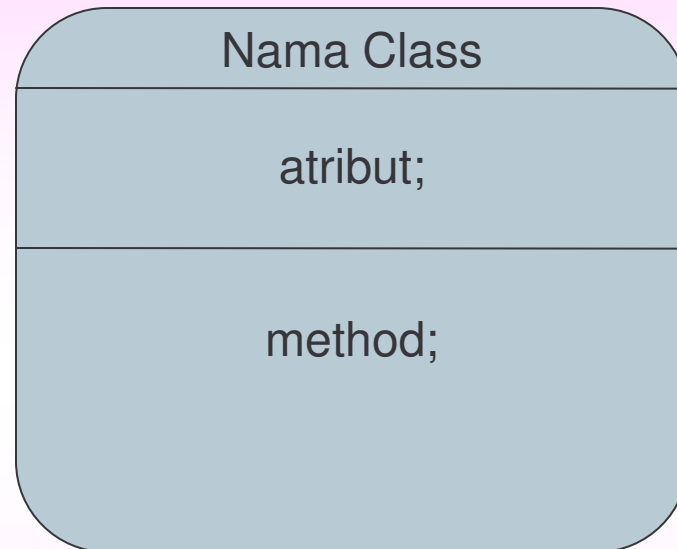
Ke-8 Diagram UML

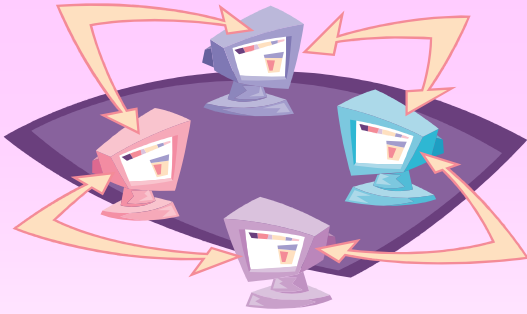
- Use case Diagram
- Class Diagram
- Object Diagram
- State Diagram
- Activity Diagram
- Sequence Diagram
- Collaboration diagram
- Component diagram
- Deployment diagram
 - pada mata kuliah pemrograman berbasis obyek ini kita hanya menggunakan Class Diagram



Class Diagram

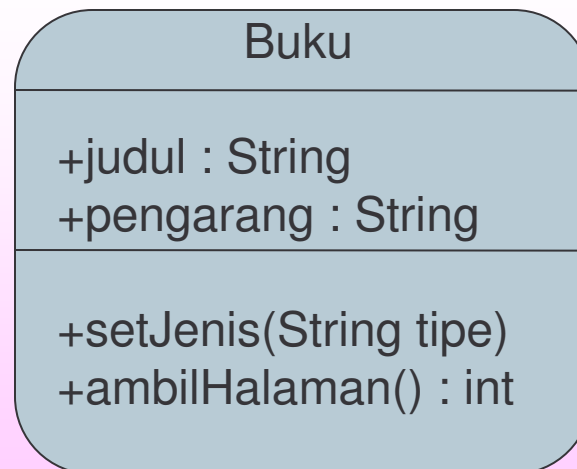
- Bentuk :

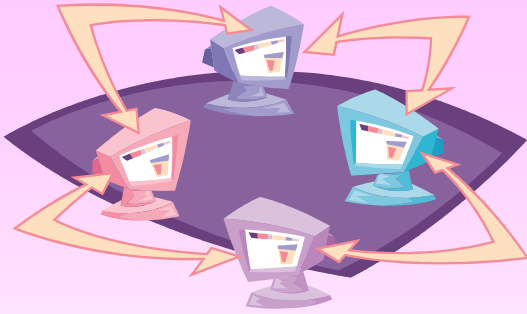




Contoh Class Diagram

- Class Buku terdiri dari :
 - Atribut : judul, pengarang
 - Method : ambilJenis(), ambilHalaman();
- Penggambaran dalam Class Diagram :

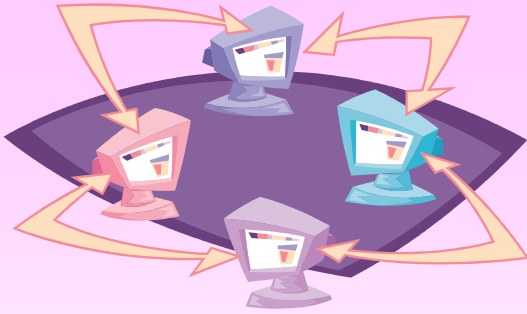




Percobaan 1

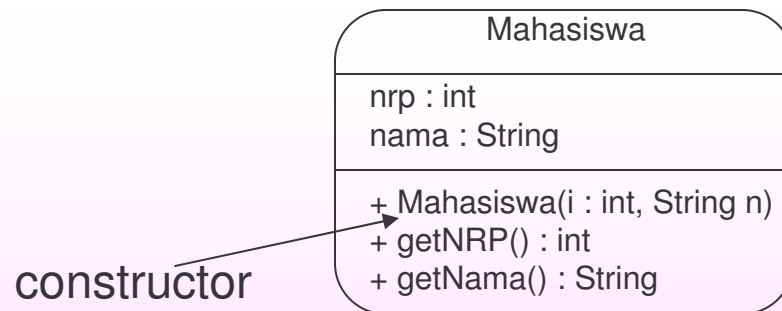
- Mengimplementasikan Class Diagram Buku ke dalam program
- Keterangan : untuk judul buku “Pemrograman Java” jenisnya = “Prog” dan jumlah halaman = 100;

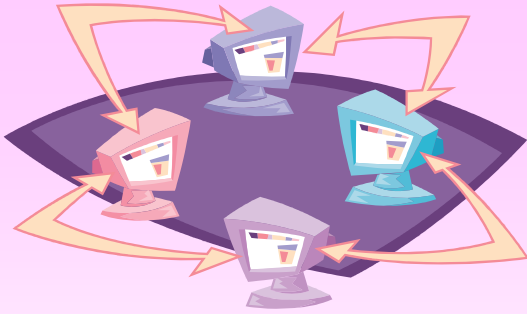
```
Buku.java  
  
public class Buku {  
    String judul, pengarang;  
    public void setJenis(String tipe) {  
        if (judul=="Pemrograman Java")  
            jenis="Prog";  
    }  
    public int ambilHalaman() {  
        if (judul=="Pemrograman Java")  
            return 100;  
    }  
}
```

Class Diagram dengan Constructor

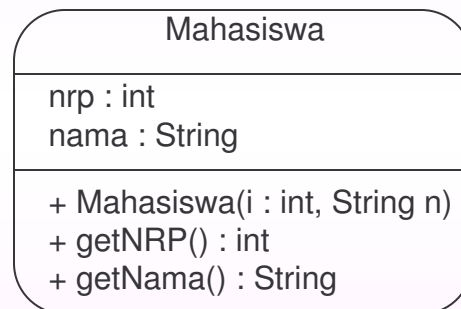
- Class diagram dengan constructor mempunyai method yang sama dengan nama class-nya.

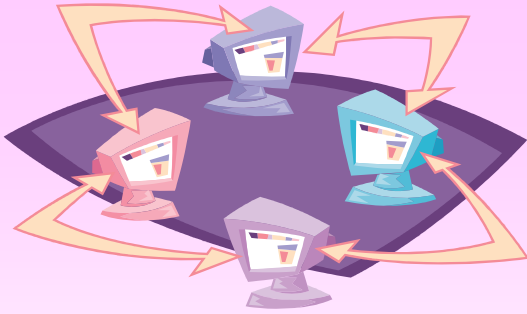




Percobaan 2

- Class Diagram Mahasiswa dengan constructor

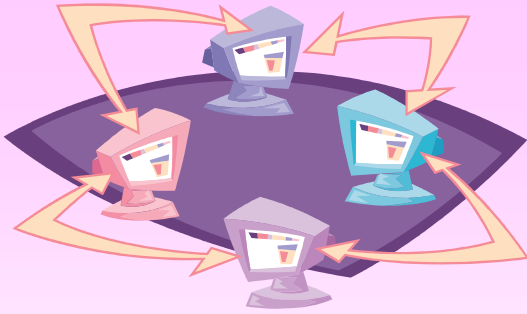




Percobaan 2

```
Mahasiswa.java
public class Mahasiswa {
    public int nrp;
    public String nama;
    public Mahasiswa(int i, String n) {
        this.nrp=i;
        this.nama=n;
    }
    public int getNRP() {
        return nrp;
    }
    public String getNama() {
        return nama;
    }
}

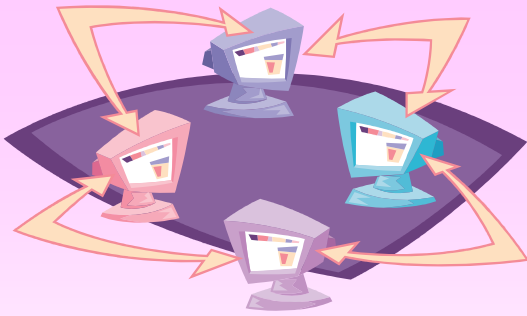
class TesMahasiswa {
    public static void main(String [] args) {
        Mahasiswa siswa = new Mahasiswa(123,"SAYA");
        System.out.println("NRP    = "+siswa.getNRP());
        System.out.println("Nama   = "+siswa.getNama());
    }
}
```



Percobaan 3

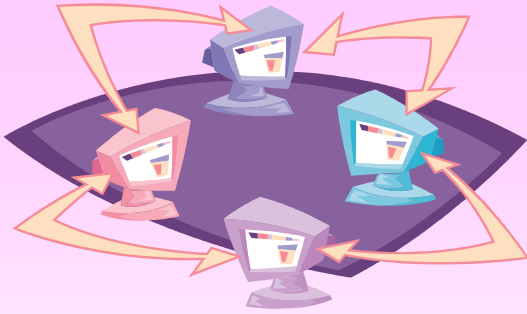
- Class Diagram Tanggal dengan constructor





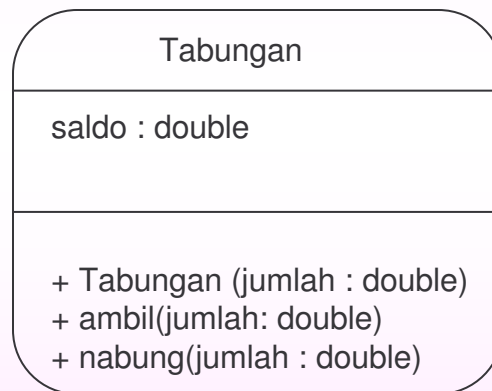
Percobaan 3

```
Tanggal.java
public class Tanggal {
    public int tgl;
    public int bulan;
    public int tahun;
    public Tanggal(int tgl, int bulan, int tahun) {
        this.tgl=tgl;
        this.bulan=bulan;
        this.tahun=tahun;
    }
}
class TesTanggal {
    public static void main(String [] args) {
        Tanggal tanggalku=new Tanggal(3,5,2007);
        System.out.println(tanggalku.tgl);
        System.out.println(tanggalku.bulan);
        System.out.println(tanggalku.tahun);
    }
}
```



Percobaan 4

- Class Diagram Tabungan



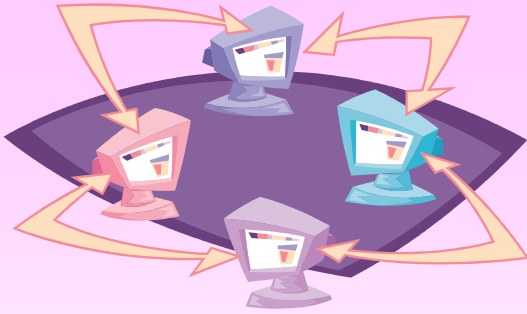


Percobaan 4

```
Tabungan.java
public class Tabungan {
    public double saldo;
    public Tabungan(double jumlah) {
        saldo=jumlah;
    }
    public void ambil(double jumlah) {
        saldo=saldo-jumlah;
    }
    public void nabung(double jumlah) {
        saldo=saldo+jumlah;
    }
}
class TesTabungan {
    public static void main(String [] args) {
        Tabungan rekeningku = new Tabungan(1000000);
        System.out.println("Jumlah saldo : "+rekeningku.saldo);
        rekeningku.ambil(800000);
        System.out.println("Jumlah saldo : "+rekeningku.saldo);
        rekeningku.nabung(100000);
        System.out.println("Jumlah saldo : "+rekeningku.saldo);
    }
}
```

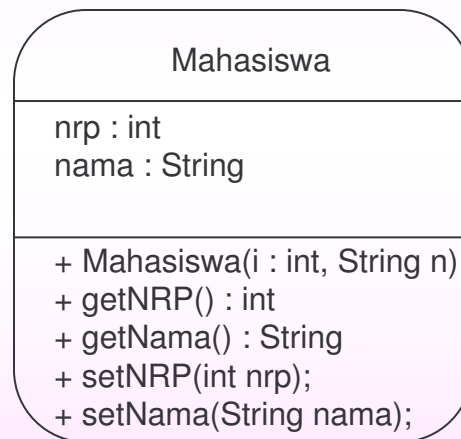
TUGAS



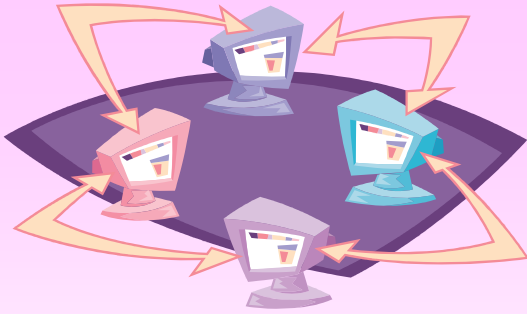


Tugas 1

- Implementasikan Perluasan Class Diagram Mahasiswa seperti berikut :

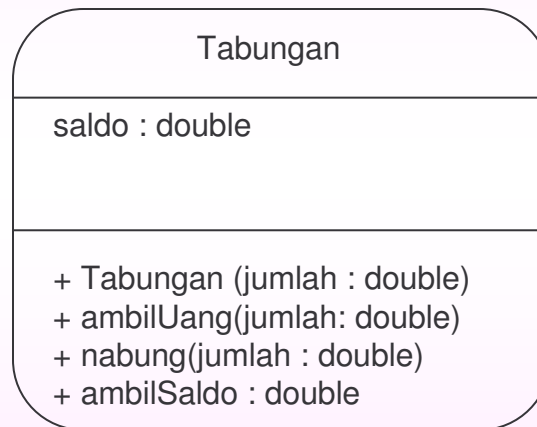


- Buat program untuk menguji Class yang telah dibuat



Tugas 2

- Implementasikan Class Diagram Tabungan seperti berikut :



- Buat program untuk menguji Class yang telah dibuat.

Selesai

4-5-2007

