



# LAPORAN RESMI

*PRAKTIKUM PEMROGRAMAN  
BERBASIS OBJEK*

## Enkapsulasi

Fandi Ahmad  
2110171033

## Percobaan

### 1. Percobaan 1 : Melakukan enkapsulasi pada suatu class

#### Listing Program

```
public class Mahasiswa{  
    private int nrp;  
    private String nama;  
  
    public Mahasiswa(int nrp, String nama){  
        this.nrp = nrp;  
        this.nama = nama;  
    }  
  
    public int getNrp(){  
        return nrp;  
    }  
  
    public String getNama(){  
        return nama;  
    }  
  
    public void setNrp(int nrp){  
        this.nrp = nrp;  
    }  
  
    public void setNama(String nama){  
        this.nama = nama;  
    }  
}
```

**Mahasiswa.java**

```
import java.util.Scanner;  
  
public class Percobaan1 {  
    public static void main(String[] a){  
        int nrp;  
        String nama;  
        Mahasiswa mhs = new Mahasiswa(0, "");  
        Scanner scan = new Scanner(System.in);  
  
        System.out.print("Masukan NRP : ");  
        nrp = Integer.parseInt(scan.nextLine());  
        mhs.setNrp(nrp);  
  
        System.out.print("Masukan Nama : ");  
        nama = scan.nextLine();  
        mhs.setNama(nama);  
    }  
}
```

**Percobaan1.java**

```
        System.out.println(mhs.getNrp() + " " + mhs.getNama());
    }
}
```

#### Capture

```
fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah
1
$ javac Mahasiswa.java Percobaan1.java
fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah
1
$ java Percobaan1
Masukan NRP : 21133
Masukan Nama : Fandi Ahmad
21133 Fandi Ahmad
```

#### Analisa

Dalam percobaan diatas, melakukan penerapan enkapsulasi dari UML diagram yang belum menerapkan enkapsulasi lalu kita mengubah UML diagram tersebut dan menerapkan enkapsulasi.

## 2. Percobaan 2 : Melakukan overloading constructor

#### Listing Program

```
public class Mahasiswa{
    private int nrp;
    private String nama;

    public Mahasiswa(){
        nrp = 0;
        nama = "";
    }

    public Mahasiswa(String nama){
        nrp = 0;
        this.nama = nama;
    }

    public Mahasiswa(int nrp, String nama){
        this.nrp = nrp;
        this.nama = nama;
    }

    public void getData(){
        System.out.println("NRP : " + nrp);
        System.out.println("Nama : " + nama);
    }
}
```

```
public class Percobaan2{

    public static void main(String[] a){

        Mahasiswa mhs1 = new Mahasiswa();
        Mahasiswa mhs2 = new Mahasiswa("Fandi");
        Mahasiswa mhs3 = new Mahasiswa(36, "Ananda");

        mhs1.getData();
        mhs2.getData();
        mhs3.getData();

    }

}
```

## Capture

```
fndiamd on X555QG in /media/fndiamd/Fandi/Kulial
2
$ javac Mahasiswa.java Percobaan2.java
fndiamd on X555QG in /media/fndiamd/Fandi/Kulial
2
$ java Percobaan2
NRP : 0
Nama :
NRP : 0
Nama : Fandi
NRP : 36
Nama : Ananda
```

## Analisa

Dalam percobaan diatas, menerapkan overloading constructor dimana overloading constructor memungkinkan kita mempunyai lebih dari 1 constructor yang dapat kita sesuaikan dengan keinginan kita, ketika pembuatan object dari sebuah class tersebut kita bisa sesuaikan akan memanggil constructor yang mana.

## Latihan

1. **Latihan 1** : Mengimplementasikan UML class diagram dalam program untuk class Kalender

### Listing Program

```
public class Kalender{
    private int tanggal, bulan, tahun;

    public Kalender(int tanggal){
        this.tanggal = tanggal;
        bulan = 1;
        tahun = 2000;
    }

    public Kalender(int bulan, int tahun){
        this.bulan = bulan;
        this.tahun = tahun;
        tanggal = 1;
    }

    public Kalender(int tanggal, int bulan, int tahun){
        this.tanggal = tanggal;
        this.bulan = bulan;
        this.tahun = tahun;
    }

    public void setTanggal(int tanggal){
        this.tanggal = tanggal;
    }

    public void setBulan(int bulan){
        this.bulan = bulan;
    }

    public void setTahun(int tahun){
        this.tahun = tahun;
    }

    public int getTanggal(){
        return tanggal;
    }

    public int getBulan(){
```

```

        return bulan;
    }

    public int getTahun(){
        return tahun;
    }
}

```

```

public class TesKalender{
    public static String getTime(Kalender kal){
        String tmp;
        tmp = kal.getTanggal() + "-" +
            kal.getBulan() + "-" +
            kal.getTahun();
        return tmp;
    }

    public static void main(String[] arg){
        Kalender kal = new Kalender(8);

        System.out.println("Waktu awal : " + getTime(kal));

        kal.setTanggal(9);
        System.out.println("1 Hari setelah waktu awal : " +
            getTime(kal));

        kal = new Kalender(6, 2003);
        System.out.println("Waktu berubah " + getTime(kal));

        kal.setBulan(7);
        System.out.println("1 Bulan setelah itu : " + getTime(kal));

        kal = new Kalender(20, 10, 2004);
        System.out.println("Waktu berubah " + getTime(kal));

        kal.setTahun(2005);
        System.out.println("1 tahun setelah itu : " + getTime(kal));
    }
}

```

### Capture

```

fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah,
$ javac Kalender.java TesKalender.java
fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah,
$ java TesKalender
Waktu awal : 8-1-2000
1 Hari setelah waktu awal : 9-1-2000
Waktu berubah 1-6-2003
1 Bulan setelah itu : 1-7-2003
Waktu berubah 20-10-2004
1 tahun setelah itu : 20-10-2005

```

### Analisa

Dalam latihan diatas, membuat class berdasarkan UML class diagram dengan menerapkan proses enkapsulasi untuk class Kalender.

## 2. Latihan 2 : Mengimplementasikan UML class diagram dalam program untuk class MataKuliah

### Listing Program

```
Mahasiswa.java

public class Mahasiswa {
    private String nrp;

    public Mahasiswa(String nrp){
        this.nrp = nrp;
    }

    public String getNrp(){
        return nrp;
    }
}
```

```
Matakuliah.java

import java.util.List;
import java.util.ArrayList;

public class Matakuliah{
    private String nrp;
    private String kode_makul;
    private int js_makul = 4;
    private String dosen_makul;
    private int angkatanMhs;
    private int maksKapasitas = 40;
    private int jumlahMhs = 0;
    private String password = "kepo";
    private List<Mahasiswa> mhs = new ArrayList<Mahasiswa>();

    public Matakuliah(String kode_makul, int js_makul,
        String dosen_makul, int angkatanMhs, int maksKapasitas){
        this.kode_makul = kode_makul;

        if(js_makul>=4){
            this.js_makul = 4;
        }else{
            this.js_makul = js_makul;
        }

        this.dosen_makul = dosen_makul;
        this.angkatanMhs = angkatanMhs;

        if(maksKapasitas>=60){
            this.maksKapasitas = 60;
        }
        else{
            this.maksKapasitas = maksKapasitas;
        }
    }

    public Matakuliah(String kode_makul, int js_makul,
        String dosen_makul, int angkatanMhs){
        this.kode_makul = kode_makul;

        if(js_makul>=4){
            this.js_makul = 4;
        }else{
            this.js_makul = js_makul;
        }

        this.dosen_makul = dosen_makul;
    }
}
```

```

        this.angkatanMhs = angkatanMhs;
    }

    public Matakuliah(String kode_makul, int js_makul, int
angkatanMhs){
        this.kode_makul = kode_makul;

        if(js_makul>=4){
            this.js_makul = 4;
        }else{
            this.js_makul = js_makul;
        }
        this.angkatanMhs = angkatanMhs;
    }

    public void cetak(){
        for(int i = 0 ; i < mhs.size() ; i++){
            System.out.println("Mahasiswa angkatan
"+angkatanMhs+" dengan nrp "
                                +mhs.get(i).getNrp()+" berhasil ditambahkan
pada matakuliah "
                                +"dengan kode "+kode_makul);
        }
    }

    public String cekKodeMakul(){
        return kode_makul;
    }

    public boolean ubahKodeMakul(String kode_makul, String
password){
        if(this.password.equals(password)){
            this.kode_makul = kode_makul;
            return true;
        }
        else{
            return false;
        }
    }

    public int cekJSMakul(){
        return js_makul;
    }

    public boolean ubahJSMakul(int js_makul, String password){
        if(this.password.equals(password)){
            if(js_makul>=4){
                this.js_makul = 4;
            }
            else{
                this.js_makul = js_makul;
            }
            return true;
        }
        else{
            return false;
        }
    }

    public String cekDosenMakul(){
        return dosen_makul;
    }

    public boolean ubahDosenMakul(String dosen_makul, String
password){
        if(this.password.equals(password)){
            this.dosen_makul = dosen_makul;
            return true;
        }
        else{
            return false;
        }
    }
}

```

```

        public int cekAngkatanMhs(){
            return angkatanMhs;
        }
        public boolean ubahAngkatanMhs(int angkatanMhs, String
password){
            if(this.password.equals(password)){
                this.angkatanMhs = angkatanMhs;
                return true;
            }
            else{
                return false;
            }
        }
        public int cekMaksKapasitas(){
            return maksKapasitas;
        }
        public boolean ubahMaksKapasitas(int maksKapasitas, String
password){
            if(this.password.equals(password)){
                if(maksKapasitas>=60){
                    this.maksKapasitas = 60;
                }
                else{
                    this.maksKapasitas = maksKapasitas;
                }
                return true;
            }
            else{
                return false;
            }
        }
        public int cekJumlahMhs(){
            return jumlahMhs;
        }
        public boolean tambahMhs(String nrp, String password){
            if(this.password.equals(password) && jumlahMhs <=
maksKapasitas){
                mhs.add(new Mahasiswa(nrp));
                jumlahMhs++;
                return true;
            }
            else{
                return false;
            }
        }
    }
}

```

#### KelolaMatakuliah.java

```

import java.util.Scanner;

public class KelolaMatakuliah {
    public static void main(String[] ggwp){
        String pass, nrp;
        char ulang = 'y';
        Scanner scan = new Scanner(System.in);
        Matakuliah matkul = new Matakuliah("MT002", 1, "Santoso
Purnomo", 2017, 30);

        System.out.println("Kode Matkul : " +
matkul.cekKodeMakul());
        System.out.println("JS Matkul : " + matkul.cekJSMakul());
        System.out.println("Dosen Matkul : " +
matkul.cekDosenMakul());
        System.out.println("Angkata Mahasiswa " +
matkul.cekAngkatanMhs());
    }
}

```



```

        System.out.println("Kapasitas Maksimal Matkul : " +
matkul.cekMaksKapasitas());

        while(ulang == 'y'){
            System.out.println("Tambah Mahasiswa");
            System.out.print("Masukan Password : ");
            pass = scan.nextLine();

            System.out.print("Masukan NRP : ");
            nrp = scan.nextLine();

            if(matkul.tambahMhs(nrp, pass)){
                System.out.println("Berhasil");
            }else{
                System.out.println("Gagal");
            }

            System.out.print("Lagi ? (y/n) : ");
            ulang = scan.next().charAt(0);
            scan.nextLine();

        }
        System.out.println("Jumlah Mahasiswa : " +
matkul.cekJumlahMhs());
        matkul.cetak();
    }
}

```

### Capture

```

fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah/SMT 3/PBO/Praktikum 9/Latihan2
$ javac Matakuliah.java KelolaMatakuliah.java
fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah/SMT 3/PBO/Praktikum 9/Latihan2
$ java KelolaMatakuliah
Kode Matkul : MT002
JS Matkul : 1
Dosen Matkul : Santoso Purnomo
Angkata Mahasiswa 2017
Kapasitas Maksimal Matkul : 30
Tambah Mahasiswa
Masukan Password : kepo
Masukan NRP : 2110171033
Berhasil
Lagi ? (y/n) : y
Tambah Mahasiswa
Masukan Password : kepo
Masukan NRP : 2110171036
Berhasil
Lagi ? (y/n) : n
Jumlah Mahasiswa : 2
Mahasiswa angkatan 2017 dengan nrp 2110171033 berhasil ditambahkan pada matakuliah dengan kode MT002
Mahasiswa angkatan 2017 dengan nrp 2110171036 berhasil ditambahkan pada matakuliah dengan kode MT002

```

### Analisa

Dalam latihan diatas, kita menerapkan UML class diagram Matakuliah menjadi sebuah class Matakuliah.java dengan menerapkan proses enkapsulasi.



```

        System.out.println("Uang berhasil
disimpan");
        System.out.println("Saldo : Rp " +
tabungan.getSaldo());
        break;
    case 2 :
        System.out.print("Uang diambil (USD) :
");
        uang = scan.nextFloat();
        uang = uang * 14900;

        if(tabungan.ambilUang((int) uang)){
            System.out.println("Penarikan uang
sukses");
        }else{
            System.out.println("Saldo anda tidak
mencukupi");
        }

        System.out.println("Saldo : Rp " +
tabungan.getSaldo());
        break;
    default :
        break;
    }

    System.out.print("\nlagi ? (y/n) : ");
    ulang = scan.next().charAt(0);
}
break;
case 2 :
    while(ulang == 'y'){
        System.out.println("1. Simpan uang");
        System.out.println("2. Ambil uang\n");
        System.out.print("Pilih : ");
        aksi = scan.nextInt();

        switch(aksi){
            case 1 :
                System.out.print("Uang disimpan (AUD) :
");
                uang = scan.nextFloat();
                uang = uang * 10000;
                tabungan.simpanUang((int) uang);

                System.out.println("Uang berhasil
disimpan");
                System.out.println("Saldo : Rp " +
tabungan.getSaldo());
                break;
            case 2 :
                System.out.print("Uang diambil (AUD) :
");
                uang = scan.nextFloat();
                uang = uang * 10000;

                if(tabungan.ambilUang((int) uang)){
                    System.out.println("Penarikan uang
sukses");
                }else{
                    System.out.println("Saldo anda tidak
mencukupi");
                }

                System.out.println("Saldo : Rp " +
tabungan.getSaldo());

```

```

        break;
        default :
            break;
    }

    System.out.print("\nlagi ? (y/n) : ");
    ulang = scan.next().charAt(0);
}
break;
case 3 :
    while(ulang == 'y'){
        System.out.println("1. Simpan uang");
        System.out.println("2. Ambil uang\n");
        System.out.print("Pilih : ");
        aksi = scan.nextInt();

        switch(aksi){
            case 1 :
                System.out.print("Uang disimpan (IDR) : ");
                uang = scan.nextFloat();
                tabungan.simpanUang((int) uang);

                System.out.println("Uang berhasil
                disimpan");
                System.out.println("Saldo : Rp " +
                tabungan.getSaldo());
                break;
            case 2 :
                System.out.print("Uang diambil (IDR) : ");
                uang = scan.nextFloat();

                if(tabungan.ambilUang((int) uang)){
                    System.out.println("Penarikan uang
                    sukses");
                }else{
                    System.out.println("Saldo anda tidak
                    mencukupi");
                }

                System.out.println("Saldo : Rp " +
                tabungan.getSaldo());
                break;
            default :
                break;
        }

        System.out.print("\nlagi ? (y/n) : ");
        ulang = scan.next().charAt(0);
    }
    break;
default:
    System.out.println("Error! Pilihan tidak ada");
    break;
}

}

}

```

## Capture

```
fndiamd on X555QG in /media/fndiamd/Fandi/Kuliah/SMT 3/PBO/P
$ java MultiTabungan
===== Program Multitabungan =====

Pilih Mata Uang :
1. USD (Dollar Amerika)
2. AUD (Dollar Australia)
3. IDR (Rupiah Indonesia)

Pilihan Anda : 1
1. Simpan uang
2. Ambil uang

Pilih : 1
Uang disimpan (USD) : 2.4
Uang berhasil disimpan
Saldo : Rp 35760

lagi ? (y/n) : y
1. Simpan uang
2. Ambil uang

Pilih : 2
Uang diambil (USD) : 1.1
Penarikan uang sukses
Saldo : Rp 19370

lagi ? (y/n) : n
```

## Analisa

Dalam tugas diatas, melakukan implementasi dari UML class diagram menjadi sebuah class java, yang kemudian class tersebut diolah. Dalam kasus diatas terdapat UML class Tabungan yang dibuat program untuk menjadi sebuah aplikasi bank sederhana multitabungan, dimana nasabah bisa menyimpan / mengambil uang dalam bentuk USD, AUD atau IDR yang kemudian tabungan tersebut dikonversi menjadi IDR.