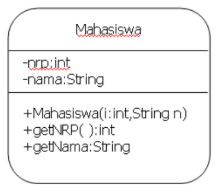
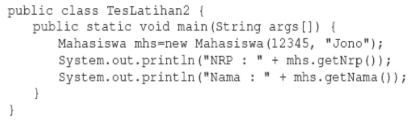
**Laporan Resmi**

1. Mengimplementasikan UML class diagram dalam program untuk class Mahasiswa



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



Sehingga didapat hasil :



Jawaban :

class Mahasiswa {

    private int nrp;

    private String nama;

    public Mahasiswa(int i, String n){

        nrp = i;

        nama = n;

    }

    public int getNrp(){

        return nrp;

    }

    public String getNama(){

        return nama;

    }

}

public class resmi1 {

    public static void main(String[] args) {

        Mahasiswa mhs = new Mahasiswa(12345, "Jono");

        System.out.println("NRP : " + mhs.getNrp());

        System.out.println("Nama : " + mhs.getNama());

    }

}

Output :

NRP : 12345

Nama : Jono

**Analisa**

Pada praktikum kali ini membahas Constructor atau Konstruktor adalah suatu method yang pertama kali dijalankan pada saat pembuatan suatu obyek. Konstruktor mempunyai ciri yaitu :

* mempunyai nama yang sama dengan nama class
* tidak mempunyai return type (seperti void, int, double dll)

Setiap class pasti mempunyai konstruktor. Jika kita membuat suatu class tanpa menuliskan konstruktornya, maka kompiler dari Java akan menambahkan sebuah konstruktor kosong.

Overloading constructor

Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Misalnya saja kita ingin menginisialisasi data member nrp dengan 2 cara. Pertama, jika user tidak memberikan nilai inisialisasi nrp, maka nrp akan diset dengan nilai 0. Kedua, jika user ingin menginisialisasi nrp sesuai dengan nilai yang diinginkan, maka nrp akan diisi sesuai nilai yang diinginkan oleh user. Sehingga class Siswa diatas dapat kita deklarasikan 2 buah konstruktor.

Pada tugas pendahuluan diminta untuk mengubah UML class ke dalam bentuk program dengan hasil :

class Lingkaran {

private double jari;

public void isijari(double jari){

this.jari = jari;

}

public double getluas(){

return 3.14 \* jari \* jari;

}

public double getkeliling(){

return 2 \* 3.14 \* jari;

}

public void gethasil(){

System.out.println("Jadi Luasnya " + getluas() + " sedangkan Kelilingnya " + getkeliling());

}

}

Pada tugas percobaan pertama diminta untuk mengubah UML class ke dalam bentuk program dengan nama class tabungan hasilnya :

class Tabungan {

int saldo;

Tabungan(int initsaldo) {

saldo = initsaldo;

}

void ambilUang(int jumlah) {

saldo = saldo - jumlah;

}

}

Pada tugas percobaan kedua diminta untuk mengubah UML class ke dalam bentuk program dengan nama class Kenaikangaji :

class Kenaikangaji {

    private double gajipokok;

    private double persen;

    private String jabatan;

    private String id;

    public Kenaikangaji(String id, String jabatan, double gajipokok, double persen){

        this.id = id;

        this.jabatan = jabatan;

        this.gajipokok = gajipokok;

        this.persen = persen;

    }

    public double getGajiPokok(){

        return gajipokok + (persen / 100) \* gajipokok;

    }

    public String getJabatan(){

        return jabatan;

    }

    public String getId(){

        return id;

    }

}

Pada tugas laporan resmi diminta untuk mengimplementasikan UMLclass diagram ke dalam bentuk program dengan nama class Mahasiswa :

class Mahasiswa {

    private int nrp;

    private String nama;

    public Mahasiswa(int i, String n){

        nrp = i;

        nama = n;

    public int getNrp(){

        return nrp;

    }

    public String getNama(){

        return nama;

    }

}

**Kesimpulan**

Pada praktikum Konstruktor dan Overloading ini diharapkan mahasiswa dapat memahami Konstuktor, Overloading terhadap Konstruktor, Overloading pada Metode.