

Tugas Proyek

Makassar, 20 Juni 2020

**PEMROGRAMAN BERORIENTASI OBJEK  
(MENGHITUNG NILAI AKHIR MAHASISWA PROGRAM STUDI  
TEKNIK INFORMATIKA)**



Nama : Yulianti A

Stambuk : 13020180125

Kelas : B2

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MUSLIM INDONESIA  
MAKASSAR  
2020**

## **KATA PENGANTAR**

Puji syukur saya ucapkan atas kehadiran Allah SWT, karena dengan rahmat dan karunia-Nya, saya masih diberi kesempatan untuk menyelesaikan laporan ini tentang konsep *Menghitung Nilai Akhir Mahasiswa* sebagai salah satu Tugas Project praktikum Pemrograman Berorientasi Objek.

Tidak lupa saya ucapkan terima kasih kepada Bapak/Ibu selaku dosen pengampu mata kuliah Pemrograman Berorientasi Objek serta berbagai pihak yang telah memberikan dukungan dan memberikan petunjuk dalam menyelesaikan laporan ini. Saya menyadari bahwa dalam penulisan laporan ini masih banyak kekurangan, oleh sebab itu saya sangat mengharapkan kritik dan saran yang membangun.

Semoga dengan selesainya laporan ini dapat bermanfaat bagi pembaca dan teman-teman.

Makassar, 20 Juni 2020

sYulianti A

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Seiring dengan kemajuan dan perkembangan zaman di era globalisasi ini, manusia dituntut mengikuti perkembangan ilmu pengetahuan dan teknologi. Semakin banyaknya *software*, dan semakin meningkatnya kecanggihan *hardware*, maka perangkat komputer sebagai alat bantu menjadi semakin berperan dalam menyelesaikan suatu pekerjaan. Hal ini juga termasuk dalam proses perkuliahan. Contohnya, apabila proses perkuliahan di setiap semester telah selesai maka dosen akan melakukan perhitungan nilai untuk setiap mahasiswa yang telah di ajarinya, namun jika dilakukan dengan perhitungan manual tentu akan membuat semua dosen kewalahan karena banyaknya mahasiswa yang mengikuti proses belajar di setiap mata kuliah yang di ampuh pada masing-masing dosen, maka dibuatlah sebuah program untuk menghitung nilai akhir mahasiswa. Dengan adanya program seperti ini setiap dosen yang ingin menghitung nilai mahasiswa menjadi lebih efisien, karena setiap dosen tinggal menginput semua nilai-nilai ke dalam sebuah aplikasi yang telah disediakan di masing-masing kampus dan otomatis akan mengeluarkan nilai akhir beserta indek dari hasil perhitungan semua nilai yang didapatkan oleh masing-masing mahasiswa.

## **1.2 Rumusan Masalah**

Adapun rumusan masalah pada laporan ini, yaitu :

1. Bagaimana sistem perhitungan nilai akhir ?
2. Bagaimana cara menginput nilai-nilai dalam aplikasi tersebut ?

## **1.3 Tujuan**

Adapun tujuan pembuatan laporan ini adalah untuk melengkapi nilai tugas project praktikum Pemrograman Berorientasi Objek sebagai media untuk menambah ilmu pengetahuan.

## BAB II

### LANDASAN TEORI

#### 2.1 Landasan Teori

##### ❖ Konsep Pemrograman Berorientasi Objek

Pemrograman Berorientasi Object atau dalam bahasa Inggris lebih dikenal dengan Object Oriented Programming (OOP) adalah sebuah paradigma dalam pemrograman yang menyelesaikan masalah program dengan menyediakan objek-objek (terdiri dari beberapa *attribute* dan *method*) yang saling berkaitan dan disusun kedalam satu kelompok atau yang disebut dengan *class*. Nantinya objek-objek tersebut akan saling berinteraksi untuk menyelesaikan masalah program yang rumit.

Dalam pemrograman berorientasi objek terdapat konsep dasar yang penting untuk diperhatikan dipahami jika kita mempelajari pemrograman.

##### ➤ Objek

Yang dimaksud dengan objek dalam pemrograman berorientasi objek adalah kumpulan perangkat lunak yang terdiri dari metode-metode dan variabel yang saling berkaitan. Objek juga merupakan wujud nyata dari prototype atau rancangan yang disebutkan dalam class tadi.

##### ➤ Kelas

Di dalam sebuah objek kelas merupakan prototype atau rancangan yang menafsirkan variabel dan berbagai method pada sebuah objek tertentu. Fungsinya adalah untuk menampung konten atau isi dari sebuah program yang akan dijalankan. Dengan kata lain, kelas/class adalah rancangan atau prototype dari

sebuah objek dalam pemrograman berorientasi objek. Salah satu contoh dari class adalah class person, class animal, vehicle, fruit dan lain-lain. Dalam bidang pemrograman, sebuah class haruslah dapat dikenali bahkan oleh orang bukan programmer, karena class merupakan dasar dari sebuah program.

#### ➤ **Method**

Method atau metode merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek. Method didefinisikan pada class, namun dipanggil melalui objek. Contoh, pada objek buah pear: terdapat ‘metode ambil rasa’, ‘kupas kulit’ dan lain-lain.

Dalam pemrograman berorientasi objek, method merupakan kumpulan dari berbagai program yang memiliki nama. Ini adalah alat yang digunakan oleh seorang programmer dalam memecah sebuah program menjadi bagian yang kecil-kecil dan kompleks sehingga dapat digunakan berulang kali.

#### ➤ **Abstraksi**

Abstraksi merupakan kemampuan dari sebuah program untuk melewati aspek informasi yang diproses olehnya, atau bisa juga disebut sebagai kemampuan untuk memfokuskan pada intinya. Setiap objek dalam sistem OOP merupakan model dari “pelaku” abstrak yang dapat melakukan suatu pekerjaan, pelaporan dan perubahan pada keadaannya, dan melakukan komunikasi dengan objek lainnya yang ada di dalam sistem suatu program, tanpa memperlihatkan bagaimana hal ini diterapkan

#### ➤ **Enkapsulasi**

Ini adalah suatu bentuk dimana pengguna dari sebuah objek tidak dapat mengganti atau mengubah (memodifikasi) suatu keadaan dari sebuah objek dengan cara yang tidak layak, oleh karena itu keadaan ini hanya dapat dimodifikasi sesuai dengan ketentuan yang terdapat di dalam objek tersebut. Begitupun ketika kita ingin mengganti interfacenya haruslah memenuhi aturan dari objek yang bersangkutan. Objek yang lainnya tidak akan tergantung dengan objek tersebut.

#### ➤ **Pewarisan Sifat Objek (Inheritance)**

Pewarisan adalah salah satu cirri pemrograman berorientasi objek, yang menyatakan suatu kelas dapat diturunkan lagi menjadi kelas-kelas baru yang lainnya sehingga dapat membentuk sebuah hirarki. Kelas yang merupakan kelas

turunan ini bisa disebut dengan kelas anak (subclass) dan kelas yang menjadi dasar penurunan disebut kelas orang tua (superclass).

➤ **Polimorfisme**

Polimorfisme adalah kemampuan suatu objek untuk mengungkap banyak hal melalui satu cara yang sama. Polimorfisme merupakan salah satu hal esensial dalam pemograman berorientasi objek karena alasan beriku: yakni polimorfisme mengizinkan kelas induk untuk mendefinisikan sebuah method general ( bersifat umum) untuk semua kelas turunannya. Polimorfisme dapat berupa overloading atau overriding. Overloading merupakan bentuk polimorfisme yaitu beberapa metode yang dapat memiliki nama yang sama dengan isi dan parameter yang berbeda didalam sebuah kelas. Biasanya eksekusi program akan langsung mengacu pada metode yang dipanggil sesuai dengan parameter

## BAB III

### IMPLEMENTASI & PEMBAHASAN

#### 3.1 Implementasi & Pembahasan

Sebelum membuat project, terlebih dahulu kita membuat database beserta table seperti di bawa ini :

- Database dengan nama db\_mhs dan table dengan nama tbl\_mhss

YADMIN/tbl\_structure.php?db=db\_mhs&table=tbl\_mhss

Server: 127.0.0.1 » Basis data: db\_mhs » Tabel: tbl\_mhss

Struktur tabel

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 Stambuk	int(15)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	2 Nama	varchar(30) utf8mb4_general_ci		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	3 UTS	int(10)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	4 UAS	int(10)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	5 Tugas Mandiri	int(10)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	6 Tugas Proyek	int(10)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	7 Kehadiran	int(5)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	8 Nilai Akhir	float		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	9 Grade	varchar(5) utf8mb4_general_ci		Tidak	Tidak ada				Ubah Hapus Lainnya

Remove from central columns

Cetak Usulkan struktur tabel Lacak tabel Move columns Normalisasi

Tambahkan 1 kolom setelah Grade Kirim

Indeks

Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
Ubah Hapus	PRIMARY	BTREE	Ya	Tidak	Stambuk	0	A	Tidak	

Konsol

- Buat sebuah project, kemudian tambahkan package dengan nama koneksi dan buat sebuah clas untuk menghubungkan database dengan program di netbeans dan isi class nya seperti dibawah ini:

➤ koneksi.java

```
package koneksi;
```

```
import com.mysql.jdbc.Statement;
```

```
import javax.swing.JOptionPane;
```

```
import com.mysql.jdbc.Driver;
```



```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;


public class koneksi {

    static Connection koneksi;

    public static Connection getConnection(){

        try{

            koneksi=

DriverManager.getConnection("jdbc:mysql://localhost/db_mhs","r
oot","");

        }catch (Exception e){

            JOptionPane.showMessageDialog(null, "Koneksi Database
Gagal");

        }

        return koneksi;

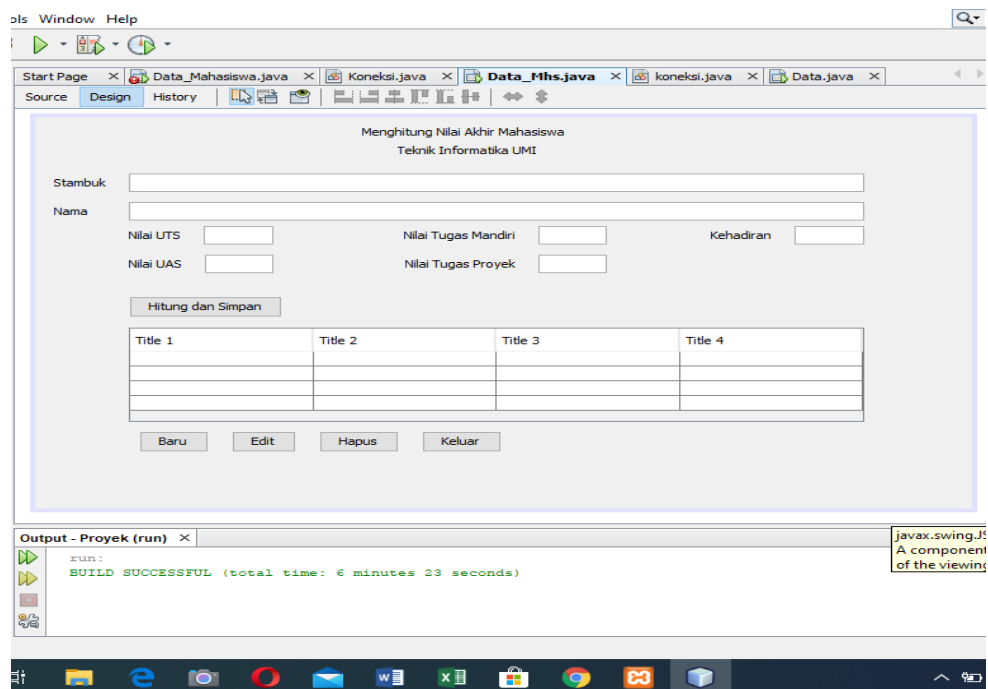
    }

}
```

- buat sebuah class frame pada default package untuk membuat design aplikasi dan mengisi programnya, seperti di bawah ini:

➤ Data\_Mhs.java

a. Tampilan design aplikasi



b. Source code

```
import com.mysql.jdbc.Statement;

import java.awt.HeadlessException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;
```

```

import javax.swing.JOptionPane;

import koneksi.koneksi;

import javax.swing.table.JTableHeader;

import javax.swing.table.DefaultTableModel;

public class Data_Mhs extends javax.swing.JFrame {

    DefaultTableModel model;

    private void setTable(){

        Object[] field = {"Stambuk", "Nama", "UTS", "UAS", "Tugas
Mandiri", "Tugas Proyek", "Kehadiran", "Nilai Akhir", "Grade"};

        model = new DefaultTableModel(field, 0);

        tbl_mhss.setModel(model);

        JTableHeader th = tbl_mhss.getTableHeader();

        th.setReorderingAllowed(false);

        tbl_mhss.setRowHeight(30);

    }

    private String calculate(String... utm){

        for (int i = 0; i < utm.length; i++){

            if(utm[i].isEmpty()){

                utm[i] = "0";

            }

        }

        double t1=0,t2=0,t3=0,t4=0,t5=0;

```

```
for (int i = 0; i < utm.length; i++) {  
    double cv = Double.valueOf(utm[i]);  
    switch (i) {  
        case 0:  
            t1 = cv * 0.20;  
            break;  
        case 1:  
            t2 = cv * 0.25;  
            break;  
        case 2:  
            t3 = cv * 0.20;  
            break;  
        case 3:  
            t4 = cv * 0.20;  
            break;  
        case 4:  
            t5 = cv * 0.15;  
            break;  
        default:  
            break;  
    }  
}  
  
double jumlah = t1+t2+t3+t4+t5;
```

```

        double as = Math.round(jumlah);

        String cvJumlah = indexOF(as);

        return cvJumlah;
    }

```

```

public String indexOF(double b){

    String hsl;

    if(b % 1 == 0){

        hsl                                     =
String.valueOf(b).substring(0,String.valueOf(b).lastIndexOf("."));

    }else{

        hsl = String.valueOf(b);

    }

    return hsl;

}

```

```

private String calAbjad(String jumlah){

    String Grade = "";

    double b = Double.valueOf(jumlah);

    if(b > 85){

        Grade = "A";

    }else if(b >= 81 && b <= 85){

        Grade = "A-";

    }
}

```

```

    }else if(b >= 76 && b <= 80){

        Grade = "B+";

    }else if(b >= 71 && b <= 75){

        Grade = "B";

    }else if(b >= 66 && b <= 70){

        Grade = "B-";

    }else if(b >= 61 && b <= 65){

        Grade = "C+";

    }else if(b >= 51 && b <= 60){

        Grade = "C";

    }else if(b >= 45 && b <= 50){

        Grade = "D";

    }else if(b < 45){

        Grade = "E";

    }

    return Grade;

}

public void kosongkan_data(){

    txtStb.setText("");

    txtNm.setText("");

    txtUts.setText("");

    txtUas.setText("");

    txtTM.setText("");

```

```

        txtTP.setText("");

        txtHadir.setText("");

        //txtStb.requestFocus();
    }

    public Data_Mhs() {

        initComponents();

        setTable();

        kosongkan_data();

    }

```

**// untuk fungsi tombol Hitung dan simpan (menghitung dan menyimpan data yang di inputkan pada table)**

```

Private void bHitungSimpanActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    String Stambuk = txtStb.getText();

    String Nama = txtNm.getText();

    String UTS = txtUts.getText();

    String UAS = txtUas.getText();

    String TugasMandiri = txtTM.getText();

    String TugasProyek = txtTP.getText();

    String Kehadiran = txtHadir.getText();

```

```

        if(Stambuk.isEmpty()){

            JOptionPane.showMessageDialog(this, "Masukan Stambuk
Terlebih Dahulu");

            txtStb.requestFocus();

        }else{

            String[] data = {UTS, UAS, TugasMandiri, TugasProyek,
Kehadiran};

            String jml = calculate(data);

            Object[] data2 = {Stambuk, Nama, UTS, UAS,
TugasMandiri, TugasProyek, Kehadiran, jml, calAbjad(jml)};

            model.addRow(data2);

        }
    }

    private void tbl_mhssMouseClicked(java.awt.event.MouseEvent
evt) {

        // TODO add your handling code here:

    }

// untuk fungsi tombol Baru (menambah data baru)

    private void bBaruActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        kosongkan_data();

```



```
}
```

**// untuk fungsi tombol Edit (mengubah data pada table)**

```
private void bEditActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    try{  
  
        String sql = "UPDATE tbl_mhss set Nama =  
        '"+txtNm.getText()+"' UTS = '"+txtUts.getText()+"' UAS =  
        '"+txtUas.getText()+"' Tugas Mandiri = '"+txtTM.getText()+"'  
        Tugas Proyek = '"+txtTP.getText()+"' Kehadiran =  
        '"+txtHadir.getText()+"' where Stambuk = '"+txtStb.getText()+"';  
  
        Statement statement =  
  
        (Statement)koneksi.getConnection().createStatement();  
  
        statement.executeUpdate(sql);  
  
        statement.close();  
  
        JOptionPane.showMessageDialog(null, "Data Berhasil  
        diubah");  
  
    }catch (Exception t){  
  
        JOptionPane.showMessageDialog(null, "Data Gagal  
        diubah");  
  
    }  
  
    setTable();  
  
    kosongkan_data();  
}
```

```
}
```

**// untuk fungsi Tombol Hapus (Menghapus data dalam table)**

```
private void bHapusActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{  
        String sql = "delete from tbl_mhss where  
Stambuk='"+txtStb.getText()+"';  
  
        Statement statement =  
(Statement)koneksi.getConnection().createStatement();  
  
        statement.executeUpdate(sql);  
  
        JOptionPane.showMessageDialog(null, "Data Berhasil  
dihapus");  
  
        txtStb.setText("");  
        txtNm.setText("");  
        txtUts.setText("");  
        txtUas.setText("");  
        txtTM.setText("");  
        txtTP.setText("");  
        txtHadir.setText("");  
  
        //txtStb.requestFocus();  
  
    }catch(Exception t){
```

```

JOptionPane.showMessageDialog(null, "    Data    Gagal
dihapus");

}

setTable();

kosongkan_data();

}

```

**// untuk fungsi tombol Keluar (Keluar dari aplikasi)**

```

private void bKeluarActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    dispose();

}

```

- Setelah selesai mengisi semua source codenya kita jalankan programnya dan akan muncul sebuah tampilan aplikasi seperti di bawah ini:

➤ Tampilan aplikasi sebelum menginputkan data

Menghitung Nilai Akhir Mahasiswa  
Teknik Informatika UMI

Stambuk

Nama

Nilai UTS  Nilai Tugas Mandiri  Kehadiran

Nilai UAS  Nilai Tugas Proyek

Stambuk	Nama	UTS	UAS	Tugas ...	Tugas ...	Kehadir...	Nilai Ak...	Grade

➤ Tampilan aplikasi setelah menginputkan data

Menghitung Nilai Akhir Mahasiswa  
Teknik Informatika UMI

Stambuk

Nama

Nilai UTS  Nilai Tugas Mandiri  Kehadiran

Nilai UAS  Nilai Tugas Proyek

Stambuk	Nama	UTS	UAS	Tugas ...	Tugas ...	Kehadir...	Nilai Ak...	Grade
130201...	Amman	95	90	70	90	100	85	A
130201...	Nur	100	95	100	90	100	97	A
130201...	Indah	45	76	50	66	80	63	C+

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Program ini di buat untuk membantu semua Dosen agar lebih mudah saat akan menghitung nilai mahasiswa.

Dari hasil pembelajaran, saya bisa memahami dan mengerti cara merancang tampilan form dari sebuah aplikasi dan mengerti fungsi dari source code masing-masing form dan menghubungkannya kedatabase.

#### **4.2 Saran**

Semoga teori & praktikum Pemrograman Berorientasi Objek dapat ditingkatkan lagi, Agar mahasiswa dapat memahami sebuah aplikasi atau sistem informasi yang kemudian bisa diterapkan dalam kehidupan sebagai kebutuhan akan informasi.

## **DAFTAR PUSTAKA**

<https://markey.id/blog/development/pemrograman-berorientasi-objek/amp#referrer=https://www.google.com>

<https://www.ismynr.xyz/2019/02/program-java-gui-menghitung-nilai-mahasiswa.html?m=1>

