

Tugas Pemrograman Berbasis Objek



Muhamad Salman Adhim Baqy

A11.2020.12641

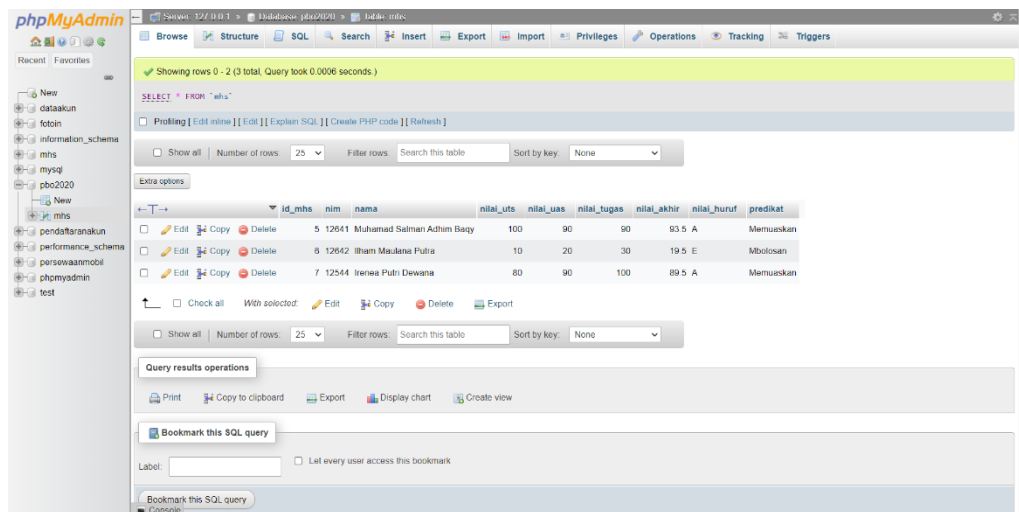
Fakultas Ilmu Komputer

Program Studi Teknik Informatika

Universitas Dian Nuswantoro

2022

1. Membuat Database.



2. Koneksikan database ke aplikasi yang akan dibuat.

```
/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package siakad;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author adhim
 */
public class DBConnection {
    public Connection getConnection(){
        Connection con = null;
        try {
            con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost/pbo2020", "root",
"");
            JOptionPane.showMessageDialog(null, "Koneksi Sukses");
        } catch (SQLException ex) {

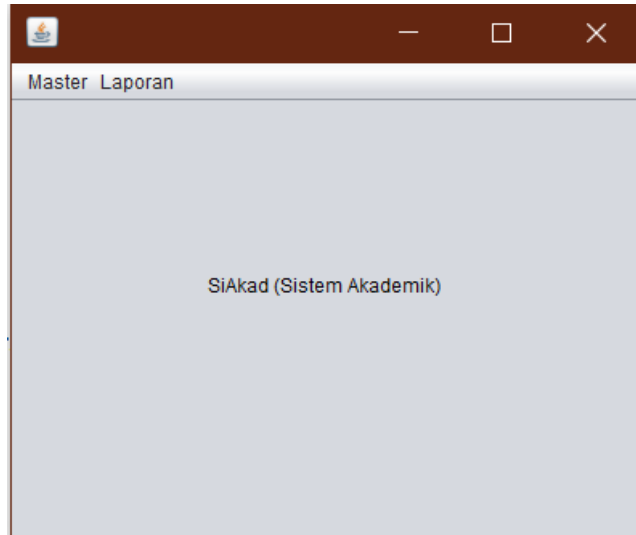
            Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,
            null, ex);
        }
    }
}
```

```

        return con;
    }
}

```

3. Membuat tampilan depan aplikasi



Source Code Depan.java:

```

private void dataMahasiswaActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        new FormNilaiMhs().setVisible(true);
        dispose();
    } catch (SQLException ex) {
        Logger.getLogger(Depan.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

4. Membuat Method

Mhs.java

```

/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package siakad;

/**
 *
 * @author adhim
 */
public class Mhs implements Nilai {

```

```

public String nim, nama;
double vUts, vUas, vTugas;
double pUts, pUas, pTugas;
double nilaiAkhir;
char nHuruf;
String predikat;

//konstruktor
public Mhs(String nim, String nama, double vUts, double vUas, double
vTugas){
    this.nim = nim;
    this.nama = nama;
    this.vUts = vUts;
    this.vUas = vUas;
    this.vTugas = vTugas;
}

public double uts(){
    return vUts * 35/100;
}

public double uas(){
    return vUas * 35/100;
}

public double tugas(){
    return vTugas * 30/100;
}

public double nilaiAkhir(){
    return uts() + uas() + tugas();
}

@Override
public char getNilHuruf(double nilaiAkhir) {
    if(nilaiAkhir >= 85.0){
        nHuruf = 'A';
    }else if(nilaiAkhir >= 70.0 && nilaiAkhir < 85.0){
        nHuruf = 'B';
    }else if(nilaiAkhir >= 60.0 && nilaiAkhir < 70.0){
        nHuruf = 'C';
    }else if(nilaiAkhir >= 40.0 && nilaiAkhir < 60.0){
        nHuruf = 'D';
    }else{
        nHuruf = 'E';
    }
    return nHuruf;
}

@Override

```

```

public String getPredikat(char nHuruf) {
    switch(nHuruf){
        case 'A':
            predikat = "Memuaskan";
            break;
        case 'B':
            predikat = "Baik";
            break;
        case 'C':
            predikat = "Cukup";
            break;
        case 'D':
            predikat = "Dongo";
            break;
        case 'E':
            predikat = "Mbolosan";
            break;
    }
    return predikat;
}
}

```

5. Membuat tampilan Data Mahasiswa serta Fungsi CRUD.

No	Nim	Nama	Uts 35%	Uas 35%	Tugas 30%	N. Akhir	N. Huruf	Predikat
1	12541	Muhamad Sal...	100	90	90	93.5	A	Memuaskan
2	12542	Ilham Maulan...	10	20	30	19.5	E	Mbolosan
3	12544	Irenea Putri D...	80	90	100	89.5	A	Memuaskan

Source Code FormNilaiMhs.java:

```

extends javax.swing.JFrame {

    DefaultTableModel tabel1;
    Object[] list_mhs = new Object[9];
    int x = 0; // int untuk nomor ascending

```

```

DBConnection dbCon = new DBConnection();
Connection con = dbCon.getConnection();

/**
 * Creates new form FormNilaiMhs
 */
public FormNilaiMhs() throws SQLException {
    initComponents();

    tabel1 = new DefaultTableModel();
    tabel.setModel(tabel1);
    tabel1.addColumn("No"); tabel1.addColumn("Nim");
    tabel1.addColumn("Nama"); tabel1.addColumn("Uts 35%");
    tabel1.addColumn("Uas 35%"); tabel1.addColumn("Tugas 30%");
    tabel1.addColumn("N. Akhir"); tabel1.addColumn("N. Huruf");
    tabel1.addColumn("Predikat");

    setResizable(false); //tdk dapat diperbesar

    //tidak bisa diketik
    uts35.setEditable(false);
    uas35.setEditable(false);
    tugas35.setEditable(false);
    nUts2.setEditable(false);
    nUas2.setEditable(false);
    nTugas2.setEditable(false);

    //tidak bisa diklik
    simpan.setEnabled(false);
    update.setEnabled(false);
    hapus.setEnabled(false);
    proses1.setEnabled(false);

    tampilkanDiTabel();
}

public void tampilkanDiTabel() throws SQLException{
    ArrayList<> list = getMhsList();
    DefaultTableModel model = (DefaultTableModel)tabel.getModel();

    Object[] row = new Object[9];
    for(int i = 0; i < list.length; i++){
        row[0] = i + 1;
        row[1] = list[i].get(0); row[2] = list[i].get(1);
        row[3] = list[i].get(2); row[4] = list[i].get(3);
        row[5] = list[i].get(4); row[6] = list[i].get(5);
        row[7] = list[i].get(6); row[8] = list[i].get(7);

        model.addRow(row);
    }
}

```

```

    }

    public void kosongkanTextField(){
        nim.setText(""); nama.setText("");
        nUts.setText(""); nUas.setText(""); nTugas.setText("");
        uts35.setText(""); uas35.setText(""); tugas35.setText("");
        nUts2.setText(""); nUas2.setText(""); nTugas2.setText("");
    }

    public void prosesHitungNilai(){
        try{
            String ni = nim.getText();
            String na = nama.getText();
            Double ts = Double.parseDouble(nUts.getText());
            Double as = Double.parseDouble(nUas.getText());
            Double tgs = Double.parseDouble(nTugas.getText());

            Mhs m = new Mhs(ni, na, ts, as, tgs);
            uts35.setText("" + m.uts()); uas35.setText("" + m.uas());
            tugas35.setText("" + m.tugas());
            nUts2.setText("" + m.nilaiAkhir());
            nTugas2.setText("" + m.getPredikat(m.getNilHuruf(m.nilaiAkhir())));
            nUas2.setText("" + m.getNilHuruf(m.nilaiAkhir()));

            simpan.setEnabled(true);
        }catch(NumberFormatException e){
            JOptionPane.showMessageDialog(null, "Inputan Anda Kosong",
"Warning", JOptionPane.WARNING_MESSAGE);
        }
    }

    public void kosongkanTabel(){
        DefaultTableModel model =
(DefaultTableModel)this.tabel.getModel();
        model.setRowCount(0);
    }

    public void terpilih(int index) throws SQLException{
        ArrayList<Mhs> list = getMhsList();
        nim.setText((String) list[index].get(0));
        nama.setText((String) list[index].get(1));
        nUts.setText((String) list[index].get(2).toString()); // karena double
        nUas.setText((String) list[index].get(3).toString());
        nTugas.setText((String) list[index].get(4).toString());

        proses.setEnabled(false);
        simpan.setEnabled(false);
    }

```

```

public ArrayList[] getMhsList() throws SQLException{
    String queryCount = "SELECT COUNT(*) AS c FROM mhs";
    Statement st;
    ResultSet rsCount, rs;

    st = con.createStatement();
    rsCount = st.executeQuery(queryCount);
    int sizeTable = 0;

    while(rsCount.next()){
        sizeTable = rsCount.getInt("c");
        //panggil alias c dari querycount
    }

    ArrayList[] mhsList = new ArrayList[sizeTable];

    String query = "SELECT * FROM mhs";
    rs = st.executeQuery(query);
    int x = 0;
    while(rs.next()){
        mhsList[x] = new ArrayList<>();
        mhsList[x].add(rs.getString("nim"));
        mhsList[x].add(rs.getString("nama"));
        mhsList[x].add(rs.getString("nilai_uts"));
        mhsList[x].add(rs.getString("nilai_uas"));
        mhsList[x].add(rs.getString("nilai_tugas"));
        mhsList[x].add(rs.getString("nilai_akhir"));
        mhsList[x].add(rs.getString("nilai_huruf"));
        mhsList[x].add(rs.getString("predikat"));
        x++;
    }

    return mhsList;
}

private void nimActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void namaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void nUtsActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

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

```



```

        // TODO add your handling code here:
    }

    private void nTugasActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void uts35ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void uas35ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void tugas35ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void nUas2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void nTugas2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void nUts2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void prosesActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        prosesHitungNilai();
    }

    private void keluarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
    }

    private void proses1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        prosesHitungNilai();
        update.setEnabled(true);
        simpan.setEnabled(false);
    }

    private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

```

```

        int index = tabel.getSelectedRow();
        try {
            terpilih(index);
        } catch (SQLException ex) {

Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
null, ex);
        }
        proses1.setEnabled(true);
        hapus.setEnabled(rootPaneCheckingEnabled);
    }

    private void tambahLainActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
        kosongkanTextField();
        proses.setEnabled(true);

        uts35.setEditable(false);                uas35.setEditable(false);
        tugas35.setEditable(false);
        nUts2.setEditable(false);                nUas2.setEditable(false);
        nTugas2.setEditable(false);

        simpan.setEnabled(false); update.setEnabled(false);
        hapus.setEnabled(false); proses1.setEnabled(false);

    }

    private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here:
            PreparedStatement ps = con.prepareStatement("INSERT INTO
mhs(nim, nama, nilai_uts, nilai_uas, nilai_tugas, nilai_akhir, nilai_huruf,
predikat) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
            ps.setString(1, nim.getText()); ps.setString(2, nama.getText());
            ps.setString(3, nUts.getText()); ps.setString(4, nUas.getText());
            ps.setString(5, nTugas.getText());
            ps.setString(6, nUts2.getText()); ps.setString(7, nUas2.getText());
            ps.setString(8, nTugas2.getText());

            ps.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data Tersimpan");
        } catch (SQLException ex) {

Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
null, ex);
            JOptionPane.showMessageDialog(null, "Data Tidak Tersimpan");
        }

        try {

```

```

        kosongkanTextField();
        kosongkanTabel();
        tampilkanDiTabel();
    } catch (SQLException ex) {

Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

private void updateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String updateQuery = null;
    PreparedStatement ps = null;

    updateQuery = "UPDATE mhs SET nim=?, nama=?, nilai_uts=?,
nilai_uas=?, nilai_tugas=?, nilai_akhir=?, nilai_huruf=?, predikat=? WHERE
nim=?";

    try {
        // TODO add your handling code here:
        ps = con.prepareStatement(updateQuery);
        ps.setString(1, nim.getText()); ps.setString(2, nama.getText());
        ps.setString(3, nUts.getText()); ps.setString(4, nUas.getText());
        ps.setString(5, nTugas.getText());
        ps.setString(6, nUts2.getText()); ps.setString(7, nUas2.getText());
        ps.setString(8, nTugas2.getText()); ps.setString(9, nim.getText());

        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Berhasil diUpdate");
    } catch (SQLException ex) {

Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
null, ex);
        JOptionPane.showMessageDialog(null, "Data Tidak Berhasil
diUpdate");
    }

    try {
        kosongkanTextField();
        kosongkanTabel();
        tampilkanDiTabel();
    } catch (SQLException ex) {

Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
null, ex);
    }

    proses.setEnabled(true);
    update.setEnabled(false);

```

```

        hapus.setEnabled(false);
        proses1.setEnabled(false);
    }

    private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        try {
            PreparedStatement ps = con.prepareStatement("DELETE FROM
mhs WHERE nim=?");
            String nimnya = nim.getText();
            ps.setString(1, nimnya);
            ps.executeUpdate();

            JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");
        } catch (SQLException ex) {

            Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
            null, ex);
            JOptionPane.showMessageDialog(null, "Data Tidak Berhasil
            Dihapus");
        }

        try{
            kosongkanTextField();
            kosongkanTabel();
            tampilkanDiTabel();
        }catch(SQLException ex
            ){

            Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,
            null, ex);
        }

    }

    private void printActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JasperReport reports;

        String path = ".\\src\\siakad\\reportMhs.jasper";
        try {
            reports = (JasperReport) JRLoader.loadObjectFromFile(path);
            JasperPrint jprint = JasperFillManager.fillReport(path, null, con);
            JasperViewer jviewer = new JasperViewer(jprint, false);

            jviewer.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
            jviewer.setVisible(true);
        } catch (JRException ex) {

```

```
Logger.getLogger(FormNilaiMhs.class.getName()).log(Level.SEVERE,  
null, ex);  
    }  
}
```