# **PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**SEMESTER GENAP T.A 2019/2020**

**LAPORAN PROYEK AKHIR**

****

**DISUSUN OLEH :**

|  |  |  |
| --- | --- | --- |
| **NAMA** | **:** | **FAUZIYAH AHMAD INAYANTI (123180097)**  **MUNADIA ROSYADA NA (123180155)** |
| **KELAS** | **:** | **F** |
| **NAMA ASISTEN** | **:** | **MUHAMMAD AFIN FAUZI** |
|  |  | **DYAH AYU SUCI ILHAMI** |

### PROGRAM STUDI INFORMATIKA

### JURUSAN TEKNIK INFORMATIKA

### FAKULTAS TEKNIK INDUSTRI

### UNIVERSITAS PEMBANGUNAN NASIONAL ”VETERAN”

#### YOGYAKARTA

#### 2019

**HALAMAN PENGESAHAN**

**LAPORAN PROYEK AKHIR**

****

Disusun oleh :

|  |  |
| --- | --- |
| *Fauziyah Ahmad Inayanti* | *Munadia Rosyada Na* |
| 123180097 | 123180155 |

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Pemrograman Berorientasi Objek

Pada Tanggal : .................................

|  |  |
| --- | --- |
| **Asisten Praktikum**  **Muhammad Afin Fauzi**  **NIM. 124170019** | **Asisten Praktikum**  **Dyah Ayu Suci Ilhami**  **NIM. 124170024** |

|  |
| --- |
| **Mengetahui,**  **Ka. Lab. Geoinformatika**  **Budi Santosa, S.Si., M.T.**  **NIK. 2 7009 02 0234 1** |

**KATA PENGANTAR**

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Pemrograman Berorientasi Objek serta laporan proyek akhir praktikum yang berjudul Program Laundry. Adapun laporan ini berisi tentang proyek akhir yang saya pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten laboratorium yang selalu membimbing dan mengajari saya dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun saya harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, saya ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 19 Mei 2020

Penyusun

**DAFTAR ISI**

**Cover**  **i**

**Halaman Pengesahan**  **ii**

**Kata** **Pengantar**  **iii**

**Daftar** **Isi iv**

**Judul Proyek Akhir 1**

1. **Pendahuluan 2**
2. Latar Belakang Masalah 2
3. Tujuan Proyek Akhir 3
4. **Pembahasan 4**
   1. Dasar Teori 4
   2. Listing dan Pembahasan Program 7
   3. Output 33
5. **Jadwal Pengerjaan dan Pembagian Tugas 37**
   1. Jadwal Pengerjaan 37
   2. Pembagian Tugas 37
6. **Kesimpulan dan Saran 38**
   1. Kesimpulan 38
   2. Saran 38

**Daftar Pustaka ................................................................................................................** 39

**JUDUL PROYEK AKHIR**

Judul Proyek yang kami pilih dalam penyelesaian praktikum Pemrograman Berorientasi Objek serta laporan proyek akhir praktikum adalah  **“PROGRAM LAUDRY”**

**BAB** **I**

**PENDAHULUAN**

1. **Latar Belakang Masalah**

Kebutuhan manusia akan teknologi informasi dan komunikasi di masa sekarang ini telah menjadi suatu hal yang wajib, dimana hampir segala aspek kehidupan sehari-hari tidak terlepas dari teknologi informasi dan komunikasi. Teknologi informasi dan komunikasi banyak dimanfaatkan manusia untuk menunjang dan memudahkan aktivitas kehidupan sehari- hari. Penggunaan internet sudah menjamur di masyarakat. Internet sudah menjadi kebutuhan sekunder bahkan sampai primer. Hal ini diikuti sistem yang awalnya manual menjadi era komputer atau digital. Pengunaan teknologi dapat meningkatkan dan menunjang suatu aktivitas bisnis, sehingga banyak pelaku bisnis menerapkan teknologi untuk bersaing dengan pelaku bisnis lainya. Dengan perkembangan teknologi yang semakin pesat, pelaku bisnis juga harus cepat megadopsi perkembangan teknologi yang ada agar tidak tertinggal dengan pelaku bisnis lainya.

Salah satu bisnis yang banyak peminatnya saat ini adalah bisnis *laundry*. Kesibukan seseorang membuatnya cenderung ingin praktis dalam mengurus pakaian sehari-hari. *Laundry* menawarkan jasa cuci pakaian hingga setrika yang terpercaya sehingga banyak orang memilih me-*laundry*-kan pakaiannya. Banyaknya peminat *laundry* mendorong bisnis laundry agar memiliki sistem yang handal untuk mengatur data pelanggannya. Atas dasar pemikiran tersebut, maka dibuat program *laundry* terkomputerisasi dengan berbagai komponennya yang dapat menyimpan dan mengatur informasi dengan baik. Program ini menggunakan Java dengan mengimplementasikan Model, View, Control. Program ini dapat digunakan untuk mengolah transaksi pendataan pelanggan laundry yang masuk dan keluar serta menyimpannya sebagai arsip bagi pemilik laundry. Sistem ini diharapkan dapat mempercepat proses pengolahan dan penyimpanan data jika sewaktu-waktu diperlukan untuk menentukan kebijakan.

1. **Tujuan Proyek Akhir**
2. Mempermudah proses pendataan pelanggan masuk dan keluar
3. Mengetahui seluruh data transaksi pelanggan
4. Mempermudah proses pengubahan dan penghapusan data pelanggan
5. Mempermudah pencarian data pelanggan
6. Mempermudah pegawai untuk menjalankan tugasnya
7. Mengurangi kesalahan dalam transaksi dengan pelanggan
8. Pengarsipan data bisnis *laundry*
9. Salah satu bentuk strategi dalam bisnis

**BAB II**

**PEMBAHASAN**

* 1. **Dasar Teori**

1. Dasar Java

Java adalah bahasa pemrograman berorientasi objek. Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas-kelas untuk mengorganisasikan kode program dalam modul-modul logikal.

1. Unsur-unsur Java:
2. Class
3. Method
4. Constructor
5. Modifier
6. Object
7. Package
8. Object Oriented Programming

Dalam bahasa pemrograman berorientasi objek, program dibagi menjadi objek-objek yang saling berinteraksi satu sama lain. Objek tersebut dapat memiliki wujud yang nyata, maupun tidak nyata (abstrak). Konsep-konsep dasar dari pemrograman berorientasi objek adalah inheritance (pewarisan), encapsulation (pengkapsulan), dan polymorphism.

1. Exception

Exception merupakan suatu objek error khusus yang dibuat ketika terjadi kesalahan pada suatu program. Setelah suatu objek exception dibuat oleh java, objek tersebut dikirimkan ke program, proses ini disebut *throwing an exception*. Objek exception tersebut kemudian diterima oleh program, dan penanganan exception selanjutnya terserah pada program.

Penanganan exception dalam java dibuat dengan menuliskan kode program exception-handling pada Source code. Exception-handling berguna untuk mencegah program menjadi “Hang”. Code program exception handling secara umum terdiri dari 3 bagian yaitu *try*, *catch* , dan *finally*.

1. Swing
2. Java Swing

Swing adalah kumpulan class–class yang digunakan untuk membuat GUI. Swing adalah GUI yang lightweight, karena lebih multiplatform dan sedikit menggunakan Native class.

1. JFrame

Dalam lingkup Java, sebuah interface window dikenal dengan sebutan frame. Frame adalah sebuah objek yang merepresentasikan suatu area di layar yang berisi sejumlah objek lain untuk menyampaikan informasi kepada user. Dengan adanya frame, berbagai komponen GUI dapat ditampilkan.

1. JLabel, JTextField , JButton, JRadioButton, JCheckbox, JComboBox

Merupakan komponen-komponen swing pada Frame.

1. Manajemen Layout

Manajemen pemasangan komponen (layout management) diperlukan untuk mengatur penempatan komponen di dalam frame agar bisa menghasilkan bentuk interface yang menarik. Penggunaan layout-manager dalam mengatur letak komponen juga akan memudahkan kita menempelkan komponen pada frame. Java menyediakan sejumlah metode dalam mengatur penempelan komponen ke dalam frame yang bisa dikombinasikan dalam penggunaannya.

Beberapa jenis layout:

1. Metode FlowLayout
2. Metode GridLayout
3. Metode BorderLayout
4. Metode NullLayout
5. Event Handling

Event Handling merupakan tindakan yang harus dilakukan oleh komponen untuk dapat berinteraksi dengan program. Event adalah peristiwa yang di-stimulasi oleh user terhadap komponen-komponen GUI. Event ditangkap oleh event listener sehingga Java akan dapat mengenali komponen mana yang menstimulasi event. Selanjutnya kita menentukan event handler, yaitu blok yang akan memproses bila terjadi suatu event. Event listener dan Event handler ini terdapat pada package java.awt.event, dan memiliki bentuk sebagai interface.

Beberapa Event Handling:

1. Action Listener
2. Adjusment Listener
3. Focus Listener
4. Item Listener
5. Key Listener
6. Java Database Connectivity

JDBC (Java DataBase Connectivity) merupakan teknologi Java yang populer sebagai standar pengaksesan database. Java tidak membedakan cara pemrograman database yang satu dengan lainnya. Sebagai contoh: cara Java mengolah tabel di Oracle dan MySQL sama saja, yang berbeda hanyalah alamat URL koneksi dan driver JDBC-nya. Sehingga kita dapat membuat sebuah program yang sama di Java untuk mengolah database yang ada di MySQL, Oracle, SQL Server dan lain-lain asalkan nama dan struktur tabelnya sama.

1. MVC

Model View Control pertama kali diperkenalkan oleh Trygve Reenskaug, seorang  pengembang perangkat lunak Smalltalk pada Xerox Palo Alto Research Center pada tahun 1979. Konsep ini memisahkan antara akses data, logika bisnis, dan antarmuka pada perangkat lunak, secara detail MVC dibagi 3, yaitu :

1. Model, merupakan bagian dari MVC yang bertugas menghubungkan langsung antara tugas-tugas dengan proses di Database (Create, Read, Update, Delete). Contoh pada Java, dibuat kelas sendiri antara koneksi dengan query akses data ke database. Jadi untuk menggunakan fungsi Insert, Update, Delete, dan Select kita tinggal memanggil method-method yang menangani hal tersebut.
2. View, merupakan bagian dari MVC yang bertugas menghubungkan antara pengguna dengan tugas perangkat lunak, dengan kata lain View menangani bagian Interface atau Antarmuka untuk mempermudah si pengguna menggunakan perangkat lunak.
3. Controller, menjadi tulang punggung perangkat lunak. Controller menghubungkan antara View dan Model yang meng-handle semua proses-proses yang terjadi didalamnya. Sebagai contoh kasus User Login, tugas Controller adalah mencocokkan data yang di-inputkan user dengan data yang ada di database.
   1. **Listing Program dan Penjelasan**

import Laundry.Login;

public class Main {

public static void main(String[] args) {

Login login = new Login();

}

}

**Listing Program 2.1** *Main* Program

Java class ini digunakan untuk memanggil java class login dan menampilkan bagian awal dari program yang bentuk form login dari program. Java class login akan di tampilkan apabila dipanggil oleh java class ini, apabila tidak memanggil java class login di bagian ini maka java class login tidak akan ditampilkan.

package Laundry;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.event.\*;

import javax.swing.\*;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

public class Login extends JFrame{

private Image image;

JLabel ljudul,lusername,lpassword;

JTextField txusername;

JPasswordField txpassword;

JButton login;

public Login(){

image = new ImageIcon(getClass().getResource("sd.jpg")).getImage();

this.setContentPane(new JPanel() {

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(image, 0, 0, null);

}

});

setTitle("LOGIN");

ljudul = new JLabel("LOGIN ADMIN");

lusername = new JLabel("Username");

lpassword = new JLabel("Password");

txusername = new JTextField("");

txpassword = new JPasswordField("");

login = new JButton("LOGIN");

setLayout(null);

add(ljudul);

add(lusername);

add(lpassword);

add(txusername);

add(txpassword);

add(login);

ljudul.setBounds(100, 20, 80, 23);

lusername.setBounds(25, 60, 80, 23);

txusername.setBounds(130, 60, 120, 23);

lpassword.setBounds(25, 100, 80, 23);

txpassword.setBounds(130, 100, 120, 23);

login.setBounds(100, 150, 70, 23);

setSize(300,250);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

login.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(txusername.getText().equals("admin") && txpassword.getText().equals("admin")){

LaundryMVC mvc = new LaundryMVC();

dispose(); }

else{

JOptionPane.showMessageDialog(null, "Username dan password salah!");

}

} }); } }

**Listing Program 2.2***Login*

*Java class* ini digunakan untuk menampilkan tampilan *login* yang mana merupakan bagian awal dari program. *Java class login* akan di tampilkan berupa *form* untuk menampilkan *username* dan *password*. Apabila *username* dan *password* benar maka ia akan menju ke java calss Laundry MVC yang kemudian akan menampilkan bagian halaman utama dari program ini. Namun, bila *username* dan *password* yang dimasukkan salah maka akan muncul notifikasi "Username dan password salah!" yang mana menandakan bahwa *username* dan *password* salah dan tidak bias masuk ke program.

package Laundry;

public class LaundryMVC {

LaundryView laundryview = new LaundryView();

LaundryModel laundrymodel = new LaundryModel();

LaundryDAO laundrydao = new LaundryDAO();

LaundryController laundrycontroller = new LaundryController(laundrymodel, laundryview, laundrydao);

}

**Listing Program 2.3** Laundry MVC

*Java class* ini digunakan untuk memisahkan antara data, logika dan *interface* yang mana nantinya ia mampu menampilkan tampilan tampilan keseluruhan dari halaman utama program. *Java class* ini mendeklarasikan dan mampu memanggil beberapa *java class* nya seperti bagian *view, model, dao* dan *controllernya*, yang mana semua itu akan berjalan pada tampilan dari program.

package Laundry;

import java.awt.Graphics;

import java.awt.Image;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

public class LaundryView extends JFrame{

private Image image;

JLabel lid;

JTextField txid;

JButton create,update,delete,exit,ambil;

JTable tabel;

DefaultTableModel tableModel;

JScrollPane scrollPane;

Object namaKolom[] = {"ID", "Nama", "Berat", "Harga", "Tanggal Masuk", "Tanggal Estimasi", "Status"};

public LaundryView(){

image = new ImageIcon(getClass().getResource("j.jpg")).getImage();

this.setContentPane(new JPanel() {

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(image, 0, 0, null);

}

});

setTitle("LAUNDRY");

tableModel = new DefaultTableModel(namaKolom,0);

tabel = new JTable(tableModel);

scrollPane = new JScrollPane(tabel);

lid = new JLabel("ID:");

txid = new JTextField("");

create = new JButton("Masukkan Data Baru");

ambil = new JButton("Ambil Laundry");

update = new JButton("Edit");

delete = new JButton("Delete");

exit = new JButton("Exit");

setLayout(null);

add(lid);

add(txid);

add(create);

add(ambil);

add(update);

add(delete);

add(exit);

add(scrollPane);

create.setBounds(40, 265, 175, 30);

ambil.setBounds(40, 320, 175, 30);

lid.setBounds(580, 265, 40, 30);

txid.setBounds(620, 265, 165, 30);

update.setBounds(530,320, 75, 30);

delete.setBounds(620, 320, 75, 30);

exit.setBounds(710, 320, 75, 30);

scrollPane.setBounds(40, 25, 750, 200);

scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS);

setSize(850,420);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

**Listing Program 2.4** Laundry View

*Java class* ini digunakan untuk menampilkan pada tampilan dari halaman utama program, yang mana berisikan tabel dan beberapa *button* untuk melakukan beberapa pilihan.

package Laundry;

public class LaundryModel {

private String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

public void setLaundryModel(String xid, String xnama, String xberat, String xharga, String xtgl\_masuk, String xtgl\_estimasi){

this.id = xid;

this.nama = xnama;

this.berat = xberat;

this.harga = xharga;

this.tgl\_masuk = xtgl\_masuk;

this.tgl\_estimasi = xtgl\_estimasi;

}

//id menggunakan auto increment

public String getId(){

return id;

}

public void setId(String id){

this.id = id;

}

public String getNama(){

return nama;

}

public void setNama(String nama){

this.nama = nama;

}

public String getBerat(){

return berat;

}

public void setBerat(String berat){

this.berat = berat;

}

public String getHarga(){

return harga;

}

public void setHarga(String harga){

this.harga = harga;

}

public String getTgl\_masuk(){

return tgl\_masuk;

}

public void setTgl\_masuk(String tgl\_masuk){

this.tgl\_masuk = tgl\_masuk;

}

public String getTgl\_estimasi(){

return tgl\_estimasi;

}

public void setTgl\_estimasi(String tgl\_estimasi){

this.tgl\_estimasi = tgl\_estimasi;

}

}

**Listing Program 2.5** Laundry Model

*Java class* ini digunakan untuk menyimpan data yang ada pada halaman utama program, yang mana ia menjadi penyimpan program selain ke *database computer* pada program.

package Laundry;

import java.sql.\*;

import javax.swing.JOptionPane;

public class LaundryDAO {

private Connection koneksi;

private Statement statement;

public LaundryDAO(){

try{

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost/projekpbo";

koneksi = DriverManager.getConnection(url, "root", "");

statement = koneksi.createStatement();

} catch (ClassNotFoundException ex){

JOptionPane.showMessageDialog(null, "Class Not Found : " + ex);

} catch (SQLException ex){

JOptionPane.showMessageDialog(null, "SQL Exception : " + ex);

} }

//utk mengambil data dari db dan mengatur ke dalam tabel

public String[][] readLaundry(){

try{

int jmlData = 0; //menampung jmlh data

String data[][] = new String[getJmldata()][7];

//pengambilan dta dlm java dari db

String query = "SELECT \* FROM laundry";

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next()){

data[jmlData][0] = resultSet.getString("id");

data[jmlData][1] = resultSet.getString("nama");

data[jmlData][2] = resultSet.getString("berat");

data[jmlData][3] = resultSet.getString("harga");

data[jmlData][4] = resultSet.getString("tgl\_masuk");

data[jmlData][5] = resultSet.getString("tgl\_estimasi");

data[jmlData][6] = resultSet.getString("status");

jmlData++;

}

return data;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return null;

} }

public int getJmldata(){

int jmlData = 0;

try{

String query = "SELECT \* FROM laundry";

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next()){

jmlData++;

}

return jmlData;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return 0;

} }

public int getIDJmldata(String ID){

int jmlData = 0;

try{

String query = "SELECT \* FROM laundry WHERE id='"+ID+"'";

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next()){

jmlData++;

}

return jmlData;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return 0;

} }

public void Delete(LaundryModel Model) {

try{

String query = "DELETE FROM laundry WHERE id = '"+Model.getId()+"'";

statement.executeUpdate(query);

JOptionPane.showMessageDialog(null, "Berhasil dihapus");

} catch (SQLException sql){

System.out.println(sql.getMessage());

}

}

}

**Listing Program 2.6** Laundry DAO

*Java class* ini digunakan untuk menyimpan data dan mengambil data dari *database* yang nantinya akan ditampilakan pada halaman utama program. *Java class* ini akan bekerja setelah dihubungkan oleh *Java class Controller*.

package Laundry;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

public class LaundryController {

LaundryModel laundrymodel;

LaundryView laundryview;

LaundryDAO laundrydao;

static String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

//,tgl\_ambil,status,bayar,kembali,search;

public LaundryController(LaundryModel laundrymodel, LaundryView laundryview, LaundryDAO laundrydao){

this.laundrymodel = laundrymodel;

this.laundryview = laundryview;

this.laundrydao = laundrydao;

if(laundrydao.getJmldata() != 0){

String dataLaundry[][] = laundrydao.readLaundry();

laundryview.tabel.setModel((new JTable(dataLaundry, laundryview.namaKolom)).getModel());

} else {

JOptionPane.showMessageDialog(null, "Data tidak ada!");

}

laundryview.create.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new InputMVC();

laundryview.setVisible(false);

}

});

laundryview.ambil.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new AmbilMVC();

laundryview.setVisible(false);

}

});

laundryview.update.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new EditMVC();

laundryview.setVisible(false);

}

});

laundryview.tabel.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

super.mouseClicked(e);

int baris = laundryview.tabel.getSelectedRow();

int kolom = laundryview.tabel.getSelectedColumn();

id = laundryview.tabel.getValueAt(baris,0).toString();

}

});

laundryview.delete.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

int input = JOptionPane.showConfirmDialog(null, "Apakah anda ingin menghapus data " + id + "?","Delete",JOptionPane.YES\_NO\_OPTION);

if (input == 0){

laundrymodel.setId(id);

laundrydao.Delete(laundrymodel); //menghapus data yg sudah ada

String dataLaundry[][] = laundrydao.readLaundry();

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

}

}

});

laundryview.exit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

//keluar dari program

System.exit(0);

}

});

laundryview.txid.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void removeUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void insertUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void changedUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

}

public void change() {

String ID = laundryview.txid.getText();

if(ID.isEmpty()) {

String dataLaundry[][] = laundrydao.readLaundry();

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

}

else {

String dataLaundry[][] = laundrydao.readIDLaundry(laundryview);

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

}

}

});

}

}

laundryview.delete.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

int input = JOptionPane.showConfirmDialog(null, "Apakah anda ingin menghapus data " + id + "?","Delete",JOptionPane.YES\_NO\_OPTION);

if (input == 0){

laundrymodel.setId(id);

laundrydao.Delete(laundrymodel); //menghapus data yg sudah ada

String dataLaundry[][] = laundrydao.readLaundry();

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

} }

});

laundryview.exit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

//keluar dari program

System.exit(0);

}

});

laundryview.txid.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void removeUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change(); }

@Override

public void insertUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change(); }

@Override

public void changedUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

}

public void change() {

String ID = laundryview.txid.getText();

if(ID.isEmpty()) {

String dataLaundry[][] = laundrydao.readLaundry();

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

}

else {

String dataLaundry[][] = laundrydao.readIDLaundry(laundryview);

laundryview.tabel.setModel((new JTable (dataLaundry, laundryview.namaKolom)).getModel());

}

}

});

}

}

**Listing Program 2.7** Laundry Controller

*Java class* ini digunakan untuk menghubungkan data dari *database* ke program yang nantinya akan ditampilakan pada halaman utama program dan data yang akan disimpan dari halaman utama program. *Java class* ini akan bekerja dengan menghubungkan *java*  dengan *database* hingga dapat menampilkan data-data yang telah ada di dalam *database* dan juga akan menyimpan data-data yang akan disimpan pada *database*. Selain itu *controller* juga berfungsi untuk refresh otomatis untuk menampilkan data yang terbaru.

package Laundry;

public class InputMVC {

InputView inputview = new InputView();

InputModel inputmodel = new InputModel();

InputDAO inputdao = new InputDAO();

InputController inputcontroller = new InputController(inputmodel, inputview, inputdao);

}

**Listing Program 2.8** Input MVC

*Java class* ini digunakan untuk memisahkan antara data, logika dan *interface* yang mana nantinya ia mampu menampilkan tampilan tampilan keseluruhan dari halaman *input* data pada program. *Java class* ini mendeklarasikan dan mampu memanggil beberapa *java class* nya seperti bagian *view, model, dao* dan *controllernya*, yang mana semua itu akan berjalan pada tampilan dari program *input* data.

package Laundry;

import java.awt.Graphics;

import java.awt.Image;

import javax.swing.\*;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

public class InputView extends JFrame{

private Image image;

JLabel lnama,lberat,lharga,ltgl\_masuk,ltgl\_est,judul;

JTextField txnama,txberat,txharga,txtgl\_masuk,txtgl\_est;

JButton create,hitung,kembali;

public InputView(){

image = new ImageIcon(getClass().getResource("z.jpg")).getImage();

this.setContentPane(new JPanel() {

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(image, 0, 0, null);

}

});

setTitle("INPUT DATA LAUNDRY");

judul = new JLabel("INPUT DATA LAUNDRY");

lnama = new JLabel("Nama");

lberat = new JLabel("Berat");

lharga = new JLabel("Harga");

ltgl\_masuk = new JLabel("Tanggal Masuk");

ltgl\_est = new JLabel("Tanggal Estimasi");

txnama = new JTextField("");

txberat = new JTextField("");

txharga = new JTextField("");

txtgl\_masuk = new JTextField("");

txtgl\_est = new JTextField("");

create = new JButton("Simpan");

kembali = new JButton ("Kembali");

setLayout(null);

add(judul);

add(lnama);

add(lberat);

add(lharga);

add(ltgl\_masuk);

add(ltgl\_est);

add(txnama);

add(txberat);

add(txharga);

add(txtgl\_masuk);

add(txtgl\_est);

add(create);

add(kembali);

judul.setBounds(130,10,300,25);

lnama.setBounds(50, 50, 50, 20);

txnama.setBounds(170, 50, 150, 20);

lberat.setBounds(50, 75, 50, 20);

txberat.setBounds(170, 75, 150, 20);

lharga.setBounds(50, 100, 50, 20);

txharga.setBounds(170, 100, 150, 20);

ltgl\_masuk.setBounds(50, 125, 150, 20);

txtgl\_masuk.setBounds(170, 125, 150, 20);

ltgl\_est.setBounds(50, 150, 150, 20);

txtgl\_est.setBounds(170, 150, 150, 20);

create.setBounds(75, 200, 100, 25);

kembali.setBounds(200,200, 100, 25);

setSize(400,300);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE); }

public String getNama(){

return txnama.getText();

}

public String getBerat(){

return txberat.getText();

}

public String getHarga(){

return txharga.getText();

}

public String getTgl\_masuk(){

return txtgl\_masuk.getText();

}

public String getTgl\_estimasi(){

return txtgl\_est.getText();

} }

**Listing Program 2.9** Input View

*Java class* ini digunakan untuk menampilkan pada tampilan dari halaman *input* data pada program ini, yang mana berisikan *form* untuk mengisi data pada *database* dan untuk ditampilkan pada tabel halaman utama.

package Laundry;

public class InputModel {

private String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

public void setInputModel(String xnama, String xberat, String xharga, String xtgl\_masuk, String xtgl\_estimasi){

this.nama = xnama;

this.berat = xberat;

this.harga = xharga;

this.tgl\_masuk = xtgl\_masuk;

this.tgl\_estimasi = xtgl\_estimasi;

} //id menggunakan auto increment

public String getId(){

return id;

}

public void setId(String id){

this.id = id;

}

public String getNama(){

return nama;

}

public void setNama(String nama){

this.nama = nama;

}

public String getBerat(){

return berat;

}

public void setBerat(String berat){

this.berat = berat;

}

public String getHarga(){

return harga;

}

public void setHarga(String harga){

this.harga = harga;

}

public String getTgl\_masuk(){

return tgl\_masuk;

}

public void setTgl\_masuk(String tgl\_masuk){

this.tgl\_masuk = tgl\_masuk;

}

public String getTgl\_estimasi(){

return tgl\_estimasi;

}

public void setTgl\_estimasi(String tgl\_estimasi){

this.tgl\_estimasi = tgl\_estimasi;

} }

**Listing Program 2.10** Input Model

*Java class* ini digunakan untuk menyimpan data yang ada pada halaman input data, yang mana ia menjadi penyimpan program selain ke *database computer* pada program.

package Laundry;

import java.sql.\*;

import javax.swing.JOptionPane;

public class InputDAO {

private Connection koneksi;

private Statement statement;

public InputDAO(){

try{

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost/projekpbo";

koneksi = DriverManager.getConnection(url, "root", "");

statement = koneksi.createStatement();

} catch (ClassNotFoundException ex){

JOptionPane.showMessageDialog(null, "Class Not Found : " + ex);

} catch (SQLException ex){

JOptionPane.showMessageDialog(null, "SQL Exception : " + ex);

}

}

public void Insert(InputModel Model){

try{

String query = "INSERT INTO laundry VALUES (NULL,'"+Model.getNama()+"','"+

Model.getBerat()+"','"+Model.getHarga()+"','"+Model.getTgl\_masuk()+

"','"+Model.getTgl\_estimasi()+"','"+"-"+"','"+"Proses"+"')";

statement.executeUpdate(query); //execute querynya

JOptionPane.showMessageDialog(null, "Data disimpan");

} catch (Exception sql){

JOptionPane.showMessageDialog(null, sql.getMessage());

}

}

}

**Listing Program 2.11** Input DAO

*Java class* ini digunakan untuk menyimpan data dan mengambil data dari *database* yang nantinya akan ditampilakan pada halaman input data dari program. *Java class* ini akan bekerja setelah dihubungkan oleh *Java class Controller*.

package Laundry;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

public class InputController {

InputModel inputmodel;

InputView inputview;

InputDAO inputdao;

static String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

//,tgl\_ambil,status,bayar,kembali,search;

public InputController(InputModel inputmodel, InputView inputview, InputDAO inputdao){

this.inputmodel = inputmodel;

this.inputview = inputview;

this.inputdao = inputdao;

inputview.create.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String nama = inputview.getNama();

String berat = inputview.getBerat();

String harga = inputview.getHarga();

String tgl\_masuk = inputview.getTgl\_masuk();

String tgl\_estimasi = inputview.getTgl\_estimasi();

if(nama.isEmpty() || berat.isEmpty() || harga.isEmpty() || tgl\_masuk.isEmpty() || tgl\_estimasi.isEmpty()){

JOptionPane.showMessageDialog(null, "Harap isi semua field");

} else {

inputmodel.setInputModel(nama,berat,harga,tgl\_masuk,tgl\_estimasi);

inputdao.Insert(inputmodel); //membuat data baru

inputview.txnama.setText("");

inputview.txberat.setText("");

inputview.txharga.setText("");

inputview.txtgl\_masuk.setText("");

inputview.txtgl\_est.setText("");

}

}

});

inputview.kembali.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new LaundryMVC();

inputview.setVisible(false);

}

});

inputview.txberat.getDocument().addDocumentListener(new

DocumentListener() {

@Override

public void removeUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void insertUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void changedUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

//change();

}

public void change() {

if(inputview.txberat.getText().isEmpty()) {

}

else {

String berat = inputview.txberat.getText();

int iberat = Integer.parseInt(berat);

iberat \*= 5000;

berat = Integer.toString(iberat);

inputview.txharga.setText(berat);

}

}

});

}

}

**Listing Program 2.12** Input Controller

*Java class* ini digunakan untuk menghubungkan data antara *database* dan program yang mana ia akan meyimpannya dari halaman input program ke dalam *database*. *Java class* ini akan bekerja dengan menghubungkan *java*  dengan *database* hingga dapat menampilkan data-data yang telah ada di dalam *database* dan juga akan menyimpan data-data yang akan disimpan pada *database*. Selain itu *controller* juga berfungsi untuk melakukan refresh otomatis untuk menampilkan data yang terbaru.

package Laundry;

public class EditMVC {

EditView editview = new EditView();

EditModel editmodel = new EditModel();

EditDAO editdao = new EditDAO();

EditController editcontroller = new EditController(editmodel, editview, editdao);

}

**Listing Program 2.13** Edit MVC

*Java class* ini digunakan untuk memisahkan antara data, logika dan *interface* yang mana nantinya ia mampu menampilkan tampilan tampilan keseluruhan dari halaman *edit* data pada program. *Java class* ini mendeklarasikan dan mampu memanggil beberapa *java class* nya seperti bagian *view, model, dao* dan *controllernya*, yang mana semua itu akan berjalan pada tampilan dari program *edit* data.

package Laundry;

import java.awt.Graphics;

import java.awt.Image;

import javax.swing.\*;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

public class EditView extends JFrame{

private Image image;

JLabel lnama,lberat,lharga,ltgl\_masuk,ltgl\_est,judul,lid;

JTextField txnama,txberat,txharga,txtgl\_masuk,txtgl\_est,txid;

JButton create,hitung,kembali,search;

public EditView(){

image = new ImageIcon(getClass().getResource("z.jpg")).getImage();

this.setContentPane(new JPanel() {

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(image, 0, 0, null);

} });

setTitle("EDIT DATA LAUNDRY");

judul = new JLabel("EDIT DATA LAUNDRY");

lid = new JLabel("ID:");

lnama = new JLabel("Nama");

lberat = new JLabel("Berat");

lharga = new JLabel("Harga");

ltgl\_masuk = new JLabel("Tanggal Masuk");

ltgl\_est = new JLabel("Tanggal Estimasi");

txid = new JTextField("");

txnama = new JTextField("");

txberat = new JTextField("");

txharga = new JTextField("");

txtgl\_masuk = new JTextField("");

txtgl\_est = new JTextField("");

create = new JButton("Simpan");

kembali = new JButton ("Kembali");

search = new JButton ("Cari");

setLayout(null);

add(judul);

add(lid);

add(search);

add(lnama);

add(lberat);

add(lharga);

add(ltgl\_masuk);

add(ltgl\_est);

add(txid);

add(txnama);

add(txberat);

add(txharga);

add(txtgl\_masuk);

add(txtgl\_est);

add(create);

add(kembali);

judul.setBounds(130,10,300,25);

lid.setBounds(50, 50, 50, 20);

txid.setBounds(170, 50, 70, 20);

search.setBounds(250, 50, 70, 20);

lnama.setBounds(50, 80, 50, 20);

txnama.setBounds(170, 80, 150, 20);

lberat.setBounds(50, 105, 50, 20);

txberat.setBounds(170, 105, 150, 20);

lharga.setBounds(50, 130, 50, 20);

txharga.setBounds(170, 130, 150, 20);

ltgl\_masuk.setBounds(50, 155, 150, 20);

txtgl\_masuk.setBounds(170, 155, 150, 20);

ltgl\_est.setBounds(50, 180, 150, 20);

txtgl\_est.setBounds(170, 180, 150, 20);

create.setBounds(75, 230, 100, 25);

kembali.setBounds(200,230, 100, 25);

setSize(400,330);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public String getNama(){

return txnama.getText();

}

public String getBerat(){

return txberat.getText();

}

public String getHarga(){

return txharga.getText();

}

public String getTgl\_masuk(){

return txtgl\_masuk.getText();

}

public String getTgl\_estimasi(){

return txtgl\_est.getText();

}

}

**Listing Program 2.14** Edit View

*Java class* ini digunakan untuk menampilkan pada tampilan dari halaman *edit* data pada program ini, yang mana berisikan *form* untuk mengedit data pada *database* dan untuk ditampilkan pada tabel halaman utama. Diawali dengan pencarian *id* untuk menemukan datanya, kemudian data dapat diganti yang nantinya data *edit* akan otomatis di *update* ke *database* dan halaman utama program.

package Laundry;

public class EditModel {

private String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

;

public void setLaundryModel(String xid, String xnama, String xberat, String xharga, String xtgl\_masuk, String xtgl\_estimasi){

this.id = xid;

this.nama = xnama;

this.berat = xberat;

this.harga = xharga;

this.tgl\_masuk = xtgl\_masuk;

this.tgl\_estimasi = xtgl\_estimasi;

//id menggunakan auto increment

public String getId(){

return id;

}

public void setId(String id){

this.id = id;

}

public String getNama(){

return nama;

}

public void setNama(String nama){

this.nama = nama;

}

public String getBerat(){

return berat;

}

public void setBerat(String berat){

this.berat = berat;

}

public String getHarga(){

return harga;

}

public void setHarga(String harga){

this.harga = harga;

}

public String getTgl\_masuk(){

return tgl\_masuk;

}

public void setTgl\_masuk(String tgl\_masuk){

this.tgl\_masuk = tgl\_masuk;

}

public String getTgl\_estimasi(){

return tgl\_estimasi;

}

public void setTgl\_estimasi(String tgl\_estimasi){

this.tgl\_estimasi = tgl\_estimasi;

}

**Listing Program 2.15** Edit Model

*Java class* ini digunakan untuk menyimpan data yang ada pada halaman *edit* data, yang mana ia menjadi penyimpan program selain ke *database computer* pada program.

package Laundry;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class EditDAO {

private Connection koneksi;

private Statement statement;

public EditDAO(){

try{

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost/projekpbo";

koneksi = DriverManager.getConnection(url, "root", "");

statement = koneksi.createStatement();

} catch (ClassNotFoundException ex){

JOptionPane.showMessageDialog(null, "Class Not Found : " + ex);

} catch (SQLException ex){

JOptionPane.showMessageDialog(null, "SQL Exception : " + ex);

}

}

public String[][] readLaundry(EditView view){

try{

String ID = view.txid.getText();

int jmlData = 0; //menampung jmlh data

String data[][] = new String[1][6];

//pengambilan dta dlm java dari db

String query = "SELECT \* FROM laundry WHERE id='"+ID+"'";

ResultSet resultSet = statement.executeQuery(query);

if(resultSet.next()){

data[jmlData][0] = resultSet.getString("id");

data[jmlData][1] = resultSet.getString("nama");

data[jmlData][2] = resultSet.getString("berat");

data[jmlData][3] = resultSet.getString("harga");

data[jmlData][4] = resultSet.getString("tgl\_masuk");

data[jmlData][5] = resultSet.getString("tgl\_estimasi");

jmlData++;

} else {

JOptionPane.showMessageDialog(null, "Data tidak ditemukan");

view.txid.setText("");

}

return data;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return null;

}

}

public void update(EditView view) {

try {

String ID = view.txid.getText();

String a = view.txnama.getText();

String b = view.txberat.getText();

String c = view.txharga.getText();

String d = view.txtgl\_masuk.getText();

String e = view.txtgl\_est.getText();

String query = "UPDATE laundry SET nama='"+a+"' , berat='"+b+"' , harga='"+c+"' , tgl\_masuk='"+d+"' , tgl\_estimasi='"+e+"' WHERE id='"+ID+"'";

statement.execute(query);

JOptionPane.showMessageDialog(null, "Data disimpan");

}catch(Exception e) {

JOptionPane.showInputDialog(e.getMessage());

}

}

}

**Listing Program 2.16** Edit DAO

*Java class* ini digunakan untuk menyimpan data dan mengambil data dari *database* yang nantinya akan ditampilakan pada halaman *edit*  data dari program. *Java class* ini akan bekerja setelah dihubungkan oleh *Java class Controller*.

package Laundry;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

public class EditController {

EditModel editmodel;

EditView editview;

EditDAO editdao;

static String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

public EditController(EditModel editmodel , EditView editview , EditDAO editdao) {

this.editmodel = editmodel;

this.editview = editview;

this.editdao = editdao;

editview.search.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

String data[][] = new String[1][6];

data = editdao.readLaundry(editview);

editview.txnama.setText(data[0][1]);

editview.txberat.setText(data[0][2]);

editview.txharga.setText(data[0][3]);

editview.txtgl\_masuk.setText(data[0][4]);

editview.txtgl\_est.setText(data[0][5]);

}

});

editview.create.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

editdao.update(editview);

editview.txid.setText("");

editview.txnama.setText("");

editview.txberat.setText("");

editview.txharga.setText("");

editview.txtgl\_masuk.setText("");

editview.txtgl\_est.setText("");

}

});

editview.kembali.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new LaundryMVC();

editview.setVisible(false);

}

});

editview.txberat.getDocument().addDocumentListener(new DocumentListener() {

@Override

public void removeUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void insertUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

change();

}

@Override

public void changedUpdate(DocumentEvent e) {

// TODO Auto-generated method stub

//change();

}

public void change() {

if(editview.txberat.getText().isEmpty()) {

}

else {

String berat = editview.txberat.getText();

int iberat = Integer.parseInt(berat);

iberat \*= 5000;

berat = Integer.toString(iberat);

editview.txharga.setText(berat);

}

}

});

}

}

**Listing Program 2.17** Edit Controller

*Java class* ini digunakan untuk menghubungkan data antara *database* dan program yang mana ia akan meyimpannya dari halaman *edit* program ke dalam *database*. *Java class* ini akan bekerja dengan menghubungkan *java*  dengan *database* hingga dapat menampilkan data-data yang telah ada di dalam *database* dan juga akan menyimpan data-data yang akan disimpan pada *database*. Selain itu *controller* juga berfungsi untuk melakukan refresh otomatis untuk menampilkan data yang terbaru.

package Laundry;

public class AmbilMVC {

AmbilView ambilview = new AmbilView();

AmbilModel ambilmodel = new AmbilModel();

AmbilDAO ambildao = new AmbilDAO();

AmbilController ambilcontroller = new AmbilController(ambilmodel, ambilview, ambildao);

}

**Listing Program 2.18** Ambil MVC

*Java class* ini digunakan untuk memisahkan antara data, logika dan *interface* yang mana nantinya ia mampu menampilkan tampilan tampilan keseluruhan dari halaman ambil pada program. *Java class* ini mendeklarasikan dan mampu memanggil beberapa *java class* nya seperti bagian *view, model, dao* dan *controllernya*, yang mana semua itu akan berjalan pada tampilan dari program ambil.

package Laundry;

import java.awt.Graphics;

import java.awt.Image;

import javax.swing.\*;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

public class AmbilView extends JFrame{

private Image image;

JLabel lid,lnama,lberat,lharga,ltgl\_masuk,ltgl\_ambil,lbayar,lkembali,judul;

JTextField txid,txnama,txberat,txharga,txtgl\_masuk,txtgl\_ambil,txbayar,txkembali;

JButton simpan,hitung,kembali,search;

public AmbilView(){

image = new ImageIcon(getClass().getResource("z.jpg")).getImage();

this.setContentPane(new JPanel() {

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(image, 0, 0, null);

}

});

setTitle("AMBIL LAUNDRY");

judul = new JLabel("AMBIL LAUNDRY");

lid = new JLabel("ID:");

lnama = new JLabel("Nama");

lberat = new JLabel("Berat");

lharga = new JLabel("Harga");

tgl\_masuk = new JLabel("Tanggal Masuk");

ltgl\_ambil = new JLabel("Tanggal Ambil");

lbayar = new JLabel("Bayar");

lkembali = new JLabel("Kembali");

txid = new JTextField("");

txnama = new JTextField("");

txberat = new JTextField("");

txharga = new JTextField("");

txtgl\_masuk = new JTextField("");

txtgl\_ambil = new JTextField("");

txbayar = new JTextField("");

txkembali = new JTextField("");

simpan = new JButton("Simpan");

hitung = new JButton("Hitung");

kembali = new JButton ("Kembali");

search = new JButton ("Cari");

setLayout(null);

add(judul);

add(lid);

add(lnama);

add(lberat);

add(lharga);

add(ltgl\_masuk);

add(ltgl\_ambil);

add(lbayar);

add(lkembali);

add(txid);

add(txnama);

add(txberat);

add(txharga);

add(txtgl\_masuk);

add(txtgl\_ambil);

add(txbayar);

add(txkembali);

add(simpan);

add(hitung);

add(kembali);

add(search);

judul.setBounds(220,10,300,20);

lid.setBounds(30, 50, 50, 20);

txid.setBounds(130, 50, 120, 20);

search.setBounds(260, 50, 70, 20);

lnama.setBounds(30, 85, 50, 20);

txnama.setBounds(130, 85, 120, 20);

lberat.setBounds(340, 85, 50, 20);

txberat.setBounds(400, 85, 90, 20);

ltgl\_masuk.setBounds(30, 115, 150, 20);

txtgl\_masuk.setBounds(130, 115, 120, 20);

lharga.setBounds(340, 115, 50, 20);

txharga.setBounds(400, 115, 90, 20);

ltgl\_ambil.setBounds(30, 145, 150, 20);

txtgl\_ambil.setBounds(130, 145, 120, 20);

lbayar.setBounds(30, 175, 150, 20);

txbayar.setBounds(130, 175, 120, 20);

hitung.setBounds(260, 175, 70, 20);

lkembali.setBounds(340, 175, 150, 20);

txkembali.setBounds(400, 175, 90, 20);

simpan.setBounds(180, 225, 80, 25);

kembali.setBounds(280, 225, 80, 25);

setSize(540,320);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

**Listing Program 2.19** Ambil View

*Java class* ini digunakan untuk menampilkan pada tampilan dari halamanambil pada program ini, yang mana berisikan *form* untuk mengambil *laundry*. Diawali dengan pencarian *id* untuk menemukan datanya, kemudian data tanggal pengambilan disi untuk dapat mengubah *status* data dari *proses* menjadi *complete* pada halaman utama program. Selain mengisi data pengambilan, diisi juga disi pembayaran untuk menyelesaikan transaksi pada program *laundry* ini.

Package Laundry;

public class AmbilModel {

private String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

public void setLaundryModel(String xid, String xnama, String xberat, String xharga, String xtgl\_masuk, String xtgl\_estimasi){

this.id = xid;

this.nama = xnama;

this.berat = xberat;

this.harga = xharga;

this.tgl\_masuk = xtgl\_masuk;

this.tgl\_estimasi = xtgl\_estimasi;

//id menggunakan auto increment

public String getId(){

return id;

}

public void setId(String id){

this.id = id;

}

public String getNama(){

return nama;

}

public void setNama(String nama){

this.nama = nama;

}

public String getBerat(){

return berat;

}

public void setBerat(String berat){

this.berat = berat;

}

public String getHarga(){

return harga;

}

public void setHarga(String harga){

this.harga = harga;

}

public String getTgl\_masuk(){

return tgl\_masuk;

}

public void setTgl\_masuk(String tgl\_masuk){

this.tgl\_masuk = tgl\_masuk;

}

public String getTgl\_estimasi(){

return tgl\_estimasi;

}

public void setTgl\_estimasi(String tgl\_estimasi){

this.tgl\_estimasi = tgl\_estimasi;

}

**Listing Program 2.20** Ambil Model

*Java class* ini digunakan untuk menyimpan data yang ada pada halaman ambil *laundry*, yang mana ia menjadi penyimpan program selain ke *database computer* pada program.

package Laundry;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class AmbilDAO {

private Connection koneksi;

private Statement statement;

public AmbilDAO(){

try{

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost/projekpbo";

koneksi = DriverManager.getConnection(url, "root", "");

statement = koneksi.createStatement();

//JOptionPane.showMessageDialog(null, "Koneksi Berhasil");

} catch (ClassNotFoundException ex){

JOptionPane.showMessageDialog(null, "Class Not Found : " + ex);

} catch (SQLException ex){

JOptionPane.showMessageDialog(null, "SQL Exception : " + ex);

}

}

public String[][] readLaundry(AmbilView view){

try{

String ID = view.txid.getText();

int jmlData = 0; //menampung jmlh data

String data[][] = new String[1][6];

//pengambilan dta dlm java dari db

id");

data[jmlData][1] = resultSet.getString("nama");

data[jmlData][2] = resultSet.getString("berat");

data[jmlData][3] = resultSet.getString("harga");

data[jmlData][4] = resultSet.getString("tgl\_masuk");

data[jmlData][5] = resultSet.getString("tgl\_estimasi");

jmlData++;

} else {

JOptionPane.showMessageDialog(null, "Data tidak ditemukan");

view.txid.setText("");

}

return data;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return null;

}

}

String query = "SELECT \* FROM laundry WHERE id='"+ID+"'";

ResultSet resultSet = statement.executeQuery(query);

if(resultSet.next()){

data[jmlData][0] = resultSet.getString("id");

data[jmlData][1] = resultSet.getString("nama");

data[jmlData][2] = resultSet.getString("berat");

data[jmlData][3] = resultSet.getString("harga");

data[jmlData][4] = resultSet.getString("tgl\_masuk");

data[jmlData][5] = resultSet.getString("tgl\_estimasi");

jmlData++;

} else {

JOptionPane.showMessageDialog(null, "Data tidak ditemukan");

view.txid.setText("");

}

return data;

} catch (SQLException e){

System.out.println(e.getMessage());

System.out.println("SQL Error");

return null;

}

}

public void update(AmbilView view) {

try {

String ID = view.txid.getText();

String a = view.txtgl\_ambil.getText();

String query = "UPDATE laundry SET status='Completed' , tgl\_ambil='"+a+"' WHERE id='"+ID+"'";

statement.execute(query);

JOptionPane.showMessageDialog(null, "Data disimpan");

}catch(Exception e) {

JOptionPane.showInputDialog(e.getMessage());

}

}

}

**Listing Program 2.21** Ambil DAO

*Java class* ini digunakan untuk menyimpan data dan mengambil data dari *database* yang nantinya akan ditampilakan pada halamanambil *laundry* dari program. *Java class* ini akan bekerja setelah dihubungkan oleh *Java class Controller*.

package Laundry;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class AmbilController {

AmbilModel ambilmodel;

AmbilView ambilview;

AmbilDAO ambildao;

static String id,nama,berat,harga,tgl\_masuk,tgl\_estimasi;

//,tgl\_ambil,status,bayar,kembali,search;

public AmbilController( AmbilModel ambilmodel , AmbilView ambilview , AmbilDAO ambildao) {

this.ambilmodel = ambilmodel;

this.ambilview = ambilview;

this.ambildao = ambildao;

ambilview.search.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

String data[][] = new String[1][6];

data = ambildao.readLaundry(ambilview);

ambilview.txnama.setText(data[0][1]);

ambilview.txberat.setText(data[0][2]);

ambilview.txharga.setText(data[0][3]);

ambilview.txtgl\_masuk.setText(data[0][4]);

}

});

ambilview.hitung.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

int bayar = Integer.parseInt(ambilview.txbayar.getText());

int harga = Integer.parseInt(ambilview.txharga.getText());

int kembalian = bayar - harga;

ambilview.txkembali.setText(Integer.toString(kembalian));

}

});

ambilview.simpan.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

ambildao.update(ambilview);

ambilview.txid.setText("");

ambilview.txnama.setText("");

ambilview.txberat.setText("");

ambilview.txharga.setText("");

ambilview.txtgl\_masuk.setText("");

ambilview.txtgl\_ambil.setText("");

ambilview.txbayar.setText("");

ambilview.txkembali.setText("");

}});

ambilview.kembali.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new LaundryMVC();

ambilview.setVisible(false);

}

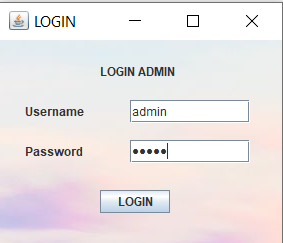
});

}

**Listing Program 2.22** Ambil Controller

* 1. **Output**

1. Login



**Gambar 2.1** Login

1. Halaman Utama

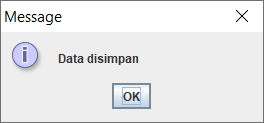


**Gambar 2.2** Halaman Utama

1. Proses Memasukkan Data Laundry Baru



**Gambar 2.3** Input Data



**Gambar 2.4** Notifikasi Data Tersimpan

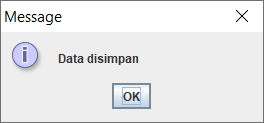


**Gambar 2.5** Output Data di halaman Utama

1. Proses Edit Data Laundry



**Gambar 2.6** Edit Data



**Gambar 2.7** Notifikasi Data Tersimpan

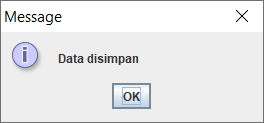


**Gambar 2.8** Output Data di halaman Utama

1. Proses Ambil Laundry



**Gambar 2.9** Ambil Laundry

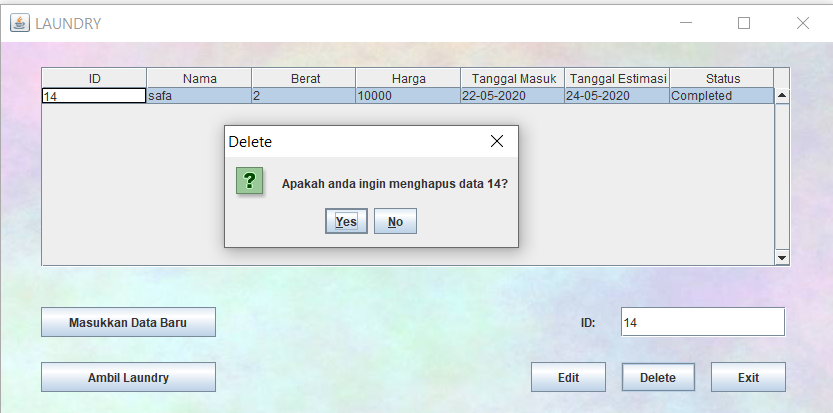


**Gambar 2.10** Notifikasi Data Tersimpan

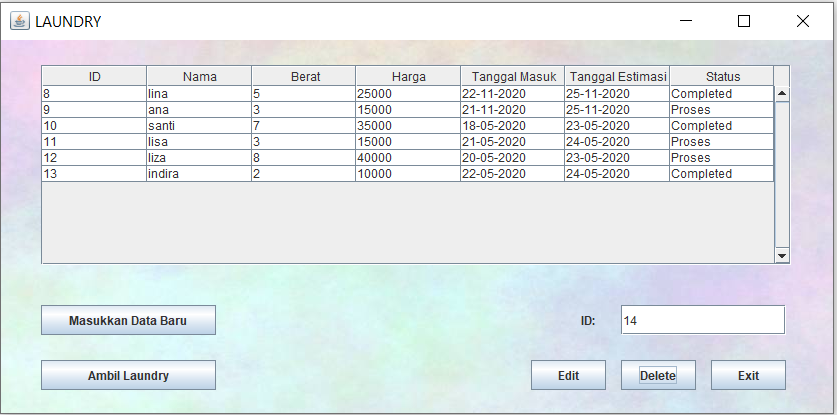


**Gambar 2.11** Output Data di halaman Utama

1. Proses Cari dan Hapus Data Laundry



**Gambar 2.12** Ambil Laundry



**Gambar 2.13** Ambil Laundry

**BAB III**

**JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS**

* 1. **Jadwal Pengerjaan**

Berikut merupakan jadwal pengerjaan yang kami lakukan untuk meyelesaikan proyek ini:

**Tabel 3.1** Tabel Jadwal Pengerjaan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Tugas** | **Minggu Ke-** | |
| **Mei** | |
| **1** | **2** |
| 1. | Penentuan konsep proyek akhir |  |  |
| 2. | Perancangan ide |  |  |
| 3. | Perancangan program |  |  |
| 4. | Pembuatan Program |  |  |
| 5. | Penyelesaian dan testing |  |  |
| 6. | Pengerjaan laporan |  |  |

* 1. **Pembagian Tugas**

Berikut merupakan pembagian tugas yang kita lakukan untuk menyelesaikan proyek ini:

**Tabel 3.2** Tabel Pembagian Tugas

|  |  |  |
| --- | --- | --- |
| **No** | **Aktivitas** | **Penanggung Jawab** |
| 1. | Penentuan tema dan judul | Ina, Nadia |
| 2. | Perancangan ide | Ina, Nadia |
| 3. | Perancangan program | Ina,Nadia |
| 4. | Pembuatan Program | Ina,Nadia |
| 5. | Penyelesaian dan testing | Ina,Nadia |
| 6. | Pengerjaan laporan | Ina,Nadia |

**BAB IV**

**KESIMPULAN DAN SARAN**

1. **Kesimpulan**

Dengan adanya program ini, proses pendataan hingga pengarsipan data *laundry* dapat dengan mudah dilakukan karena menggunakan sistem berbasis komputer. Segala proses bisnis dalam usaha *laundry* dapat dilakukan dengan lebih cepat dan efektif dengan hasil yang lebih akurat. Pengolahan data berbasis komputer ini terbukti lebih teliti, aman, dan terhindar dari hal-hal yang tidak diinginkan sehingga data tersebut dapat dilihat kembali di lain waktu jika perlu ditinjau lebih dalam.

1. **Saran**

Dalam pengembangannya, program ini dapat dimanfaatkan dalam operasional laundry di kehidupan nyata sehingga dapat membantu pendataan pelanggan dengan lebih mudah dan mengorganisir pengarsipan data bisnis laundry dengan lebih baik.

**DAFTAR PUSTAKA**

Penyusun. 2019. *MODUL PRAKTIKUM Pemrograman Berorientasi Objek*. Yogyakarta: UPN Veteran Yogyakarta.