

**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

LAPORAN PROYEK AKHIR

PROGRAM PENYEWAAN ZOOM



DISUSUN OLEH :

ANGGOTA	:	REYHAN DEVARA S.	124190023
		REIHAN HERIANTONO	124190034
KELAS/PLUG	:	A	
NAMA ASISTEN	:	RAMA TRI AGUNG	
		LIKE ALLYANDARU	

**PROGRAM STUDI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA
2020**

HALAMAN PENGESAHAN

LAPORAN PROYEK AKHIR

Disusun oleh :

Reyhan Devara Samudra

124190023

Reihan Heriantono

124190034

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Pemrograman Berorientasi Objek

Pada Tanggal :

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Rama Tri Agung
NIM. 123180053

Like Allyandaru
NIM. 123180054

Mengetahui,

Ka. Lab. Digital

Mangaras Yanu Florestivanto, S.T., M.Eng.
NIK. 2 8201 13 0425 1

KATA PENGANTAR

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

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

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 20 Januari 2021

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
BAB II PEMBAHASAN	2
2.1 Dasar Teori	2
2.2 Deskripsi Umum Proyek Akhir.....	8
2.3 Inti Pembahasan	9
BAB III JADWAL Pengerjaan dan Pembagian Tugas	30
3.1 Jadwal Pengerjaan	30
3.2 Pembagian Tugas	30
BAB IV KESIMPULAN DAN SARAN.....	31
4.1 Kesimpulan	31
4.2 Saran.....	31
DAFTAR PUSTAKA	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di masa pandemi seperti sekarang ini masyarakat tidak mudah untuk dapat melakukan kegiatan secara tatap muka dan mengumpulkan banyak. Banyak kegiatan yang harus dilakukan secara virtual seperti seminar dan kelas *online*. Terdapat beberapa pilihan platform yang dapat digunakan, tetapi yang menjadi favorit masyarakat kebanyakan adalah Zoom. Untuk menggunakan Zoom sebagai media penyelenggaraan kegiatan secara online dengan durasi lebih dari 20 menit dan jumlah partisipan diatas 100 orang perlu menggunakan akun Zoom yang berbayar. Masalahnya mayoritas masyarakat tidak menggunakan Zoom untuk keperluan mengumpulkan orang banyak dan dalam waktu lama setiap hari, biasanya dilakukan sesekali dalam sebulan, yang mengakibatkan mengeluarkan uang yang tidak sedikit untuk berlangganan akun Zoom premium selama satu bulan dirasa terlalu mahal.

Terdapat jenis usaha baru yang muncul akibat dari fenomena tersebut, yaitu jasa sewa akun Zoom yang disesuaikan kebutuhan, pembeli bisa menyewa akun Zoom premium sesuai dengan kebutuhan mereka. Maka dari itu, untuk memudahkan penjual jasa sewa akun Zoom, penulis tertarik untuk membuat sebuah program yang membantu pemilik usaha sewa Zoom untuk pencatatan data penjualan akun Zoom menggunakan bahasa pemrograman Java.

1.2 Tujuan

Tujuan dari proyek Program Penyewaan Zoom ini adalah untuk membuat sistem yang memudahkan pemilik usaha sewa akun zoom dalam proses pencatatan data penyewaan akun zoom serta menyelesaikan tugas akhir dari mata kuliah Pemrograman Berorientasi Objek.

BAB II PEMBAHASAN

2.1 Dasar Teori

2.1.1 Bahasa Pemrograman Java

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

1.1 Aturan Dasar Java

Berikut adalah sejumlah aturan umum dari penggunaan sintaks bahasa Java :

1. Java = Case sensitive. “Perhatikan apa yang Anda ketik. $A \neq a$.”
2. Nama file dokumen java **HARUS** sama dengan nama Kelas.
3. Secara umum Java mengabaikan spasi kosong. Spasi kosong terdiri dari karakter yang tidak muncul ada layar misalnya tab, spasi, baris baru. Gunakan spasi kosong untuk membuat program lebih menarik dan mudah di baca.
4. Tanda { } adalah simbol pengelompokan dalam Java. Berfungsi untuk menandai awal dan akhir seksi program. Jumlah { dan jumlah } adalah sama.
5. Setiap pernyataan Java diakhiri dengan tanda ;
6. Program komputer biasanya juga berisi komentar-komentar. Java menawarkan dua cara untuk menyisipkan komentar, yaitu :
 - a. simbol // digunakan untuk penulisan satu baris komentar
 - b. simbol /* dan */ digunakan untuk penulisan komentar lebih dari satu baris.

Contoh :

```
/* ini adalah contoh komentar yang
   Lebih dari satu baris */
```

7. Perlu diingat pula aturan-aturan berikut ini hanya berlaku untuk Kelas :
 - a. dalam satu file .java diperbolehkan memiliki lebih dari satu kelas
 - b. dalam satu file .java, jumlah kelas yang boleh dideklarasikan sebagai `public` **HANYA SATU** dan nama dari file .java ini harus sama dengan nama kelas yang dideklarasikan sebagai `public`.

1.2 Atribut dan Tipe Data

Java memiliki tipe data, syntax keputusan dan perulangan yang sama dengan C dan C++. Bagi yang telah mempelajari C atau C++ sebelumnya, mempelajari Java seharusnya menjadi hal yang mudah.

Cara mendeklarasikan atribut:

1. `tipeData namaAtribut;`
2. `tipeData namaAtribut = nilaiAwal;`

Contoh:

1. `int i;`
2. `String nama = "Java";`

Atribut dalam Java bisa dijadikan konstanta, sehingga nilainya tidak akan berubah. Untuk membuat nilai konstanta, gunakan kata kunci `final` yang diikuti dengan tipe data dan nama atribut dalam huruf besar:

Contoh: `final double PHI = 3.14;`

Tabel 1.1 Perbandingan Tipe Data

Kelompok Tipe Data	C++	Java
bilangan bulat	-	byte
	Short	short
	Int	int
	Long	long
bilangan pecahan	Float	float
	Double	double
Karakter	Char	char
	-	String
Boolean	Boolean	boolean

1.3 Perulangan

Tabel 1.2 Perbandingan Syntax Perulangan

Perulangan	C++	Java
FOR	<code>for (inisialisasi;kondisi;i terasi) pernyataan;</code>	<code>for (inisialisasi;kondisi;iter asi) pernyataan;</code>
WHILE	<code>while (kondisi) { pernyataan; }</code>	<code>while (kondisi) { pernyataan; }</code>
DO WHILE	<code>do { pernyataan; } while (kondisi);</code>	<code>do { pernyataan; } while (kondisi);</code>

1.4 Keputusan

Tabel 1.3 Perbandingan Syntax Keputusan

Keputusan	C++	Java
IF – ELSE	<pre>if (kondisi) Pernyataan jika kondisi benar; else Pernyataan jika kondisi salah;</pre>	<pre>if (kondisi) Pernyataan jika kondisi benar; else Pernyataan jika kondisi salah;</pre>
SWITCH – CASE	<pre>switch (ekspresi switch) { case nilai1 : pernyataan1; break; case nilai2 : pernyataan2; break; ... case nilaiN : pernyataanN; break; default : pernyataan default; }</pre>	<pre>switch (ekspresi switch) { case nilai1 : pernyataan1; break; case nilai2 : pernyataan2; break; ... case nilaiN : pernyataanN; break; default : pernyataan default; }</pre>

2.1.2 Pemrograman Berorientasi Objek

2.1. CLASS

Deklarasi class pada Java:

```
[modifier1] class NamaKelas [modifier2] {
    class body;
}
```

Keterangan :

1. Tanda [] bersifat optional.
2. Modifier1 dapat berupa : public, abstract, final
3. Modifier2 dapat berupa : extends, implements
4. Class body merupakan isi dari program yang terdiri dari constructor, atribut dan method.

2.2 METHOD

Deklarasi method:

```
[modifier] tipeNilaiKembalian namaMethod (parameter input) {
    method body;
```



```
}
```

Keterangan :

- a) Tanda [] bersifat optional.
- b) Modifier dapat berupa : `public`, `protected`, `private`, `static`, `abstract`, `final`, `native` dan `synchronized`
- c) Dilihat dari parameter input, method dibagi menjadi dua yaitu membutuhkan input atau tidak. Bila membutuhkan input, maka tipe data dari parameter input harus dituliskan. Gunakan koma untuk memisahkan setiap parameter input. Bila tidak membutuhkan input, maka bagian parameter input dikosongkan.

Cara pemanggilan method:

1. `NamaKelas>NamaMethod(parameterInput);`
2. `NamaObjek>NamaMethod(parameterInput);`

2.3. CONSTRUCTOR

Constructor merupakan method yang namanya sama dengan nama kelas dimana method tersebut berada. Constructor dieksekusi pertama kali saat suatu kelas diinstansiasi menjadi objek. Biasanya, Constructor digunakan untuk menginisialisasi nilai awal (memberikan nilai default) pada atribut-atribut yang dimiliki oleh suatu objek saat objek itu pertama kali dibuat.

2.4. MODIFIER

Modifier adalah sifat yang dimiliki oleh setiap atribut, method maupun kelas dalam java. Modifier akses adalah modifier yang selalu digunakan, modifier akses terdiri dari `private`, `default`, `protected` dan `public`. Kriteria modifier-modifier tersebut adalah :

Wilayah Akses	public	protected	default	Private
Kelas yang sama	ya	ya	ya	ya
Antar kelas dalam package yang sama	ya	ya	ya	Tidak
SubKelas di package yang berbeda	ya	ya	Tidak	Tidak
Bukan subkelas, package yang berbeda	ya	Tidak	Tidak	Tidak

Beberapa modifier penting lainnya :

static yaitu modifier yang digunakan agar suatu atribut maupun method dapat diakses oleh objek atau kelas lain meski tanpa dilakukan instansiasi terhadap kelas dimana atribut maupun method itu berada, method main adalah salah satu contoh method bermodifier static yang sering digunakan.

final yaitu modifier yang digunakan untuk mencegah kemungkinan modifikasi terhadap atribut maupun method, dengan modifier ini suatu atribut akan berlaku sebagai konstanta

2.5. OBJECT

Sebuah kelas dapat digunakan untuk membuat banyak objek. Setiap objek dapat diperlakukan secara berbeda oleh objek-objek lain yang menggunakannya. Dibutuhkan operator `new` untuk membuat objek dari suatu kelas.

Objek dibuat dengan cara menuliskan :

1. `NamaKelas NamaObjek = new NamaConstructor();`
2. `Nama Kelas NamaObjek;`
`NamaObjek = new NamaConstructor();`

Proses diatas disebut **instansiasi**. Ketika objek selesai dibuat, objek tersebut disimpan di dalam memori dan dapat diakses oleh objek-objek lain melalui `NamaObjek` nya.

2.6. PACKAGE

Package digunakan untuk mengelompokkan file kelas (*.class) yang terkait (karena jenisnya, fungsinya, atau karena alasan lainnya) pada folder yang sama, dimana di dalam setiap kelasnya terdapat directive (statement java dalam source code yang digunakan untuk membuat kelas) package yang mengacu pada folder tersebut. Deklarasi package:

```
package namapackage;
```

Untuk bisa mengakses kelas yang berbeda package, diperlukan pernyataan `import` baik pada kelas yang ingin diakses, maupun package yang menampung kelas tersebut, berikut ini contohnya:

1. `import namaPackage.namaKelas; // untuk mengimport suatu kelas`
2. `import namaPackage.*; // untuk mengimport semua kelas dalam package tersebut`

2.7 Inheritance

Inheritance adalah penurunan atribut dan method dari suatu kelas induk (superclass) ke kelas anak (subclass). Pewarisan bersifat menyeluruh dan beruntun, sehingga jika A merupakan superclass dari B, B merupakan superclass dari C, maka A juga merupakan superclass dari C. Jika ada atribut atau method dari C dan B yang ingin dimodifikasi, kita hanya perlu memodifikasi atribut/method pada A. Pernyataan `extends` digunakan untuk melakukan pewarisan ini.

Deklarasi class:

```
[modifier1] class NamaSubKelas extends NamaKelasSuper {
    classbody;
}
```

2.8. Encapsulation

Encapsulation atau pengkapsulan digunakan untuk menyembunyikan atribut atau method pada suatu objek, dari objek lainnya, atau dengan kata lain yakni *Information hiding*. Pengkapsulan dilakukan dengan memberikan modifier `private` pada atribut maupun method. Kemudian, untuk memberikan nilai pada atribut `private`, digunakan method '`setter`', dan untuk mengambil nilai dari atribut `private`, digunakan method '`getter`'. Method `setter` dan `getter` ini sama seperti method biasa, hanya berbeda pada pemanfaatan yang dituliskan programmernya saja.

2.9. Polymorphism

Polymorphism dapat diartikan sebagai kemampuan untuk membuat banyak objek berbeda dari satu sumber yang sama. Dalam Java, contoh yang mudah adalah membuat beberapa objek berbeda dari beberapa subclass yang mewarisi satu superclass yang sama.

2.10. Abstract Class

Abstract class adalah kelas yang tidak dapat diinstansiasi karena terlalu umum cakupannya. Misalnya `MakhlukHidup`, makhluk hidup bisa berupa burung, reptil, mamalia, dsb yang terlalu berbeda jenis dan sifatnya, sehingga abstract class memerlukan kelas lain yang lebih nyata dan lebih khusus untuk menjadi pewarisnya (`extends`). Method yang abstract adalah method yang belum memiliki implementasi (isi) dan menggunakan modifier `abstract`. Method ini diisi nilainya oleh kelas yang menjadi turunannya, hal ini disebut dengan `overriding`.

2.11. Interface

Interface merupakan suatu kelas yang berisi method-method tanpa implementasi, namun tanpa modifier abstract. Apabila terdapat atribut, maka atribut tersebut berlaku sebagai konstanta (static final). Interface bersifat seperti kontrak yang bisa digunakan oleh setiap kelas yang nyata, dan penggunaannya bukan diwariskan (extends) melainkan diimplementasikan (implements).

Analogi interface misalnya sebagai kontrak yang dapat dipakai oleh setiap kelas. Kita tahu ada berbagai macam profesi seperti penyanyi, pengacara, pelawak, dsb, namun selain menjalani profesinya, mereka juga memiliki sifat dan ciri standar sebagai manusia. Sehingga, untuk membuat objek Mario, Penyanyi, kita harus membuat kelas yang merupakan turunan kelas manusia yang mengimplementasikan interface penyanyi.

2.1.3 Model View Controller

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). Sebagai 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,

2.2 Deskripsi Umum Proyek Akhir

Program Penyewaan Zoom ini merupakan program yang berguna untuk memudahkan pemilik usaha penyewaan akun zoom dalam pencatatan data. Pengguna dapat memasukkan data paket akun zoom. Pada saat pertama kali program dijalankan, pengguna akan dihadapkan pada halaman login. Kemudian setelah login pada tampilan awal terdapat menu Pembelian, Inut Paket, Edit Paket, dan Riwayat Transaksi. Pada menu Pembelian, pengguna dapat memilih paket yang tersedia lalu memasukkan data pembeli. Lalu pada menu input data, pembeli dapat memasukkan data paket. Kemudian pada menu Edit Paket pengguna dapat merubah data paket yang tersedia. Dan yang terakhir terdapat menu Riwayat Transaksi yang dapat digunakan untuk melihat riwayat dari penyewaan akun zoom.

2.3 Inti Pembahasan

2.3.1 Listing Program

```
package projectakhir;

public class Main {
    public static void main(String[] args) {
        MVC mvc = new MVC();
    }
}
```

Listing Program 2.1 Main.java

```
package projectakhir;

public class MVC {
    ModelSQL Koneksi = new ModelSQL();
    ViewDaftar vd = new ViewDaftar();
    ViewHome vh = new ViewHome();
    ViewPembelian vp = new ViewPembelian();
    ViewInput vi = new ViewInput();
    ViewEdit ve = new ViewEdit();
    ViewTampil vt = new ViewTampil();

    Model m = new Model();

    Controller c = new Controller(vd, vh, vt, ve, vi, vp, m);
}
```

Listing Program 2.2 MVC.java

```
package projectakhir;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
```

```

public class Model extends ModelSQL {
    String username,password;

    protected boolean login(String username,String password)
    {
        boolean sukses=false;
        try{
            String cek = "SELECT count(*) as jumlah from user
where username='"+username+"' and password='"+password+"'";
            statement = (Statement) koneksi.createStatement();
            ResultSet hasil = statement.executeQuery(cek);
            hasil.next();
            System.out.println(username);
            if(hasil.getInt("jumlah")==1)
            {
                sukses=true;
            }

            else

            {
                sukses=false;
            }

        }
        catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
        return sukses;
    }

    public boolean inputpaket(String kode, String nama, String
maks, String durasi, String harga, String stok)
    {
        boolean status=false;
        try {
            String input = "insert into paket
(kode,nama,maks_participant,durasi,harga,stok) VALUES ('" + kode +
"','" + nama + "','" + maks + "','" + durasi + "','" + harga + "','"
+ stok + "')";
            statement = (Statement) koneksi.createStatement();
            statement.execute(input);
            status=true;
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
    }

```

Listing Program 2.3 Model.java

```

return status;
    }

    public int getjumlahpaket()
    {
        int jumlah=0;
        try {
            String cek = "select count(*) as jumlah from paket";

```

```

        statement = (Statement) koneksi.createStatement();
        ResultSet hasil = statement.executeQuery(cek);
        hasil.next();
        jumlah=hasil.getInt("jumlah");
        return jumlah;
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
    return jumlah;
}

public String[][] getpaket()
{
    String data[][] = new String[getjumlahpaket()][6];
    //baris, kolom nya ada 3

    try {
        int jmlData=0;
        String query = "Select * from paket";
        ResultSet resultSet = statement.executeQuery(query);
        while (resultSet.next()){

            data[jmlData][0] = resultSet.getString("kode");
            //harus sesuai nama kolom di mysql
            data[jmlData][1] = resultSet.getString("nama");
            data[jmlData][2] =
resultSet.getString("maks_participant");
            data[jmlData][3] = resultSet.getString("durasi");
            data[jmlData][4] = resultSet.getString("harga");
            data[jmlData][5] = resultSet.getString("stok");

            if("0".equals(resultSet.getString("stok")))
            {
                data[jmlData][5] = "Habis";
            }

            jmlData++;
        }
        return data;
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
    return data;
}

public void inputpembelian(String kode,String nama,String
Notelp,String Alamat,String TanggalPembelian) throws SQLException
{

```

Listing Program 2.4 Lanjutan Model.java

```

String query = "insert into pembelian
(Nama,Alamat,NoTelp,Kode,Tanggal)
Values
('"+nama+"','"+Alamat+"','"+Notelp+"','"+kode+"','"+TanggalPembelian+"')";

statement = (Statement) koneksi.createStatement();
statement.execute(query);

String query1 = "update paket set stok=stok-1 where
kode = '"+kode+"'";
statement = (Statement) koneksi.createStatement();

```

```

        statement.execute(query1);
    }

    public void editpaket(String kode, String nama, String
maks, String durasi, String harga, String stok)
    {
        try {
            String query = "update paket set nama = '"+nama+"',
maks_participant='"+maks+"', durasi = '"+durasi+"', harga = '
'+harga+"', stok = '" + stok + "' where kode = '"+kode+"'";
            statement = (Statement) koneksi.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
    }

    public String[][] getpembelian()
    {
        String[][] data = new String[getjumlahpembelian()][5];
        try {
            int jmlData=0;
            String query = "select * from pembelian inner join
paket on pembelian.kode = paket.kode";
            ResultSet resultSet = statement.executeQuery(query);
            while (resultSet.next()){
                data[jmlData][0] =
resultSet.getString("Tanggal"); //harus sesuai nama kolom di mysql
                data[jmlData][1] = resultSet.getString("Nama");
                data[jmlData][2] = resultSet.getString("kode");
                data[jmlData][3] = resultSet.getString("durasi");
                data[jmlData][4] = resultSet.getString("harga");
                jmlData++;
            }
        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
        return data;
    }

    public int getjumlahpembelian()
    {
        int jumlah=0;
        try {
            String cek = "select count(*) as jumlah from
pembelian";
            statement = (Statement) koneksi.createStatement();

```

Listing Program 2.5 Lanjutan Model.java

```

        ResultSet hasil = statement.executeQuery(cek);
        hasil.next();
        jumlah=hasil.getInt("jumlah");
        return jumlah;
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
    return jumlah;
}

}

```


Listing Program 2.6 Lanjutan Model.java

```

package projectakhir;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

/**
 *
 *
 */
public class ModelSQL {

    static final String JDBC_DRIVER =
"com.mysql.jdbc.Driver";
    static final String DB_URL =
"jdbc:mysql://localhost/zoom";
    static final String USER = "root";
    static final String PASS = "";

    Connection koneksi;
    Statement statement;

    public ModelSQL() {
        try {
            Class.forName(JDBC_DRIVER);
            koneksi = (Connection)
DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Koneksi Berhasil");
        }
        catch (ClassNotFoundException | SQLException ex)
        {
            JOptionPane.showMessageDialog(null,
ex.getMessage());
            System.out.println("Koneksi Gagal");
        }
    }
}

```

Listing Program 2.7 ModelSQL.java

```

package projectakhir;
import javax.swing.*;
public class ViewDaftar extends JFrame {
    //Header
    JLabel jlheader = new JLabel("SISTEM INFORMASI");
    JLabel jlheader2 = new JLabel("PENYEWAAN ZOOM");
    //Username
    JLabel jlusername = new JLabel("Username");
    JTextField jtusername = new JTextField();
    JPasswordField jppassword = new JPasswordField();
    //Password
    JLabel jlpassword = new JLabel("Password");
}

```

```

JButton jblogin = new JButton("Login");
JButton jbdaftar = new JButton("Daftar");
public ViewDaftar()
{
    setTitle("Penyewaan Zoom/login");
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(true);
    setLayout(null);
    setSize(1000,600);

    //Header
    add(jlheader);
    jlheader.setBounds(420, 140, 200, 20);
    add(jlheader2);
    jlheader2.setBounds(420, 160, 200, 20);
    //Username
    add(jlusername);
    add(jtusername);
    jlusername.setBounds(450, 200, 200, 20);
    jtusername.setBounds(380, 220, 200, 20);
    //Password
    add(jlpassword);
    add(jppassword);
    jlpassword.setBounds(450, 240, 200, 20);
    jppassword.setBounds(380, 260, 200, 20);
    //Login
    add(jblogin);
    jblogin.setBounds(380, 300, 200, 20);
    //Daftar
    // add(jbdaftar);
    //jbdaftar.setBounds(380, 320, 200, 20);

}
public String getusername()
{
    return jtusername.getText();
}

public String getpassword()
{
    return jppassword.getText();
}
}

```

Listing Program 2.8 ViewDaftar.java

```

package projectakhir;
import javax.swing.*.*;
import javax.swing.table.DefaultTableModel;
public class ViewEdit extends JFrame{
    //Kembali
    JButton jbkembali = new JButton("<");
    //Header
    JLabel jlheader = new JLabel("Edit Paket");
    JLabel jlheader2 = new JLabel("Masukkan Data Paket yang Baru");

```

```

//Kode Paket
JLabel jlkode = new JLabel("Kode Paket:");
JTextField jtkode = new JTextField();
//Nama Paket
JLabel jlpaket = new JLabel("Nama Paket:");
JTextField jtpaket = new JTextField();
//Maximal Participant
JLabel jlpartisipan = new JLabel("Maksimal Partisipan:");
JTextField jtpartisipan = new JTextField();
//Durasi Paket
JLabel jldurasi = new JLabel("Durasi:");
JTextField jtdurasi = new JTextField();
//Harga
JLabel jlharga = new JLabel("Harga:");
JTextField jtharga = new JTextField();
//Stock
JLabel jlstok = new JLabel("Stok:");
JTextField jtstok = new JTextField();
//Edit
JButton jbedit = new JButton("Edit");
//Tabel
JTable tabel;
DefaultTableModel dtm;
JScrollPane scrollPane;
Object namaKolom[] = {"Kode", "Paket", "Maks Partisiapan",
"Durasi", "Harga", "Stok"};
public ViewEdit(){
    setTitle("Penyewaan Zoom/edit_paket");

```

Listing Program 2.9 ViewEdit.java

```

setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(false);
setLayout(null);
setSize(1000,600);
dtm = new DefaultTableModel(namaKolom, 0);
tabel = new JTable(dtm);
scrollPane = new JScrollPane(tabel);
//Kembali
add(jbkembali);
jbkembali.setBounds(20, 20, 100, 20);

```

```

//Header
add(jlheader);
jlheader.setBounds(460, 20, 200, 20);
add(jlheader2);
jlheader2.setBounds(70, 80, 200, 20);
//Kode Paket
add(jlkode);
add(jtkode);
jlkode.setBounds(60, 120, 100, 20);
jtkode.setBounds(60, 140, 200, 20);
//Nama Paket
add(jlpaket);
add(jtpaket);
jlpaket.setBounds(60, 180, 100, 20);
jtpaket.setBounds(60, 200, 200, 20);
//Maksimal Partisipan
add(jlpartisipan);
add(jtpartisipan);
jlpartisipan.setBounds(60, 240, 200, 20);
jtpartisipan.setBounds(60, 260, 200, 20);
//Durasi Paket
add(jldurasi);
add(jtdurasi);
jldurasi.setBounds(60, 300, 200, 20);
jtdurasi.setBounds(60, 320, 200, 20);
//Harga
add(jlharga);

```

Listing Program 2.10 Lanjutan ViewEdit.java

```

add(jtharga);
jlharga.setBounds(60, 360, 200, 20);
jtharga.setBounds(60, 380, 200, 20);
//Stok
add(jlstok);
add(jtstok);
jlstok.setBounds(60, 420, 200, 20);
jtstok.setBounds(60, 440, 200, 20);
//Button Edit
add(jbedit);
jbedit.setBounds(80, 480, 160, 20);

```

```

        //Tabel
        add(scrollPane);
        scrollPane.setBounds(340, 80, 600, 380);
    }

    public String getkode()
    {
        return jtkode.getText();
    }

    public String getpaket()
    {
        return jtpaket.getText();
    }

    public String getmaks()
    {
        return jtpartisipan.getText();
    }

    public String getdurasi()
    {
        return jtdurasi.getText();
    }

    public String getharga()
    {
        return jtharga.getText();
    }

    public String getstok();

```

Listing Program 2.11 Lanjutan ViewEdit.java

```

    {
        return jtstok.getText();
    }
}

```

Listing Program 2.12 Lanjutan ViewEdit.java

```

package projectakhir;
import javax.swing.*.*;
public class ViewHome extends JFrame{
    //Header
    JLabel jlheader = new JLabel("MENU");
    //Button

```

```

        JButton jbpembelian = new JButton("Pembelian");
        JButton jbinput = new JButton("Input Paket");
        JButton jbedit = new JButton("Edit Paket");
        JButton jibriwayat = new JButton("Riwayat Transaksi");
        public ViewHome() {
            setTitle("Penyewaan Zoom");
            setDefaultCloseOperation(EXIT_ON_CLOSE);
            setVisible(false);
            setLayout(null);
            setSize(1000,600);
            //Header
            add(jlheader);
            jlheader.setBounds(460, 160, 200, 20);
            //Pembelian
            add(jbpembelian);
            jbpembelian.setBounds(380, 200, 200, 20);
            //Input
            add(jbinput);
            jbinput.setBounds(380, 240, 200, 20);
            //Edit
            add(jbedit);
            jbedit.setBounds(380, 280, 200, 20);
            //Riwayat
            add(jibriwayat);
            jibriwayat.setBounds(380, 320, 200, 20);
        }
    }
}

```

Listing Program 2.13. ViewHome.java

```

package projectakhir;
import javax.swing.*;
public class ViewInput extends JFrame {
    //Header
    JLabel jlheader = new JLabel("Tambahkan Paket Zoom");
    //Kembali
    JButton jbkembali = new JButton("←");
    //Kode Paket
    JLabel jlkode = new JLabel("Kode:");
    JTextField jtkode = new JTextField();
    //Nama Paket
    JLabel jlpaket = new JLabel("Nama:");
}

```

```

JTextField jtpaket = new JTextField();
//Maximal Participant
JLabel jlpartisipasi = new JLabel("Maksimal Partisipan:");
JTextField jtpartisipasi = new JTextField();
//Durasi Paket
JLabel jldurasi = new JLabel("Durasi:");
JTextField jtdurasi = new JTextField();
//Harga
JLabel jlharga = new JLabel("Harga:");
JTextField jtharga = new JTextField();
//Stock
JLabel jlstok = new JLabel("Stok:");
JTextField jtstok = new JTextField();
//Tambahkan
JButton jbtambah = new JButton("Tambahkan");
public ViewInput(){
setTitle("Penyewaan Zoom/input_paket");
setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(false);
setLayout(null);
setSize(1000,600);
//Header
add(jlheader);
jlheader.setBounds(380, 60, 200, 20);
//Kembali

```

Listing Program 2.14. ViewInput.java

```

add(jbkembali);
jbkembali.setBounds(20, 20, 100, 20);
//Nama Paket
add(jlkode);
add(jtkode);
jlkode.setBounds(360, 100, 200, 20);
jtkode.setBounds(360, 120, 200, 20);
//Nama Paket
add(jlpaket);
add(jtpaket);
jlpaket.setBounds(360, 160, 200, 20);
jtpaket.setBounds(360, 180, 200, 20);
//Maksimal Partisipan

```

```

add(jlpartisipan);
add(jtpartisipan);
jlpartisipan.setBounds(360, 220, 200, 20);
jtpartisipan.setBounds(360, 240, 200, 20);
//Durasi Paket
add(jldurasi);
add(jtdurasi);
jldurasi.setBounds(360, 280, 200, 20);
jtdurasi.setBounds(360, 300, 200, 20);
//Harga
add(jlharga);
add(jtharga);
jlharga.setBounds(360, 340, 200, 20);
jtharga.setBounds(360, 360, 200, 20);
//Stok
add(jlstok);
add(jtstok);
jlstok.setBounds(360, 400, 200, 20);
jtstok.setBounds(360, 420, 200, 20);
//Button Tambah
add(jbtambah);
jbtambah.setBounds(400, 460, 120, 20);
}
public String getkode()

```

Listing Program 2.15. Lanjutan ViewInput.java

```

{
return jtkode.getText();
}
public String getnama()
{
return jtpaket.getText();
}
public String getmaksparticipant()
{
return jtpartisipan.getText();
}
public String getdurasi()
{
return jtdurasi.getText();
}

```



```

    }
    public String getharga()
    {
        return jtharga.getText();
    }
    public String getstok()
    {
        return jtstok.getText();
    }
}

```

Listing Program 2.16. Lanjutan ViewInput.java

```

package projectakhir;
import javax.swing.*.*;
import javax.swing.table.DefaultTableModel;
public class ViewPembelian extends JFrame{
    //Header
    JLabel jlheader = new JLabel("Daftar Paket");
    JLabel jlheader2 = new JLabel("Masukkan Data Pembeli");
    //Kembali
    JButton jbkembali = new JButton("←");
    //Table
    JTable tabel;

```

Listing Program 2.17. ViewPembelian.java

```

DefaultTableModel dtm;
JScrollPane scrollPane;
Object namaKolom[] = {"Kode", "Paket", "Maks Partisiapan", "Durasi",
"Harga", "Stok"};
//Input Nama Pembeli
JLabel jlnama = new JLabel("Nama");
JTextField jtnama = new JTextField();
//Input Nomor Telpon Pembeli
JLabel jlnotlp = new JLabel("Nomor Telpon");
JTextField jtnotlp = new JTextField();
//Input Alamat Pembeli
JLabel jlalamat = new JLabel("Alamat");
JTextField jtalamat = new JTextField();
//Input Tanggal Pembelian

```

```

JLabel jltanggal = new JLabel("Tanggal Pembelian");
JTextField jttanggal = new JTextField();
//Button Pembelian
JButton jbpembelian = new JButton("Pembelian");
public ViewPembelian(){
    setTitle("Penyewaan Zoom/pembelian");
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(false);
    setLayout(null);
    setSize(1000,600);
    dtm = new DefaultTableModel(namaKolom, 0);
    tabel = new JTable(dtm);
    scrollPane = new JScrollPane(tabel);
    //Header
    add(jlheader);
    jlheader.setBounds(460, 20, 200, 20);
    //Kembali
    add(jbkembali);
    jbkembali.setBounds(20, 20, 100, 20);
    //Tabel
    add(scrollPane);
    scrollPane.setBounds(200, 60, 600, 220);
    //Header 2

```

Listing Program 2.18. Lanjutan ViewPembelian.java

```

add(jlheader2);
    jlheader2.setBounds(440, 300, 200, 20);
    //Nama
    add(jlnama);
    add(jtnama);
    jlnama.setBounds(320, 340, 100, 20);
    jtnama.setBounds(460, 340, 220, 20);
    //Nomor Telpon
    add(jlnotlp);
    add(jtnotlp);
    jlnotlp.setBounds(320, 380, 200, 20);
    jtnotlp.setBounds(460, 380, 220, 20);
    //Alamat
    add(jlalamat);
    add(jtalamat);

```

```

        jalamat.setBounds(320, 420, 200, 20);
        jalamat.setBounds(460, 420, 220, 20);
        //
        add(jltanggal);
        add(jttanggal);
        jltanggal.setBounds(320, 460, 200, 20);
        jttanggal.setBounds(460, 460, 220, 20);
        //Pembelian
        add(jbpembelian);
        jbpembelian.setBounds(420, 520, 200, 20);
    }
    public String getnama()
    {
        return jtnama.getText();
    }
    public String getnomertelepon()
    {
        return jtnotlp.getText();
    }
    public String getalamat()
    {
        return jtalamat.getText();
    }

```

Listing Program 2.19. Lanjutan ViewPembelian.java

```

    }
    public String gettanggalpembelian()
    {
        return jttanggal.getText();
    }
}

```

Listing Program 2.20. Lanjutan ViewPembelian.java

```

package projectakhir;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class ViewTampil extends JFrame {
    //Kembali
    JButton jbkembali = new JButton("<");
    //Header
    JLabel jlheader = new JLabel("Riwayat Transaksi");

```

```

//Table
JTable tabel;
DefaultTableModel dtm;
JScrollPane scrollPane;
Object namaKolom[] = {"Tanggal", "Nama", "Paket", "Durasi",
"Harga"};
//Button Menu
JButton jbmenu = new JButton("Menu");
public ViewTampil(){
    setTitle("Penyewaan Zoom/riwayat_transaksi");
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(false);
    setLayout(null);
    setSize(1000,600);
    //Kembali
    add(jbkembali);
    jbkembali.setBounds(20, 20, 100, 20);
    dtm = new DefaultTableModel(namaKolom, 0);
    tabel = new JTable(dtm);
    scrollPane = new JScrollPane(tabel);
    //Header

```

Listing Program 2.21. ViewTampil.java

```

    add(jlheader);
    jlheader.setBounds(460, 20, 200, 20);
    //Tabel
    add(scrollPane);
    scrollPane.setBounds(200, 60, 600, 380);
    //Pembelian
}
}

```

Listing Program 2.22. Lanjutan ViewTampil.java

2.3.2 Penjelasan Program

a. Login

Pada bagian ini, terdapat tampilan login, pengguna diharapkan untuk memasukkan username dan password sesuai dengan database yang telah

dibuat. Jika ingin menambahkan akun maka pengguna dapat menambahkannya lewat database MySQL.

b. Menu Awal

Pada tampilan Menu Awal terdapat empat menu, yaitu:

1. Pembelian
2. Input Paket
3. Edit Paket
4. Riwayat Transaksi

c. Menu Pembelian

Pada Menu Pembelian, pengguna dapat memilih paket yang tersedia pada tabel bagian atas lalu memasukkan data pembeli, setelah itu pembeli dapat klik tombol pembelian seperti nama, nomor telepon, alamat, dan tanggal pembelian. Maka setelah itu data pembelian akan tersimpan kedalam database.

d. Menu Input Paket

Pada Menu Pembelian, pengguna dapat menambahkan paket zoom beserta kode paket, nama paket, jumlah maksimal partisipan, durasi paket, harga paket, serta stok yang tersedia.

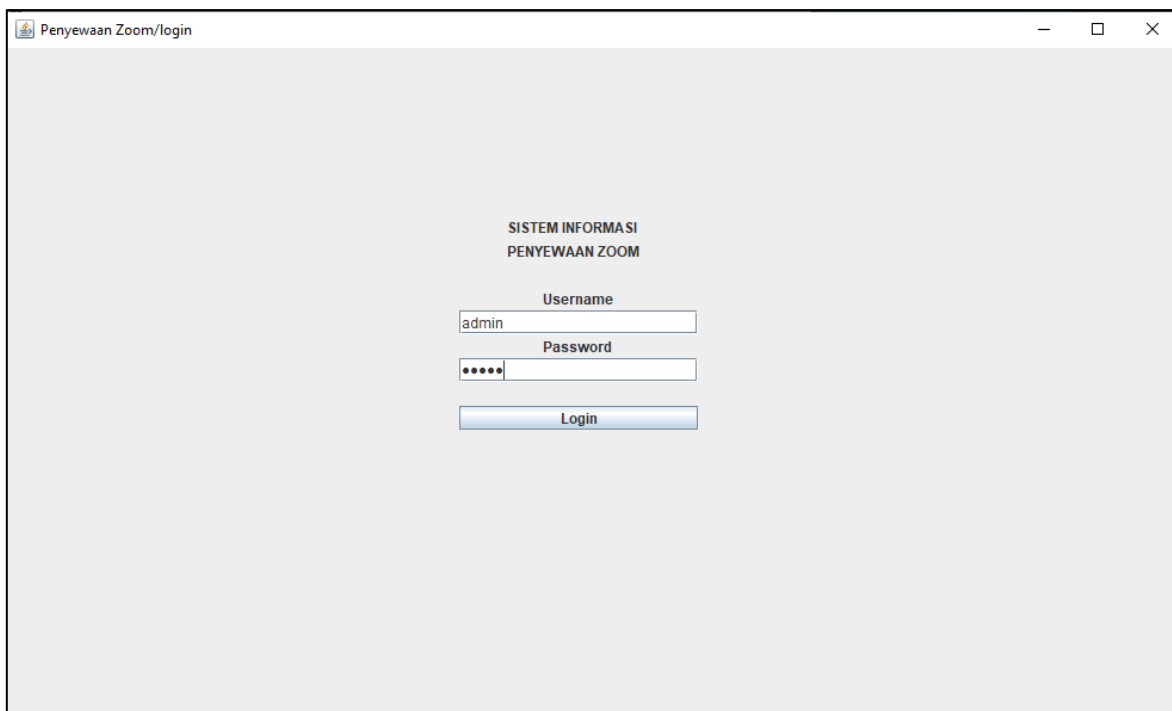
e. Menu Edit Paket

Pada Menu Edit Paket, pengguna dapat mengubah data paket yang tersedia, pengguna tidak dapat mengubah kode paket karena kode paket merupakan *primary key*. Pengguna dapat merubah nama paket, jumlah maksimal partisipan, durasi paket, harga paket, dan jumlah stok yang tersedia.

f. Menu Riwayat Paket

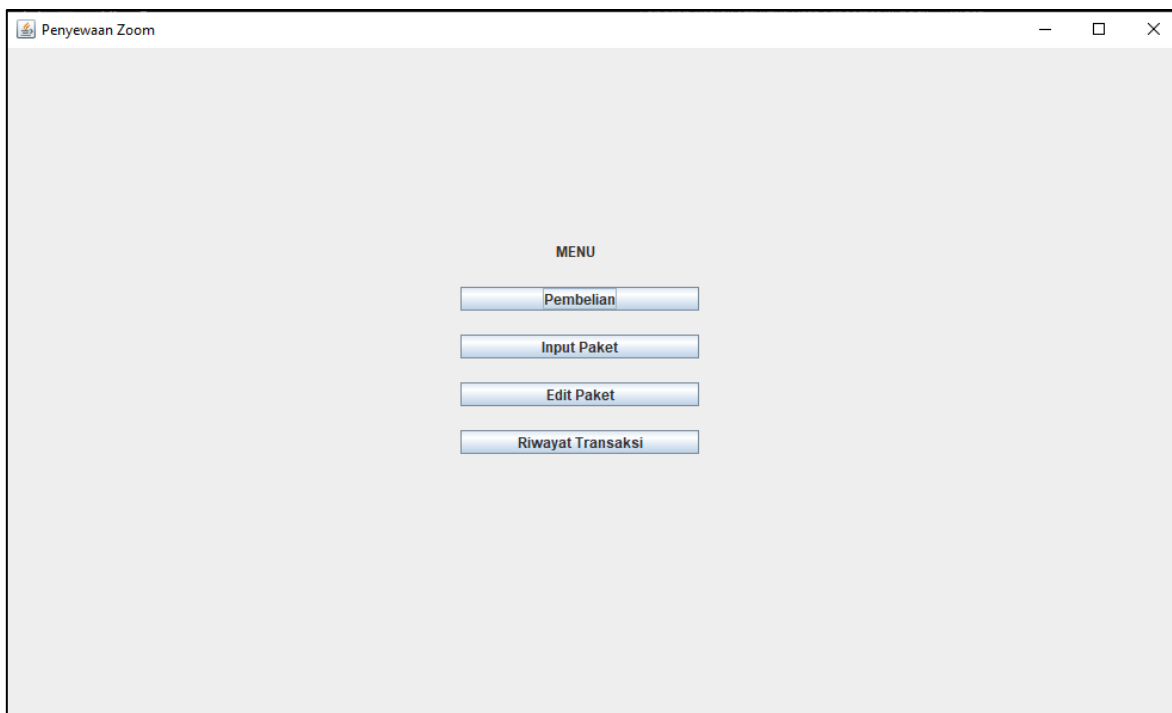
Pada Menu Riwayat Paket, pengguna dapat melihat data riwayat transaksi dari seluruh pembelian paket yang tersimpan pada database.

2.3.3 Output Program



The screenshot shows a login window titled "Penyewaan Zoom/login". The window has a light gray background. In the center, the text "SISTEM INFORMASI" and "PENYEWAAN ZOOM" is displayed. Below this, there are two input fields: "Username" with the text "admin" and "Password" with masked characters ".....". A "Login" button is positioned below the password field.

Gambar 2.1 Output Login



The screenshot shows the main menu window titled "Penyewaan Zoom". The window has a light gray background. In the center, the text "MENU" is displayed. Below this, there are four buttons: "Pembelian", "Input Paket", "Edit Paket", and "Riwayat Transaksi".

Gambar 2.2 Output Menu Utama

Penyewaan Zoom/pembelian

Daftar Paket

Kode	Paket	Maks Partisiapan	Durasi	Harga	Stok
Zoom	Zoom	215	1 Hari	50000	1
Zoom1	zoom	300	2 hari	30000	Habis
Zoom2	ZoomMurah	155	1 Hari	50000	10

Masukkan Data Pembeli

Nama:

Nomor Telpon:

Alamat:

Tanggal Pembelian:

Gambar 2.3 Output Menu Pembelian

Penyewaan Zoom/input_paket

Tambahkan Paket Zoom

Kode:

Nama:

Maksimal Partisipan:

Durasi:

Harga:

Stok:

Gambar 2.4 Output Menu Input Paket

Penyewaan Zoom/edit_paket

←

Edit Paket

Masukkan Data Paket yang Baru

Kode Paket:
Zoom

Nama Paket:
Zoom

Maksimal Partisipan:
215

Durasi:
1 Hari

Harga:
50000

Stok:
1

Edit

Kode	Paket	Maks Partisipan	Durasi	Harga	Stok
Zoom	Zoom	215	1 Hari	50000	1
Zoom1	zoom	300	2 hari	30000	Habis
Zoom2	ZoomMurah	155	1 Hari	50000	10

Gambar 2.5 Output Menu Edit Paket

Penyewaan Zoom/riwayat_transaksi

←

Riwayat Transaksi

Tanggal	Nama	Paket	Durasi	Harga
24 Januari 2021	reyhan	zoom	1 Hari	50000
25 Januari 2021	Reihan	Zoom2	1 Hari	50000
30 Januari 2021	ReyhanDs	Zoom1	2 hari	30000
30 Januari 2021	ReyhanDs1	Zoom1	2 hari	30000
30 Januari 2022	ReyhanDs1	Zoom2	1 Hari	50000
30 Januari 2022	ReyhanDs1	Zoom	1 Hari	50000
121	Rei	Zoom2	1 Hari	50000
1	testing	Zoom2	1 Hari	50000
2	re	Zoom1	2 hari	30000

Gambar 2.6 Output Menu Riwayat Transaksi

BAB III

JADWAL Pengerjaan DAN PEMBAGIAN TUGAS

3.1 Jadwal Pengerjaan

Jadwal pengerjaan proyek Praktikum Pemrograman Berorientasi Objek *Program Penyewaan Zoom* ini sebagai berikut:

Tabel 3.1 Tabel Jadwal Pengerjaan

No	Nama Tugas	Januari		
		Minggu ke-		
		1	2	3
1	Menentukan Judul Proyek			
2	Perancangan Konsep Proyek			
3	Desain Skenario Program			
4	Halaman Login			
5	Register Akun			
6	Halaman Input Data			
7	Halaman Review Anda			
8	Halaman Profile			
9	Halaman Daftar Film			
10	Pembuatan Laporan			

3.2 Pembagian Tugas

Pembagian tugas proyek Praktikum Pemrograman Berorientasi Objek *Program Penyewaan Zoom* ini sebagai berikut:

Tabel 3.2 Tabel Pembagian Tugas

no	aktivitas	penanggung jawab
1	Pembahasan Judul	Ucup & Fulan
2	Perancangan Konsep	Ucup & Fulan
3	Desain Skenario	Ucup & Fulan
4	Halaman Login	Ucup & Fulan
5	Halaman Register	Fulan
6	Halaman Input Data	Fulan
7	Halaman Review Anda	Fulan
8	Halaman Profile	Ucup
9	Halaman Daftar Film	Ucup
10	Pembuatan Laporan	Ucup & Fulan

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Kesimpulan yang dapat diambil dari pengerjaan ini adalah *project* program Penyewaan Zoom yang dibuat dapat melakukan penginputan data paket zoom yang disesuaikan dengan kebutuhan pasar. Program ini juga dapat melakukan Penginputan data pelanggan, sehingga memudahkan pemilik usaha dalam menjalankan usahanya. Juga terdapat menu Edit yang memberikan keleluasaan pada pemilik usaha untuk merubah data paket sesuai keinginan dan kebutuhan pemilik usaha. Dan yang terakhir dengan Menu Riwayat Transaksi dapat membantu pemilik usaha dalam proses pendataan dan pencatatan transaksi pembelian.

4.2 Saran

Adapun saran yang dapat dipertimbangkan untuk penyempurnaan dan pengembangan program Penyewaan Zoom lebih lanjut adalah sebagai berikut:

- a. Program Penyewaan Zoom ini lebih disesuaikan lagi dengan perkembangan teknologi sehingga lebih efektif dan efisien dalam penggunaannya.
- b. Penambahan fitur yang memungkinkan pemilik usaha tidak hanya menjual satu buah produk.

DAFTAR PUSTAKA

Anonim.2019.*Modul Praktikum Pemrograman Berorientasi Objek Program Studi Teknik Infromatika Jurusan Teknik Informatika Fakultas Tekik Industri UPN “Veteran” Yogyakarta*. Yogyakarta: UPN “Veteran” Yogyakarta.