

Tugas Akhir Pengembangan Aplikasi

“RENTAL MOTOR”



Disusun oleh:

Tegar Setiono / 682021018

Hilarius Juan Setio Putranto / 682021070

Amatya Keiko Paramasatya / 682021008

Program Studi Sistem Informasi

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Salatiga

2023

Daftar Isi

Daftar Isi	0
BAB I PENDAHULUAN	1
1.1 Profil Bisnis.....	1
1.2 Proses Bisnis	1
1.3 Identifikasi Kegunaan	2
2 BAB II DESAIN BASIS DATA	3
2.1 Desain ERD.....	3
2.2 Penjelasan setiap entitas	3
• Entitas Admin.....	3
• Entitas Motor.....	4
• Entitas Penyewa.....	5
• Entitas Transaksi.....	5
• Entitas Pengembalian.....	6
2.3 Desain UI.....	7
2.4 Kode Program.....	9
2.5 Kode Program Aplikasi	10
2.6 Demo Aplikasi	39
DAFTAR PUSTAKA	46

BAB I PENDAHULUAN

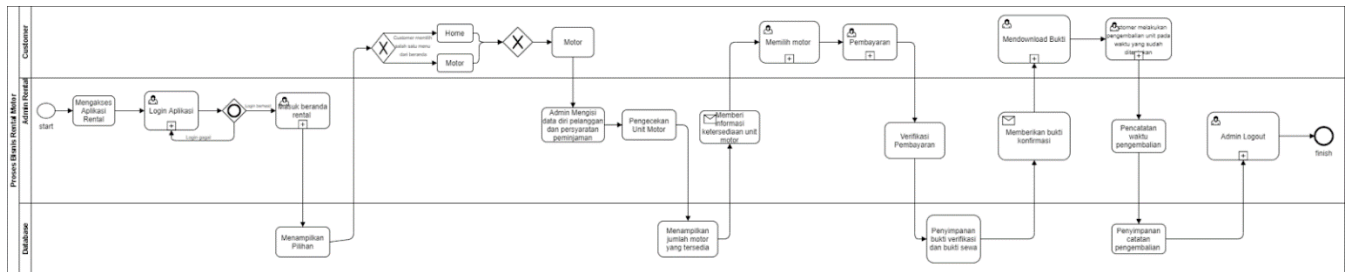
1.1 Profil Bisnis

Motor merupakan salah satu modal transportasi yang paling banyak digunakan oleh masyarakat di Indonesia. Kondisi jalan di Indonesia dimana sering macet, membuat motor menjadi alat transportasi yang paling efektif.

Perkembangan zaman saat ini dengan diiringi kemajuan teknologi menuntut kita untuk mengikuti iramanya, karena untuk sebuah kemajuan yang lebih baik. Jika kita sedang berpergian jauh dari rumah dan kebetulan tidak ada motor, maka rental kendaraan motor menjadi pilihan. Penyewaan motor adalah suatu usaha yang bergerak di bidang jasa persewaan sepeda motor.

Dengan system yang masih manual maka akan berjalan sangat lama dan memakan banyak waktu serta harus sangat teliti dalam pencatatannya. Karena itu, jika manusia mengalami sebuah kesalahan sedikit saja, akan berakibat buruk dan menimbulkan ketidak-efektifan dalam pelaksanaannya. Karena itu, kami membuat aplikasi sederhana untuk pencatatan penyewaan motor yang nantinya akan tercatat masuk ke dalam database. Hal itu memberikan kemudahan saat terjadinya transaksi sewa motor dan membantu pengolahan serta pencatatan data transaksi sewa motor.

1.2 Proses Bisnis



Gambar 1.2.1 swimlane

- Alur proses bisnis rental motor :
 1. Admin mengakses aplikasi rental kendaraan
 2. Admin login ke aplikasi
 3. Admin masuk ke beranda rental kendaraan
 4. Admin Rental kemudian menampilkan pilihan motor
 5. Pelanggan memilih motor karena akan rental motor
 6. Admin kemudian mengisi form data diri pelanggan dan persyaratan peminjaman
 7. Admin rental melakukan pengecekan unit motor melalui database

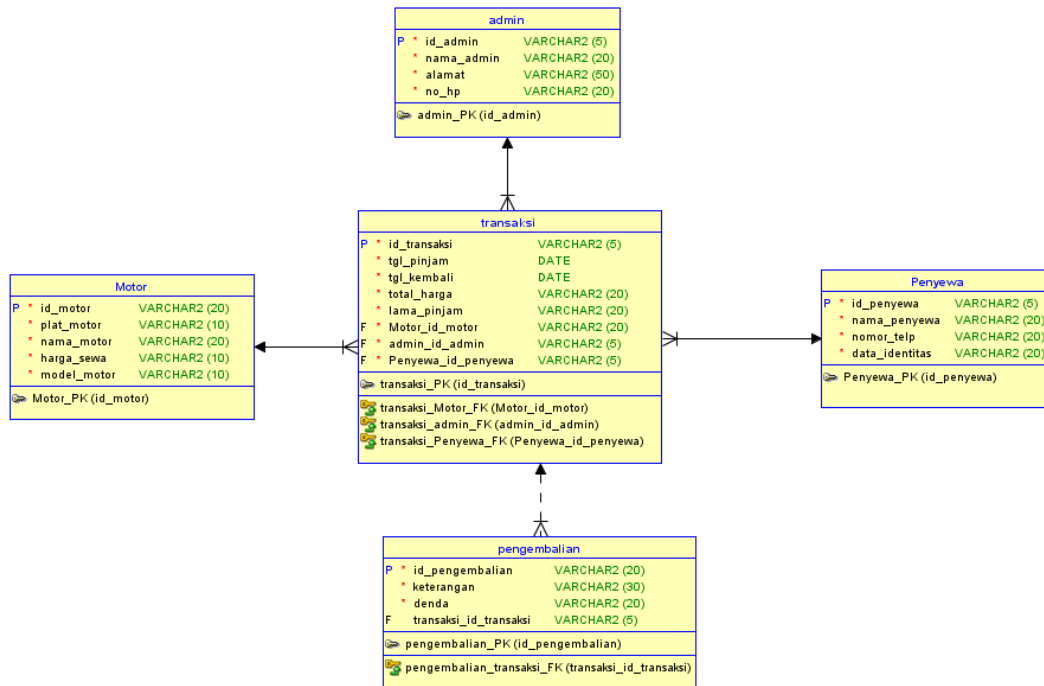
8. Admin rental memberikan informasi ketersediaan unit motor pada customer
9. Pelanggan melakukan transaksi
10. Admin rental menyimpan bukti transaksi dan bukti sewa di database
11. Admin memberikan kwitansi bukti pembayaran dan unit motor sewa kepada pelanggan
12. Setelah itu pelanggan mengembalikan motor di tanggal yang sudah di tentukan
13. Admin mencatat pengembalian motor dan keterangan pengembalian
14. Admin Logout
15. Proses selesai

1.3 Identifikasi Kegunaan

Dalam laporan ini kami ingin meningkatkan penyewaan motor melalui proses bisnis dan mengembangkan suatu program dengan mengimplementasikan kerja database dari rental motor menggunakan SQL dan bahasa pemrograman Java dalam bentuk GUI (UI) untuk membuat aplikasi sederhana guna bisa digunakan untuk penyewaan motor, yang didalam aplikasi tersebut bisa digunakan untuk input data admin, motor, mobil, data transaksi, dan data dari pengembalian motor.

2 BAB II DESAIN BASIS DATA

2.1 Desain ERD



Gambar 2.1.1 ERD rental motor

2.2 Penjelasan setiap entitas

- Entitas Admin

Penjelasan: entitas ini berfungsi untuk data admin yang akan mengurus rental motor. Dalam entitas ini memiliki relasi one to many terhadap entitas transaksi, maka dari itu atribut id_admin menjadi foreign key di entitas transaksi.

Nama Atribut	Data Type	Length	Keterangan
id_admin	VARCHAR	5	Berisi ID admin dan menjadi Primary Key
nama_admin	VARCHAR	20	Berisi nama admin yang bertugas di

			rental motor
alamat	VARCHAR	50	Berisi alamat tempat tinggal dari admin
no_hp	VARCHAR	20	Berisi nomor HP dari admin

Tabel 2.2.1 entitas admin

- **Entitas Motor**

Penjelasan: entitas ini berfungsi untuk data motor yang akan di sewakan. Dalam entitas ini memiliki relasi one to many terhadap entitas transaksi, maka dari itu atribut id_motor menjadi foreign key di entitas transaksi.

Nama Atribut	Data Type	Length	Keterangan
id_motor	VARCHAR	5	Berisi ID Motor yang akan disewakan dan sebagai Primary Key
plat_motor	VARCHAR	10	Berisi nomor polisi dari motor yang akan di rental
merk_motor	VARCHAR	20	Berisi nama motor dari kendaraan yang akan di rental
harga_sewa	VARCHAR	20	Berisi keterangan harga sewa dari tiap motor
no_mesin	VARCHAR	20	Berisi nomor mesin dari tiap motor yang akan disewakan

Tabel 2.2.2 entitas motor

- **Entitas Penyewa**

Penjelasan: Entitas ini berfungsi untuk data penyewa yang akan menyewa motor. Dalam entitas ini memiliki relasi one to many terhadap entitas transaksi, maka dari itu atribut id_penyewa menjadi foreign key di entitas transaksi.

Nama Atribut	Data Type	Length	Keterangan
id_penyewa	VARCHAR	20	Berisi ID Penyewa, sebagai primary key
nama_penyewa	VARCHAR	20	Berisi data nama dari penyewa motor
nomor_telp	VARCHAR	20	Berisi data nomor HP dari penyewa
data_identitas	VARCHAR	20	Berisi data NIK dari penyewa

Tabel 2.2.3 entitas penyewa

- **Entitas Transaksi**

Penjelasan: Entitas ini berfungsi untuk data transaksi motor yang akan di sewakan. Dalam entitas ini memiliki foreign key dari entitas admin (id_admin), penyewa (id_penyewa), dan motor (id_motor).

Nama Atribut	Data Type	Length	Keterangan
id_transaksi	VARCHAR	5	Berisi ID transaksi dari tiap transaksi sewa motor dan menjadi primary key
tgl_pinjam	date		Berisi keterangan tanggal pinjam dari motor yang akan di pinjam

tgl_kembali	date		Berisi keterangan tanggal kembali dari motor yang akan di pinjam
lama_pinjam	VARCHAR	20	Berisi keterangan berapa lama penyewa pinjam motor
total_harga	VARCHAR	20	Berisi keterangan total harga setelah kita rental motor
id_admin	VARCHAR	5	Foreign Key
id_pelanggan	VARCHAR	5	Foreign Key
id_motor	VARCHAR	20	Foreign Key

Tabel 2.2.4 entitas transaksi

- Entitas Pengembalian

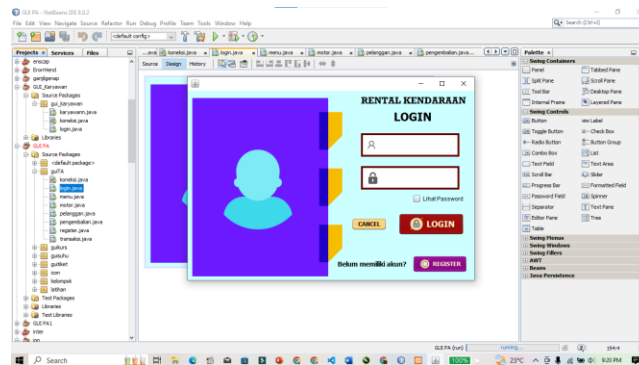
Penjelasan: tabel ini berfungsi untuk data motor yang akan di sewakan. Dalam entitas ini memiliki relasi one to many terhadap entitas transaksi, maka dari itu atribut id_motor menjadi foreign key di entitas transaksi.

Nama Atribut	Data Type	Length	Keterangan
id_pengembalian	VARCHAR	20	Berisi ID Pengembalian motor dan menjadi primary key
keterangan	VARCHAR	20	Berisi keterangan kondisi motor yang sudah selesai di pinjam
id_transaksi	VARCHAR	5	Foreign Key

Tabel 2.2.5 entitas pengembalian

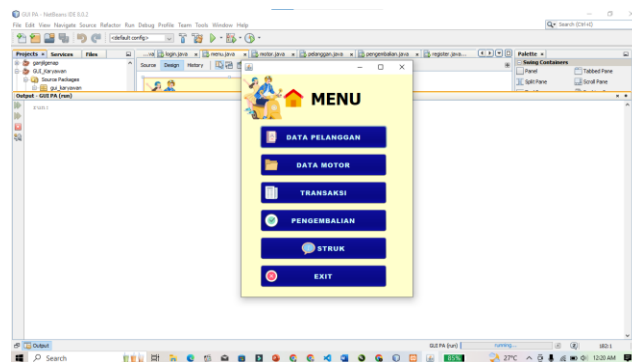
2.3 Desain UI

- Gui Login



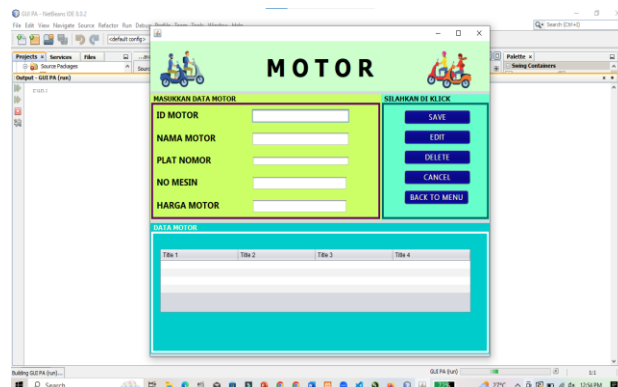
Gambar 2.3.1 GUI login

- Gui Menu



Gambar 2.3.2 GUI menu

- Gui Motor



Gambar 2.3.3 GUI motor

- Gui Pelanggan



Gambar 2.3.4 GUI pelanggan

- Gui Transaksi



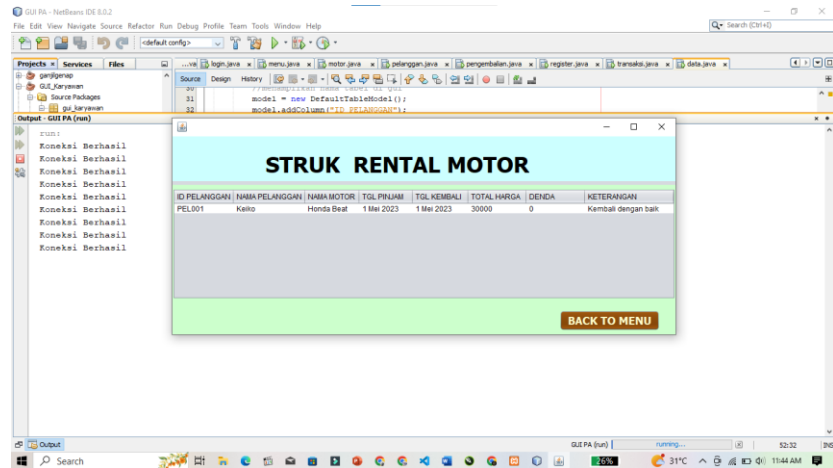
Gambar 2.3.5 GUI transaksi

- Gui Pengembalian



Gambar 2.3.6 GUI pengembalian

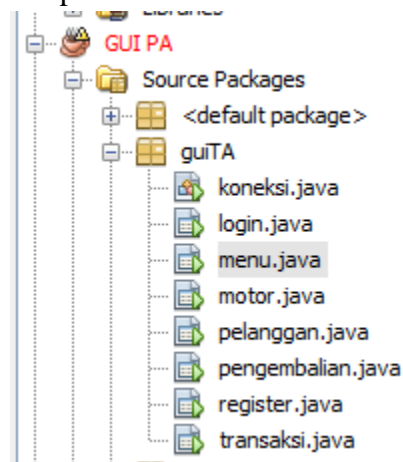
- Gui Struk Rental Motor



Gambar 2.3.7 GUI struk rental motor

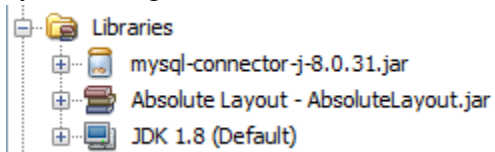
2.4 Kode Program

- Struktur Project Package, terdapat JFrame Form dan Java Class sebagai berikut:



Gambar 2.4.1 struktur project packages

- Libraries, yang digunakan yaitu sebagai berikut:



Gambar 2.4.2 libraries

2.5 Kode Program Aplikasi

1. *Koneksi*

```
package guiTA;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class koneksi {
    //membuat variabel untuk menyimpan koneksinya serta variabel
    static Connection con;
    //membuat function
    public static Statement stm;
    public static Connection koneksiDB() {
        try{
            //stlh itu masukkan driver serta deskripsikan database yg kita buat
            String url ="jdbc:mysql://localhost:3306/rentaldb";
            String user="root";
            String pass="";
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pass);
            stm = con.createStatement();
            System.out.println("Koneksi Berhasil");
        }catch (Exception e){
            System.out.println("Koneksi Gagal" + e.getMessage());
        }
        return con;
    }
    public static void main(String[] args) {
        try{
            Connection c = koneksi.koneksiDB();
            System.out.println(String.format("Connected to database " + c.getCatalog()));
        }catch(SQLException e){
            System.out.println(e);
        }
    }
}
```

2. *Login*

```
package guiTA;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class login extends javax.swing.JFrame {
    private Connection con;
    private Statement stm;
    private ResultSet rs;

    public login() {
        initComponents();
    }

    public static String hashSHA256(String password) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(password.getBytes());
            StringBuilder hexString = new StringBuilder();
            for (int i = 0; i < hash.length; i++) {
                String hex = Integer.toHexString(0xff & hash[i]);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }

    public void lihatpass(){
        if(cek.isSelected()){
            pass.setEchoChar((char)0);
        }
    }
}
```

```

    }else{
        pass.setEchoChar('*');
    }
}

private void loginActionPerformed(java.awt.event.ActionEvent evt) {
    String password = pass.getText().toString().trim();
    String hashedPassword = hashSHA256(password);
    //menggunakan login enkripsi dengan SHA 256
    try{
        //mengambil data dari tabel registrasi
        String sql = "SELECT * FROM registrasi WHERE username='" + user.getText()
            + "' and password='" + hashedPassword + "'";
        //mengambil koneksi yg kita buat
        con = koneksi.koneksiDB();
        //persiapkan statement
        stm = con.createStatement();
        //program dijalankan
        rs = stm.executeQuery(sql);
        if (rs.next()) {
            //login berhasil, maka otomatis ke gui menu / memanggil menu
            JOptionPane.showMessageDialog(null, "Login Berhasil", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
            menu m = new menu();
            m.setVisible(true);
            dispose();

        } else {
            //jika gagal, mungkin user dan pass salah
            JOptionPane.showMessageDialog(null, "Login Gagal. Username dan Password
Salah", "Perhatian", JOptionPane.WARNING_MESSAGE);
            user.setText("");
            pass.setText("");
            user.requestFocus();
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e, "Pesan",
JOptionPane.INFORMATION_MESSAGE);
    }

}

```

```

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

private void regisActionPerformed(java.awt.event.ActionEvent evt) {
    // jika tidak memiliki akun, maka kita harus register
    this.dispose();
    register r = new register();
    r.setVisible(true);
}

private void cekActionPerformed(java.awt.event.ActionEvent evt) {
    //agar passbisa terlihat
    lihatpass();
}

private void batalActionPerformed(java.awt.event.ActionEvent evt) {
    // kosongkan textfield
    user.setText(null);
    pass.setText(null);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new login().setVisible(true);
        }
    });
}

```

3. *Register*

```

package guiTA;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

```

```

import java.sql.Statement;
import javax.swing.JOptionPane;

public class register extends javax.swing.JFrame {
    public register() {
        initComponents();
    }
    //membuat enkripsi SHA256
    public static String hashSHA256(String password) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(password.getBytes());
            StringBuilder hexString = new StringBuilder();
            for (int i = 0; i < hash.length; i++) {
                String hex = Integer.toHexString(0xff & hash[i]);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }
    private void regisActionPerformed(java.awt.event.ActionEvent evt) {
        //membuat atribut
        String username = user.getText().toString().trim();
        String password = pass.getText().toString().trim();
        String hashedPassword = hashSHA256(password);
        String alamat = txtalamat.getText().toString().trim();
        String nohp = txtnohp.getText().toString().trim();
        try{
            //mengambil data
            String sql = "INSERT INTO registrasi VALUES (?, ?, ?, ?)";
            java.sql.Connection con = koneksi.koneksiDB();
            java.sql.PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, username);
            ps.setString(2, hashedPassword);
            ps.setString(3, alamat);
            ps.setString(4, nohp);
        }
    }
}

```



```

        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Registrasi akun berhasil !");

        }catch(SQLException e){
        System.out.println("Error");
        }finally{
        //jika berhasil register, akan otomatis terpanggil halaman login
        this.dispose();
        login l = new login();
        l.setVisible(true);
        }
    }

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

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

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

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new register().setVisible(true);
            }
        });
    }
}

```

4. Menu

```

package guiTA;

import java.awt.Component;
import javax.swing.JOptionPane;

public class menu extends javax.swing.JFrame {
    public menu() {

```

```

        initComponents();
    }

    private void exitActionPerformed(java.awt.event.ActionEvent evt) {
        // untuk keluar dari program
        int a = JOptionPane.showConfirmDialog((Component)null,"Apakah anda ingin
keluar ?", "EXIT",JOptionPane.YES_NO_OPTION);

        if (a==0){
            System.exit(0);
        }
    }

    private void pelangganActionPerformed(java.awt.event.ActionEvent evt) {
        ///jika kita klic pelanggan, maka akan memanggil gui pelanggan, dan terbuka
        pelanggan fromm = new pelanggan();
        fromm.setVisible(true);
        dispose();
    }

    private void motorActionPerformed(java.awt.event.ActionEvent evt) {
        ///jika kita klic motor maka akan memanggil gui motor, dan terbuka
        motor fromm = new motor();
        fromm.setVisible(true);
        dispose();
    }

    private void transaksiActionPerformed(java.awt.event.ActionEvent evt) {
        //jika kita klic transaksi, maka akan memanggil gui transaksi, dan terbuka
        transaksi fromm = new transaksi();
        fromm.setVisible(true);
        dispose();
    }

    private void kembaliActionPerformed(java.awt.event.ActionEvent evt) {
        //jika kita klic pengembalian, maka akan memanggil gui pengembalian, dan terbuka
        pengembalian fromm = new pengembalian();
        fromm.setVisible(true);
        dispose();
    }

```

```

    }
    public static void main(String args[]) {
        try {
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new menu().setVisible(true);
                }
            });
        }
    }
}

```

5. **Motor**

```

package guiTA;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class motor extends javax.swing.JFrame {
    private DefaultTableModel model;
    public Connection conn = koneksi.koneksiDB();
    private String SQL;

    public motor() {
        initComponents();
        tampil();
        autonumber();
    }

    public void tampil(){
        //menamai kolom tabel yang ada di gui
        model = new DefaultTableModel();
        model.addColumn("ID MOTOR");
        model.addColumn("NAMA MOTOR");
        model.addColumn("PLAT MOTOR");
        model.addColumn("NO MESIN");
        model.addColumn("HARGA MOTOR");
        tabel.setModel(model);

        //cara untuk memunculkan data
    }
}

```

```

try{
    java.sql.Statement stmt = conn.createStatement();
    SQL = "select * from motor";
    java.sql.ResultSet res = stmt.executeQuery(SQL);
    while (res.next()){
        model.addRow(new Object []{
            //nama2ini harus sesuai dengan database yg kita buat
            res.getString("id_motor"),
            res.getString("nama_motor"),
            res.getString("plat_motor"),
            res.getString("no_mesin"),
            res.getString("harga_motor"),});
    }
} catch(SQLException e){
    System.out.println(e.getMessage());
}
}

void tabel(){
    //codingan ini berfungsi untuk apa yang sudah diinputkan di textfield, bisa terbaca di
tabel
    txtid.setText(tabel.getValueAt(tabel.getSelectedRow(), 0)+"");
    txtmotor.setText(tabel.getValueAt(tabel.getSelectedRow(), 1)+"");
    txtplat.setText(tabel.getValueAt(tabel.getSelectedRow(), 2)+"");
    txtmesin.setText(tabel.getValueAt(tabel.getSelectedRow(), 3)+"");
    txtharga.setText(tabel.getValueAt(tabel.getSelectedRow(), 4)+"");
    ubah.setEnabled(true);
    hapus.setEnabled(true);
    simpan.setEnabled(true);

}

private void autonumber(){
    //agar ID motor bisa langsung otomatis dan bisa urut
    String noMot = "MTR000";
    int i = 0;
    try{
        Statement st = conn.createStatement();
        //mengambil id motor dan diurutkan
        String sql = "select id_motor from motor";
        ResultSet rs = st.executeQuery(sql);

```

```

        while(rs.next()){
            noMot = rs.getString("id_motor");
        }
        noMot = noMot.substring(4);
        i = Integer.parseInt(noMot)+1;
        noMot = "00" +i;
        noMot = "MTR"+ noMot.substring(noMot.length()-3);
        txtid.setText(noMot);
    }catch (SQLException e){
        System.out.println(e.getMessage());
    }
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // untuk kembali ke menu
    menu home = new menu();
    home.setVisible(true);
    dispose();}

private void txtidActionPerformed(java.awt.event.ActionEvent evt) {
    // memanggil method autonumber, agar id motor bisa terbaca dan otomatis
    autonumber();
}

private void resetActionPerformed(java.awt.event.ActionEvent evt) {
    // untuk mengosongkan textfield
    txtid.setText(null);
    txtmotor.setText(null);
    txtplat.setText(null);
    txtmesin.setText(null);
    txtharga.setText(null);
}

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // untuk menambah data
    try{
        String sql = " INSERT INTO motor VALUES ('" + txtid.getText()
            + "','" + txtmotor.getText()
            + "','" + txtplat.getText()
            + "','" + txtmesin.getText()
            + "','" + txtharga.getText() + "')";
        //untuk menyiapkan statementnya
        java.sql.PreparedStatement ps = conn.prepareStatement(sql);
    }
}

```

```

        //untuk menjalankan statementnya
        ps.execute();
        JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
        tampil();

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Simpan Gagal" + e.getMessage(),
        "Failed", JOptionPane.ERROR_MESSAGE);
    }
}

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

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

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // hapus data
    try {
        //hapus data dari id motor berapa
        String sql = "delete from motor where id_motor =" + txtid.getText() + """;
        java.sql.PreparedStatement s = conn.prepareStatement(sql);
        s.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Motor Berhasil Dihapus",
        "Pesan", JOptionPane.INFORMATION_MESSAGE);
        tampil();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
        JOptionPane.INFORMATION_MESSAGE);
    }
}

private void ubahActionPerformed(java.awt.event.ActionEvent evt) {
    // ubah data
    try {
        //untuk apa saja yang mau diubah, dari id_motor berapa

```

```
String sql ="UPDATE motor set nama_motor=?, plat_motor=?, no_mesin=?,
harga_motor=? where id_motor=?";
```

```
PreparedStatement s = conn.prepareStatement(sql);
```

```
String kod = txtid.getText();
s.setString(1, txtmotor.getText());
s.setString(2, txtplat.getText());
s.setString(3, txtmesin.getText());
s.setString(4, txtharga.getText());
s.setString(5, kod);
s.executeUpdate();
JOptionPane.showMessageDialog(null, "Data Berhasil Dirubah",
"Pesan",JOptionPane.INFORMATION_MESSAGE);
tampil();

}catch (SQLException e){
    JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE);
    System.out.println(e.getMessage());
} }

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

6. *Pelanggan*

```
package guiTA;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;
import java.sql.ResultSet;

public class pelanggan extends javax.swing.JFrame {
private DefaultTableModel model;
```

```

public Connection conn = koneksi.koneksiDB();
private String SQL;
    public pelanggan() {
        initComponents();
        tampil();
        autonumber();
    }

    public void tampil(){
        //untuk menampilkan nama tabel di gui
        model = new DefaultTableModel();
        model.addColumn("ID PELANGGAN");
        model.addColumn("NAMA PELANGGAN");
        model.addColumn("NIK");
        model.addColumn("JENIS KELAMIN");
        model.addColumn("ALAMAT");
        model.addColumn("NO TELEPON");
        tabel.setModel(model);

        try{
            //cara untuk memunculkan data
            java.sql.Statement stmt = conn.createStatement();
            SQL = "select * from pelanggan";
            java.sql.ResultSet res = stmt.executeQuery(SQL);
            while (res.next()){
                model.addRow(new Object []{
                    res.getString("id_pelanggan"),
                    res.getString("nama_pelanggan"),
                    res.getString("nik"),
                    res.getString("jenis_kelamin"),
                    res.getString("alamat"),
                    res.getString("nomor_telepon"),});
            }
        }catch(SQLException e){
            System.out.println(e.getMessage());
        }
    }

    void tabel(){

```


//codingan ini berfungsi untuk apa yang sudah diinputkan di textfield, bisa terbaca di tabel

```
txtid.setText(tabel.getValueAt(tabel.getSelectedRow(), 0)+"");
txtnama.setText(tabel.getValueAt(tabel.getSelectedRow(), 1)+"");
txtnik.setText(tabel.getValueAt(tabel.getSelectedRow(), 2)+"");
jk.setSelectedItem(tabel.getValueAt(tabel.getSelectedRow(),3) + "");
txtalamat.setText(tabel.getValueAt(tabel.getSelectedRow(), 4)+"");
txtno.setText(tabel.getValueAt(tabel.getSelectedRow(), 5)+"");
ubah.setEnabled(true);
hapus.setEnabled(true);
simpan.setEnabled(false);

}
```

```
private void autonumber(){
    //untuk membuat autonumber di id-pelanggan
    String noPel = "PEL000";
    int i = 0;
    try{
        Statement st = conn.createStatement();
        //mengambil dari id_pelanggan dan diurutkan
        String sql = "select id_pelanggan from pelanggan";
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            noPel = rs.getString("id_pelanggan");
        }
        noPel = noPel.substring(4);
        i = Integer.parseInt(noPel)+1;
        noPel = "00" +i;
        noPel = "PEL"+ noPel.substring(noPel.length()-3);
        txtid.setText(noPel);
    }catch (SQLException e){
        System.out.println(e.getMessage()); } }
```

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

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

```

    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // kembali ke menu
        menu home = new menu();
        home.setVisible(true);
        dispose();
    }

    private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
        // Tcara untuk menyimpan data
        try{
            String sql = " INSERT INTO pelanggan VALUES (" + txtid.getText()
                + "," + txtnama.getText()
                + "," + txtnik.getText()
                + "," + jk.getSelectedIndex()
                + "," + txtalamat.getText()
                + "," + txtno.getText() + ")";
            //menyiapkan statement
            java.sql.PreparedStatement ps = conn.prepareStatement(sql);
            //menjalankan statement
            ps.execute();
            JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
            tampil();

        }catch(SQLException e){
            JOptionPane.showMessageDialog(null, "Simpan Gagal" + e.getMessage(),
                "Failed", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void resetActionPerformed(java.awt.event.ActionEvent evt) {
        // untuk mengosongkan textfield
        txtid.setText(null);
        txtnama.setText(null);
        txtnik.setText(null);
        jk.setSelectedIndex("Pilih");
        txtalamat.setText(null);
        txtno.setText(null);
    }

```

```

private void ubahActionPerformed(java.awt.event.ActionEvent evt) {
    // Tuntuk ubah data
    try{
        String sql ="update pelanggan set nama_pelanggan=?, nik=?, jenis_kelamin=?,
alamat=?, nomor_telepon=? where id_pelanggan=?";
        //menyiapkan statement
        PreparedStatement s = conn.prepareStatement(sql);

        String kod = txtid.getText();
        s.setString(1, txtnama.getText());
        s.setString(2, txtnik.getText());
        s.setObject(3, jk.getSelectedItem());
        s.setString(4, txtalamat.getText());
        s.setString(5, txtno.getText());
        s.setString(6, kod);
        //menjalankan statement
        s.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Berhasil Dirubah",
"Pesan",JOptionPane.INFORMATION_MESSAGE);
        tampil();

    }catch (SQLException e){
        JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE);
        System.out.println(e.getMessage());
    }
}

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // hapus data
    try{
        String sql = "delete from pelanggan where id_pelanggan =" + txtid.getText()+
"";
        java.sql.PreparedStatement s = conn.prepareStatement(sql);
        s.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Pelanggan Berhasil Dihapus",
"Pesan",JOptionPane.INFORMATION_MESSAGE);
        tampil();
    }catch(SQLException e){

```

```

        JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE); }}
    private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        tabel();
    }

    private void txtidActionPerformed(java.awt.event.ActionEvent evt) {
autonumber();
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new pelanggan().setVisible(true);
            }
        });
    }
}

```

7. *Transaksi*

```

package guiTA;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;

public class transaksi extends javax.swing.JFrame {
    private DefaultTableModel model;
    public Connection conn = koneksi.koneksiDB();
    private String SQL;

    public transaksi() {
        initComponents();
        autonumber();
        tampil();
        tabel2();
        tabel3();
    }

    private void autonumber(){
        //membuat autonumber di id_transaksi
        String noTr = "TR000";
    }
}

```

```

int i = 0;
try{
    Statement st = conn.createStatement();
    String sql = "select id_transaksi from transaksi";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()){
        noTr = rs.getString("id_transaksi");
    }
    noTr = noTr.substring(4);
    i = Integer.parseInt(noTr)+1;
    noTr = "00" +i;
    noTr = "TR"+ noTr.substring(noTr.length()-3);
    txtid.setText(noTr);
}catch (SQLException e){
    System.out.println(e.getMessage());
}
}

public void tampil(){
    //menampilkan nama tabel
    model = new DefaultTableModel();
    model.addColumn("ID TRANSAKSI");
    model.addColumn("ID PELANGGAN");
    model.addColumn("TGL PINJAM");
    model.addColumn("TGL KEMBALI");
    model.addColumn("ID MOTOR");
    model.addColumn("HARGA MOTOR");
    model.addColumn("LAMA PINJAM");
    model.addColumn("TOTAL HARGA");
    tabel.setModel(model);

    try{
        //mengambil data dari data transaksi
        java.sql.Statement stmt = conn.createStatement();
        SQL = "select * from transaksi";
        java.sql.ResultSet res = stmt.executeQuery(SQL);
        while (res.next()){
            model.addRow(new Object []{
                res.getString(1),
                res.getString(2),

```

```

        res.getString(3),
        res.getString(4),
        res.getString(5),
        res.getString(6),
        res.getString(7),
        res.getString(8),});
    }
} catch(SQLException e){
    System.out.println(e.getMessage());
}
}

void tabel(){
    txtid.setText(tabel.getValueAt(tabel.getSelectedRow(), 0)+"");
    idpelangganfk.setText(tabel.getValueAt(tabel.getSelectedRow(), 1)+"");
    tglpinjam.setText(tabel.getValueAt(tabel.getSelectedRow(), 3)+"");
    tglkembali.setText(tabel.getValueAt(tabel.getSelectedRow(), 4)+"");
    idmotorfk.setText(tabel.getValueAt(tabel.getSelectedRow(), 5)+"");
    hargafk.setText(tabel.getValueAt(tabel.getSelectedRow(), 6)+"");
    lamapinjam.setSelectedItem(tabel.getValueAt(tabel.getSelectedRow(),7) + "");
    txttotal.setText(tabel.getValueAt(tabel.getSelectedRow(), 8)+"");
    ubah.setEnabled(true);
    hapus.setEnabled(true);
    simpan.setEnabled(false);

}

private void tabel2(){
    //memanggil data dari pelanggan
    DefaultTableModel model = new DefaultTableModel();
    //hanya id_pelanggan saja yg terpanggil
    model.addColumn("id_pelanggan");
    try{

        java.sql.Connection c = koneksi.koneksiDB();
        java.sql.Statement s = c.createStatement();
        String sql = "SELECT * FROM pelanggan";
        java.sql.ResultSet r = s.executeQuery(sql);
        while(r.next()) {
            model.addRow(new Object[]{

```

```

        r.getString("id_pelanggan")
    });
}
tpel.setModel(model);
} catch (Exception ex) {
    JOptionPane.showMessageDialog(this, ex.getMessage());
}
}

void klik2(){
    //memanggil id_pelanggan
    idpelangganfk.setText(tpel.getValueAt(tpel.getSelectedRow(),0)+"");
}

private void tabel3(){
    //memanggil data dari data motor
    DefaultTableModel model = new DefaultTableModel();
    //manamai tabel
    model.addColumn("id_motor");
    model.addColumn("harga_motor");
    try{
        //mengambil data motor, yg diambil hanya id_motor dan harga_motor
        java.sql.Connection c = koneksi.koneksiDB();
        java.sql.Statement s = c.createStatement();
        String sql = "SELECT * FROM motor";
        java.sql.ResultSet r = s.executeQuery(sql);
        while(r.next()) {
            model.addRow(new Object[]{

                r.getString("id_motor"),
                r.getString("harga_motor")
            });
        }
        tmotor.setModel(model);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

void klik3(){
    //menampilkan

```

```

        idmotorfk.setText(tmotor.getValueAt(tmotor.getSelectedRow(),0)+"");
        hargafk.setText(tmotor.getValueAt(tmotor.getSelectedRow(),1)+"");
    }

    @SuppressWarnings("unchecked")

    private void menuActionPerformed(java.awt.event.ActionEvent evt) {
        // untuk kembali ke menu
        menu home = new menu();
        home.setVisible(true);
        dispose();
    }

    private void resetActionPerformed(java.awt.event.ActionEvent evt) {
        //untuk mengosongkan textfield
        txtid.setText(null);
        idpelangganfk.setText(null);
        tglpinjam.setText(null);
        tglkembali.setText(null);
        idmotorfk.setText(null);
        hargafk.setText(null);
        lamapinjam.setSelectedItem("Pilih");
        txttotal.setText(null);
    }

    private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            //hapus data
            String sql = "delete from transaksi where id_transaksi =" + txtid.getText() + """;
            java.sql.PreparedStatement s = conn.prepareStatement(sql);
            s.executeUpdate();
            JOptionPane.showMessageDialog(null, "Data Pelanggan Berhasil Dihapus",
            "Pesan",JOptionPane.INFORMATION_MESSAGE);
            tampil();
        }catch(SQLException e){
            JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
            JOptionPane.INFORMATION_MESSAGE);
        }
    }

    private void ubahActionPerformed(java.awt.event.ActionEvent evt) {
        try{

```



```

//rubah data
String sql ="update transaksi set nama_pelanggan=?, nik=?, jenis_kelamin=?,
alamat=?, no_telp=? where id_pelanggan=?";

PreparedStatement s = conn.prepareStatement(sql);

String kod = txtid.getText();
s.setString(1, idpelangganfk.getText());
s.setString(2, tglpinjam.getText());
s.setString(3, tglkembali.getText());
s.setString(4, idmotorfk.getText());
s.setString(5, hargafk.getText());
s.setObject(6, lamapinjam.getSelectedItem());
s.setString(7, txttotal.getText());
s.setString(8, kod);
s.executeUpdate();
JOptionPane.showMessageDialog(null, "Data Berhasil Dirubah",
"Pesan",JOptionPane.INFORMATION_MESSAGE);
tampil();

} catch (SQLException e){
    JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE);
    System.out.println(e.getMessage());
}
}

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
try{
//menambah data
String sql = " INSERT INTO transaksi VALUES ('" + txtid.getText()
+ "','" + idpelangganfk.getText()
+ "','" + tglpinjam.getText()
+ "','" + tglkembali.getText()
+ "','" + idmotorfk.getText()
+ "','" + hargafk.getText()
+ "','" + lamapinjam.getSelectedItem()
+ "','" + txttotal.getText() + "')";
java.sql.PreparedStatement ps = conn.prepareStatement(sql);
ps.execute();

```

```

        JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
        tampil();

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Simpan Gagal" + e.getMessage(),
        "Failed", JOptionPane.ERROR_MESSAGE);
    }
}

private void txtidActionPerformed(java.awt.event.ActionEvent evt) {
    autonumber();
}

private void tpelMouseClicked(java.awt.event.MouseEvent evt) {
    klik2();    // TODO add your handling code here:
}

private void tmotorMouseClicked(java.awt.event.MouseEvent evt) {
    klik3();    // TODO add your handling code here:
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new transaksi().setVisible(true);
        }
    });
}

```

8. *Pengembalian*

```

package guiTA;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class pengembalian extends javax.swing.JFrame {
    private DefaultTableModel model;
    public Connection conn = koneksi.koneksiDB();
    private String SQL;
    public pengembalian() {

```

```

initComponents();
autonumber();
tampil();
tabel2();
}

private void autonumber(){
    //membuat auto number di id_pengembalian
String noKmb = "KMB000";
int i = 0;
try{
    Statement st = conn.createStatement();
    String sql = "select id_pengembalian from pengembalian";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()){
        noKmb = rs.getString("id_pengembalian");
    }
    noKmb = noKmb.substring(4);
    i = Integer.parseInt(noKmb)+1;
    noKmb = "00" +i;
    noKmb = "KMB"+ noKmb.substring(noKmb.length()-3);
    txtid.setText(noKmb);
}catch (SQLException e){
    System.out.println(e.getMessage());
}
}

public void tampil(){
    //menampilkan nama tabel di gui
    model = new DefaultTableModel();
    model.addColumn("ID PENGEMBALIAN");
    model.addColumn("ID TRANSAKSI");
    model.addColumn("TGL PINJAM");
    model.addColumn("TGL KEMBALI");
    model.addColumn("DENDA");
    model.addColumn("KETERANGAN");
    tabel.setModel(model);

    try{

```

```

        //mengambil data
        java.sql.Statement stmt = conn.createStatement();
        SQL = "select * from pengembalian";
        java.sql.ResultSet res = stmt.executeQuery(SQL);
        while (res.next()){
            model.addRow(new Object []{
                res.getString("id_pengembalian"),
                res.getString("id_transaksi"),
                res.getString("tgl_pinjam"),
                res.getString("tgl_kembali"),
                res.getString("denda"),
                res.getString("keterangan"),});
        }
    } catch (SQLException e){
        System.out.println(e.getMessage());
    }
}

void tabel(){
    //aga data bisa terbaca di tabel gui pengembalian
    txtid.setText(tabel.getValueAt(tabel.getSelectedRow(), 0)+"");
    idtransfk.setText(tabel.getValueAt(tabel.getSelectedRow(), 1)+"");
    tglpinjamfk.setText(tabel.getValueAt(tabel.getSelectedRow(), 2)+"");
    tglkembalikf.setText(tabel.getValueAt(tabel.getSelectedRow(), 4)+"");
    txtidenda.setText(tabel.getValueAt(tabel.getSelectedRow(), 5)+"");
    txtket.setText(tabel.getValueAt(tabel.getSelectedRow(), 6)+"");
    ubah.setEnabled(true);
    hapus.setEnabled(true);
    simpan.setEnabled(false);
}

private void tabel2(){
    //untuk menampilkan data transaksi
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("id_transaksi");
    model.addColumn("tgl_pinjam");
    model.addColumn("tgl_kembali");
    try{
        //mengambil data transaksi

```

```

        java.sql.Connection c = koneksi.koneksiDB();
        java.sql.Statement s = c.createStatement();
        String sql = "SELECT * FROM transaksi";
        java.sql.ResultSet r = s.executeQuery(sql);
        while(r.next()) {
            model.addRow(new Object[]{

                r.getString("id_transaksi"),
                r.getString("tgl_pinjam"),
                r.getString("tgl_kembali")
            });
        }
        t_trx.setModel(model);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

void klik2(){
    //agar data dari transaksi bisa terbaca
    idtransfk.setText(t_trx.getValueAt(t_trx.getSelectedRow(),0)+"");
    tglpinjamfk.setText(t_trx.getValueAt(t_trx.getSelectedRow(),1)+"");
    tglkembalikf.setText(t_trx.getValueAt(t_trx.getSelectedRow(),2)+"");
}

private void menuActionPerformed(java.awt.event.ActionEvent evt) {
    // kembali memanggil menu
    menu home = new menu();
    home.setVisible(true);
    dispose();
}

private void resetActionPerformed(java.awt.event.ActionEvent evt) {
    //mengosongkan textfield
    txtid.setText(null);
    idtransfk.setText(null);
    tglpinjamfk.setText(null);
    tglkembalikf.setText(null);
    txtenda.setText(null);
    txtket.setText(null);
}

```

```

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //simpan data
        String sql = " INSERT INTO pengembalian VALUES (" + txtid.getText()
        + "," + idtransfk.getText()
        + "," + tglpinjamfk.getText()
        + "," + tglkembalifk.getText()
        + "," + txtkdenda.getText()
        + "," + txtket.getText() + ")";
        //menyiapkan statement
        java.sql.PreparedStatement ps = conn.prepareStatement(sql);
        //statement dijalankan
        ps.execute();
        JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
        tampil();

    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Simpan Gagal" + e.getMessage(),
        "Failed", JOptionPane.ERROR_MESSAGE);
    }
}

private void ubahActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        //rubah data
        String sql ="update pengembalian set nama_pelanggan=?, nik=?,
jenis_kelamin=?, alamat=?, no_telp=? where id_pelanggan=?";

        PreparedStatement s = conn.prepareStatement(sql);

        String kod = txtid.getText();
        s.setString(1, idtransfk.getText());
        s.setString(2, tglpinjamfk.getText());
        s.setString(3, tglkembalifk.getText());
        s.setString(4, txtkdenda.getText());
        s.setString(5, txtket.getText());
        s.setString(6, kod);
        s.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Berhasil Dirubah",
        "Pesan",JOptionPane.INFORMATION_MESSAGE);
    }
}

```

```

        tampil();

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE);
        System.out.println(e.getMessage());
    }
}

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        //hapus data
        String sql = "delete from pengembalian where id_pengembalian =" +
txtid.getText()+ """;
        java.sql.PreparedStatement s = conn.prepareStatement(sql);
        s.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Pelanggan Berhasil Dihapus",
"Pesan",JOptionPane.INFORMATION_MESSAGE);
        tampil();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Gagal " + e.getMessage(), "Pesan",
JOptionPane.INFORMATION_MESSAGE);
    }
}

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

private void txtidActionPerformed(java.awt.event.ActionEvent evt) {
    autonumber();
}

private void t_trxMouseClicked(java.awt.event.MouseEvent evt) {
    klik2();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

```

```

        public void run() {
            new pengembalian().setVisible(true);
        }
    });
}

```

9. Data (Join)

```

public class data extends javax.swing.JFrame {
    private DefaultTableModel model;
    public Connection conn = koneksi.koneksiDB();
    private String SQL;
    public data() {
        initComponents();
        tampil();
    }
    public void tampil(){
        DefaultTableModel model = new DefaultTableModel();
        //menampilkan nama tabel di gui
        model = new DefaultTableModel();
        model.addColumn("ID PELANGGAN");
        model.addColumn("NAMA PELANGGAN");
        model.addColumn("NAMA MOTOR");
        model.addColumn("TGL PINJAM");
        model.addColumn("TGL KEMBALI");
        model.addColumn("TOTAL HARGA");
        model.addColumn("DENDA");
        model.addColumn("KETERANGAN");
        tabel.setModel(model);

        try{
            //mengambil data
            java.sql.Statement stmt = conn.createStatement();
            SQL = "SELECT p.id_pelanggan, p.nama_pelanggan, m.nama_motor,
t.tgl_pinjam, t.tgl_kembali, t.total_harga, a.denda, a.keterangan
FROM transaksi t JOIN pelanggan p ON (p.id_pelanggan = t.id_pelanggan)
JOIN motor m ON (m.id_motor = t.id_motor)
JOIN pengembalian a ON (a.id_transaksi = t.id_transaksi)";
            java.sql.ResultSet res = stmt.executeQuery(SQL);
            while (res.next()){
                model.addRow(new Object []{

```



```

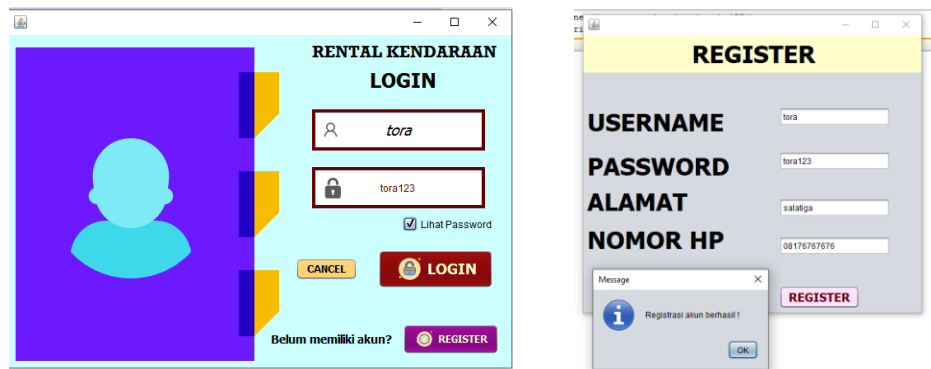
        res.getString("id_pelanggan"),
        res.getString("nama_pelanggan"),
        res.getString("nama_motor"),
        res.getString("tgl_pinjam"),
        res.getString("tgl_kembali"),
        res.getString("total_harga"),
        res.getString("denda"),
        res.getString("keterangan"),});
    }
} catch(SQLException e){
    System.out.println(e.getMessage());
}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    menu home = new menu();
    home.setVisible(true);
    dispose();
}

```

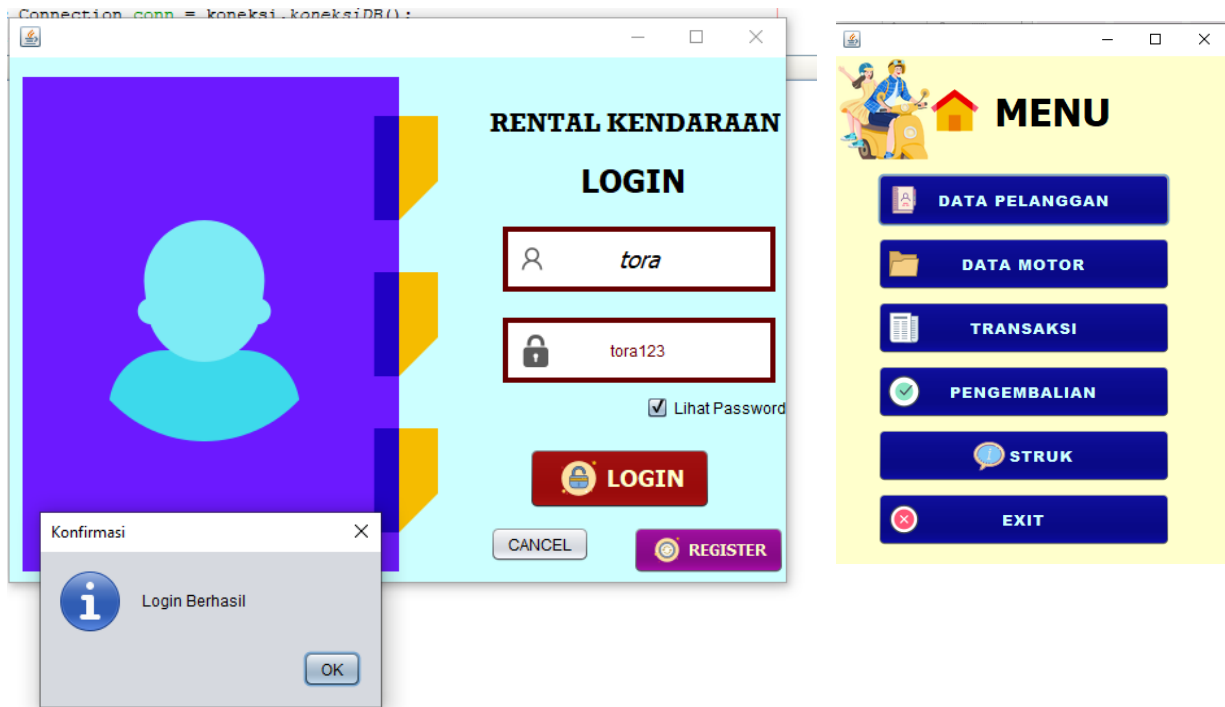
2.6 Demo Aplikasi

- Pertama jalankan login, jika belum memiliki akun maka harus register terlebih dahulu



Gambar 2.6.1 GUI login dan register

- Setelah login berhasil, otomatis program akan membuka menu, yang akan menampilkan data pelanggan, data motor, transaksi, pengembalian dan exit.



Gambar 2.6.2 GUI login beserta menu

- Setelah membuka menu, jika ada pelanggan yang ingin memesan motor, kita kan menambah data pelanggan dahulu, untuk mengisi data pelanggan yang akan sewa motor.

ID PELANGGAN	NAMA PELANGG.	NIK	JENIS KELAMIN	ALAMAT	NO TELEPON
PEL001	Kelko	5656565655	P	Salatiga	081222232
PEL002	Juan	90909078787	L	Semarang	081790908787
PEL003	Tegar	7676767677	L	Ambarawa	09871212121

Gambar 2.6.3 GUI proses input data pelanggan

ID PELANGGAN	NAMA PELANGG...	NIK	JENIS KELAMIN	ALAMAT	NO TELEPON
PEL002	Ajeng	909090909	P	Salatiga	081789890707
PEL001	dion	22222222	L	bali	090909091

Gambar 2.6.4 GUI data pelanggan berhasil diinput

- Setelah data ditambahkan, maka admin akan menginputkan data motor yang akan di sewa, Dalam gui motor ini, kita bisa menambah, menghapus, rubah data motor.

ID MOTOR	NAMA MOTOR	PLAT MOTOR	NO MESIN	HARGA MOTOR
MTR001	Honda Beat	H 3455 JKL	90KLP	30000
MTR002	Honda Vario 150	H 6789 JHG	KL0987	45000
MTR003	Honda Vario 125	H 5454 HGF	KL090	40000

Gambar 2.6.5 GUI penambahan data motor

- Setelah udah melihat motor, maka lanjut ke menu transaksi, untuk pencatatan peminjaman motor dan mengetahui transaksinya. Di gui transaksi ini, bisa langsung terbaca data pelanggan dan data motor yang sudah di insert, jadi bisa langsung otomatis

terpanggil di tabel transaksi ini, karena mengambil Foreign Key dari tabel transaksi, yaitu terdapat Id_pelanggan dan id_motor

TRANSAKSI

MASUKKAN DATA TRANSAKSI

ID TRANSAKSI: TR002 LAMA PINJAM: 12 Jam

ID PELANGGAN: PEL002 TOTAL HARGA: 60000

ID MOTOR: MTR002

HARGA: 30000

TGL PINJAM: 2 Mei 2023 TGL KEMBALI: 2 Mei 2023

SILAHKAN DI KLIK

SAVE EDIT DELETE CANCEL BACK TO MENU

DATA TRANSAKSI

ID TRANSAKSI	ID PELANGG...	TGL PINJAM	TGL KEMBALI	ID MOTOR	HARGA MOT...	LAMA PINJAM	TOTAL HARGA
TR001	PEL001	1 Mei 2023	1 Mei 2023	MTR001	30000	6 Jam	30000

id_pelanggan: PEL001, PEL002

id_pelanggan: MTR001, MTR002, MTR003, MTR004, MTR005, MTR006

harga_motor: 30000, 30000, 40000, 40000, 40000, 35000

Gambar 2.6.6 GUI transaksi

TRANSAKSI

MASUKKAN DATA TRANSAKSI

ID TRANSAKSI: TR004 LAMA PINJAM: 6 Jam

ID PELANGGAN: TOTAL HARGA:

ID MOTOR:

HARGA:

TGL PINJAM: TGL KEMBALI:

SILAHKAN DI KLIK

SAVE EDIT DELETE CANCEL BACK TO MENU

DATA TRANSAKSI

ID TRANSAKSI	ID PELANGG...	TGL PINJAM	TGL KEMBALI	ID MOTOR	HARGA MOT...	LAMA PINJAM	TOTAL HARGA
TR001	PEL001	1 Mei 2023	1 Mei 2023	MTR001	30000	6 Jam	30000
TR002	PEL002	2 Mei 2023	2 Mei 2023	MTR002	45000	6 Jam	45000
TR003	PEL003	4 Mei 2023	4 Mei 2023	MTR003	40000	6 Jam	40000

id_pelanggan: PEL001, PEL002, PEL003

id_motor: MTR001, MTR002, MTR003

harga_motor: 30000, 45000, 40000

Gambar 2.6.7 GUI transaksi setelah data diinput

- Setelah dari data transaksi, dan data sudah tercatat, jika pelanggan ingin mengembalikan motor maka kembali ke menu dan pilih menu pengembalian. Di dalam GUI pengambilan ini terdapat Foreign Key dari tabel motor, yaitu id_transaksi. ID tersebut juga bisa langsung terpanggil di tabel transaksi

PENGEMBALIAN MOTOR

MASUKAN DATA PENGEMBALIAN

ID PENGEMBALIAN: KMB002

ID TRANSAKSI:

TGL PINJAM:

TGL KEMBALI:

DENDA:

KETERANGAN:

SILAHKAN DI KLIK

SAVE

EDIT

DELETE

CANCEL

BACK TO MENU

DATA PENGEMBALIAN

id_transaksi	tgl_pinjam	tgl_kembali
TR001	1 Mei 2023	1 Mei 2023
TR002	2 Mei 2023	2 Mei 2023

ID PENGEMBA...	ID TRANSAKSI	TGL PINJAM	TGL KEMBALI	DENDA	KETERANGAN
KMB001	TR001	1 Mei 2023	1 Mei 2023	0	Motor dalam ko...

Gambar 2.6.8 GUI pengembalian motor

- Setelah itu data bisa masuk dan tercatat dalam tabel pengembalian motor. Sesudah itu, bisa kembali ke munu.

PENGEMBALIAN MOTOR

MASUKAN DATA PENGEMBALIAN

ID PENGEMBALIAN: KMB003

ID TRANSAKSI: TR003

TGL PINJAM: 4 Mei 2023

TGL KEMBALI: 4 Mei 2023

DENDA: 0

KETERANGAN: Kondisi motor baik

SILAHKAN DI KLIK

SAVE

EDIT

DELETE

CANCEL

BACK TO MENU

DATA PENGEMBALIAN

id_transaksi	tgl_pinjam	tgl_kembali
TR001	1 Mei 2023	1 Mei 2023
TR002	2 Mei 2023	2 Mei 2023
TR003	4 Mei 2023	4 Mei 2023

ID PENGEMBA...	ID TRANSAKSI	TGL PINJAM	TGL KEMBALI	DENDA	KETERANGAN
KMB001	TR001	1 Mei 2023	1 Mei 2023	0	Kembali denga...
KMB002	TR002	2 Mei 2023	2 Mei 2023	0	Baik seperti se...
KMB003	TR003	4 Mei 2023	4 Mei 2023	0	Kondisi motor b...

Gambar 2.6.9 GUI pengembalian setelah data diinput

- Kemudian, setelah kembali ke menu, terdapat pilihan struk, di situ akan mengetahui stuk penyewaan dari rental motor, dan proses selesai.

ID PELANGGAN	NAMA PELANG...	NAMA MOTOR	TGL PINJAM	TGL KEMBALI	TOTAL HARGA	DENDA	KETERANGAN
PEL001	Keiko	Honda Beat	1 Mei 2023	1 Mei 2023	30000	0	Kembali denga...
PEL002	Juan	Honda Vario 150	2 Mei 2023	2 Mei 2023	45000	0	Baik seperti se...
PEL003	Tegar	Honda Vario 125	4 Mei 2023	4 Mei 2023	40000	0	Kondisi motor ...

BACK TO MENU

Gambar 2.6.10 GUI melihat data master / struk

- Semua data yang diinputkan akan masuk dan tercatat di dalam database
 - Data register, di database password tercatat huruf acak, dikarenakan dalam login menggunakan SHA-256.

username	password	alamat	no_hp
amatya	dbbd370f2f4c9fde0632b7526995162989b8dbbaf5af785bba...	salatiga	08211111
tora	f26faa4016ed70e1b6d1f48257a2b0ad3b47278fa6af2aebc8...	salatiga	08176767676

Gambar 2.6.11 database enkripsi 256 login

- Data pelanggan

id_pelanggan	nama_pelanggan	NIK	jenis_kelamin	alamat	nomor_telepon
PEL001	Keiko	5656565655	P	Salatiga	0812222232
PEL002	Juan	90909078787	L	Semarang	081790908787
PEL003	Tegar	76767676677	L	Ambarawa	09871212121

Gambar 2.6.12 database pelanggan

c. Data motor

Showing rows 0 - 2 (3 total, Query took 0.0091 seconds.)

`SELECT * FROM `motor``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	ID_motor	nama_motor	plat_motor	no_mesin	harga_motor
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	MTR001	Honda Beat	H 3456 JKL	90KLP	30000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	MTR002	Honda Vario 150	H 6789 JHG	KL0987	45000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	MTR003	Honda Vario 125	H 5454 HGF	KL090	40000

Gambar 2.6.13 database motor

d. Data transaksi

Showing rows 0 - 1 (2 total, Query took 0.0014 seconds.)

`SELECT * FROM `transaksi``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id_transaksi	id_pelanggan	tgl_pinjam	tgl_kembali	id_motor	harga_motor	lamapinjam	total_harga
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	TR001	PEL001	1 Mei 2023	1 Mei 2023	MTR001	30000	6 Jam	30000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	TR002	PEL002	2 Mei 2023	2 Mei 2023	MTR002	30000	12 Jam	60000

Gambar 2.6.14 database transaksi

e. Data pengembalian

Showing rows 0 - 2 (3 total, Query took 0.0026 seconds.)

`SELECT * FROM `pengembalian``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

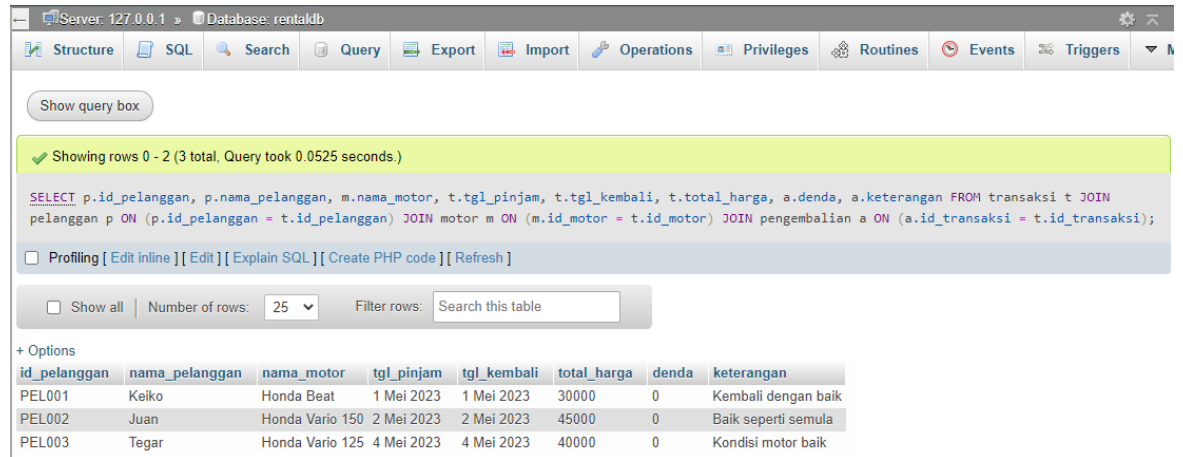
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id_pengembalian	id_transaksi	tgl_pinjam	tgl_kembali	denda	keterangan
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	KMB001	TR001	1 Mei 2023	1 Mei 2023	0	Kembali dengan baik
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	KMB002	TR002	2 Mei 2023	2 Mei 2023	0	Baik seperti semula
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	KMB003	TR003	4 Mei 2023	4 Mei 2023	0	Kondisi motor baik

Gambar 2.6.15 database pengembalian

f. Data dari table join



The screenshot shows a MySQL database interface with the following components:

- Server:** 127.0.0.1, **Database:** rentaladb
- Navigation Bar:** Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers.
- Query Box:** Contains the SQL query:


```
SELECT p.id_pelanggan, p.nama_pelanggan, m.nama_motor, t.tgl_pinjam, t.tgl_kembali, t.total_harga, a.denda, a.keterangan FROM transaksi t JOIN pelanggan p ON (p.id_pelanggan = t.id_pelanggan) JOIN motor m ON (m.id_motor = t.id_motor) JOIN pengembalian a ON (a.id_transaksi = t.id_transaksi);
```
- Results:** Showing rows 0 - 2 (3 total). Query took 0.0525 seconds.
- Options:**
 - Show all
 - Number of rows: 25
 - Filter rows: Search this table
- Table Data:**

id_pelanggan	nama_pelanggan	nama_motor	tgl_pinjam	tgl_kembali	total_harga	denda	keterangan
PEL001	Keiko	Honda Beat	1 Mei 2023	1 Mei 2023	30000	0	Kembali dengan baik
PEL002	Juan	Honda Vario 150	2 Mei 2023	2 Mei 2023	45000	0	Baik seperti semula
PEL003	Tegar	Honda Vario 125	4 Mei 2023	4 Mei 2023	40000	0	Kondisi motor baik

Gambar 2.6.16 dsatabase join dari beberapa tabel

DAFTAR PUSTAKA

Laporan ini diambil dari gambaran rental motor di website berikut:

<http://rental-mobil.org>