

Nama : Bayu Pangestu
NIM : A11.2018.11031
Kelompok : A11.4423

Software yang digunakan :

- Netbeans IDE 12.4
- JDK 15.0.2
- Driver mysql connector
- Apache
- Mysql

Package Controller

- cSewaBuku

```
public class cSewaBuku {
    TabelSewaBuku tabelSewaBuku = new TabelSewaBuku(list_sewaBuku);
    _tampilan.GetTabelData().setModel(tabelSewaBuku);

    public cSewaBuku(mainGUI _tampil) {
        this._tampilan = _tampil;
        iDAOSewaBuku = new DAOSewaBuku();
    }
}
```

Controller dari sewa buku yang digunakan untuk menampilkan isi table.

- IsiTable

```
public void IsiTable() {
    list_sewaBuku = iDAOSewaBuku.GetAll();
    TabelSewaBuku tabelSewaBuku = new TabelSewaBuku(list_sewaBuku);
    _tampilan.GetTabelData().setModel(tabelSewaBuku);
}
}
```

Berfungsi untuk menampilkan isi table dari Database

KONEKSI

- Koneksi

```

package Koneksi;

import com.mysql.cj.jdbc.MysqlDataSource;
import java.sql.Connection;
import java.sql.SQLException;

/**
 *
 * @author Ida Bagus Putu Wira Kusuma
 */
public class Koneksi {
    static Connection con;
    static String db_name = "sewabuku";
    static String db_user = "root";
    static String db_pass = "";

    public static Connection GetConnection() {
        if (con == null) {
            MysqlDataSource data = new MysqlDataSource();
            data.setDatabaseName(db_name);
            data.setUser(db_user);
            data.setPassword(db_pass);

            try {
                con = data.getConnection();
                System.out.println("Database connect");
            } catch (SQLException e) {
                System.out.println("Database tidak connect");
            }

            return con;
        }
    }
}

```

Berfungsi untuk koneksi ke database

MODEL

- TableSewaBuku

```
package Hodel:
```

```
import java.util.List:  
import javax.swing.table.AbstractTableModel:
```

```
• @author Ida Bagus Putu Wira Kusuma
```

```
public class TabelSewaBoho extends AbstractTableModel  
    List<MTable> list sewaBubu:
```

```
    public TabelSewaBoho(List<MTable> list_sewaBuku) {  
        this.list sewaBulu = list sewaBuIcu:
```

```
    @Override  
    public int getRowCount() {  
        return this.list_sewaBubu.size():
```

```
    @Override  
    public int getColumnCount() {  
        return 7:
```

```
    @Override  
    public Object getValueAt(int rowlodex, int columolodex) {  
        switch(columolodex) {  
            case 0:  
                return list sewaBubu.get frowlodex).id:
```

```

        case 1:
            return list_sewaBuku.get(rowIndex).judul;
        case 2:
            return list_sewaBuku.get(rowIndex).tanggal_pinjam;
        case 3:
            return list_sewaBuku.get(rowIndex).tanggal_harus_kembali;
        case 4:
            return list_sewaBuku.get(rowIndex).tanggal_kembali;
        case 5:
            return list_sewaBuku.get(rowIndex).denda;
        case 6:
            return list_sewaBuku.get(rowIndex).biaya_sewa;
        default :
            return null;
    }
}

@Override
public String getColumnName(int column){
    switch(column){
        case 0:
            return "ID";
        case 1:
            return "Judul";
        case 2:
            return "Tanggal Pinjam";
        case 3:
            return "Tanggal Harus Kembali";
        case 4:
            return "Tanggal Kembali";
        case 5:
            return "Denda";
        case 6:
            return "Biaya Sewa";
        default :
            return null;
    }
}
}

```

Berfungsi untuk membuat tampilan table dari table sql ke UI

- mTable

```

package Model;

/**
 *
 * @author Ida Bagus Putu Wira Kusuma
 */
public class mTable {
    public int id;
    public String judul;
    public String tanggal_pinjam;
    public String tanggal_kembali;
    public String tanggal_harus_kembali;
    public int denda;
    public int biaya_sewa;
}

```

Digunakan untuk membuat variabel model table yang nantinya akan dijadikan list dan menampung data dari sql

IDAO

- IDAOSewaBuku

```

package IDAO;

import Model.mTable;
import java.util.List;

/**
 *
 * @author Ida Bagus Putu Wira Kusuma
 */
public interface IDAOSewaBuku {
    public List<mTable> GetAll();
}

```

Berfungsi sebagai interface dari DAOSewaBuku

DAO

- DAOSewaBuku

- GetAll

```

@Override
public List<mTable> GetAll() {
    SimpleDateFormat format_waktu = new SimpleDateFormat("dd MMMM yyyy");
    List<mTable> list_sewaBuku = null ;
}

```

Berfungsi untuk mendapatkan isi dari database sewabuku.sql

- InsertNewData

```
//sql query
String query_InsertData = "INSERT INTO sewabuku (judul,tanggal_pinjam,tanggal_harus_kembali, biaya_sewa) VALUES (?,?,?,?)";
public void InserNewData(String _judul,String _tanggal_pinjam, String _tanggal_harus_kembali){
    PreparedStatement statement = null;
    int biaya_sewa = 5000;
    try{
        statement = con.prepareStatement(query_InsertData);
        statement.setString(1, _judul);
        statement.setString(2, _tanggal_pinjam);
        statement.setString(3, _tanggal_harus_kembali);
        statement.setInt(4, biaya_sewa);
        statement.execute();
    }
}
```

berfungsi untuk membuat data baru pinjaman buku baru, mengambil dari tanggal saat diinputkan

- UpdateData

```
String query_updateData = "UPDATE sewabuku SET judul = ? WHERE id = ?";
public void UpdateData(int _id,String _judul){
    PreparedStatement statement = null;
```

Berfungsi untuk mengupdate data pada data yang sudah diinput

- DeleteData

```
String query_deleteData = "DELETE FROM sewabuku where id = ?";
public void DeleteData(int id){
    PreparedStatement statement = null;
```

Berfungsi untuk menghapus data yang sudah diinput sebelumnya.

- GetTanggalHarusKembali

```

public String GetTanggalHarusKembali(int id){
    String tanggal_harusKembali = "";
    try{
        Statement st = con.createStatement();
        String query = "SELECT tanggal_harus_kembali FROM sewabuku WHERE id = "+id;
        System.out.println(query);
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            tanggal_harusKembali = rs.getString("tanggal_harus_kembali");
        }
    }
    catch(SQLException e){
        System.out.println("Error");
    }
    return tanggal_harusKembali;
}

```

Berfungsi untuk menentukan tanggal buku harus dikembalikan, dihitung dari ketika buku dipinjam.

○ KembalikanBuku

```

public void KembalikanBuku(int id) throws ParseException{
    long denda = 2000;
    String _harusKembali = GetTanggalHarusKembali(id);
    Date harusKembali =new SimpleDateFormat("yyyy-MM-dddd").parse(_harusKembali);
    Date kembali = new Date();

    if(kembali.after(harusKembali)){

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
        String str1 = formatter.format(harusKembali);
        String str2 = formatter.format(kembali);

        LocalDate d1 = LocalDate.parse(str1, DateTimeFormatter.ISO_LOCAL_DATE);
        LocalDate d2 = LocalDate.parse(str2, DateTimeFormatter.ISO_LOCAL_DATE);

        Duration diff = Duration.between(d1.atStartOfDay(), d2.atStartOfDay());

        long diffDays = diff.toDays();
        denda= denda * diffDays;
    }else{
        denda = 0;
    }
}

```

Berfungsi untuk meenghitung denda jika buku terlambat dikembalikan, dihitung dari selisih tanggal harus kembali dan tanggal kembali.

○ UpdateTanggalKembaliBuku

```

String query_updateKembaliBuku = "UPDATE sewabuku SET tanggal_kembali = ?,denda = ? WHERE id = ?";
public void UpdateTanggalKembaliBuku(long _denda,int id){
    PreparedStatement statement = null;

    try{
        statement = con.prepareStatement(query_updateKembaliBuku);

        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
        Date dateNow = new Date();

        statement.setString(1,formatter.format(dateNow));
        statement.setLong(2, _denda);
        statement.setInt(3, id);
        statement.execute();
    }
    catch(SQLException e){
        System.out.println("Gagal");
    }
    finally
    {
        try
        {
            statement.close();
        } catch (SQLException ex)
        {
            System.out.println("Gagal");
        }
    }
}
}

```

Berfungsi untuk membuat tanggal kembali jika buku telah dikembalikan.

GUI

- maingui.java

- jTable1MouseClicked

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    int row = jTable1.rowAtPoint(evt.getPoint());
    id_selected = (int)jTable1.getValueAt(row,0);
    tf_judul.setText(jTable1.getValueAt(row, 1).toString());
    tgl_kembali_selected = jTable1.getValueAt(row, 4).toString();
    int denda = (int)jTable1.getValueAt(row, 5);
    int sewa = (int)jTable1.getValueAt(row, 6);
    jLabel6.setText("Rp " +(denda+sewa));
}

```

Berfungsi untuk mendapatkan id row, judul, dan tgl_kembali pada table yg dipilih

- btn_kembaliActionPerformed


```

private void btn_kembaliActionPerformed(java.awt.event.ActionEvent evt) {
    if(id_selected != 0){
        //kalau masih kosong
        if(tgl_kembali_selected.equals("")){
            DAOsewaBuku dao = new DAOsewaBuku();
            try {
                dao.KembalikanBuku(id_selected);
                JOptionPane.showMessageDialog(this, "Buku berhasil dikembalikan!!");
                CreateTable();
                id_selected = 0;
                tf_judul.setText("");
                jLabel6.setText("Rp");
            } catch (ParseException ex) {
                Logger.getLogger(maingui.class.getName()).log(Level.SEVERE, null, ex);
            }
        }else{
            JOptionPane.showMessageDialog(this, "Buku sudah dikembalikan","warning",JOptionPane.WARNING_MESSAGE);
        }
    }
    else{
        JOptionPane.showMessageDialog(this, "Tidak ada data yang terpilih","error",JOptionPane.ERROR_MESSAGE);
    }
}

```

Berfungsi untuk mengembalikan buku dari id row yang dipilih pada table. Jika buku berhasil dikembalikan maka akan muncul pesan / notif “Buku berhasil dikembalikan”. Jika tidak ada row yg dipilih akan muncul notif “Buku sudah dikembalikan”. Jika buku sudah dikembalikan sebelumnya maka akan muncul notif “Buku sudah dikembalikan”. Melakukan cek dari tanggal dikembalikannya buku atau belum pada tabel.

- initComponents

```

public maingui() {
    initComponents();
    waktu();
    CreateTable();
}

```

Merupakan build in komponen dari UI yang telah dibuat.

- btn_simpanActionPerformed

```

private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    String _judulBuku = tf_judul.getText().trim();
    if(_judulBuku.equals(""))
    {
        JOptionPane.showMessageDialog(this, "Input judul tidak boleh kosong!!", "fail", JOptionPane.ERROR_MESSAGE);
        return;
    }
    DAOsewaBuku DAO = new DAOsewaBuku();
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dddd");

    String _tgl_Pinjam = formatter.format(tanggal);

    //bikin tanggal kembali
    Calendar c = Calendar.getInstance();
    c.setTime(tanggal);
    c.add(Calendar.DATE, 7);
    String _tgl_Kembali = formatter.format(c.getTime());
    DAO.InsertNewData(_judulBuku, _tgl_Pinjam, _tgl_Kembali);
    JOptionPane.showMessageDialog(this, "Data berhasil dibuat!!");
    CreateTable();
    id_selected = 0;
    tf_judul.setText("");
    jLabel6.setText("Rp");
}

```

Berfungsi untuk menyimpan judul buku yang diinputkan. Jika tidak ada inputan maka akan muncul notif “Input judul tidak boleh kosong” dan tidak akan diproses. Tanggal harus kembali dihitung 7 hari setelah input data buku. Jika data berhasil diproses dan selesai dibuat maka akan muncul notif “Data berhasil dibuat”.

○ btn_editActionPerformed

```

private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    if(id_selected != 0){
        //cek apakah sudah dikembalikan
        if(tgl_kembali_selected.equals("")){
            //cek apakah judul kosong
            String _judulBuku = tf_judul.getText();
            if(_judulBuku.equals("")){
                JOptionPane.showMessageDialog(this, "Input judul tidak boleh kosong!!", "error", JOptionPane.ERROR_MESSAGE);
                return;
            }
            else{
                DAOsewaBuku dao = new DAOsewaBuku();
                dao.UpdateData(id_selected, _judulBuku);
                JOptionPane.showMessageDialog(this, "Data berhasil diupdate!!");
                CreateTable();
                id_selected = 0;
                tf_judul.setText("");
                jLabel6.setText("Rp");
            }
        }
        else{
            JOptionPane.showMessageDialog(this, "Buku sudah dikembalikan, data tidak dapat diedit", "warning", JOptionPane.WARNING_MESSAGE);
        }
    }
    else{
        JOptionPane.showMessageDialog(this, "Tidak ada data yang terpilih", "error", JOptionPane.ERROR_MESSAGE);
    }
}

```

Berfungsi untuk mengedit judul buku dari row yg sudah dipilih pada table. Jika inputan kosong maka tidak akan diproses dan muncul notif “Input judul tidak boleh kosong”. Jika buku telah berhasil diedit pada table maka akan muncul notif “Data berhasil diupdate”. Jika buku sudah dikembalikan maka data tidak dapat diedit serta akan muncul notif “Buku sudah dikembalikan, data tidak dapat diedit”. Pengecekannya dari sudah ada atau tidak adanya tanggal dikembalikan pada table.

Jika tidak ada row yang dipilih maka otomatis akan muncul pesan “Tidak ada data yang terpilih”.

- btn_deleteActionPerformed

```
private void btn_deleteActionPerformed(java.awt.event.ActionEvent evt) {  
    if(id_selected != 0){  
        DAOsewaBuku dao = new DAOsewaBuku();  
        dao.DeleteData(id_selected);  
        JOptionPane.showMessageDialog(this, "Data berhasil terhapus!!");  
        CreateTable();  
        id_selected = 0;  
        tf_judul.setText("");  
        jLabel6.setText("Rp");  
    }else{  
        JOptionPane.showMessageDialog(this, "Tidak ada data yang terpilih", "error", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Berguna untuk menghapus data pada row yang sudah dipilih. Jika tidak ada row yang dipilih maka secara otomatis muncul notif “Tidak ada data yang terpilih.”

- line 418 - 425

merupakan deklarasi variable

```
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
private javax.swing.JLabel lbl_harga;  
private javax.swing.JLabel lbl_jam;  
private javax.swing.JLabel lbl_tanggal;  
private javax.swing.JTextField tf_judul;  
// End of variables declaration
```

- cSewabuku : kontroller dari sewa buku yang berfungsi untuk menampilkan table
- simppldateformat formmat tanggal : berfungsi untuk membuat format tanggal sekarang
- siimpledatformmatt formmat jam : berfungsi untuk format jam sekarang.
- date tanggal : berguna untuk mengambil data tanggal sekarang/
- int id_selected : berfungsi untuk mengambil id dari row pada table yang dipilih
- str tgl_kembali_selecetted : berfungsi mengecek apakah buku sudah dikembalikan atau belum
- int biaya : menentukan biaya sewa pada buku yang dipinjam

- fungsi waktu

```

//variable
cSewaBuku c_sewaBuku;
SimpleDateFormat format_tanggal;
SimpleDateFormat format_jam;
Date tanggal;
int id_selected;
String tgl_kembali_selected;
int biaya_sewa = 5000;

    void waktu() {
        format_tanggal = new SimpleDateFormat("dd MMMM yyyy");
        format_jam = new SimpleDateFormat("HH:mm");
        tanggal = new Date();
        lbl_tanggal.setText(format_tanggal.format(tanggal));
        lbl_jam.setText(format_jam.format(tanggal));
    }

```

Fungsi untuk mengambil data tanggal sekarang dan jam sekarang yang akan ditampilkan pada UI.

- fungsi GetTabelData

```

public JTable GetTabelData(){
    return jTable1;
}

```

Berfungsi untuk mengambil data variable pada tabel

- fungsi CreateTable

```

//fungsi menampilkan tabel data pinjaman buku
public void CreateTable(){
    c_sewaBuku = new cSewaBuku(this);
    c_sewaBuku.IsiTable();
}
}

```

Berfungsi untuk membuat table pada UI