

File koneksi.java

```
public class koneksi {
    public static Connection con;
    public static Connection koneksiDB(){
        if(con==null){
            try {

                con =
DriverManager.getConnection("jdbc:mysql://localhost/projectpbo", "root",
null);

                System.out.println("Connection successfully created !");
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(null,"gagal koneksi");
            }
        }
        return con;
    }
    public static void main(String args[]) {
        koneksi db = new koneksi();
        db.koneksiDB();

    }

}
```

A. LOGIN

File MenuLogin.java

```
private void masuk(){
    try {

        String sql = "SELECT * FROM user where nik =
'" + nik.getText() + "' and password = '" + password.getText() + "'";
        Connection con = koneksi.koneksiDB();
        Statement stat = con.createStatement();
        ResultSet rs = stat.executeQuery(sql);

        if(rs.next()){

            if(nik.getText().equals(rs.getString("nik")) && password
.getText().equals(rs.getString("password"))){
                MenuUtama fa = new MenuUtama();
                fa.show();
                this.dispose();
            }
        }else{
            JOptionPane.showMessageDialog(null, "Maaf password atau
username anda salah");
            nik.setText("");
            password.setText("");
        }

    }

    catch (SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}
```

```

    }

}

```

```

private void loginMouseClicked(java.awt.event.MouseEvent evt) {
    masuk();
}

```

B. REGISTRASI

File memasukkan data Model.java

```

public void tambah_user (String nama, String nik, String Tempat_lahir,
                          String Tanggal_lahir, String alamat, String
                          jns_kelamin, String Password) throws
                          SQLException {
    String query = "insert into user values
('"+nama+"','"+nik+"','"+Tempat_lahir+"','"+Tanggal_lahir+"','"+a
alamat+
                        "','"+jns_kelamin+"','"+Password+"')";
    System.out.print(query);
    Connection con = koneksi.koneksiDB();
    Statement st = con.createStatement();
    st.execute(query);
    JOptionPane.showMessageDialog(null, "Data Disimpan");
}

```

File memasukkan data MenuRegistrasi.java

```

String nama, tempat_lahir, tanggal_lahir, alamat, jenis_kelamin, nik,
password;

private void simpan() {
    if (lakilaki.isSelected()){
        jenis_kelamin = "Laki-Laki";
    } else {
        jenis_kelamin = "Perempuan";
    }

    nama = name.getText();
    nik = niik.getText();
    tempat_lahir = tmpt_lahir.getText();
    tanggal_lahir = tgl_lahir1.getText();
    alamat = alamatt.getText();
    password = passwordd.getText();

    Model a = new Model();
    try {
        a.tambah_user(nama, nik, tempat_lahir, tanggal_lahir,
alamat, jenis_kelamin, password);
        MenuAwal p = new MenuAwal();
        p.setVisible(true);
        this.dispose();
    } catch (SQLException ex) {

```

```

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

    }

private void SimpanMouseClicked(java.awt.event.MouseEvent evt) {
    simpan();
}

```

C. INSERT DATA PERJALANAN

Memasukkan Data Perjalanan Model.java

```

public void tambah_data (int id, String tanggal, String lokasi, String
        Suhu) throws SQLException {
    String query = "insert into catatan values
('"+id+"', '"+tanggal+"', '"+lokasi+"', '"+Suhu+"')";
    System.out.print(query);
    Connection con = koneksi.koneksiDB();
    Statement st = con.createStatement();
    st.execute(query);
    JOptionPane.showMessageDialog(null, "Data Disimpan");
}

```

Memasukkan Data Perjalanan IsiDataPerjalanan.java

```

int id;
String tanggall, lokasii, suhuu;
private void simpan () {
    tanggall = tanggal.getText();
    lokasii = lokasi.getText();
    suhuu = suhu.getText();
    Model a = new Model();
    try {
        a.tambah_data(id, tanggall, lokasii, suhuu);
        MenuCatatan p = new MenuCatatan();
        p.setVisible(true);
        this.dispose();
    } catch (SQLException ex) {

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

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    simpan();
}

```

D. MENAMPILKAN DATA PADA MENU UTAMA

File Menampilkan Data pada MenuUtama.java

```
public void getData(){
    DefaultTableModel tabel1= new DefaultTableModel();
    tabel1.addColumn("Tanggal");
    tabel1.addColumn("Lokasi");
    tabel1.addColumn("Suhu");
    try {

        Connection conn = koneksi.koneksiDB();
        Statement stm = conn.createStatement();
        ResultSet sql = stm.executeQuery("select tanggal, lokasi, suhu
        from catatan limit 5");
        while(sql.next()){
            tabel1.addRow(new Object[]{
                sql.getString(1),
                sql.getString(2),
                sql.getString(3),
            });
        }
        tabel.setModel(tabel1);
    }catch (SQLException e) {
    }
}
```

E. MENAMPILKAN DATA PADA MENU CATATAN

File Menampilkan Data Pada MenuCatatan.java

```
public void getData(){
    DefaultTableModel tabel1= new DefaultTableModel();
    tabel1.addColumn("id");
    tabel1.addColumn("Tanggal");
    tabel1.addColumn("Lokasi");
    tabel1.addColumn("Suhu");
    try {

        Connection conn = koneksi.koneksiDB();
        Statement stm = conn.createStatement();
        ResultSet sql = stm.executeQuery("select * from catatan");
        while(sql.next()){
            tabel1.addRow(new Object[]{
                sql.getString(1),
                sql.getString(2),
                sql.getString(3),
                sql.getString(4),
            });
        }
        tabel.setModel(tabel1);
    }catch (SQLException e) {
    }
}
```

F. MENGHAPUS DATA

Mengambil data table dengan id MenuCatatan.java

```
static String id;
public void getTabel(){
    try {
        int row=tabel.getSelectedRow();
        id=(tabel.getModel().getValueAt(row, 0).toString());
        System.out.print(id);

    }catch(Exception e){

    }

}
```

File Menghapus Data Pada Model.java

```
public void hapus_data (String id) {
    int ok=JOptionPane.showConfirmDialog(null,"Yakin Ingin
    Menghapus???", " ",JOptionPane.YES_NO_OPTION);
    if (ok==0)
    try
    {
        String query = "DELETE FROM catatan WHERE id ='"+id+"'";
        System.out.print(query);
        Connection con = koneksi.koneksiDB();
        Statement st1 = con.createStatement();

        st1.execute(query);

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

    }
    catch (SQLException ex)
    {
        JOptionPane.showMessageDialog(null, ex);
    }
}
```

Menghapus Data MenuCatatan.java

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    Model u = new Model();
    u.hapus_data(id);
    MenuCatatan a = new MenuCatatan();
    a.setVisible(true);
    this.dispose();          // TODO add your handling code here:
}

private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
    getTabel();
}
```

G. MENGUPDATE DATA

Update Data Model.java

```
public void update(String id, String tanggal, String lokasi, String
                    Suhu) {
    try
    {
        String query = "update catatan set tanggal = '"+tanggal+"',
            lokasi = '"+lokasi+"', suhu = '"+Suhu+"' where id = '"+id+"'";
        System.out.print(query);
        Connection con = koneksi.koneksiDB();
        Statement st = con.createStatement();
        st.execute(query);

        JOptionPane.showMessageDialog(null, "Data Update");
    }
    catch (SQLException ex)
    {
        JOptionPane.showMessageDialog(null, ex);
    }
}
```

MenuCatatan.java

```
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {
    new MenuUpdate().setVisible(true);
    this.dispose();
}
```

Getdata pada MenuUpdate.java

```
public void getData() {
    try {
        Connection conn = koneksi.koneksiDB();
        Statement stm = conn.createStatement();
        ResultSet sql= stm.executeQuery("select * from catatan where
            id = '"+MenuCatatan.id+"'");
        if (sql.next()){
            id.setText(sql.getString(1));
            tanggal.setText(sql.getString(2));
            lokasi.setText(sql.getString(3));
            suhu.setText(sql.getString(4));
        }

    }catch(Exception e){

    }
}
```

```

    }

    private void UpdateMouseClicked(java.awt.event.MouseEvent evt) {
        String id1, tanggal1, lokasi1, suhu1;

        id1 = id.getText();
        tanggal1 = tanggal.getText();
        lokasi1 = lokasi.getText();
        suhu1 = suhu.getText();

        Model m = new Model();
        m.update(id1, tanggal1, lokasi1, suhu1);
        MenuCatatan d = new MenuCatatan();
        d.setVisible(true);
        this.dispose();
    }

    private void CancellMouseClicked(java.awt.event.MouseEvent evt) {
        new MenuCatatan().setVisible(true);
        this.dispose();
    }
}

```

H. LAINNYA

Link untuk ke MenuCatatan Pada Menu Bar

```

    private void CatatanMouseClicked(java.awt.event.MouseEvent evt) {
        MenuCatatan p = new MenuCatatan();
        p.setVisible(true);
        this.dispose();
    }
}

```

Memberikan Nama Pada Nama Lengkap dan NIK

```

public void getNama(){
    try {

        Connection conn = koneksi.koneksiDB();
        Statement stm = conn.createStatement();
        ResultSet sql = stm.executeQuery("select * from user");
        while(sql.next()){
            nama.setText(sql.getString("nama"));
            nik.setText(sql.getString("nik"));
        }

    }catch (SQLException e) {
    }
}

```

Mengurutkan Data MenuCatatan.java

```
public void urutan() {
    DefaultTableModel tabel1= new DefaultTableModel();
    tabel1.addColumn("id");
    tabel1.addColumn("Tanggal");
    tabel1.addColumn("Lokasi");
    tabel1.addColumn("Suhu");
    try {

        Connection conn = koneksi.koneksiDB();
        Statement stm = conn.createStatement();
        ResultSet sql = stm.executeQuery("select * from catatan  order
by tanggal asc");
        while(sql.next()){
            tabel1.addRow(new Object[]{
                sql.getString(1),
                sql.getString(2),
                sql.getString(3),
                sql.getString(4),
            });
        }
        tabel.setModel(tabel1);
    }catch (SQLException e) {
    }

}

private void CariMouseClicked(java.awt.event.MouseEvent evt) {
    if(urutan.getSelectedItemAt() == "Tanggal") {
        urutan();
    }
}
```