

Table of Contents

DAO Pattern	3
Contoh Kasus	4
Diagram UML dan Database	4
Use Case Diagram Perpustakaan	4
Class Diagram Entity Perpustakaan.....	5
Class Diagram DAO.....	5
Diagram Database	6
Target dari perkuliahan ini adalah	7
Memulai Project SistemPerpustakaan.....	7
Membuat project baru di Netbeans	7
Membuat Tampilan	9
Mendesain Form Utama.....	9
Mendesain Form Data Mahasiswa	16
Menghubungkan FrmMain dengan FrmDataMahasiswa	23
Mengatur Main kelas	24
Membuat Entity	25
Class Diagram Entity Perpustakaan	25
Kode Program	26
• Kelas Anggota	26
• Kelas Mahasiswa	27
• Kelas PetugasPerpus	27
• Kelas Koleksi	28
• Kelas ItemKoleksi	29
• Kelas Transaksi	30
• Kelas TransaksiPeminjaman	31
• Kelas TransaksiPengembalian	31
• Kelas Karyawan	31
• Kelas Denda	32
• Kelas Buku	33

• ItemTransaksi	34
Membuat DAO	35
Diagram Database.....	35
Class Diagram DAO Perpustakaan	36
Kode Program	37
• Kelas Database	37
• Kelas AbstractDao	39
• Kelas DaoMahasiswa.....	40
Untuk DAO Selanjutnya silahkan dikembangkan sendiri.....	42
• Kelas DaoPetugasPerpus.....	42
• Kelas DaoPengembalian.....	42
• Kelas DaoPeminjaman	42
• Kelas DaoItemKoleksi	42
• Kelas DaoPetugasPerpus.....	42
• Kelas DaoKaryawan.....	42

DAO Pattern

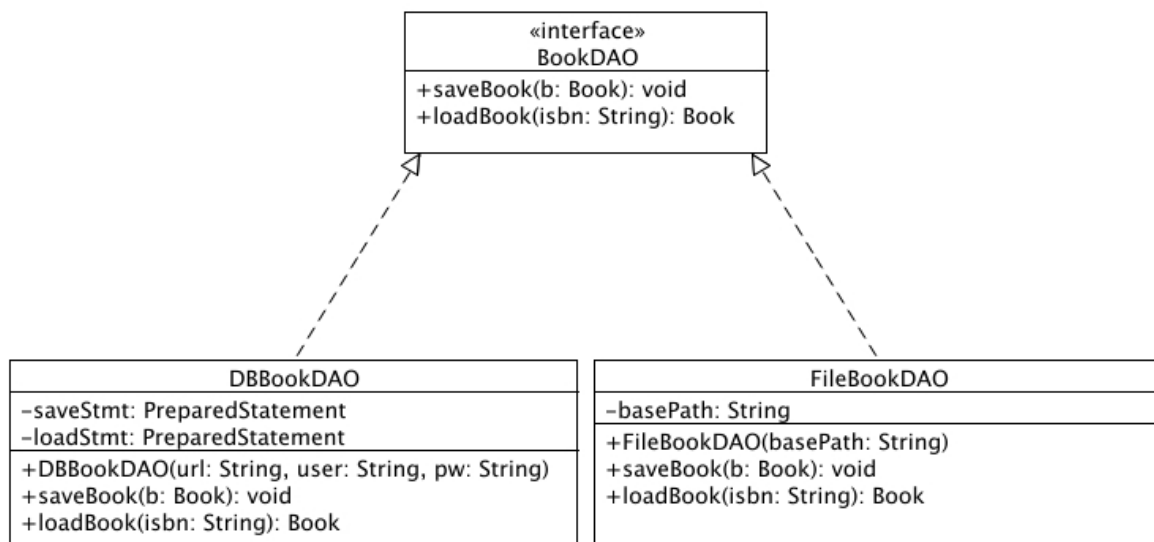
Salah satu aspek dari lapisan bisnis adalah lapisan akses data yang menghubungkan layanan dengan database. Mengakses data bervariasi tergantung pada sumber data. Akses ke data persisten sangat bervariasi tergantung pada jenis penyimpanan (database, flat file, file xml, dan sebagainya) dan bahkan berbeda dari pelaksanaannya (misalnya berbeda-dialek SQL). Tujuannya adalah untuk merangkul semua akses ke data dan menyediakan interface. Ini disebut pola DAO.

Dari sudut pandang aplikasi, tidak ada bedanya ketika mengakses sebuah database relasional atau mem-parsing file xml (menggunakan DAO). DAO ini biasanya mampu menciptakan sebuah instance dari objek data ("untuk membaca data") dan juga untuk menyimpan data sebagai sumber data.

DAO Digunakan ketika

- Membutuhkan akses data lebih dari satu kali dan ada kemungkinan pertukaran data antar aplikasi
- Menginginkan pemisahan antarmuka dengan mekanisme pengaksesan data
- Pada tim pengembang yang besar, memungkinkan untuk melakukan pemisahan pekerjaan

Contoh kelas diagram untuk DAO



Contoh Kasus

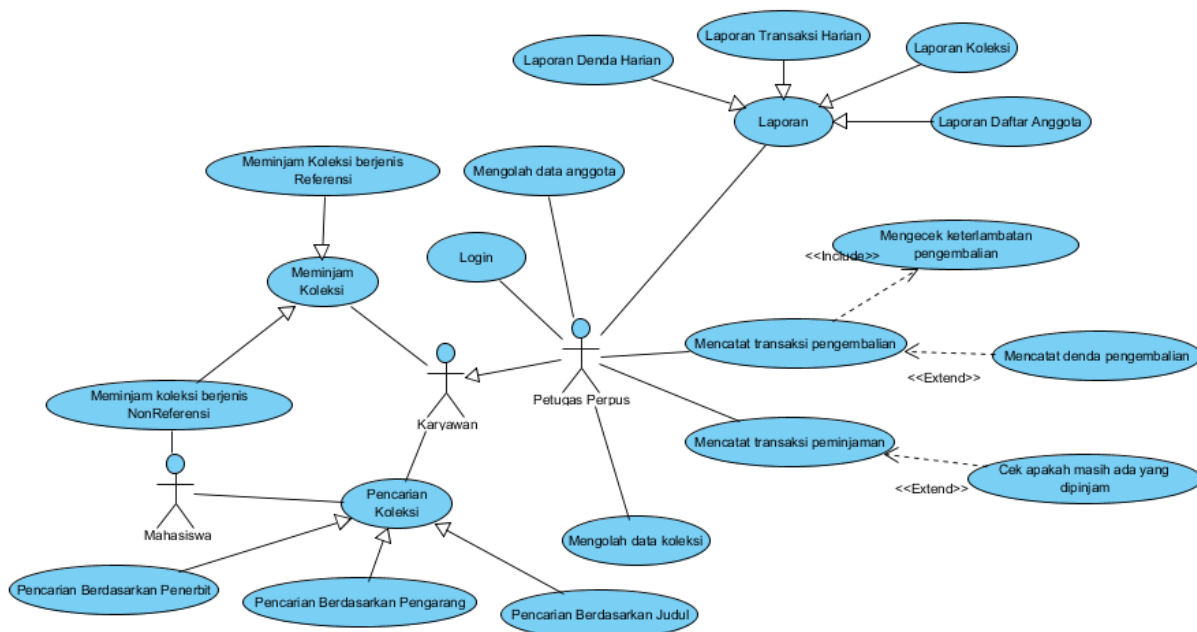
Perpustakaan STMIK AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian.

Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda.

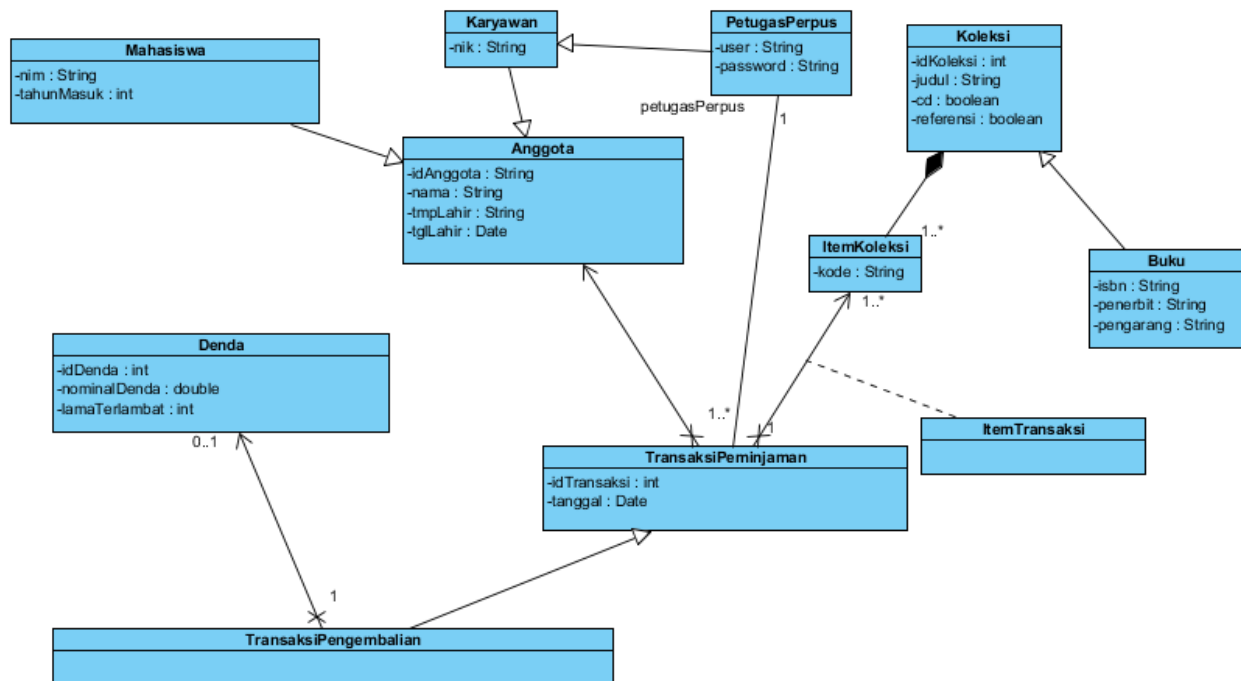
Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencari koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda. Berikut ini adalah beberapa diagram UML yang akan menjadi acuan kita dalam pengembangan Sistem Perpustakaan ini

Diagram UML dan Database

Use Case Diagram Perpustakaan



Class Diagram Entity Perpustakaan



Class Diagram DAO

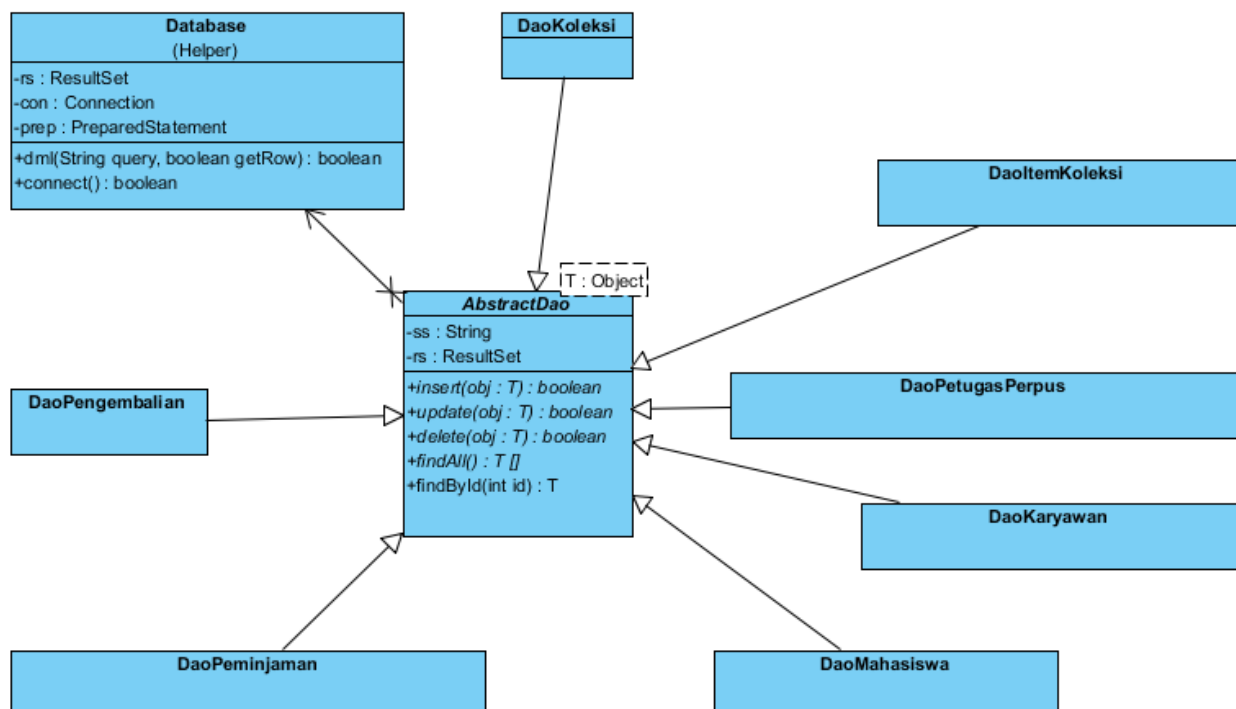
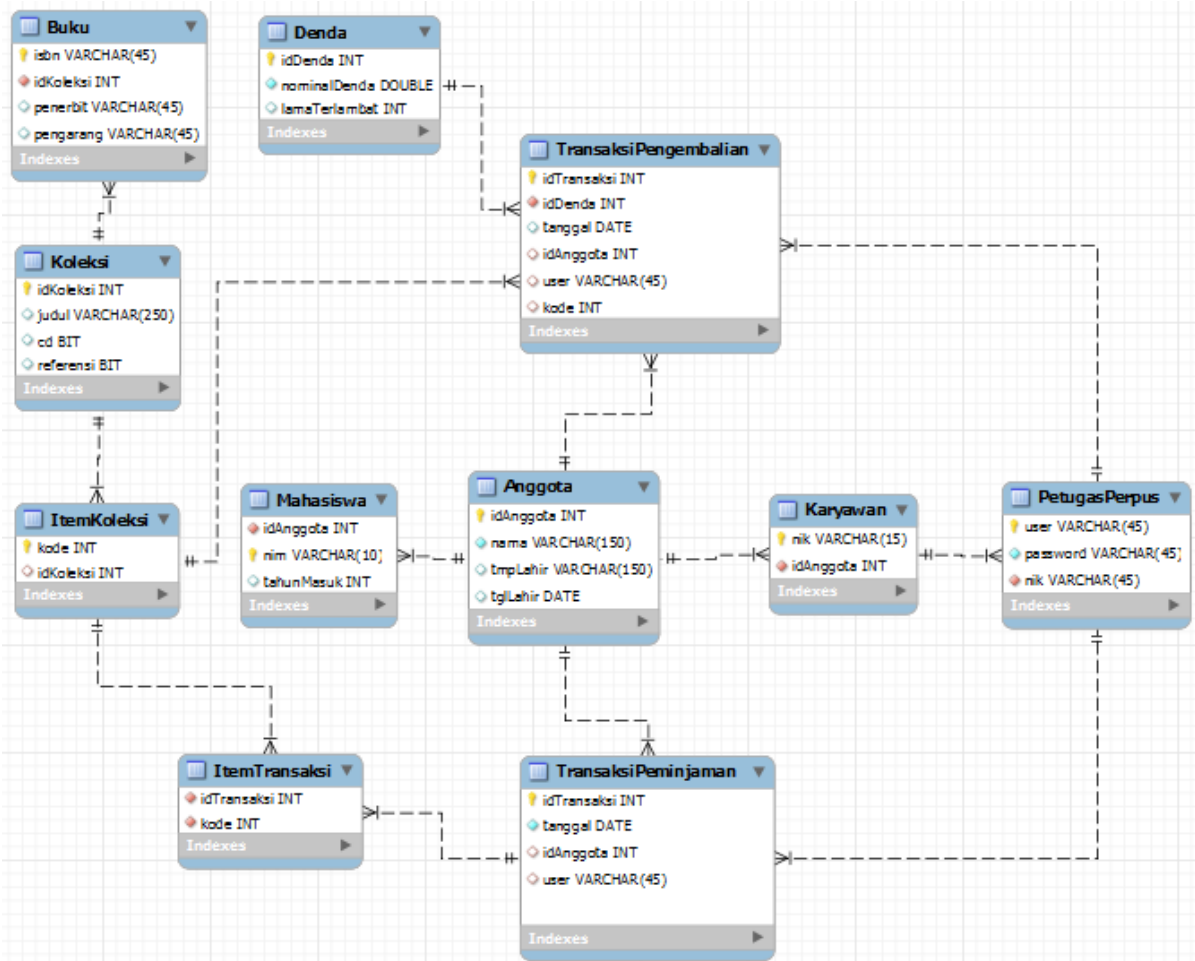


Diagram Database



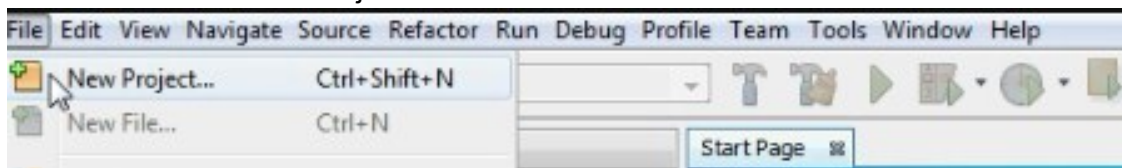
Target dari perkuliahan ini adalah

- Memahami konsep DAO
- Membuat koneksi database ke mysql
- Merancang tampilan
- Menyelesaikan minimal modul pengolahan data mahasiswa
- Menyelesaikan minimal modul peminjaman
- Menyelesaikan minimal report peminjaman harian

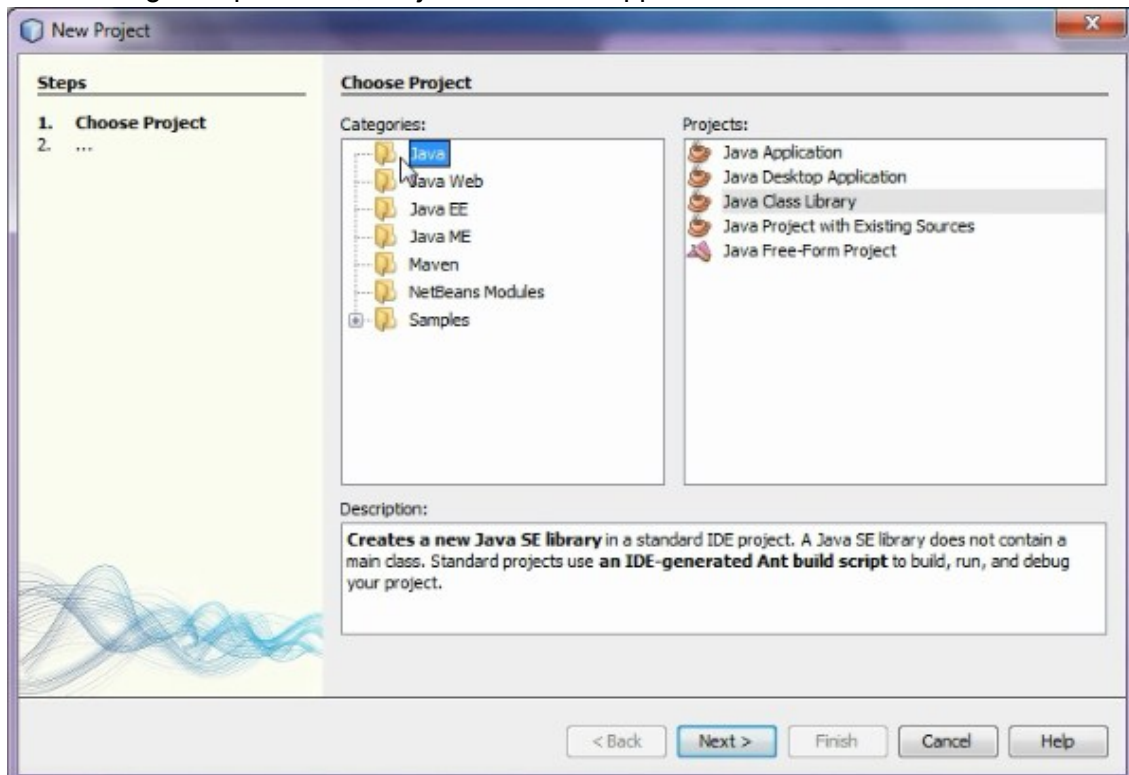
Memulai Project Sistem Perpustakaan

Membuat project baru di Netbeans

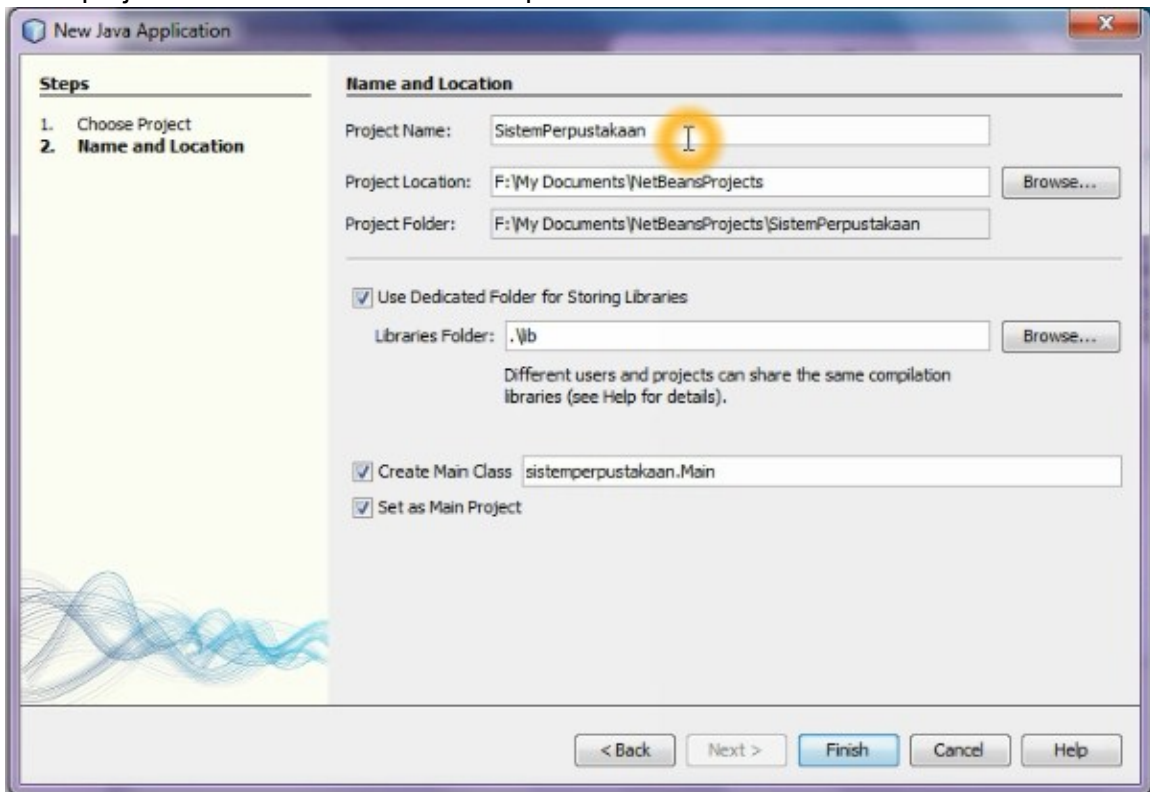
1. Jalankan Netbeans Anda
2. Pilih menu File → New Project



3. Pada Categories pilih Java, Project Pilih Java Application → Next



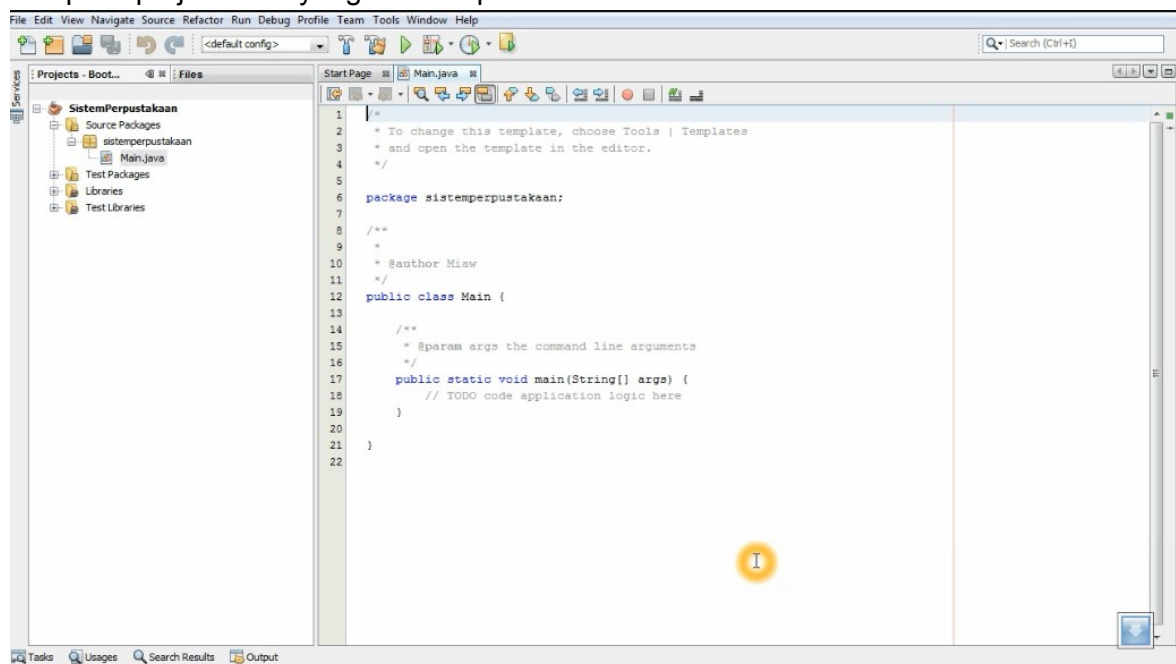
4. Pada project name bernama SistemPerpustakaan



Project Location diisi dengan folder anda di lab.

Beri tanda check pada Use Dedicated Folder for Storing Libraries, Create Main Class dan Set as Main Project. Pilih Finish

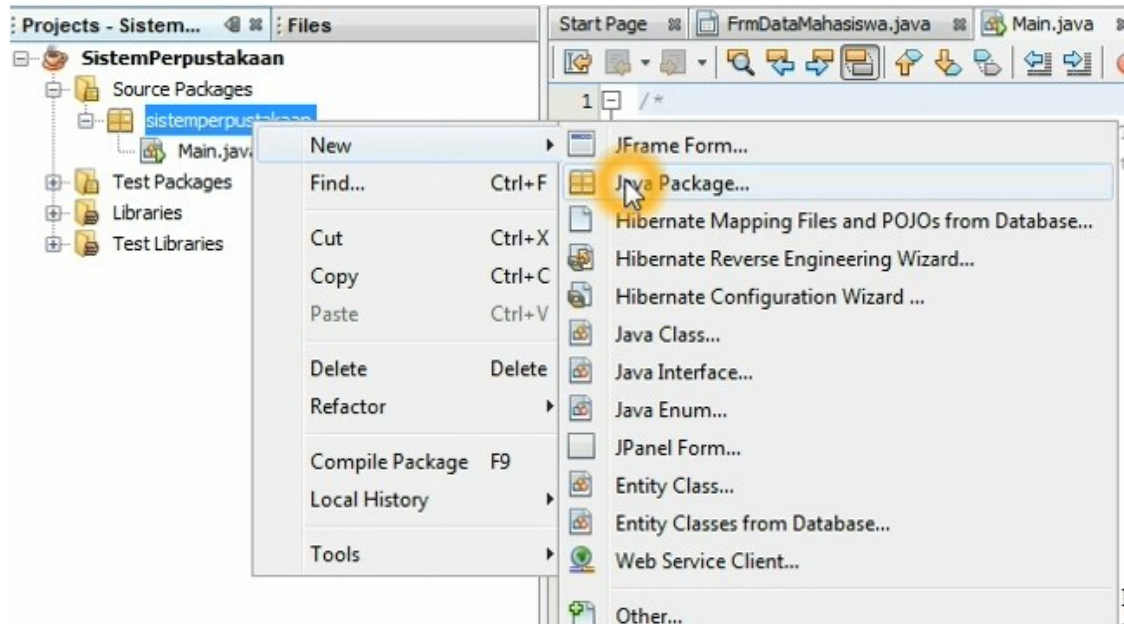
5. Tampilan project baru yang anda dapatkan



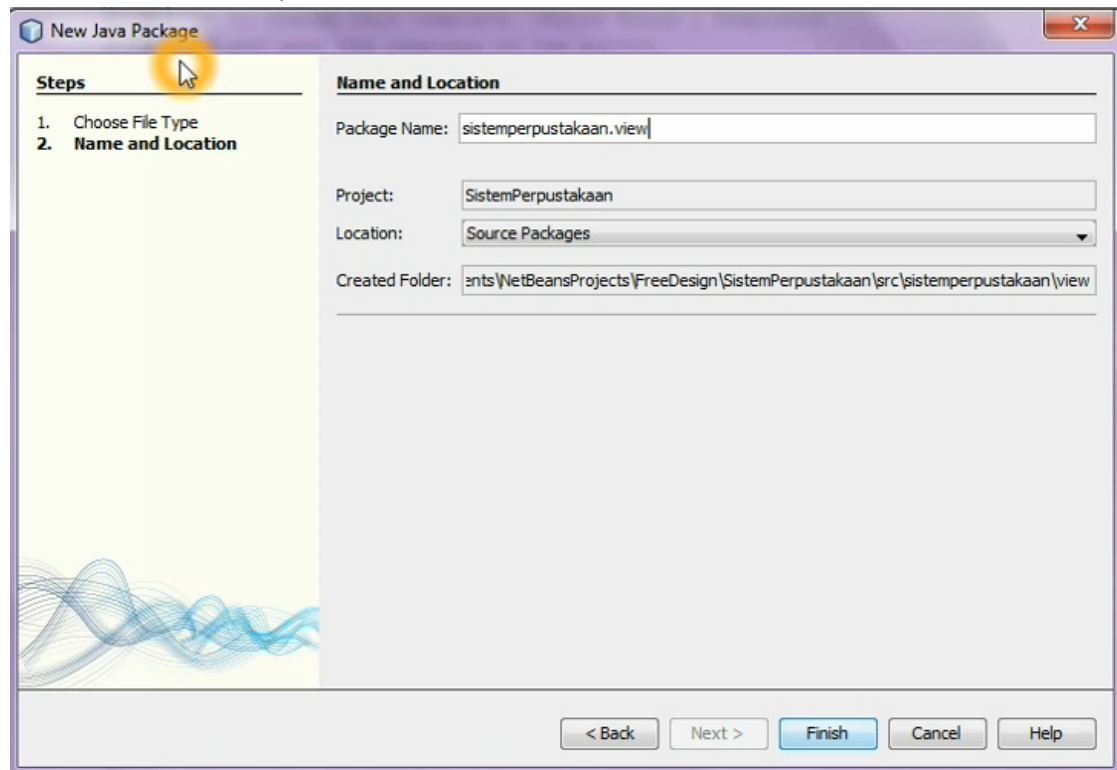
Membuat Tampilan

Mendesain Form Utama

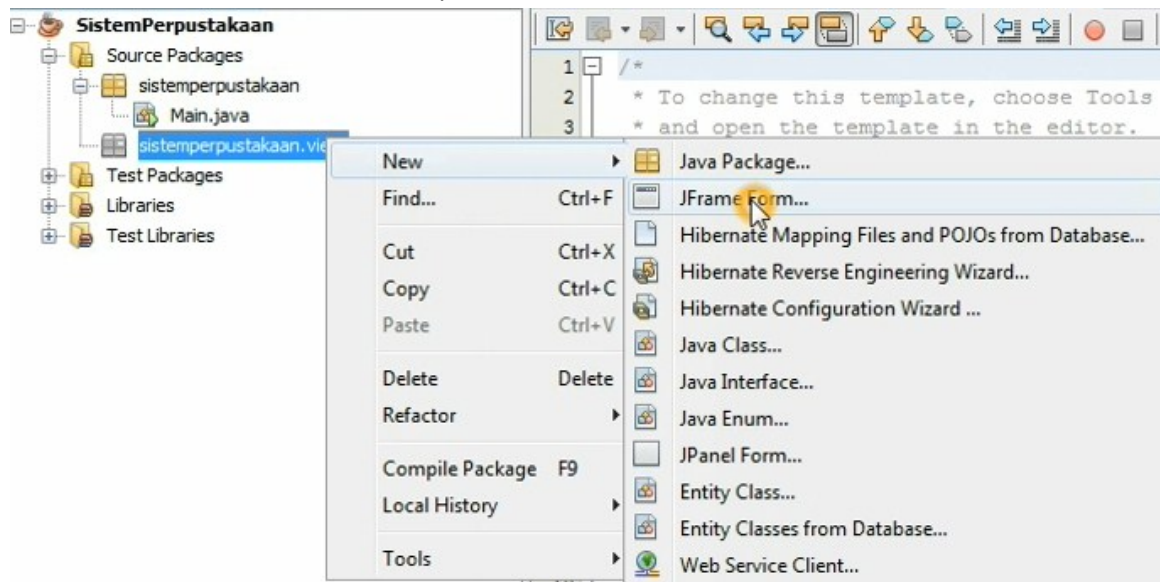
1. Buat 1 buah paket baru didalam paket sistemperpustakaan



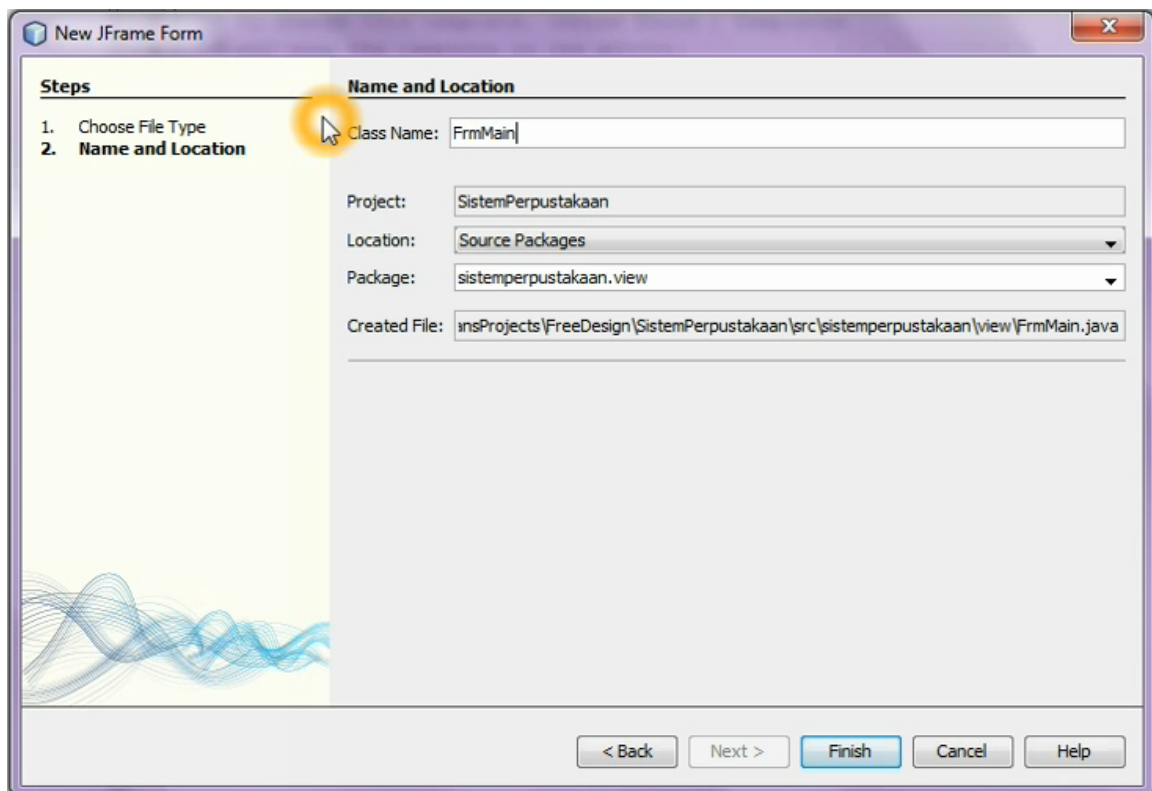
2. Beri nama **view** untuk paket baru → Finish



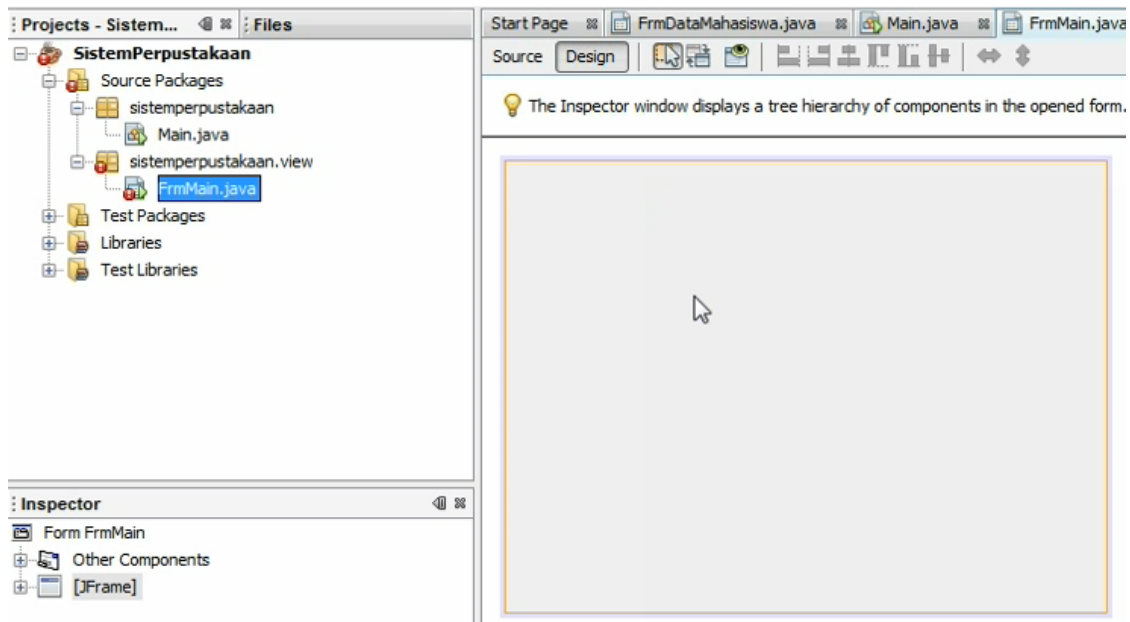
3. Tambahkan 1 buah JFrame didalam paket view



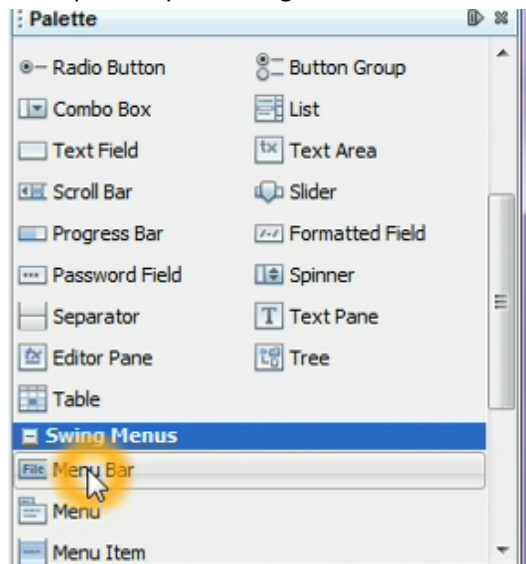
4. Beri nama FrmMain → Finish



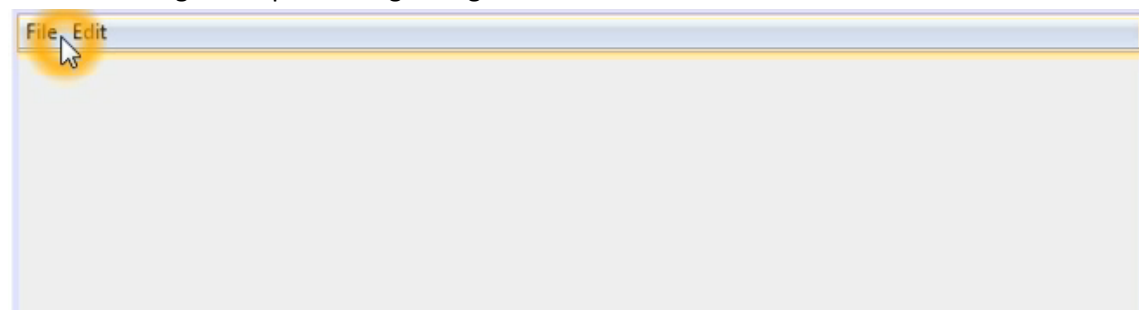
5. Maka anda akan mendapatkan tampilan sebagai berikut



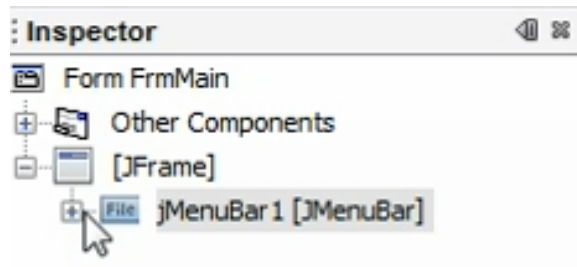
6. Pada palette pilih Swing Menu → Menu Bar



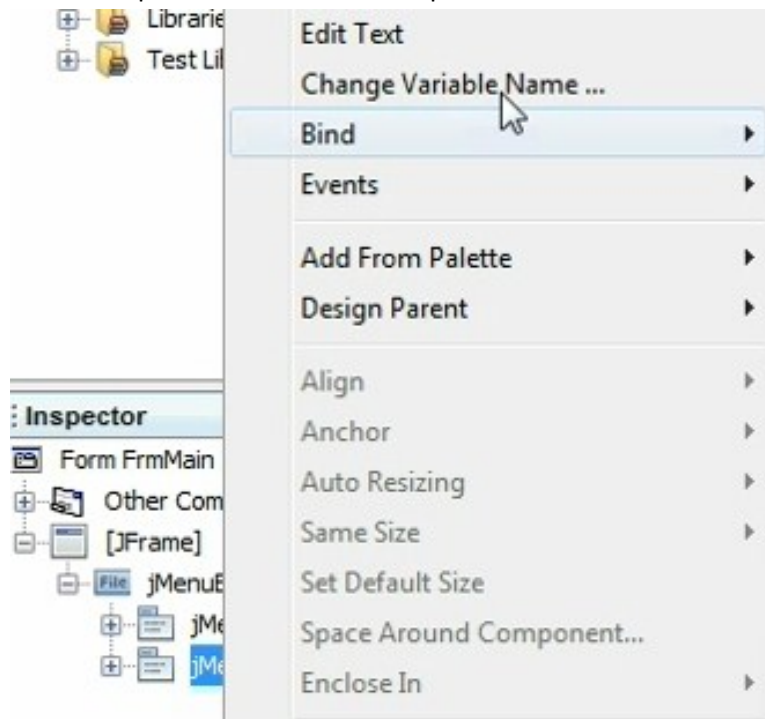
7. Kemudian drag di tempat kosong di bagian atas



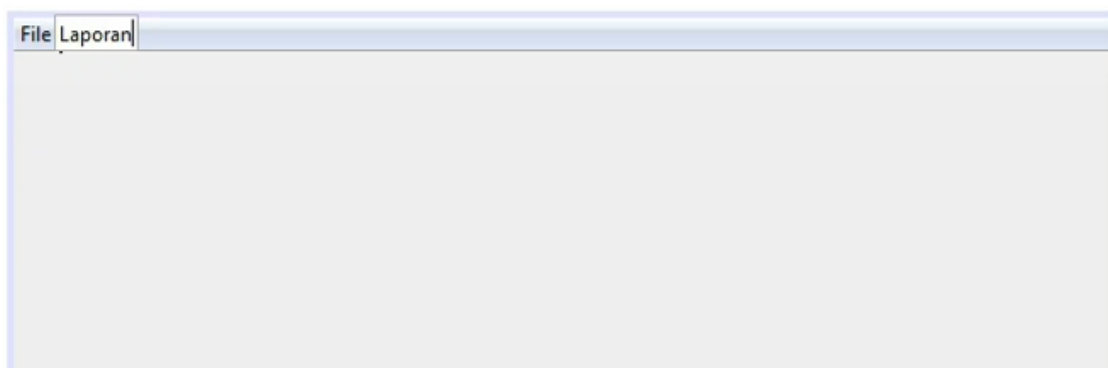
8. Cari window inspector di kiri bawah



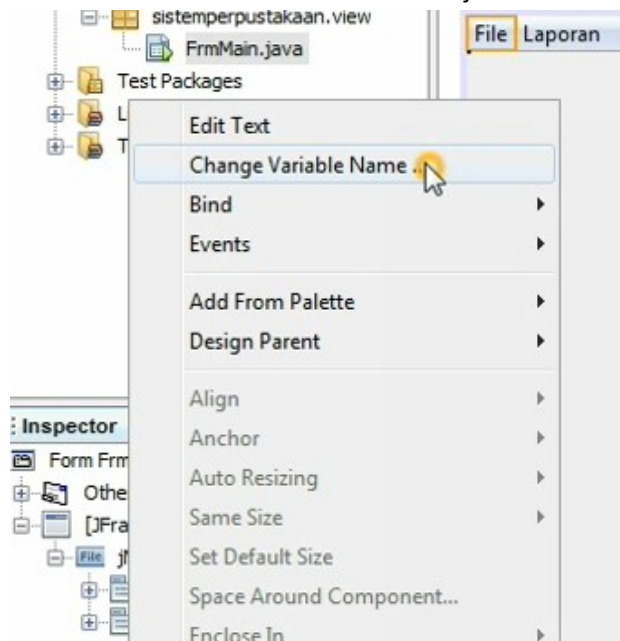
9. Silahkan tanda + pada jMenuBar1 diklik, maka anda akan mendapatkan JMenu1 dan JMenu2
10. Klik kanan pada JMenu2 kemudian pilih Edit Text



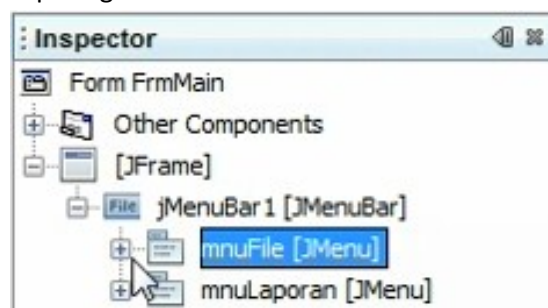
11. Ubah dari tulisan Edit menjadi Laporan



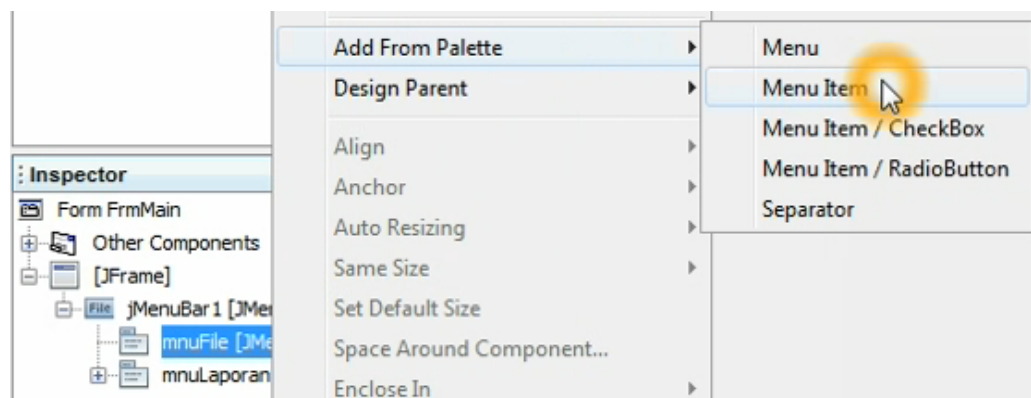
12. Ubah variabel name dari JMenu1 menjadi mnuFile kemudian pilih OK



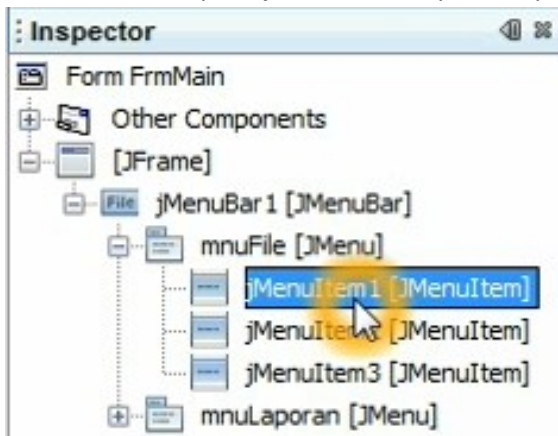
13. Untuk JMenu2 ubah variabel namanya menjadi mnuLaporan, sehingga di inspector tampak seperti gambar berikut



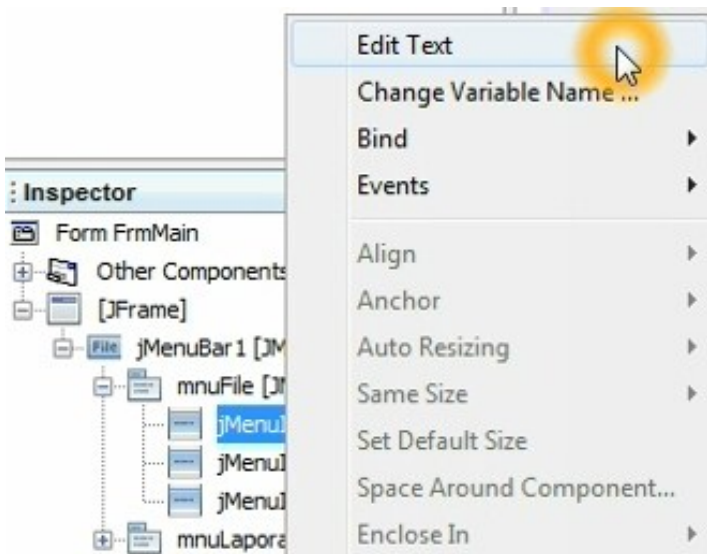
14. Tambahkan MenuItem di mnuFile dengan cara klik kanan mnuFile → Add From Palette → Menu Item



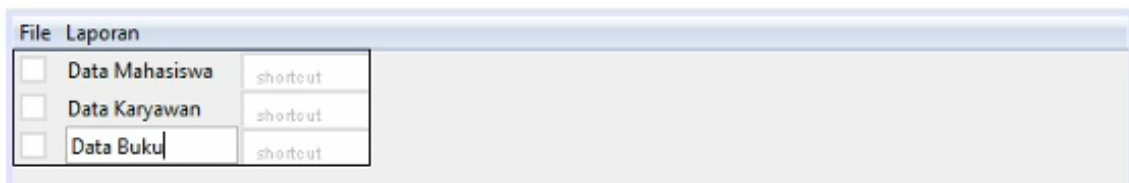
15. Tambahkan sampai 3 JMenuItem, seperti inspector berikut ini



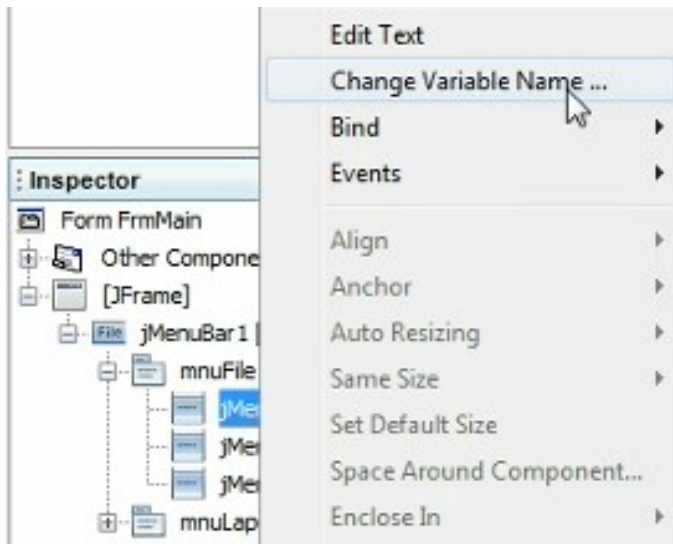
16. Ubah text dari jMenuItem1 menjadi Data Mahasiswa dengan cara klik kanan jMenuItem1 → Edit Text



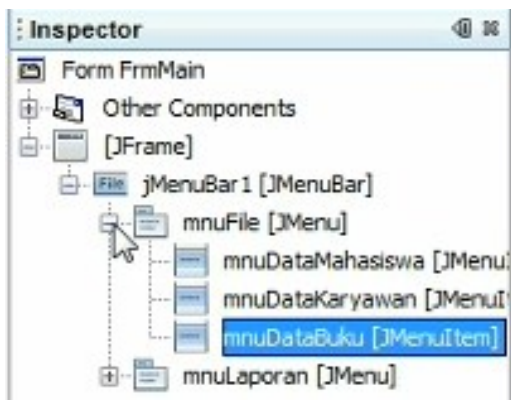
17. Untuk jMenuItem2 ubah textnya menjadi Data Karyawan, jMenuItem2 ubah textnya menjadi Data Buku



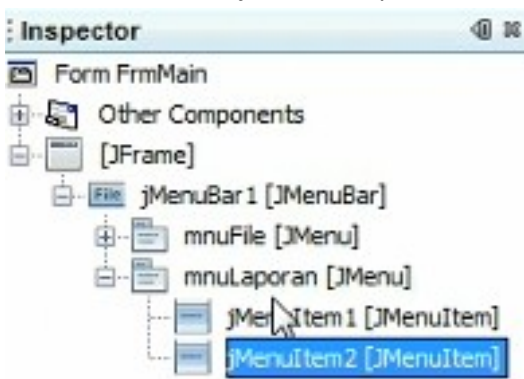
18. Ubah variabel name dari jMenuItem1 menjadi mnuDataMahasiswa dengan cara klik kanan jMenuItem1 → Change Variable Name



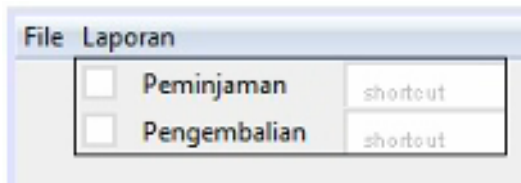
19. Ubah juga variabel name jMenuItem2 menjadi mnuDataKaryawan dan jMenuItem3 menjadi mnuDataBuku



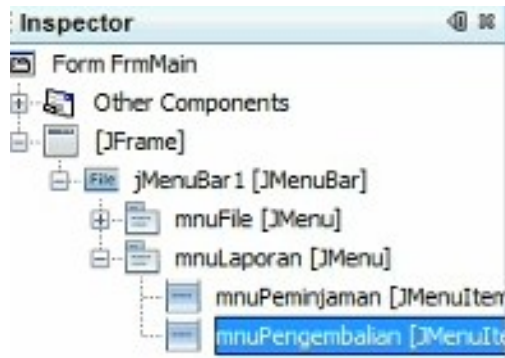
20. Tambahkan 2 buah jMenuItem pada mnuLaporan, yaitu jMenuItem1 dan jMenuItem2



21. Ubah text menjadi Peminjaman dan Pengembalian

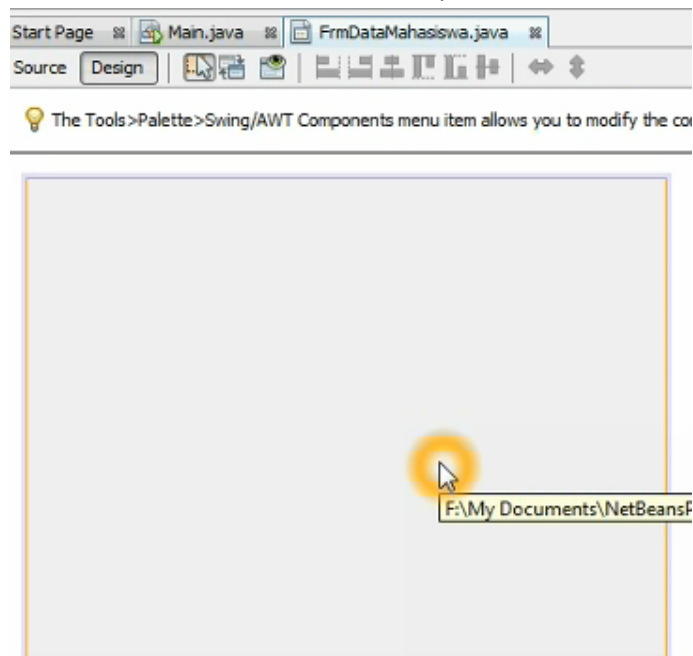


22. Ubah variabel name menjadi seperti berikut

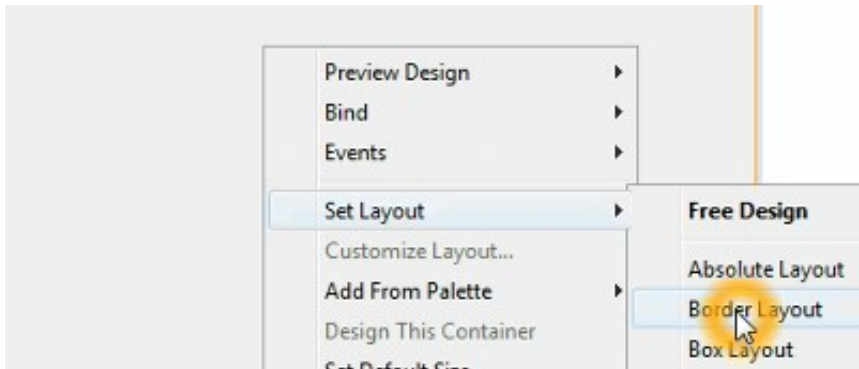


Mendesain Form Data Mahasiswa

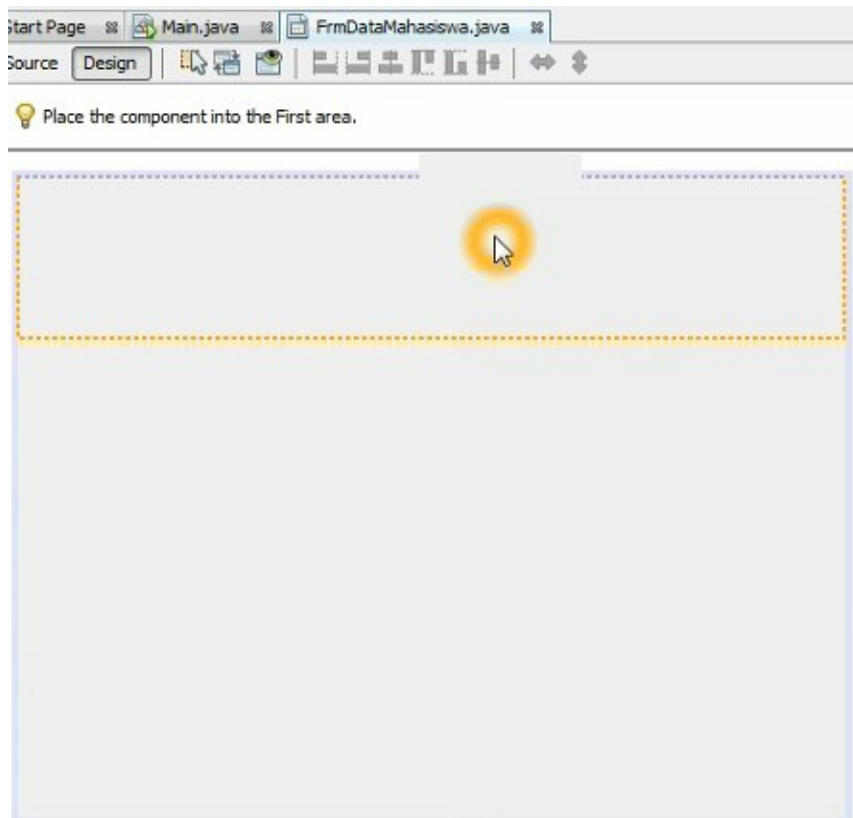
1. Tambahkan 1 buah JFrame didalam paket view, beri nama FrmDataMahasiswa



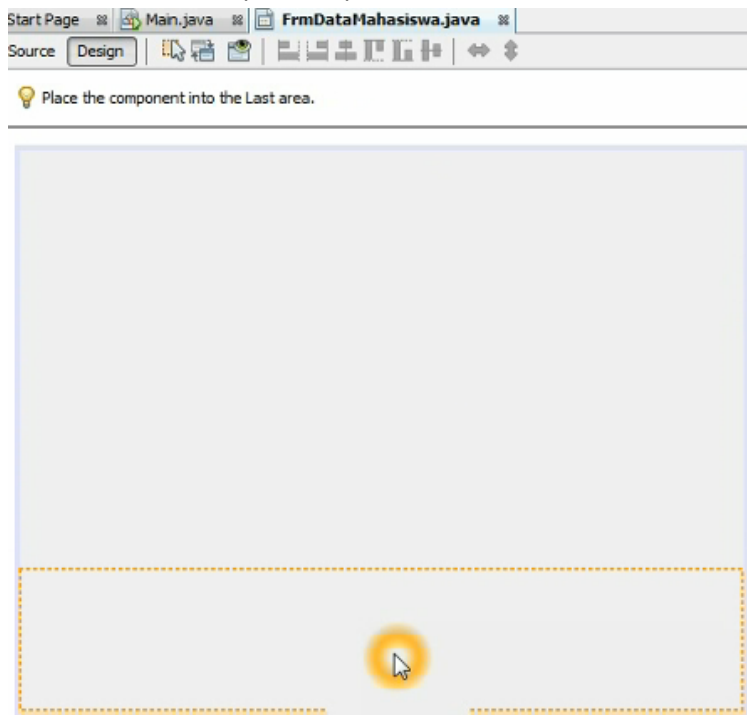
2. Klik kanan pada FrmDataMahasiswa, kemudian pilih menu Layout → BorderLayout



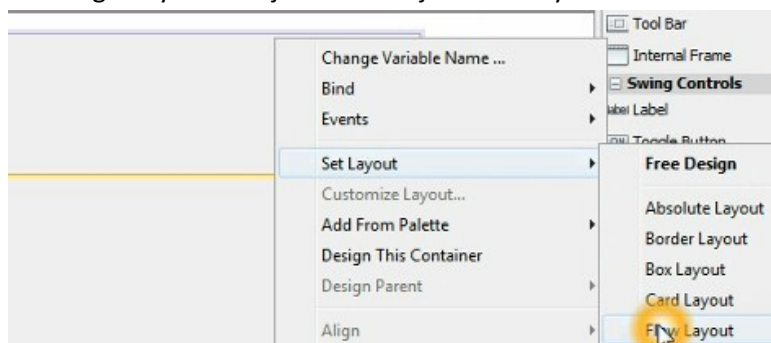
3. Tambahkan 1 buah panel di Posisi atas



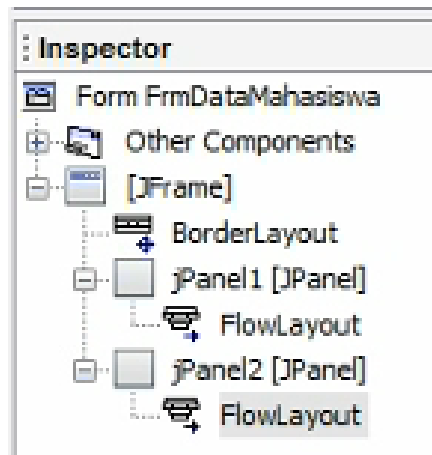
4. Tambahkan 1 buah panel di posisi bawah



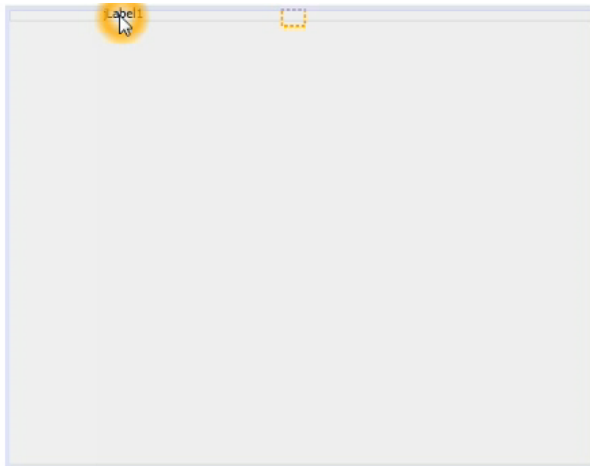
5. Ubah agar layout dari jPanel1 menjadi FlowLayout



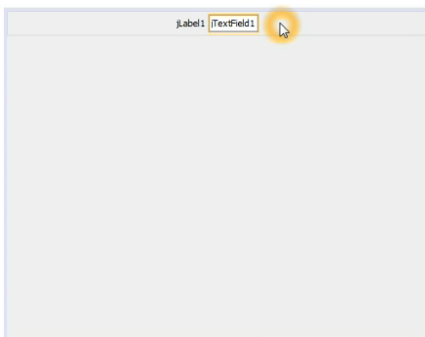
6. Ubah juga agar layout dari jPanel2 menjadi FlowLayout, sama seperti langkah 5



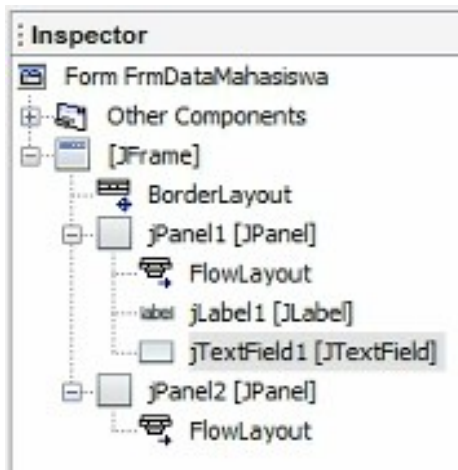
7. Tambahkan label dan jTextField ke jPanel1 seperti gambar berikut ini



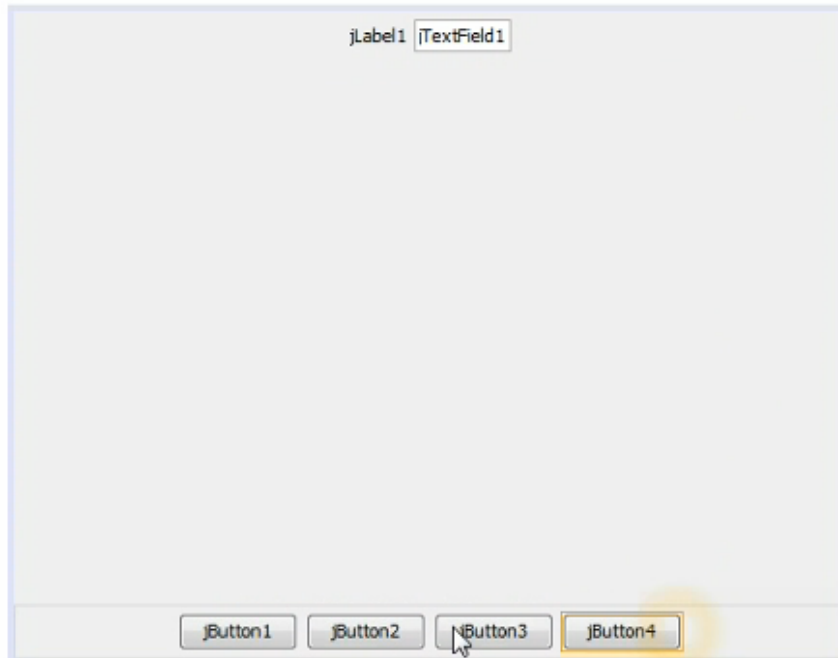
Sehingga form seperti berikut



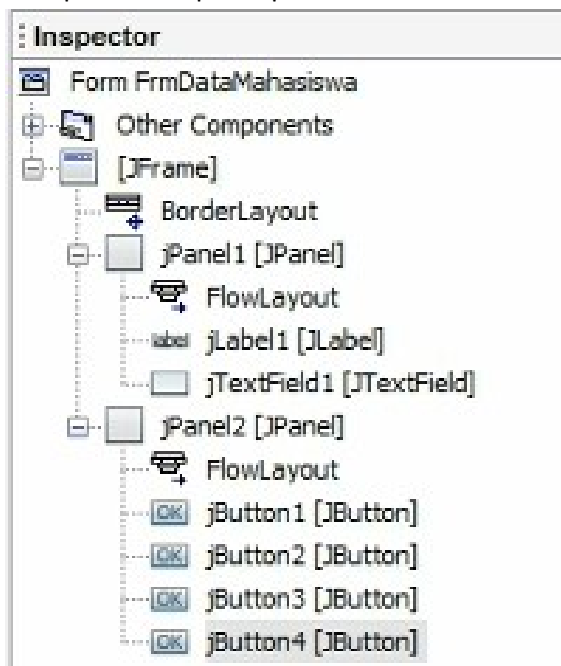
Sehingga di inspector tampak seperti gambar berikut



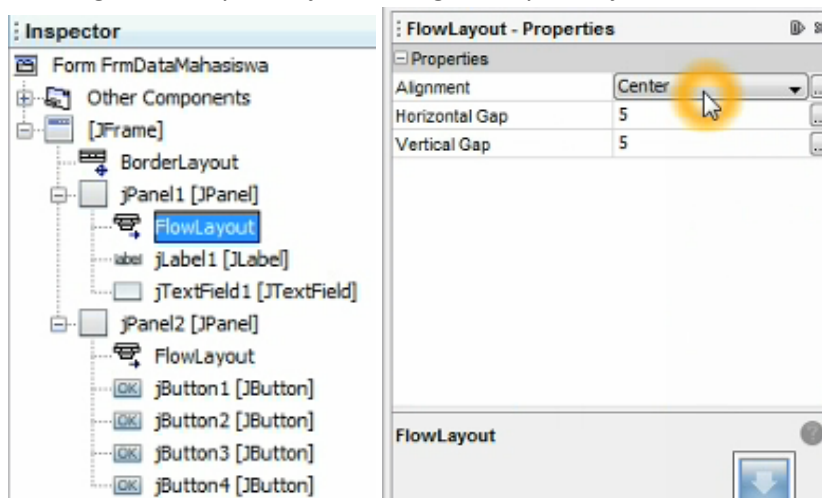
8. Tambahkan 4 buah button di jPanel2 seperti gambar berikut



Diinspector tampak seperti berikut

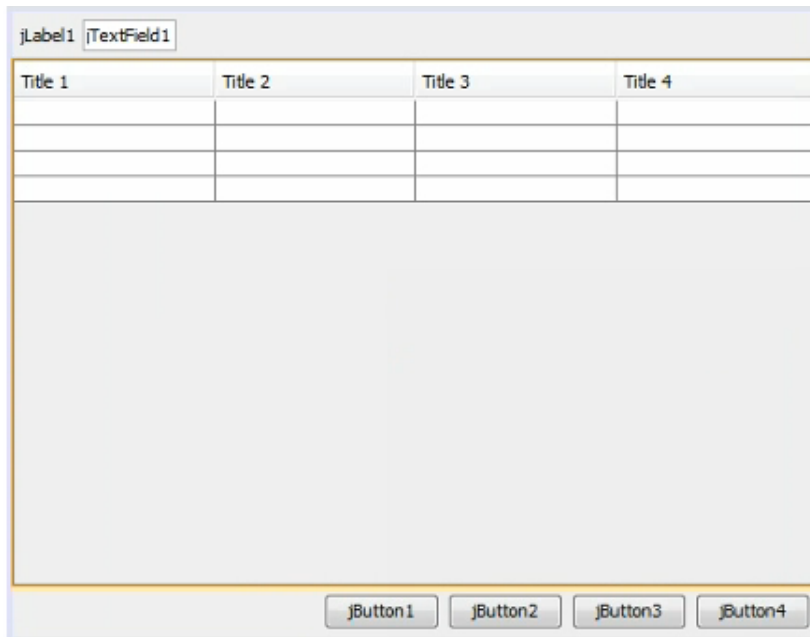


9. Ubah agar FlowLayout di jPanel1 alignmentnya menjadi **Left**



10. Lakukan hal yang sama dengan jPanel2 dengan alignment **Right**

11. Tambahkan 1 buah JTable di tengah



12. Atur agar seperti gambar berikut

The screenshot shows a Java Swing window with a search bar at the top labeled "Cari berdasarkan nama" with a text input field. Below the search bar is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table has three empty rows below the header.

13. Atur agar tombol seperti berikut

The screenshot shows a Java Swing window with four buttons at the bottom: "Tambah", "Ubah", "Hapus", and "Tutup". The "Tutup" button is highlighted with a yellow border.

14. Atur agar jTextField widthnya menjadi 100,20

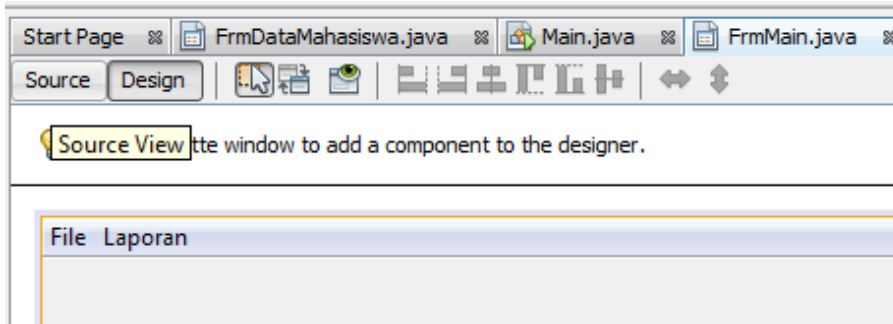
The screenshot shows the NetBeans IDE Properties window for a component named "jTextField1 [JTextField]". The "Properties" tab is selected. The "preferredSize" property is set to "[100, 20]". Other visible properties include "margin", "maximumSize", "minimumSize", "name", "navigationFilter", "nextFocusableComponent", "opaque", "paintingForPrint", and "requestFocusEnabled".

15. Ubah variabel name dari jTextField1 menjadi txtCari, kemudian button urut dari kiri ke kanan menjadi btnTambah, btnUbah, btnHapus, btnTutup dan tabel menjadi tbMhs sehingga di inspector menjadi seperti berikut

The screenshot shows the final Java Swing window design. It features a search bar at the top labeled "Cari berdasarkan nama" with a text input field. Below the search bar is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table has three empty rows below the header. At the bottom of the window are four buttons: "Tambah", "Ubah", "Hapus", and "Tutup".

Menghubungkan FrmMain dengan FrmDataMahasiswa

1. Masuk ke editor dari FrmMain dengan menekan tombol Source di desain form FrmMain



2. Tambahkan implements ActionListener
3. Tambahkan import :
 - a. import java.awt.event.ActionEvent;
 - b. import java.awt.event.ActionListener;
4. Tambahkan method actionPerformed

```
public void actionPerformed(ActionEvent e) {
    throw new UnsupportedOperationException("Not supported yet.");
}
```

5. Sesudah pemanggilan method initComponents() tambahkan kode program berikut
mnuDataMahasiswa.addActionListener(this);
sehingga kode program di Constructor menjadi seperti berikut

```
public FrmMain() {
    initComponents();
    mnuDataMahasiswa.addActionListener(this);
}
```

6. Deklarasikan objek frmDataMahasiswa dengan tipe FrmDataMahasiswa seperti kode program berikut
private FrmDataMahasiswa frmDataMahasiswa;
7. Lengkapi method actionPerformed seperti kode program dibawah ini

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(mnuDataMahasiswa)){
        if(frmDataMahasiswa==null)
            frmDataMahasiswa=new FrmDataMahasiswa();
            frmDataMahasiswa.setVisible(true);
    }
}
```

8. Sehingga kode program FrmDataMahasiswa seperti berikut

```
package sistemperpustakaan.view;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class FrmMain extends javax.swing.JFrame implements ActionListener{

    private FrmDataMahasiswa frmDataMahasiswa;

    public FrmMain() {
        initComponents();
        mnuDataMahasiswa.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource().equals(mnuDataMahasiswa)){
            if(frmDataMahasiswa==null)
                frmDataMahasiswa=new FrmDataMahasiswa();
            frmDataMahasiswa.setVisible(true);
        }
    }
}
```

Mengatur Main kelas

1. Silahkan buka kelas Main.java
2. Pada method main, tambahkan kode program yang ditebali

```
package sistemperpustakaan;

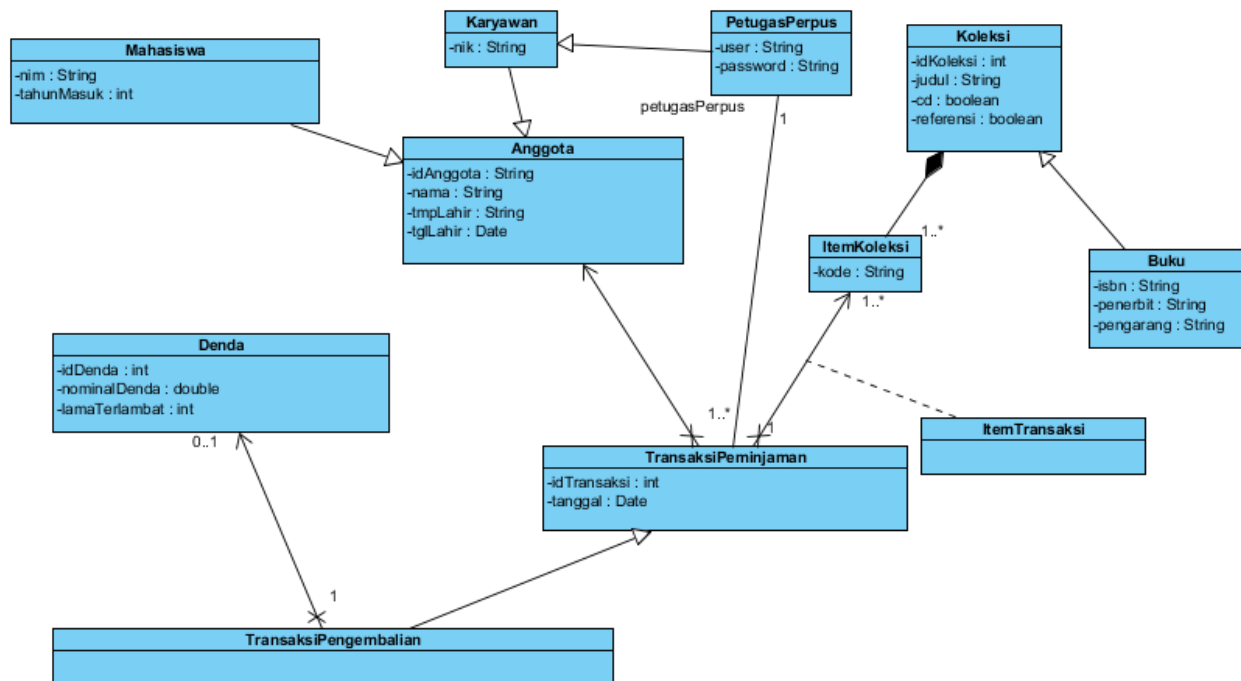
import sistemperpustakaan.view.FrmMain;

public class Main {

    public static void main(String[] args) {
        FrmMain frmMain=new FrmMain();
        frmMain.setVisible(true);
    }
}
```


Membuat Entity

Class Diagram Entity Perpustakaan



Kode Program

- **Kelas Anggota**

```
package sistemperpustakaan.entity;
import java.util.Date;
public class Anggota {
    private int id;
    private String nama;
    private String tmpLahir;
    private Date tglLahir;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public String getTmpLahir() {
        return tmpLahir;
    }
    public void setTmpLahir(String tmpLahir) {
        this.tmpLahir = tmpLahir;
    }
    public Date getTglLahir() {
        return tglLahir;
    }
    public void setTglLahir(Date tglLahir) {
        this.tglLahir = tglLahir;
    }
}
```

- **Kelas Mahasiswa**

```
package sistempustakaan.entity;
public class Mahasiswa extends Anggota {
    private String nim;
    private int tahunMasuk;
    public String getNim() {
        return nim;
    }
    public void setNim(String nim) {
        this.nim = nim;
    }
    public int getTahunMasuk() {
        return tahunMasuk;
    }
    public void setTahunMasuk(int tahunMasuk) {
        this.tahunMasuk = tahunMasuk;
    }
}
```

- **Kelas PetugasPerpus**

```
package sistempustakaan.entity;
public class PetugasPerpus extends Karyawan {
    private String user;
    private String pwd;
    public String getUser() {
        return user;
    }
    public void setUser(String user) {
        this.user = user;
    }
    public String getPwd() {
        return pwd;
    }
    public void setPwd(String pwd) {
        this.pwd = pwd;
    }
}
```

- **Kelas Koleksi**

```
package sistemperpustakaan.entity;
public class Koleksi {
    private int id;
    private String judul;
    private boolean cd;
    private boolean referensi;
    private ItemKoleksi[] itemKoleksi;
    public ItemKoleksi[] getItemKoleksi() {
        return itemKoleksi;
    }
    public void setItemKoleksi(ItemKoleksi[] itemKoleksi) {
        this.itemKoleksi = itemKoleksi;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getJudul() {
        return judul;
    }
    public void setJudul(String judul) {
        this.judul = judul;
    }
    public boolean isCd() {
        return cd;
    }
    public void setCd(boolean cd) {
        this.cd = cd;
    }
    public boolean isReferensi() {
        return referensi;
    }
    public void setReferensi(boolean referensi) {
        this.referensi = referensi;
    }
}
```

- **Kelas ItemKoleksi**

```
package sistemperpustakaan.entity;  
public class ItemKoleksi {  
    private String kode;  
    private Koleksi koleksi;  
    public Koleksi getKoleksi() {  
        return koleksi;  
    }  
    public void setKoleksi(Koleksi koleksi) {  
        this.koleksi = koleksi;  
    }  
    public String getKode() {  
        return kode;  
    }  
    public void setKode(String kode) {  
        this.kode = kode;  
    }  
}
```

- **Kelas Transaksi**

```
package sistemperpustakaan.entity;
import java.util.Date;
public class Transaksi {
    private int idTransaksi;
    private Date tanggal;
    private PetugasPerpus petugasPerpus;
    private Anggota anggota;
    private ItemKoleksi[] itemKoleksi;
    public int getIdTransaksi() {
        return idTransaksi;
    }
    public void setIdTransaksi(int idTransaksi) {
        this.idTransaksi = idTransaksi;
    }
    public Date getTanggal() {
        return tanggal;
    }
    public void setTanggal(Date tanggal) {
        this.tanggal = tanggal;
    }
    public PetugasPerpus getPetugasPerpus() {
        return petugasPerpus;
    }
    public void setPetugasPerpus(PetugasPerpus petugasPerpus) {
        this.petugasPerpus = petugasPerpus;
    }
    public Anggota getAnggota() {
        return anggota;
    }
    public void setAnggota(Anggota anggota) {
        this.anggota = anggota;
    }
    public ItemKoleksi[] getItemKoleksi() {
        return itemKoleksi;
    }
    public void setItemKoleksi(ItemKoleksi[] itemKoleksi) {
        this.itemKoleksi = itemKoleksi;
    }
}
```

- **Kelas TransaksiPeminjaman**

```
package sistemperpustakaan.entity;
public class TransaksiPeminjaman extends Transaksi{
    @Override
    public void setItemKoleksi(ItemKoleksi[] itemKoleksi) {
        // TODO Auto-generated method stub
        super.setItemKoleksi(itemKoleksi);
    }
}
```

- **Kelas TransaksiPengembalian**

```
package sistemperpustakaan.entity;
public class TransaksiPengembalian extends Transaksi{
    private Denda denda;
    @Override
    public void setItemKoleksi(ItemKoleksi[] itemKoleksi) {
        // TODO Auto-generated method stub
        super.setItemKoleksi(itemKoleksi);
    }
    public Denda getDenda() {
        return denda;
    }
    public void setDenda(Denda denda) {
        this.denda = denda;
    }
}
```

- **Kelas Karyawan**

```
package sistemperpustakaan.entity;
public class Karyawan extends Anggota {
    private String nik;
    public String getNik() {
        return nik;
    }
    public void setNik(String nik) {
        this.nik = nik;
    }
}
```

- **Kelas Denda**

```
package sistempustaka.entity;
public class Denda {
    private int idDenda;
    private double nominalDenda;
    private int lamaTerlambat;
    public int getIdDenda() {
        return idDenda;
    }
    public void setIdDenda(int idDenda) {
        this.idDenda = idDenda;
    }
    public double getNominalDenda() {
        return nominalDenda;
    }
    public void setNominalDenda(double nominalDenda) {
        this.nominalDenda = nominalDenda;
    }
    public int getLamaTerlambat() {
        return lamaTerlambat;
    }
    public void setLamaTerlambat(int lamaTerlambat) {
        this.lamaTerlambat = lamaTerlambat;
    }
}
```


- **Kelas Buku**

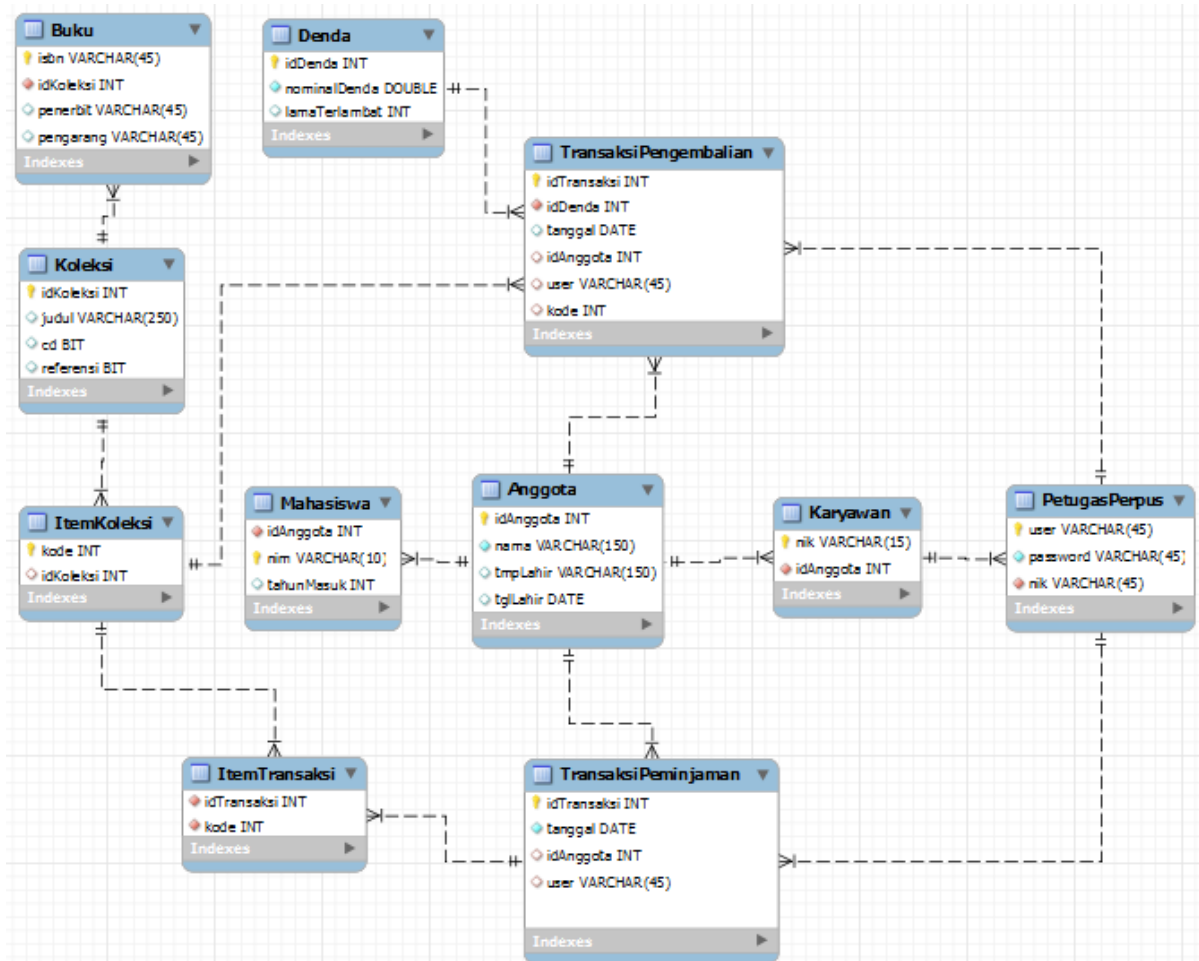
```
package sistempustaka.entity;
public class Buku extends Koleksi {
    private String isbn;
    private String penerbit;
    private String pengarang;
    public String getIsbn() {
        return isbn;
    }
    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }
    public String getPenerbit() {
        return penerbit;
    }
    public void setPenerbit(String penerbit) {
        this.penerbit = penerbit;
    }
    public String getPengarang() {
        return pengarang;
    }
    public void setPengarang(String pengarang) {
        this.pengarang = pengarang;
    }
}
```

- **ItemTransaksi**

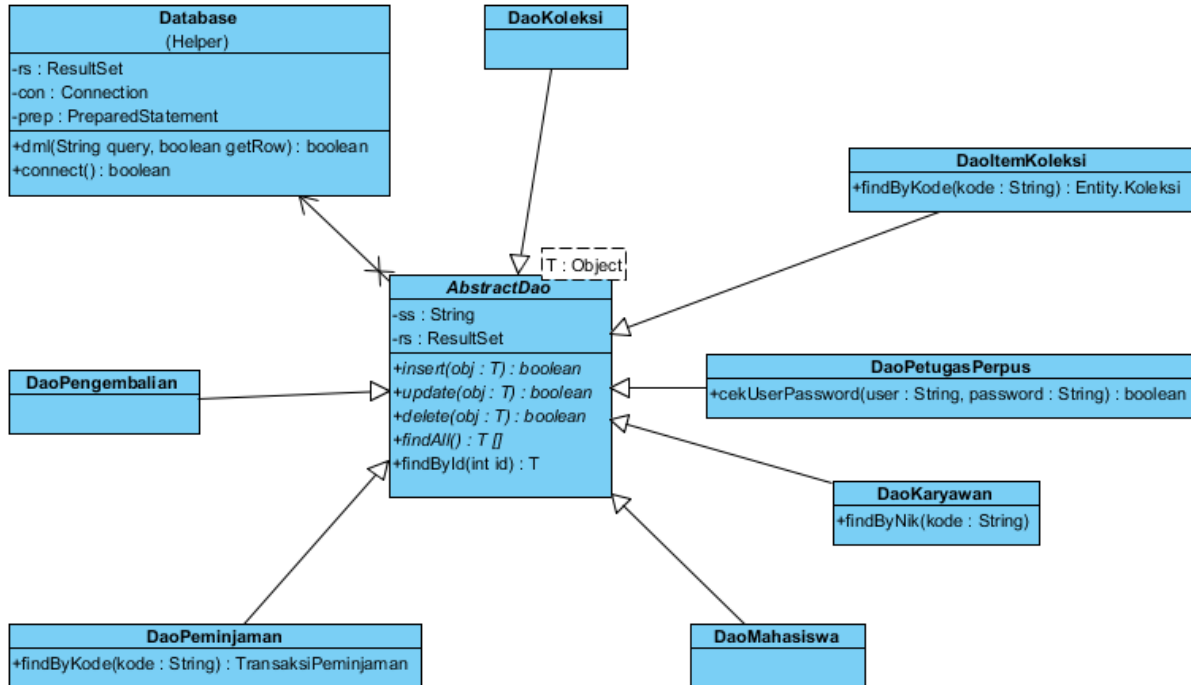
```
package sistemperpustakaan.entity;  
public class ItemTransaksi {  
    private ItemKoleksi itemKoleksi;  
    private Transaksi transaksi;  
    public ItemKoleksi getItemKoleksi() {  
        return itemKoleksi;  
    }  
    public void setItemKoleksi(ItemKoleksi itemKoleksi) {  
        this.itemKoleksi = itemKoleksi;  
    }  
    public Transaksi getTransaksi() {  
        return transaksi;  
    }  
    public void setTransaksi(Transaksi transaksi) {  
        this.transaksi = transaksi;  
    }  
}
```

Membuat DAO

Diagram Database



Class Diagram DAO Perpustakaan



Kode Program

- Kelas Database

```
package sistemperpustakaan.helper;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class Database {
    private Connection con;
    private PreparedStatement prep;
    private ResultSet rs;
    public boolean connect() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/perpustakaan", "root", "");
        } catch (SQLException ex) {
            System.out.println("Error 1 : " + ex.getMessage());
            return false;
        } catch (ClassNotFoundException ex) {
            System.out.println("Error 2 : " + ex.getMessage());
            return false;
        }
        return true;
    }
    public Database(){
        connect();
    }
    public ResultSet getRs() {
        return rs;
    }
}
```

```

public boolean dml(String query, boolean getRow, Object... objects) {
    try {
        prep = con.prepareStatement(query);
        int i = 1;
        for (Object obj : objects) {
            prep.setObject(i, obj);
            i++;
        }
        if (getRow) {
            rs = prep.executeQuery();
        } else {
            prep.executeUpdate();
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return false;
    }
    return true;
}

public int getGeneratedKeys() {
    int keys=0;
    try{
        if(dml("SELECT LAST_INSERT_ID();",true)){
            rs.next();
            keys=rs.getInt("LAST_INSERT_ID()");
            rs.close();
            return keys;
        }
        return -1;
    }catch(SQLException e){
        return -1;
    }
}
}

```

- Kelas AbstractDao

```
package sistemperpustakaan.dao;
import java.sql.ResultSet;
import helper.Database;
public abstract class AbstractDao<T> {
    protected Database database;
    protected String ss="";
    protected ResultSet rs;
    public AbstractDao(){
        database=new Database();
    }
    public abstract boolean insert(T obj);
    public abstract boolean update(T obj);
    public abstract boolean delete(T obj);
    public abstract T[] findAll();
    public abstract T findById(int id);
}
```

- Kelas DaoMahasiswa

```
package sistemperpustakaan.dao;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import entity.Mahasiswa;
public class DaoMahasiswa extends AbstractDao<Mahasiswa> {
    @Override
    public boolean delete(Mahasiswa obj) {
        obj.setId(findId(obj.getNim()));
        ss="delete mahasiswa where idanggota=?";
        if(!database.dml(ss, false, obj.getId())){
            return false;
        }
        ss="delete anggota where idanggota=?";
        if(!database.dml(ss, false, obj.getId())){
            return false;
        }
        return true;
    }
    @Override
    public Mahasiswa findById(int id) {
        return null;
    }

    @Override
    public boolean insert(Mahasiswa obj) {
        ss="insert into anggota(nama,tmp_lahir,tgl_lahir)values(?,?,?);";
        if(!database.dml(ss, false, obj.getNama(),obj.getTmpLahir(),obj.getTglLahir())){
            return false;
        }

        obj.setId(database.getGeneratedKeys());
        ss="insert into mahasiswa(idanggota,nim,tahunMasuk)values(?,?,?);";
        if(!database.dml(ss, false, obj.getId(),obj.getNim(),obj.getTahunMasuk())){
            return false;
        }
        return true;
    }
}
```



```

@Override
public Mahasiswa[] findAll() {
    ss="select a.idAnggota,nim,nama,tmplahir,tglahir,tahunmasuk ";
    ss+="from mahasiswa m ";
    ss+="inner join anggota a on m.idanggota=a.idanggota";
    if(database.dml(ss, true)){
        rs=database.getRs();
        List<Mahasiswa> lstMahasiswa=new ArrayList<Mahasiswa>();
        try {
            while(rs.next()){
                Mahasiswa mahasiswa=new Mahasiswa();
                mahasiswa.setld(rs.getInt("idAnggota"));
                mahasiswa.setNim(rs.getString("nim"));
                mahasiswa.setNama(rs.getString("nama"));
                mahasiswa.setTmpLahir(rs.getString("tmplahir"));
                mahasiswa.setTglLahir(rs.getDate("tglahir"));
                mahasiswa.setTahunMasuk(rs.getInt("tahunmasuk"));
                lstMahasiswa.add(mahasiswa);
            }
            rs.close();
            return lstMahasiswa.toArray(new Mahasiswa[lstMahasiswa.size()]);
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
    return null;
}

@Override
public boolean update(Mahasiswa obj) {
    obj.setld(findld(obj.getNim()));
    ss="update anggota set nama=?,tmplahir=?,tglahir=? where idanggota=?";
    if(!database.dml(ss, false, obj.getNama(),obj.getTmpLahir(),obj.getTglLahir(),obj.getId())){
        return false;
    }
    ss="update mahasiswa set tahunmasuk=? where idanggota=?";
    if(!database.dml(ss, false, obj.getTahunMasuk(),obj.getId())){
        return false;
    }
    return true;
}

```

```

private int findId(String nim){
    int result=0;
    try{
        ss="select idAnggota from anggota a ";
        ss+="inner join mahasiswa m on a.idanggota=m.idanggota ";
        ss+="where nim=?";
        if(database.dml(ss, true, nim)){
            rs=database.getRs();
            if(rs.next()){
                result= rs.getInt("idAnggota");
            }
            rs.close();
        }
        return result;
    }catch(SQLException ex){
        ex.printStackTrace();
        return result;
    }
}

```

Untuk DAO Selanjutnya silahkan dikembangkan sendiri

- Kelas DaoPetugasPerpus
- Kelas DaoPengembalian
- Kelas DaoPeminjaman
- Kelas DaoItemKoleksi
- Kelas DaoPetugasPerpus
- Kelas DaoKaryawan