

## MODUL 1

### IMPLEMENTASI DAO

#### Teori Singkat

DAO merupakan design pattern berkaitan dengan database access. DAO memisahkan persistence logic dari business atau presentation logic. DAO merupakan representasi fisik dan relasi antara object dari database.

DAO design pattern diimplementasi dengan beberapa langkah, yaitu :

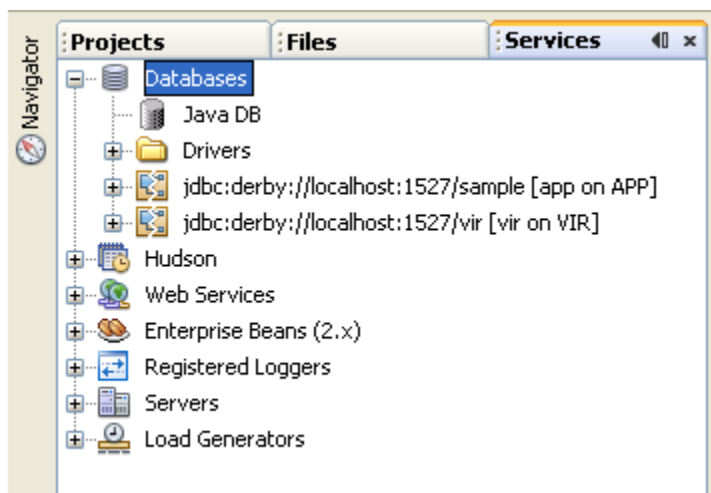
- Mendefinisikan interface DAO
- Menulis implementasi dari interface DAO
- Menuliskan business logic dari aplikasi untuk mengakses data source

#### Latihan 3

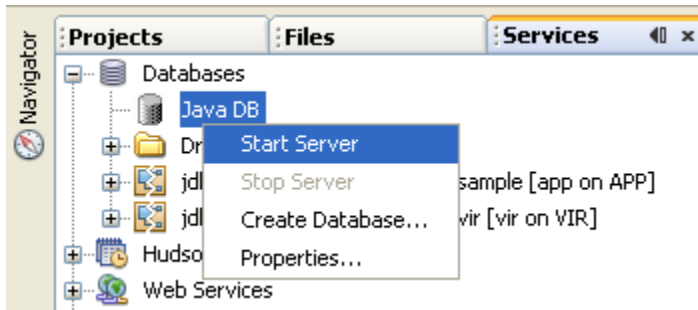
1. Buka kembali project yang telah dibuat pada latihan 2 (project akademik)
2. Pada latihan ini akan diimplementasikan penggunaan DAO untuk membangun konektivitas antara aplikasi dengan database management system (dbms)

TAHAP 1 (Menyiapkan database)

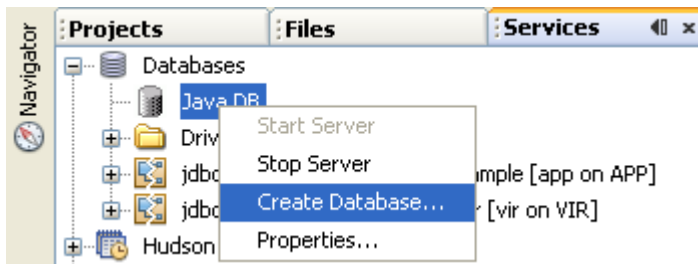
3. Untuk database akan digunakan javaDB yang sudah terintegrasi dengan netbeans 6.5
4. Pada netbeans aktifkan window services (CTRL+5)



5. Aktifkan server JavaDB, klik kanan → Start Server

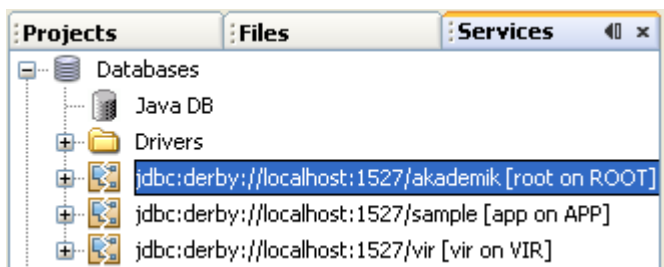
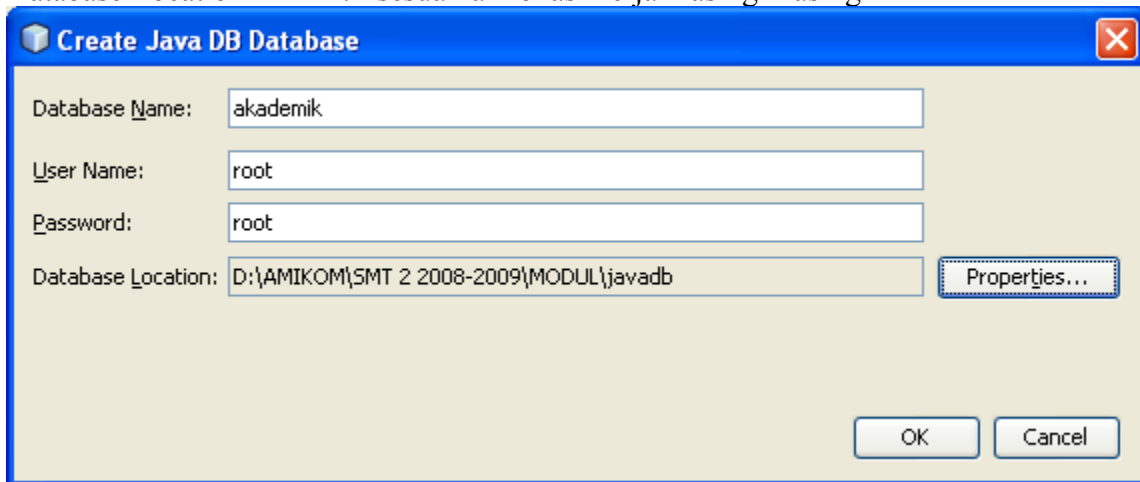


6. Buat database baru, klik kanan → Create Database

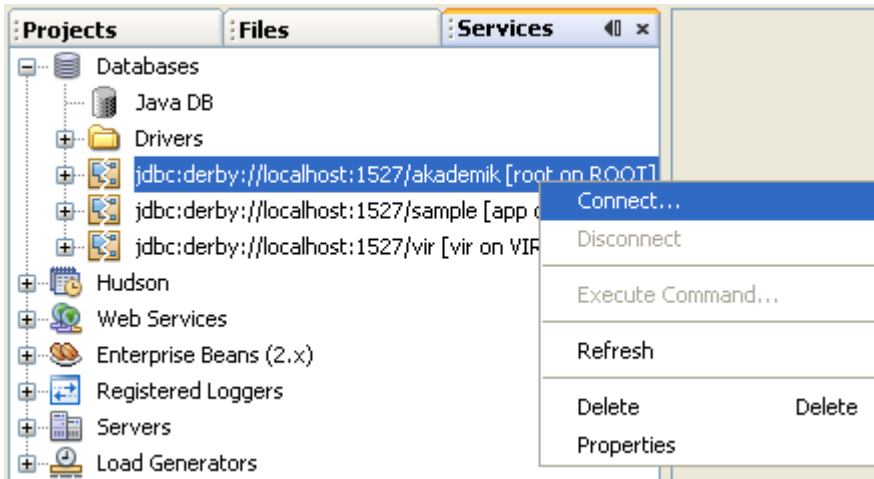


Database baru :

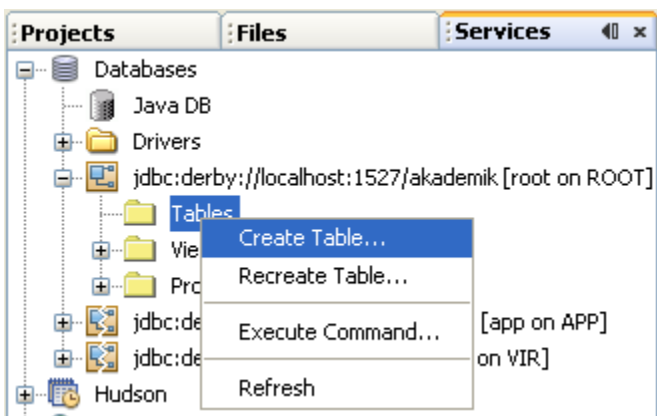
Database Name : akademik  
User Name : root  
Password : root  
Database Location : <sesuaikan lokasi kerja masing-masing>



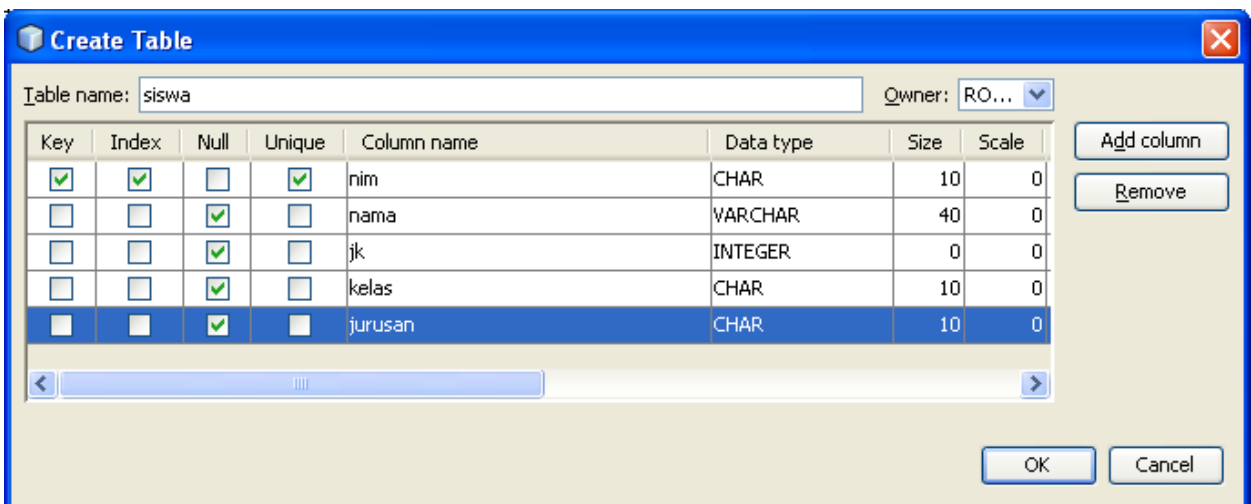
7. Buka koneksi ke database akademik yang baru saya di buat, klik kanan → Connect

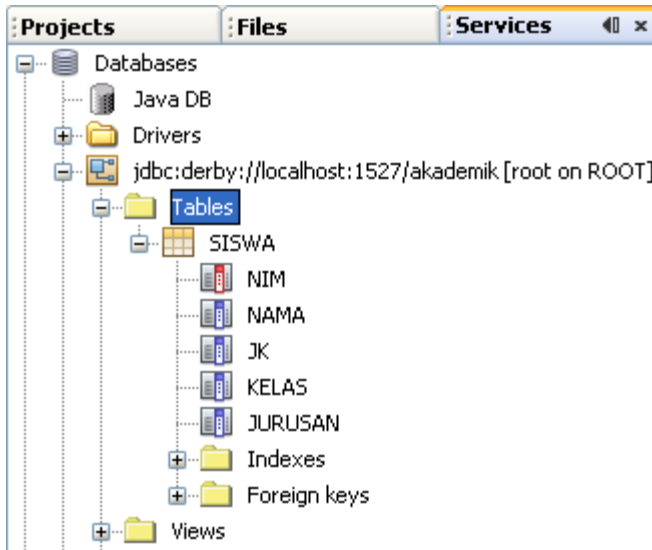


8. Buat table baru

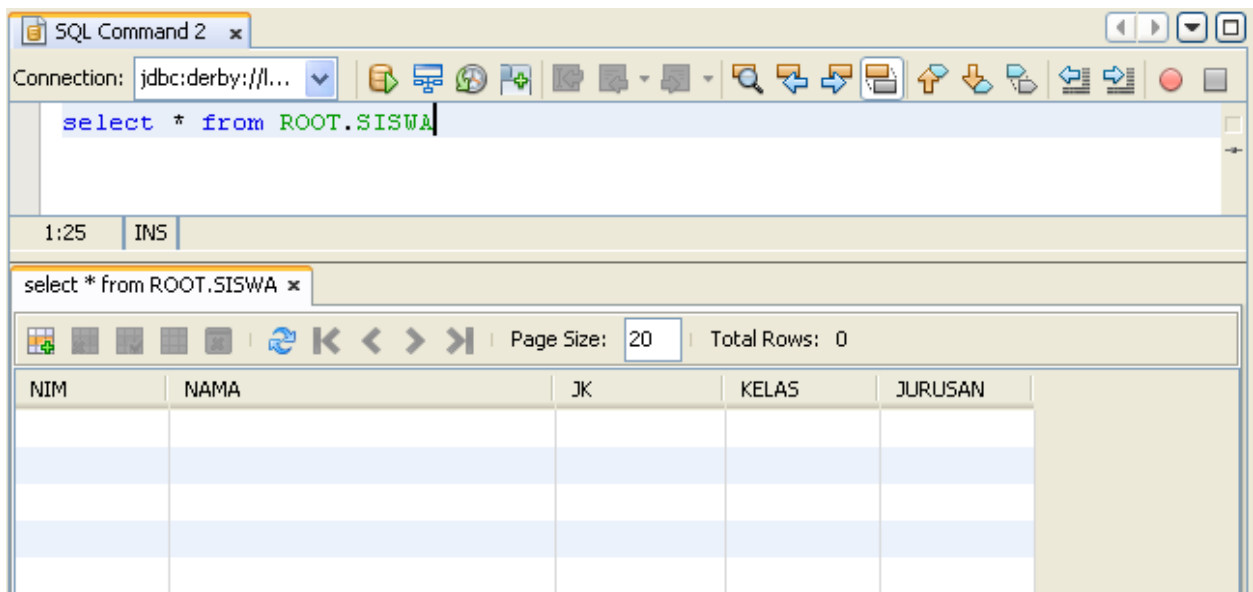
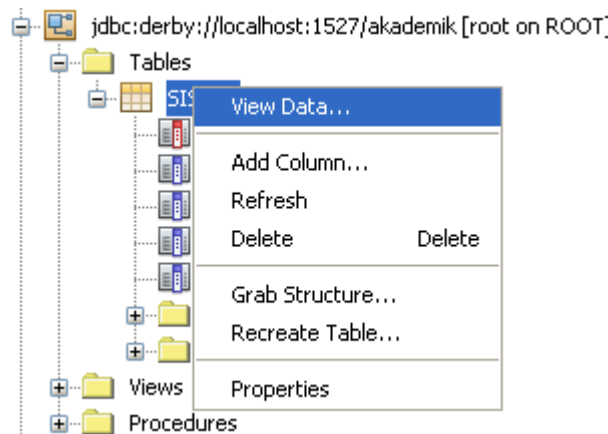



Definisikan table baru seperti dalam image dibawah ini :





9. Kita isi table siswa dengan beberapa data sample, klik kanan → View Data












10. Tambahkan 3 buah data siswa, seperti berikut : (klik tombol **insert record** )

**Insert Record** ✕

Enter values for the following columns that were found in the table. Time and date values must match the ISO date and time format. Press CTRL+0 to toggle setting NULL value for a given column.

NIM	<input type="text" value="01"/>	CHAR
NAMA	<input type="text" value="Agus"/>	VARCHAR
JK	<input type="text" value="0"/>	INTEGER
KELAS	<input type="text" value="Satu"/>	CHAR
JURUSAN	<input type="text" value="IPA"/>	CHAR

select \* from ROOT.SISWA ×

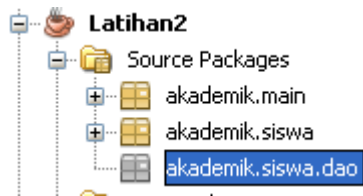
         | Page Size:  | Total Rows: 3 (Page 1 of 1)

NIM	NAMA	JK	KELAS	JURUSAN
01	Agus	0	Satu	IPA
02	Agustin	1	Dua	BAHASA
03	Jaka	0	Satu	IPS

## TAHAP 2 (Membangun DAO)

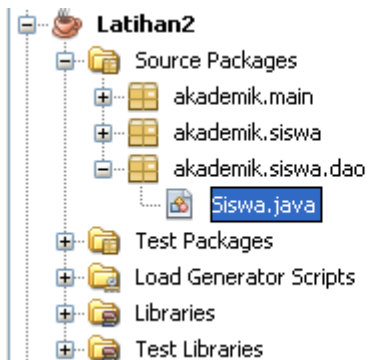
11. Membuat Entity Class, dalam hal ini adalah Entity Class **Siswa**

12. Buat Java Package Baru dengan nama : **akademik.siswa.dao**



13. Buat satu buah **class java** baru di dalam package : **akademik.siswa.dao**

Dengan nama class : **Siswa**



14. Lengkapi kode program untuk class **Siswa** menjadi seperti berikut :

```
package akademik.siswa.dao;

/**
 *
 * @author emha taufiq luthfi
 */
public class Siswa {
    private String nim;
    private String nama;
    private int jk;
    private String kelas;
    private String jurusan;

    public Siswa() {
    }

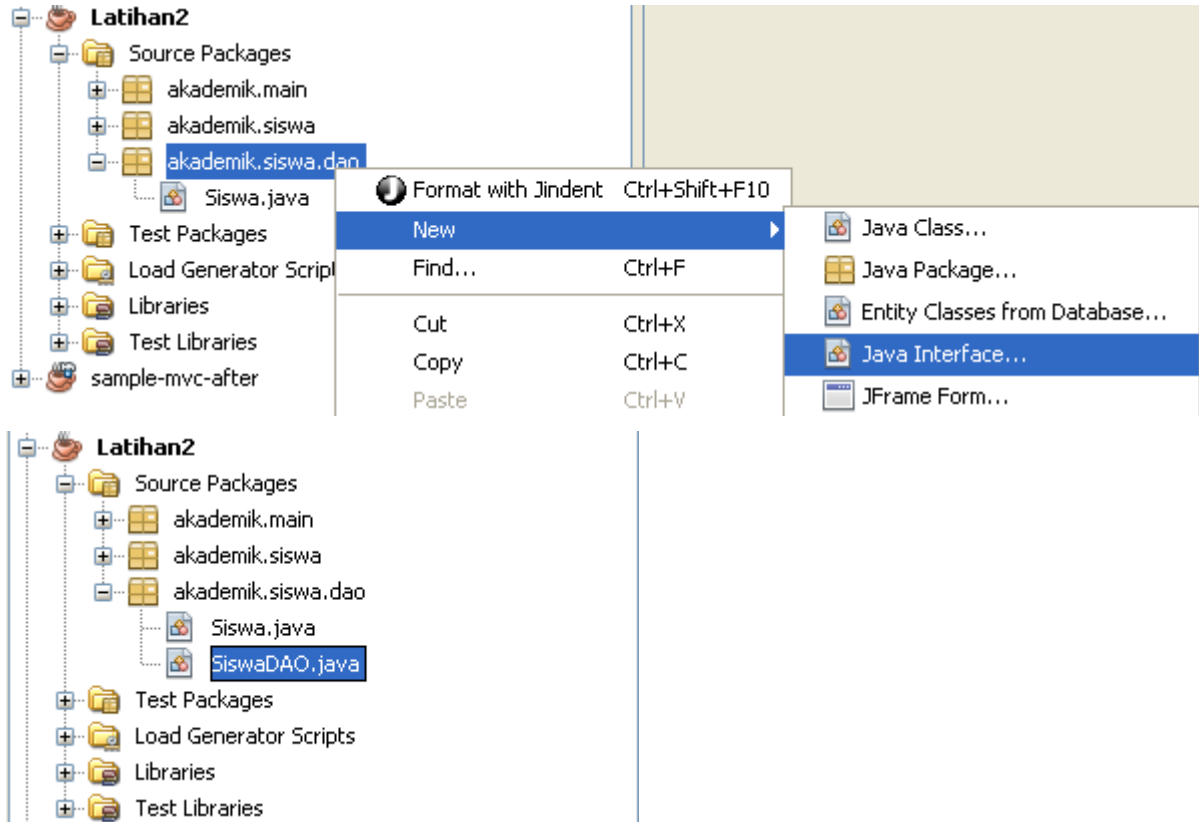
    public Siswa(String nim, String nama, int jk, String kelas, String jurusan) {
        this.nim = nim;
        this.nama = nama;
        this.jk = jk;
        this.kelas = kelas;
        this.jurusan = jurusan;
    }
}
```

```
public String getNim() {  
    return nim;  
}  
  
public String getNama() {  
    return nama;  
}  
  
public int getJk() {  
    return jk;  
}  
  
public String getKelas() {  
    return kelas;  
}  
  
public String getJurusan() {  
    return jurusan;  
}  
  
public void setNim(String nim) {  
    this.nim = nim;  
}  
  
public void setNama(String nama) {  
    this.nama = nama;  
}  
  
public void setJk(int jk) {  
    this.jk = jk;  
}  
  
public void setKelas(String kelas) {  
    this.kelas = kelas;  
}  
  
public void setJurusan(String jurusan) {  
    this.jurusan = jurusan;  
}  
}
```

15. Selanjutnya buat sebuah interface yang merupakan rancangan dari fungsionalitas DAO untuk siswa.

Buat satu **interface** dalam package **akademik.siswa.dao**

dengan nama interface : **SiswaDAO**



16. Lengkapi kode program interface SiswaDAO menjadi seperti berikut :

```
package akademik.siswa.dao;

import java.util.List;

public interface SiswaDAO {

    Siswa getSiswa(String nim);

    List getAllSiswa();

    void insertSiswa(Siswa siswa);

    void updateSiswa(Siswa siswa);

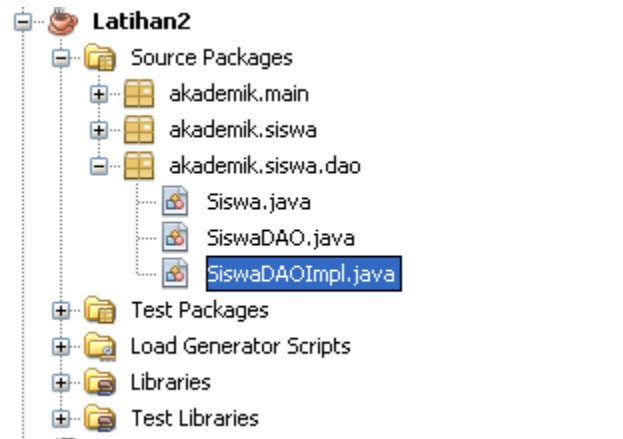
    void deleteSiswa(Siswa siswa);

}
```



17. Selanjutnya kita buat class implementasi untuk interface SiswaDAO yang sudah dibuat;

Buat satu buah **java class** baru dengan nama : **SiswaDAOImpl**



18. Untuk **tahap pertama** lengkapi kode program pada class **SiswaDAOImpl** menjadi seperti berikut :

```
package akademik.siswa.dao;

import java.util.ArrayList;
import java.util.List;

public class SiswaDAOImpl implements SiswaDAO {

    public SiswaDAOImpl() {
    }

    public Siswa getSiswa(String nim) {
        Siswa siswa = new Siswa();
        return siswa;
    }

    public List getAllSiswa() {
        List listSiswa = new ArrayList<Siswa>();
        return listSiswa;
    }

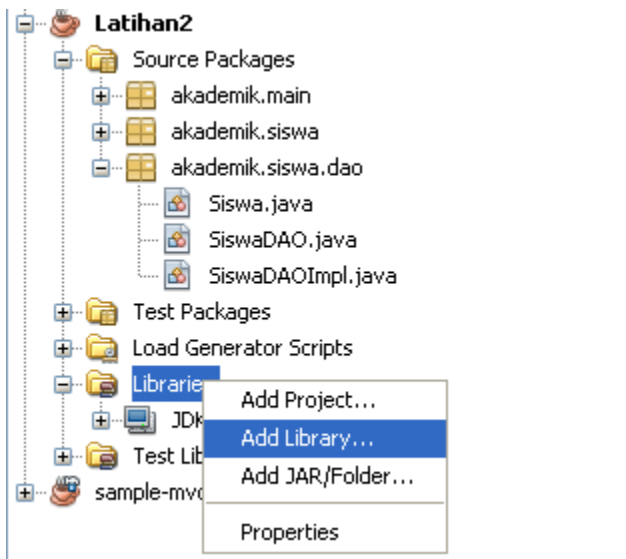
    public void insertSiswa(Siswa siswa) {
    }

    public void updateSiswa(Siswa siswa) {
    }

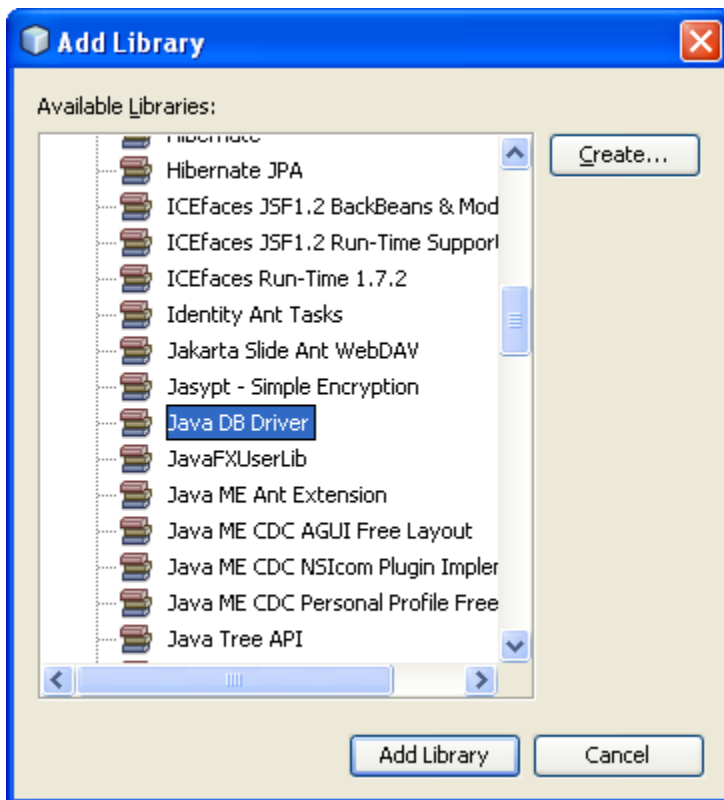
    public void deleteSiswa(Siswa siswa) {
    }
}
```

19. Selanjutnya kita bangun konektivitas dengan dbms
20. Tambahkan dulu driver JDBC javaDB ke dalam project

Pada node libraries, klik kanan → Add Library



Pilih Java DB Driver



21. Lengkapi kode pada bagian konstruktor dan field dari class **Siswa** serta buat satu buah method baru dengan nama **getConnection()**;

```
public class SiswaDAOImpl implements SiswaDAO {  
  
    private Connection connection;  
  
    public SiswaDAOImpl() {  
        try {  
            // 1 - tentukan driver yang digunakan  
            Class.forName("org.apache.derby.jdbc.ClientDriver");  
  
            // 2 - tentukan url koneksi  
            String url = "jdbc:derby://localhost:1527/akademik";  
  
            // 3 - buat object connection  
            connection = DriverManager.getConnection(url, "root", "root");  
  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
  
    public Connection getConnection() {  
        return connection;  
    }  
}
```

22. Lengkapi kode program pada metode **getSiswa()** menjadi seperti berikut :

```
public Siswa getSiswa(String nim) {  
    Siswa siswa = new Siswa();  
    Connection conn;  
    Statement stmt;  
  
    try {  
        //siapkan koneksi yang digunakan  
        conn = getConnection();  
        //siapkan statement untuk mengeksekusi query  
        stmt = conn.createStatement();  
        //ambil data dan letakkan di ResultSet  
        String query = "select * from siswa where nim='"+ nim +"'";  
        ResultSet rsSiswa = stmt.executeQuery(query);  
        rsSiswa.next();  
    }  
}
```

```

        //bentuk object siswa dari data di ResultSet
        siswa.setNim(rsSiswa.getString(1));
        siswa.setNama(rsSiswa.getString(2));
        siswa.setJk(rsSiswa.getInt(3));
        siswa.setKelas(rsSiswa.getString(4));
        siswa.setJurusan(rsSiswa.getString(5));

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return siswa;
}

```

23. Dalam tahap 1 ini telah selesai dibangun metode untuk mengambil data siswa berdasar nimnya.

Selanjutnya coba kita implementasikan dalam **modelSiswa**

24. Buka class **modelSiswa**, ubah kode di konstruktor menjadi seperti berikut :

```

package akademik.siswa;

import akademik.siswa.dao.Siswa;
import akademik.siswa.dao.SiswaDAOImpl;
import java.util.ArrayList;

public class SiswaModel {
    private String nim;
    private String nama;
    private int jenisKelamin;
    private String kelas;
    private String jurusan;
    private ArrayList<Object> listeners = new ArrayList<Object>(10);

    // deklarasikan object siswaDAO
    private SiswaDAOImpl siswaDAO;

    public SiswaModel() {
        // instansiasi object
        siswaDAO = new SiswaDAOImpl();
        //ambil data siswa yang nim nya = 01
        Siswa siswa = siswaDAO.getSiswa("01");
        setNim(siswa.getNim());
        setNama(siswa.getNama());
        setJenisKelamin(siswa.getJk());
        setKelas(siswa.getKelas());
        setJurusan(siswa.getJurusan());
    }
}

```

25. Selanjutnya coba kita jalankan dahulu, **harusnya akan tampil data siswa dengan nim 01**

```
select * from ROOT.SISWA where nim='01'
```

1:40 | INS

select \* from ROOT.SISWA ×

Page Size: 20 | Total Rows: 1 (Page 1 of 1)

NIM	NAMA	JK	KELAS	JURUSAN
01	Agus	0	Satu	IPA

NIM: 01

Nama: Agus

Jenis Kelamin:  Laki-Laki  Perempuan

Kelas: Satu

Jurusan: IPA, IPS, BAHASA

Update

NIM : 01

Nama : Agus

Jenis Kelamin : Laki-Laki

Kelas : Satu

Jurusan : IPA

26. Coba ubah dibagian konstruktor siswaModel, untuk **menampilkan siswa dengan nim=02**

```
public SiswaModel() {  
    // instansiasi object  
    siswaDAO = new SiswaDAOImpl();  
    //ambil data siswa yang nim nya = 01  
    Siswa siswa = siswaDAO.getSiswa("02");  
    setNim(siswa.getNim());  
    setName(siswa.getName());  
    setJenisKelamin(siswa.getJk());  
    setKelas(siswa.getKelas());  
    setJurusan(siswa.getJurusan());  
}
```

Jalankan :

The screenshot shows a database query tool interface. At the top, a text box contains the SQL query: `select * from ROOT.SISWA where nim='02'`. Below the query, the tool shows the execution time as 1:38 and the table name as INS. A toolbar includes icons for refresh, back, forward, and search, along with a page size of 20 and a total of 1 row. The results are displayed in a table with the following data:

NIM	NAMA	JK	KELAS	JURUSAN
02	Agustin	1	Dua	BAHASA

The screenshot shows a Java Swing window titled "Update" with a light gray background and a blue border. It contains several input fields and a button:

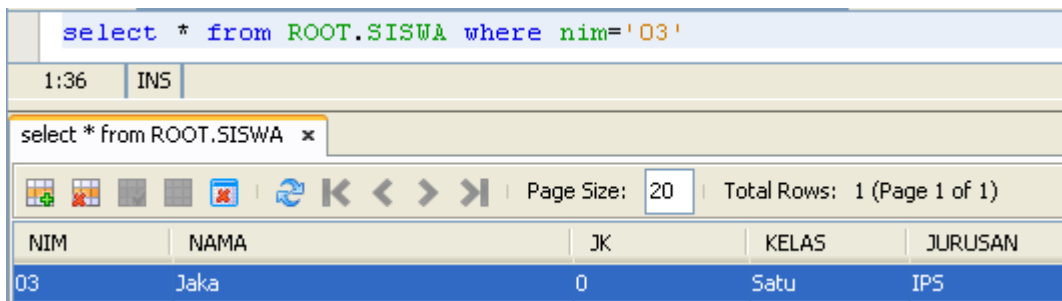
- NIM:** A text field containing the value "02".
- Nama:** A text field containing the value "Agustin".
- Jenis Kelamin:** Two radio buttons, "Laki-Laki" (unselected) and "Perempuan" (selected).
- Kelas:** A dropdown menu currently showing "Satu".
- Jurusan:** A list box containing the options "IPA", "IPS", and "BAHASA".
- Update:** A button at the bottom of the window.



27. Coba ubah dibagian konstruktor siswaModel, untuk **menampilkan siswa dengan nim=03**

```
public SiswaModel() {  
    // instansiasi object  
    siswaDAO = new SiswaDAOImpl();  
    //ambil data siswa yang nim nya = 01  
    Siswa siswa = siswaDAO.getSiswa("03");  
    setNim(siswa.getNim());  
    setName(siswa.getName());  
    setJenisKelamin(siswa.getJk());  
    setKelas(siswa.getKelas());  
    setJurusan(siswa.getJurusan());  
}
```

Jalankan :



A Java Swing window with a blue title bar and standard window controls (minimize, maximize, close). The window contains a form with the following fields and controls:

- NIM**: A text input field containing the value "03".
- Nama**: A text input field containing the value "Jaka".
- Jenis Kelamin**: Two radio buttons, "Laki-Laki" (which is selected) and "Perempuan".
- Kelas**: A dropdown menu currently showing "Satu".
- Jurusan**: A list box containing the items "IPA", "IPS", and "BAHASA".
- Update**: A button located at the bottom center of the window.

A Java Swing window with a blue title bar and standard window controls (minimize, maximize, close). The window displays the following text:

**NIM : 03**

**Nama : Jaka**

**Jenis Kelamin : Laki-Laki**

**Kelas : Satu**

**Jurusan : IPS**



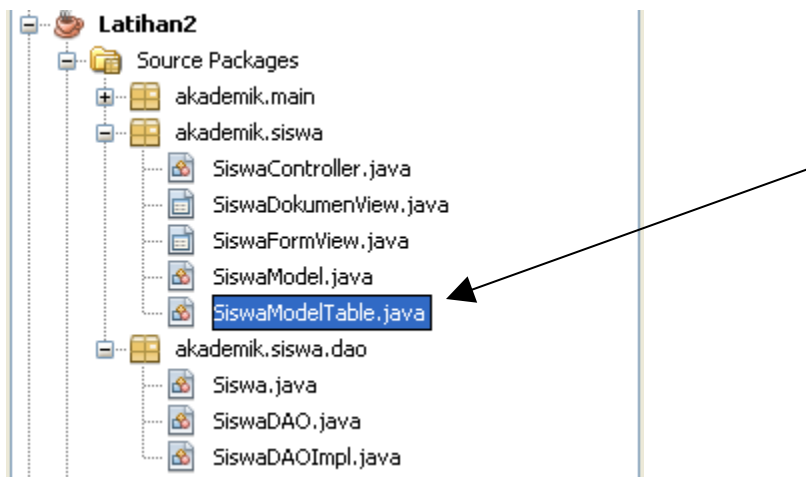
28. Lengkapi kode program pada metode `getAllSiswa()` menjadi seperti berikut :

```
public List getAllSiswa() {
    List listSiswa = new ArrayList<Siswa>();
    Connection conn;
    Statement stmt;
    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        stmt = conn.createStatement();
        //ambil data dan letakkan di ResultSet
        String query = "select * from siswa";
        ResultSet rsSiswa = stmt.executeQuery(query);
        //bentuk object list yang terdiri banyak object siswa
        //dari data di ResultSet
        while (rsSiswa.next()) {
            Siswa siswa = new Siswa();
            siswa.setNim(rsSiswa.getString(1));
            siswa.setNama(rsSiswa.getString(2));
            siswa.setJk(rsSiswa.getInt(3));
            siswa.setKelas(rsSiswa.getString(4));
            siswa.setJurusan(rsSiswa.getString(5));

            listSiswa.add(siswa);
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return listSiswa;
}
```

29. Pada package `akademik.siswa` buat sebuah class java baru yang nanti akan menjadi model untuk menampilkan data dalam bentuk tabel.

Beri nama class baru tersebut : `SiswaModelTable`



30. Lengkapi kode program class **SiswaModelTable** menjadi seperti berikut :

```
package akademik.siswa;

import akademik.siswa.dao.Siswa;
import akademik.siswa.dao.SiswaDAOImpl;
import java.util.ArrayList;
import java.util.List;
import javax.swing.table.AbstractTableModel;

/**
 *
 * @author emha taufiq luthfi
 */
public class SiswaModelTable extends AbstractTableModel {
    private static final long serialVersionUID = 2368207257439644156L;

    List<Siswa> list = new ArrayList<Siswa>();
    SiswaDAOImpl siswaDAOImpl = new SiswaDAOImpl();

    public SiswaModelTable() {
        list = siswaDAOImpl.getAllSiswa();
    }

    public int getColumnCount() {
        return 5;
    }

    public int getRowCount() {
        return list.size();
    }

    public String getColumnName(int column) {
        switch (column) {
            case 0:
                return "Nim";
            case 1:
                return "Nama";
            case 2:
                return "Jk";
            case 3:
                return "Kelas";
            case 4:
                return "Jurusan";
            default:
                return null;
        }
    }
}
```

```
public Object getValueAt(int rowIndex, int columnIndex) {
    switch (columnIndex) {
        case 0:
            return list.get(rowIndex).getNim();
        case 1:
            return list.get(rowIndex).getNama();
        case 2:
            return list.get(rowIndex).getJk();
        case 3:
            return list.get(rowIndex).getKelas();
        case 4:
            return list.get(rowIndex).getJurusan();
        default:
            return null;
    }
}
```

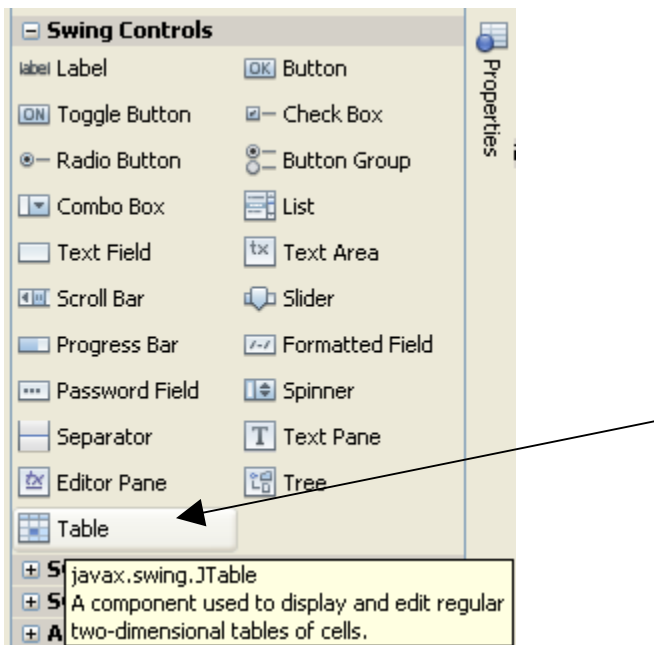
### 31. Tampilkan design GUI dari class **SiswaFormView**

Kondisi awal yang telah dibuat adalah sbb :

The screenshot shows a Java Swing window with a light beige background. It contains the following elements:

- NIM:** A text input field.
- Nama:** A text input field.
- Jenis Kelamin:** Two radio buttons labeled "Laki-Laki" and "Perempuan".
- Kelas:** A dropdown menu with "Satu" selected.
- Jurusan:** A list box containing "IPA", "IPS", and "BAHASA".
- Update:** A button at the bottom center.

Tambahkan satu buat object **Table** ke dalam frame sehingga menjadi seperti berikut :



NIM

Nama

Jenis Kelamin  Laki-Laki  Perempuan

Kelas

Jurusan   
IPS  
BAHASA

Title 1	Title 2	Title 3	Title 4

Ubah variable Name dari table menjadi : **tabelSiswa**

32. Ubah / lengkapi sedikit kode program di class SiswaFormView sehingga menjadi seperti berikut :

```
package akademik.siswa;

import java.util.ArrayList;

/**
 *
 * @author emha taufiq luthfi
 */
public class SiswaFormView extends javax.swing.JFrame {

    private SiswaModel model;
    private ArrayList<SiswaController> listeners =
        new ArrayList<SiswaController>();

    //buat object SiswaModelTable model untuk menampilkan data dalam bentuk model
    private SiswaModelTable modelTabel;

    /** Creates new form SiswaFormView */
    public SiswaFormView(SiswaModel model) {
        this.model = model;
        initComponents();

        modelTabel = new SiswaModelTable();
        tabelSiswa.getSelectionModel().addListSelectionListener(tabelSiswa);
        tabelSiswa.setModel(modelTabel);

        refresh();
        setVisible(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        model.addModelChangeListener(this);
    }
}
```

33. Coba jalankan, akan didapat hasil sbb :

The screenshot shows a Java Swing application window with a blue title bar and standard window controls (minimize, maximize, close). The window contains a form with the following fields and controls:

- NIM**: A text input field containing the value "03".
- Nama**: A text input field containing the value "Jaka".
- Jenis Kelamin**: Two radio buttons, "Laki-Laki" (which is selected) and "Perempuan".
- Kelas**: A dropdown menu currently showing "Satu".
- Jurusan**: A list box containing the items "IPA", "IPS", and "BAHASA".
- Update**: A button located below the list box.

Below the form is a table displaying a list of student records:

Nim	Nama	Jk	Kelas	Jurusan
01	Agus	0	Satu	IPA
02	Agustin	1	Dua	BAHASA
03	Jaka	0	Satu	IPS

34. Pelajari