

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### 1. Tujuan

- Instalasi MySQL Server
- Membuat program Address Book
- Mengetahui cara membuat JDBC baru
- Membuat koneksi ke Database Melalui class DriverManager
- Membuat koneksi ke Database Melalui class DataSource
- Melakukan Operasi INSERT, UPDATE, DELETE dan SELECT

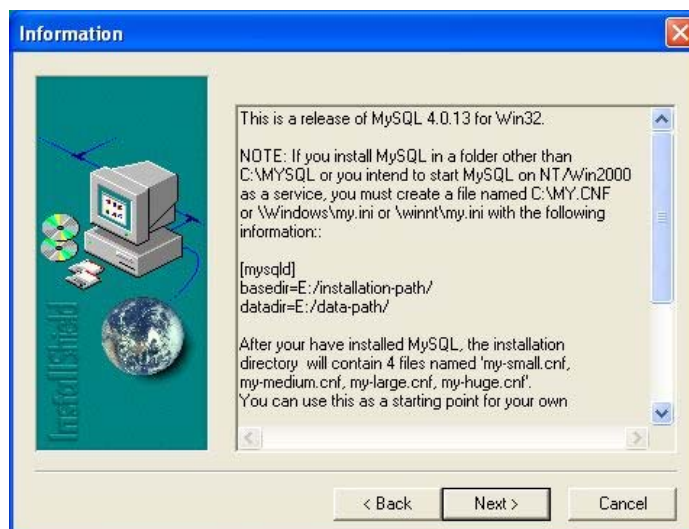
### 2. Latar Belakang

Pada modul praktikum ini, anda akan mulai membuat aplikasi yang membutuhkan database. Database server yang digunakan pada modul praktikum ini adalah MySQL Server. Modul ini menerangkan beberapa hal yang terkait seputar SQL dan JDBC, mulai dari instalasi MySQL Server, membuat program Address Book, mengetahui cara membuat JDBC baru, membuat koneksi ke Database Melalui class DriverManager, membuat koneksi ke Database Melalui class DataSource, hingga melakukan Operasi INSERT, UPDATE, DELETE dan SELECT.

### 3. Percobaan

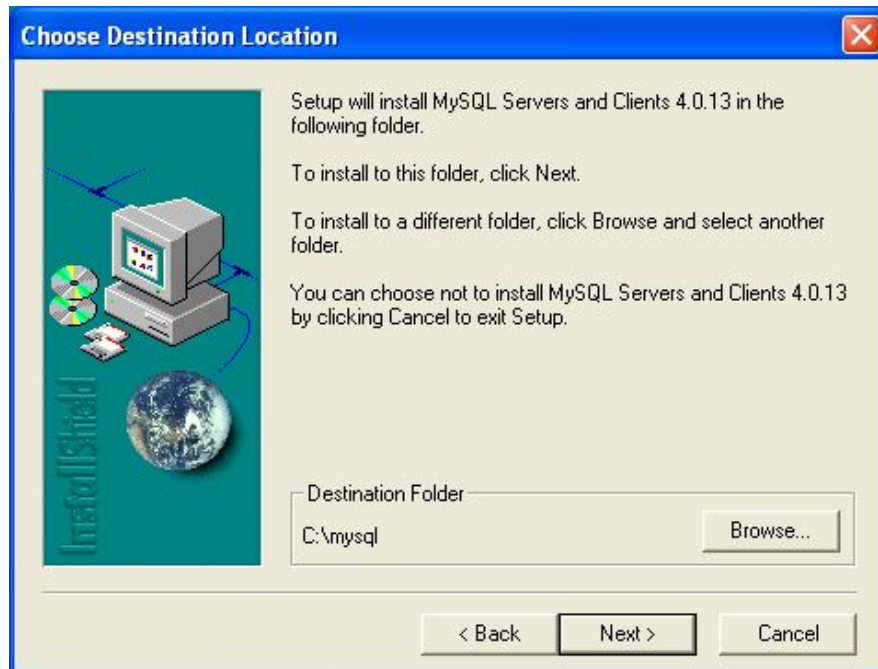
#### **Percobaan 1 – Instalasi MySQL Server:**

1. Download MySQL server dari <http://dev.mysql.com/downloads/>
2. Jika file yang anda ambil berupa file zip, ekstrak file mysql-4.x.x.zip ke direktori anda
3. Jalankan file Setup.exe

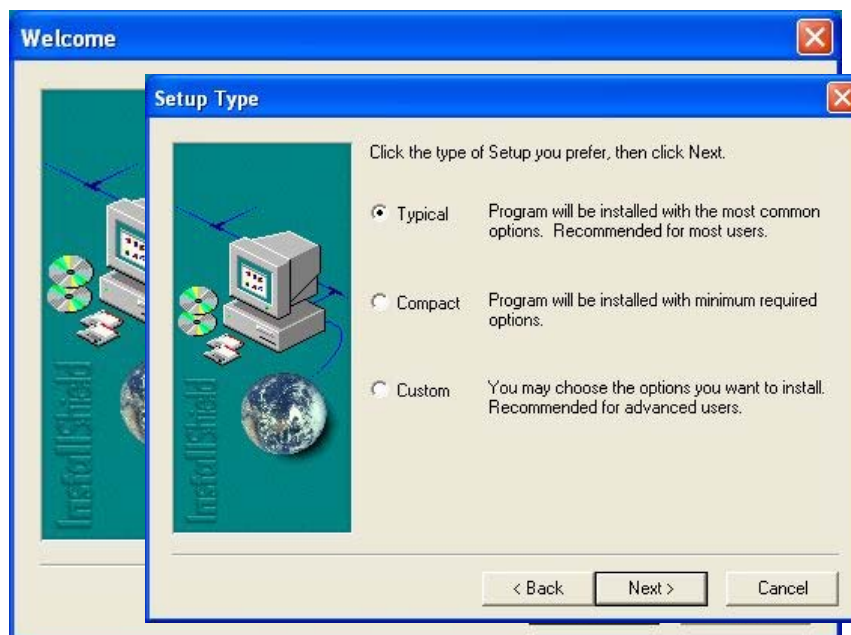


## JENI 3 – 05 Modul Praktikum SQL dan JDBC

## 4. Tentukan letak folder instalasi

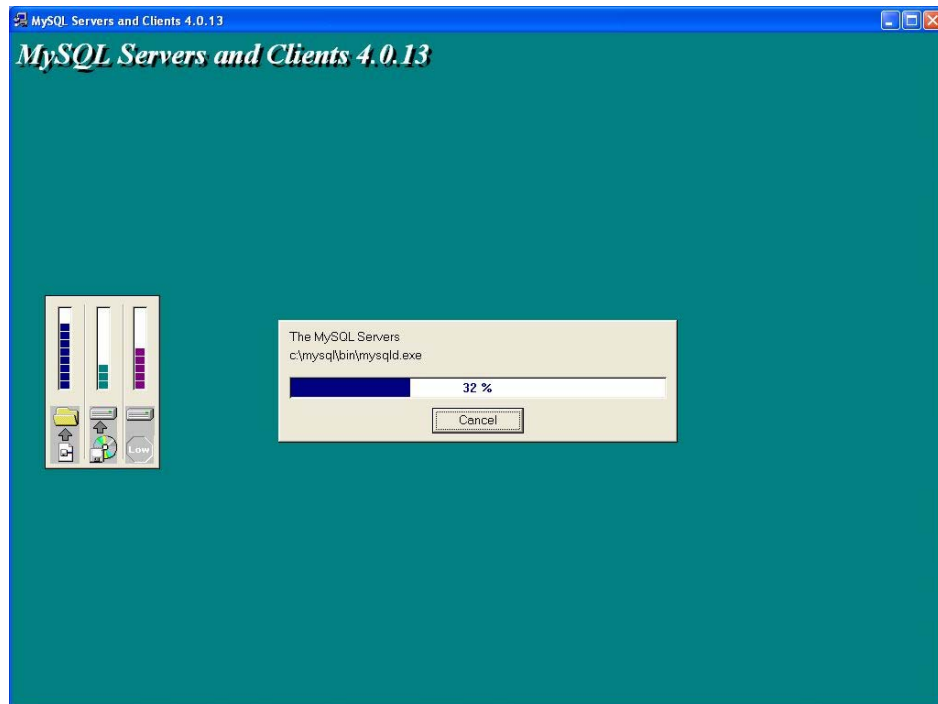


## 5. Pilih Typical (rekomen untuk pengguna normal)



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

## 6. Instalasi sedang berlangsung

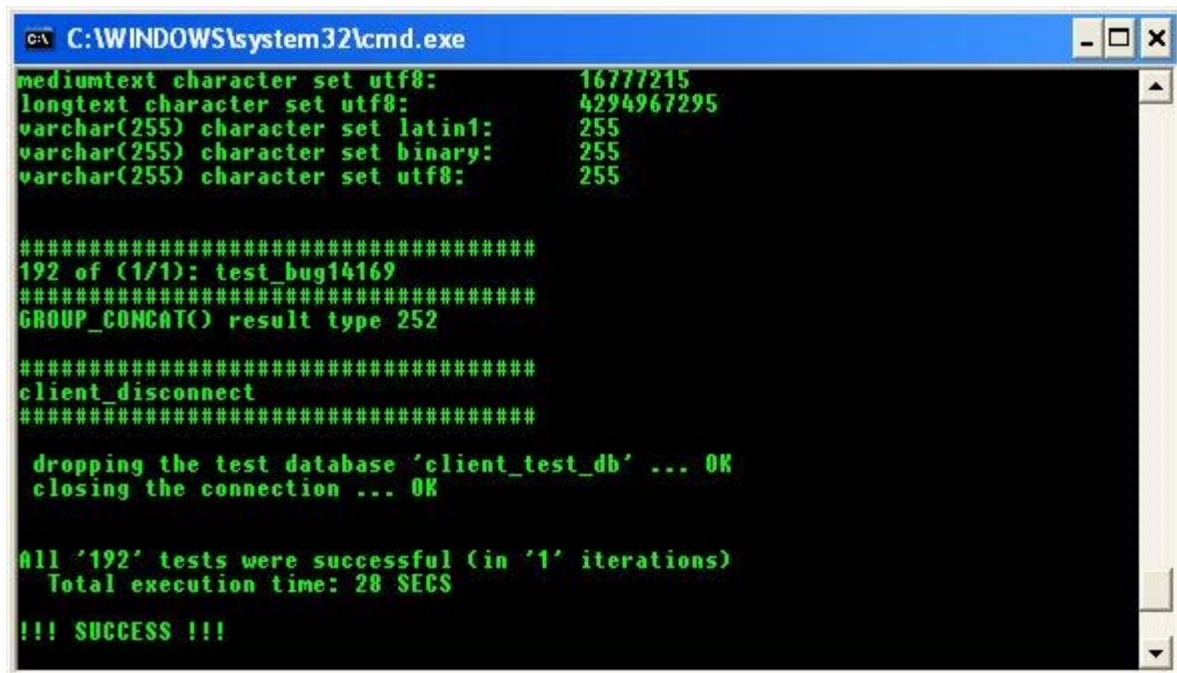


## 7. Instalasi selesai



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

8. Buka Windows Explorer anda
9. Jalankan file mysqld-nt.exe pada direktori bin (C:\mysql\bin)
10. Lakukan test mysql dengan menjalankan mysqltest.exe atau mysql\_client\_test.exe -h localhost -P 3306 -u root



```
C:\WINDOWS\system32\cmd.exe
mediumtext character set utf8:      16777215
longtext character set utf8:        4294967295
varchar(255) character set latin1:   255
varchar(255) character set binary:   255
varchar(255) character set utf8:     255

#####
192 of (1/1): test_bug14169
#####
GROUP_CONCAT() result type 252

#####
client_disconnect
#####

dropping the test database 'client_test_db' ... OK
closing the connection ... OK

All '192' tests were successful (in '1' iterations)
Total execution time: 28 SECS

!!! SUCCESS !!!
```

11. Jika berhasil akan muncul pesan

Jika tidak ada koneksi anda perlu membuat file my.ini dan ana letakkan dalam direktori [C:\Windows](#) atau [C:\Winnt](#); Berikut konfigurasinya:

```
[mysqld]
basedir=C:/mysql
#bind-address=127.0.0.1
datadir=C:\mysql\data
#language=C:/xampp/mysql/share/your language directory
#slow query log#=
#tmpdir#=
#port=3306
#set-variable=key_buffer=16M
[WinMySQLAdmin]
Server=C:/mysql/bin/mysqld-nt.exe
user=root
password=password
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### Percobaan 2 – Penggunaan Sintak Sederhana SQL (Address Book) :

#### **Syntax sederhana MySQL**

Praktekkan syntaks berikut dalam MySQL command prompt/ MySQL GUI.

#### **Login ke mysql**

- Masuk ke direktori bin mysql
- ketikkan **mysql** atau **mysql -u root**
- Jika terkoneksi, prompt akan berubah seperti berikut: **mysql>**

#### **Melihat Database yang ada**

Syntaks: **Show Databases;**

#### **Menggunakan Database**

Syntaks: **USE nama\_database;**

*Percobaan:*

**mysql>** USE mysql;

#### **Melihat Tabel dalam suatu database;**

Syntaks: **SHOW TABLES;**

*Percobaan:*

**mysql>** USE mysql;

**mysql>** Show tables;

#### **Membuat Database Baru**

Syntaks: **CREATE DATABASE nama\_database;**

*Percobaan:*

**mysql>** create database jeni\_sql;

**mysql>** show databases;

Maka database jeni\_sql akan muncul dalam list

#### **Membuat table**

Syntaks: **CREATE TABLE nama\_tabel (field1 datatype1, field2 datatype2, .....);**

*Percobaan:*

**mysql>** CREATE TABLE 'friends' (  
'id' int(10) unsigned NOT NULL auto\_increment,  
'Nama' varchar(45) NOT NULL,  
'Email' varchar(100) NOT NULL,  
'Website' varchar(55) NOT NULL,  
'Alamat' varchar(255) NOT NULL,  
PRIMARY KEY ('id'))



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

)

**INSERT**

Format:

**INSERT INTO table-name VALUES(value1, value2, ...)****INSERT INTO table-name (field1, field2, ....) VALUES (value1, value2, ...)***Percobaan:*

1. Memasukkan data sesuai urutan field.

```
mysql> INSERT INTO friends VALUES (1, 'Agung Pribadi',  
'agung@negaraku.com', 'http://www.agung.info', 'Surabaya, Indonesia');
```

2. Memasukkan data dengan mendefinisikan secara eksplisit field yang akan diisi.

```
mysql> INSERT INTO 'friends' ('id', 'Nama', 'Email', 'Website', 'Alamat') VALUES  
(2, 'Wicaksono', 'wicak@kerja-bhakti.co.id', '', 'Jakarta, Indonesia');
```

3. Memasukkan data secara eksplisit tanpa id, maka id akan diincrement secara otomatis.

```
mysql> INSERT INTO 'friends' ('Nama', 'Email', 'Website', 'Alamat') VALUES ('Jeni  
Wulandari', 'jeni@jardiknas.org', 'http://jeni.jardiknas.com', '');
```

4. Memasukkan data secara normal dengan id yang agak bernilai besar dari yang sudah ada.

```
mysql> INSERT INTO 'friends' VALUES (10, 'Prasetyo', 'setyo@kampoes.ac.id',  
'http://setyo.kampoes.ac.id', 'Malang, Indonesia');
```

5. Memasukkan data secara eksplisit mendefinisikan field yang akan diisi kecuali id. Maka id akan diincrement berdasarkan id terbesar yang ada di tabel.

```
mysql> INSERT INTO 'friends' ('Nama', 'Email', 'Website', 'Alamat') VALUES  
(Thomas Crust, 'crust@gebraucht.de', 'http://crust.gebraucht.de', 'Switzerland');
```

```
mysql> INSERT INTO 'friends' ('id', 'Nama', 'Email', 'Website', 'Alamat') VALUES  
(8, 'Alex', 'alex@imitasi.com', 'http://www.jualan.com', 'Bandung');
```

```
mysql> INSERT INTO 'friends' ('id', 'Nama', 'Email', 'Website', 'Alamat') VALUES  
(9, 'Alex', 'alex@gerbang.com', 'http://www.masadepan.com', 'Nusakambangan');
```

```
mysql> INSERT INTO 'friends' ('Nama', 'Email', 'Website', 'Alamat') VALUES  
(Narita S, 'narita@kampoes.ac.id', 'http://narita.kampoes.ac.id', 'Surabaya');
```

6. Masukkan data diri anda;
7. Masukkan 2 data rekan disamping anda (kanan & kiri anda).

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### UPDATE

Format: **UPDATE** nama\_tabel SET field1=value1, field2=value2,..... WHERE condition(s)

*Percobaan:*

```
mysql> UPDATE friends SET nama='Andreas', website='http://www.latihan.com'
WHERE id=2;
```

```
mysql> UPDATE friends SET email='aku@rumahku.net', alamat='Banjarmasin'
WHERE nama='Prasetyo';
```

### DELETE

Format:

**DELETE FROM** table-name WHERE condition(s)

*Percobaan:*

```
mysql> DELETE FROM friends WHERE id=3;
```

```
mysql> DELETE FROM friends WHERE nama='Alex' AND
alamat='Nusakambangan';
```

### SELECT

Format:

**SELECT** columns FROM tablename WHERE condition(s)

*Percobaan:*

```
mysql> SELECT * FROM friends;
```

```
mysql> SELECT * FROM friends WHERE id=10;
```

```
mysql> SELECT * FROM friends WHERE id<10;
```

```
mysql> SELECT * FROM friends WHERE id>10;
```

```
mysql> SELECT * FROM friends WHERE alamat LIKE '%Ba%';
```

```
mysql> SELECT * FROM friends WHERE alamat LIKE '%Ba%';
```

```
mysql> SELECT * FROM friends WHERE id<10 AND alamat LIKE '%Ba%';
```

```
mysql> SELECT COUNT(*) FROM friends WHERE alamat LIKE '%Ba%';
```

```
mysql> SELECT COUNT(*) FROM friends;
```

```
mysql> SELECT MAX(id) FROM friends;
```

```
mysql> SELECT MIN(id) FROM friends;
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### **DROP**

Untuk menghapus Tabel atau Database

Format:

**DROP TABLE** nama\_tabel;

**DROP DAATABASE** nama\_database;

### **Percobaan 3 – Membuat Project Java Database Connectivity (JDBC):**



Tujuan section ini adalah membuat project untuk aplikasi yang menggunakan JDBC.

- Buat project web baru dengan nama **sql\_jdbc**



Tip

File → New Project, pilih category Web → Web Application, klik Next  
Ketikkan nama project, klik Finish

- Tambahkan librari **mysql-connector-java\_xx.jar** ke dalam CLASSPATH aplikasi. Dapat dilakukan dengan 2 cara:
- Secara manual, kopikan librari tersebut ke dalam folder WEB-INF/lib.
- Melalui Netbeans IDE:  
Pada tab **Projects**, klik kanan pada **Libraries**, pilih **Add JAR/Folder**, dan pilih file librari yang akan ditambahkan.  
Ketika proses Build Project, file tadi akan dikopikan ke folder WEB-INF/lib oleh Netbeans secara otomatis.

### **Percobaan 4 – Koneksi ke Database Melalui class DriverManager :**



Tujuan section ini adalah mencontohkan koneksi ke database melalui class DriverManager dalam aplikasi Java.

### **Langkah-langkahnya:**

Mengetes koneksi DriverManager dalam aplikasi Servlet :

- Buat link dalam file index.jsp:



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

## &lt;h4&gt;Driver Manager&lt;/h4&gt;

&lt;a href="TestDMConnection"&gt;Test Koneksi Driver Manager (Servlet)&lt;/a&gt;&lt;br/&gt;

&lt;a href="TestJspDMConnection.jsp"&gt;Test Koneksi Driver Manager (JSP)&lt;/a&gt;&lt;br/&gt;

- Pada Source Packages Projects Explorer, buat class dengan nama **TestServletDMConnection**, dan isikan kode berikut:

```
import java.io.*;
import java.util.Enumeration;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.net.*;

public class TestServletDMConnection extends HttpServlet{
    Connection theConnection;
    private ServletConfig config;

    public void init(ServletConfig config)
        throws ServletException{
        this.config=config;
    }

    public void service (HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        HttpSession session = req.getSession(true);
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        out.println("<HTML><HEAD><TITLE>Email List.</TITLE>");
        out.println("</HEAD>");
        out.println("<BODY bgColor=blanchedalmond text=#008000 topMargin=0>");
        out.println("<P align=center><FONT face=Helvetica><FONT color=fuchsia style=\"BACKGROUND-COLOR: white\"><BIG><BIG>List of Address Book.</BIG></BIG></FONT></P>");
        out.println("<P align=center>");
        out.println("<TABLE align=center border=1 cellPadding=1 cellSpacing=1 width=\"75%\">");
        out.println("<TR>");
        out.println("<TD>Name</TD>");
        out.println("<TD>E-mail</TD>");
        out.println("<TD>Website</TD>");
        out.println("<TD>Alamat</TD>");
        out.println("</TR>");

        try{
            Class.forName("com.mysql.jdbc.Driver");
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```
theConnection =
DriverManager.getConnection("jdbc:mysql://localhost/jeni_sql","root","password");
Statement theStatement=theConnection.createStatement();

ResultSet theResult=theStatement.executeQuery("select * from
friends"); //Select all records from emaiillists table.
while(theResult.next()) //Fetch all the records and print in table
{
    out.println();
    out.println("<TR>");
    out.println("<TD>" + theResult.getString("nama") +
"</TD>");
    out.println("<TD>" + theResult.getString("email") +
"</TD>");
    String s=theResult.getString("website");
    out.println("<TD><a href=" + s + ">" + s + "</a></TD>");
    out.println("<TD>" + theResult.getString("alamat") +
"</TD>");
    out.println("</TR>");
}

theResult.close();//Close the result set
theStatement.close();//Close statement
theConnection.close(); //Close database Connection
} catch (Exception e) {
    out.println(e.getMessage()); //Print trapped error.
}

    out.println("</TABLE></P>");
    out.println("<P>&nbsp;</P></FONT></BODY></HTML>");
}
public void destroy(){
}
}
```

- Buka file **web.xml**, tambahkan konfigurasi servlet berikut:

```
<web-app>
  <servlet>
    <servlet-name>TestDM</servlet-name>
    <servlet-class>TestServletDMConnection</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>TestDM</servlet-name>
    <url-pattern>/TestDMConnection</url-pattern>
  </servlet-mapping>
</web-app>
```

- Run → Run Project

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

Mengetes koneksi DriverManager dalam aplikasi JSP:

- Buat file JSP dengan nama **TestJspDMConnection.jsp** dalam folder Web Pages

```
<%@ page import="java.sql.*" %>
<%
String connectionURL = "jdbc:mysql://localhost:3306/jeni_sql";
Connection connection = null;
Statement statement = null;
ResultSet rs = null;
%>
<html><body>
<p align=center>
<font face="Helvetica" color="fuchsia" style="background-color:
white"><big><big>List of Address Book.</big></big></font></p>
<p align=center>
<table align=center border=1 cellPadding=1 cellSpacing=1 width="75%">
<tr align="center">
<td><b>Name</b></td>
<td><b>E-mail</b></td>
<td><b>Website</b></td>
<td><b>Alamat</b></td>
</tr>
<%
Class.forName("com.mysql.jdbc.Driver").newInstance();
connection = DriverManager.getConnection(connectionURL, "root", "password");
statement = connection.createStatement();
rs = statement.executeQuery("SELECT * FROM friends");
while (rs.next()) {
    out.println("<tr>");
    out.println("<td>" + rs.getString("nama") + "</td>");
    out.println("<td>" + rs.getString("email") + "</td>");
    String s=rs.getString("website");
    out.println("<td><a href=" + s + ">" + s + "</a></td>");
    out.println("<td>" + rs.getString("alamat") + "</TD>");
    out.println("</tr>");
}
rs.close();
statement.close();
connection.close();
%>
</table>
</body>
</html>
```

- Run → Run Project

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### Percobaan 5 – Konfigurasi DataSource JNDI secara Konvensional :



Tujuan section ini adalah menunjukkan bagaimana melakukan konfigurasi DataSource JNDI secara konvensional.

#### Langkah-langkahnya:

a) Buka file **server.xml** dalam direktori **\$CATALINA\_HOME/conf/** dan tambahkan kode berikut:

```
<Host ....>
  <Context crossContext="true" docBase="C:/programme/app/sql_jdbc/build/web"
    path="/sql_jdbc">
    <Resource auth="Container" driverClassName="com.mysql.jdbc.Driver"
      factory="org.apache.commons.dbcp.BasicDataSourceFactory"
      name="jdbc/sql_jdbc" password="password" scope="Shareable"
      type="javax.sql.DataSource"
      url="jdbc:mysql://localhost:3306/jeni_sql?autoReconnect=true"
      username="root"/>
    </Context>
  </Host>
```

dimana,

docBase = Path direktori dimana aplikasi web kita berada.

path = path aplikasi saat diakses dari web browser

driverClassName = class driver connector ke database

name = nama DataSource JNDI

url = alamat pengaksesan database

username = nama user database

password = password user database

b) Buka file **web.xml** aplikasi anda (dalam folder WEB-INF) dan tambahkan kode berikut:

```
<resource-ref>
  <res-ref-name>jdbc/sql_jdbc</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

dimana,

res-ref-name = nama DataSource JNDI

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

c) Buka file Context.xml aplikasi anda dan tambahkan kode berikut:

```
<ResourceLink global="jdbc/sql_jdbc" name="jdbc/sql_jdbc"
type="javax.sql.DataSource"/>
```

- d) Aplikasi Java untuk dapat melakukan koneksi ke database MySQL membutuhkan driver konektor untuk Java. MySQL menyediakan driver konektor secara gratis dan bisa didownload di <http://dev.mysql.com/downloads/connector/j/3.0.html>. Sesuaikan aja dengan versi MySQL server yang kompatibel.
- e) Jika sudah mendownloadnya, anda ekstrak file mysql-connector-java-3.xx.zip dan ambil **mysql-connector-java.jar**. Kopikan file jar ini ke dalam folder lib aplikasi anda atau kopikan ke direktori **\$CATALINA\_HOME/common/lib** atau **\$CATALINA\_HOME/lib** kalau tidak menyediakan folder common. Sangat disarankan untuk meletakkan file mysql-connector-java.jar dalam direktori common/lib Tomcat server.
- f) Selain librari itu tomcat menggunakan DataBase Connection Pool (DBCP) untuk menangani basic data source yang mendukung JDBC 2.0. Commons-DBCP sendiri membutuhkan librari lain yaitu **Jakarta-Commons-Dbcp.jar**, **Jakarta-Common-Collections.jar** dan **Jakarta-Common-pool.jar**.
- g) Restart Tomcat Server anda.

### Percobaan 6–Konfigurasi DataSource JNDI dengan Tomcat Administration Tool :



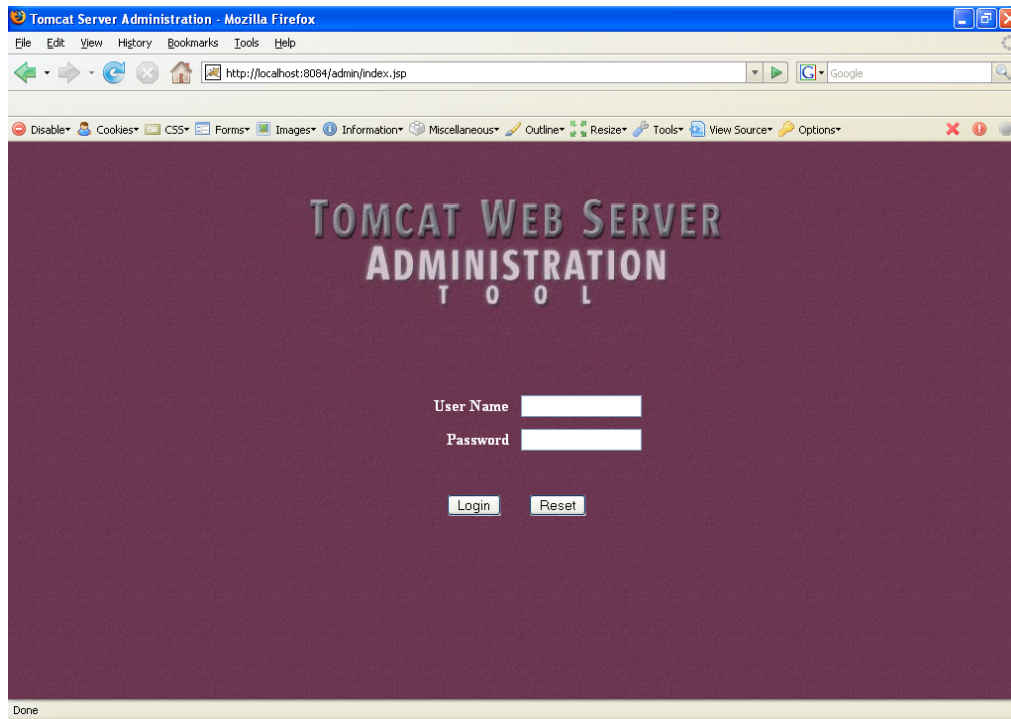
Info

Tujuan section ini adalah menunjukkan bagaimana melakukan konfigurasi DataSource JNDI melalui Tomcat Administration Tool.

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

**Langkah-langkahnya:**

- 1) Akses halaman <http://localhost:8084/admin>



- 2) Login sebagai admin.

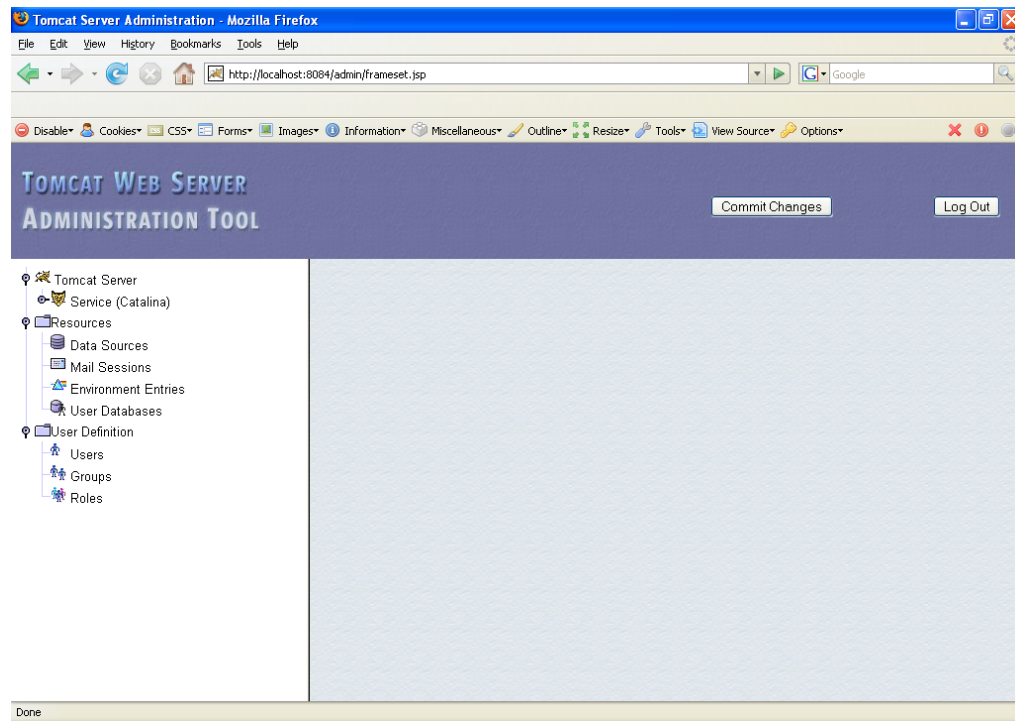
Jika anda belum mengetahui username dan password administrator, bisa anda lihat dalam file tomcat-users.xml di dalam folder \$TOMCAT\_HOME/conf. Lihat username dan password yang memiliki role manager/admin. Jika tidak anda temukan role manager atau admin bisa anda tambahkan kode berikut:

```
<role rolename="manager"/>
<role rolename="admin"/>
<user username="ide" password="netbeans" roles="manager,admin"/>
```



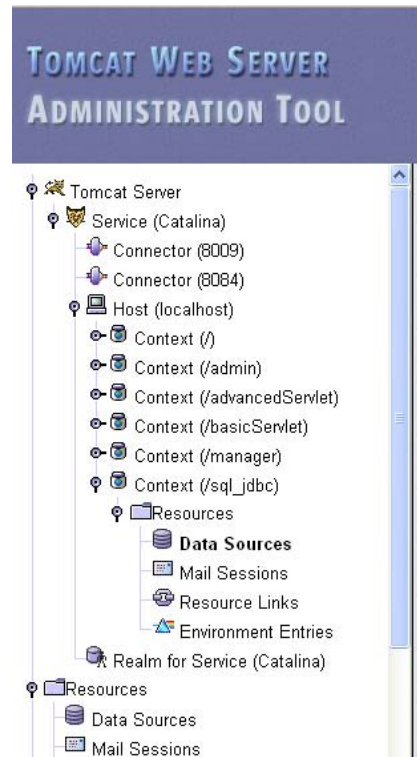
## JENI 3 – 05 Modul Praktikum SQL dan JDBC

3) Jika login anda berhasil, akan anda dapatkan halaman seperti berikut:



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

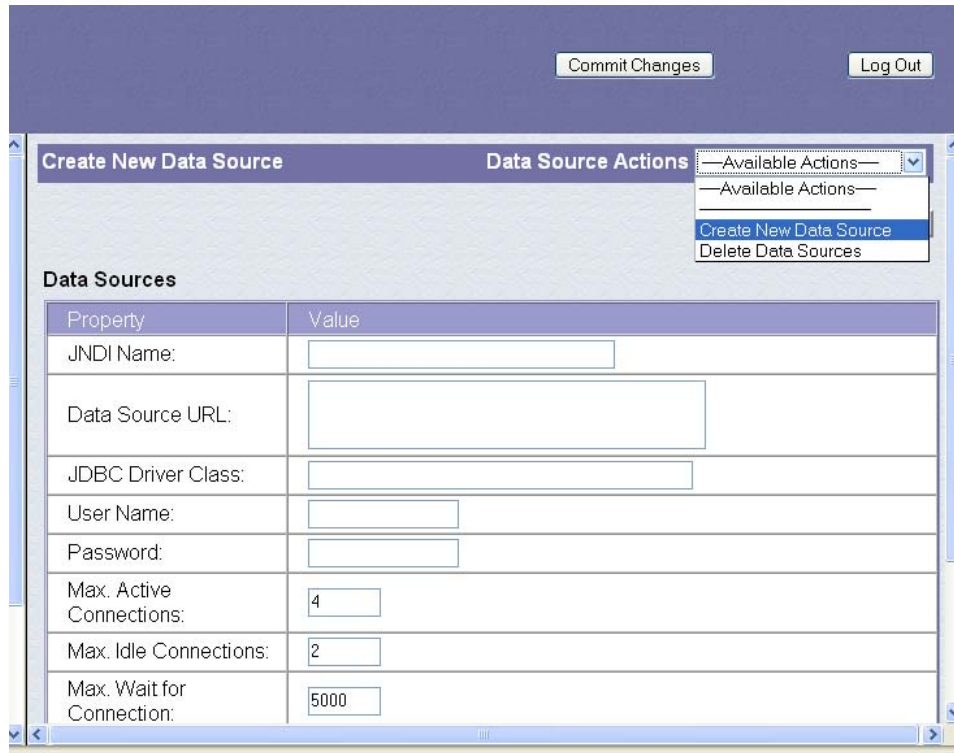
- 4) Untuk membuat DataSource dapat anda lakukan: klik Tomcat Server → Service (Catalina) → Host (localhost) → Context (/sql\_jdbc) → Resources → Data Sources, sehingga tree yang terbuka seperti terlihat pada gambar berikut:



- 5) Pada menu Data Source Actions pilih Create New Data Source. Isikan data-data berikut:

JNDI Name	= jdbc/sql_jdbc
Data Source URL	= jdbc:mysql://localhost:3036/jeni_sql?autoReconnect=true
JDBC Driver Class	= com.mysql.jdbc.Driver
User Name	= root (username user database anda)
Password	= password (password user database anda)
Max Active Connections	= 4
Max Idle Connections	= 2
Max Wait for Connection	= 5000
Validation Query	=

## JENI 3 – 05 Modul Praktikum SQL dan JDBC



Property	Value
JNDI Name:	<input type="text"/>
Data Source URL:	<input type="text"/>
JDBC Driver Class:	<input type="text"/>
User Name:	<input type="text"/>
Password:	<input type="text"/>
Max. Active Connections:	<input type="text" value="4"/>
Max. Idle Connections:	<input type="text" value="2"/>
Max. Wait for Connection:	<input type="text" value="5000"/>

6) Setelah selesai, klik Save. Jika berhasil menyimpan JNDI name akan muncul di halaman Data Source:

Pembuatan Data Source JNDI cukup.

7) Selanjutnya tambahkan resource reference pada file web.xml aplikasi anda.

```
<resource-ref>
  <res-ref-name>jdbc/sql_jdbc</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

dimana, res-ref-name = JNDI name

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### Percobaan 7 – Koneksi ke Database melalui Class DataSource :



Tujuan section ini adalah menunjukkan bagaimana membuat koneksi ke database melalui class DataSource.

#### Langkah-langkahnya:

##### 1) Mengetes koneksi via JNDI menggunakan Servlet

###### a) Buat link dalam file index.jsp:

```
<h4>Data Source - JNDI</h4>
<a href="TestJNDI">Test Koneksi DataSource JNDI (Servlet)</a><br/>
<a href="TestJspJNDIConnection">Test Koneksi DataSource JNDI
(JSP)</a><br/>
```

###### b) Buat class Java dengan nama **TestJNDIConnection**, dan berikan kode berikut:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.sql.*;
import javax.naming.*;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class TestJNDIConnection extends HttpServlet{
    Connection theConnection;
    private ServletConfig config;

    public void init(ServletConfig config)
    throws ServletException{
        this.config=config;
    }

    public void service(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
        Connection connection = null;
        DataSource ds = null;

        HttpSession session = req.getSession(true);
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```
out.println("<HTML><HEAD><TITLE>Email List.</TITLE>");
out.println("</HEAD>");
out.println("<BODY bgcolor=blanchedalmond text=#008000
topMargin=0>");
    out.println("<P align=center><FONT face=Helvetica><FONT
color=fuchsia style=\"BACKGROUND-COLOR: white\"><BIG><BIG>List of
Address Book.</BIG></BIG></FONT></P>");
    out.println("<P align=center>");
    out.println("<TABLE align=center border=1 cellPadding=1
cellSpacing=1 width=\"75%\">");
    out.println("<TR>");
    out.println("<TD>Name</TD>");
    out.println("<TD>E-mail</TD>");
    out.println("<TD>Website</TD>");
    out.println("<TD>Alamat</TD>");
    out.println("</TR>");

try{
    try {
        Context ctx = new InitialContext();
        Context envCtx ;

        envCtx = (Context) ctx.lookup("java:comp/env");
        ds = (DataSource)envCtx.lookup("jdbc/sql_jdbc");
    } catch (NamingException ne) {
        System.out.println("Naming Exception" + ne);
    }

    if (ds == null){
        System.out.println("Data Source Null");
    }
    connection = ds.getConnection();

    Statement theStatement=theConnection.createStatement();

    ResultSet theResult=theStatement.executeQuery("select * from
friends"); //Select all records from emailists table.
    while(theResult.next()) //Fetch all the records and print in
table
    {
        out.println();
        out.println("<TR>");
        out.println("<TD>" + theResult.getString("nama") +
"</TD>");
        out.println("<TD>" + theResult.getString("email") +
"</TD>");
        String s=theResult.getString("website");
        out.println("<TD><a href=\"" + s + ">" + s + "</a></TD>");
        out.println("<TD>" + theResult.getString("alamat") +
"</TD>");
        out.println("</TR>");
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```

    }

    theResult.close();//Close the result set
    theStatement.close();//Close statement
    theConnection.close(); //Close database Connection
} catch (Exception e) {
    out.println(e.getMessage()); //Print trapped error.
}
out.println("</TABLE></P>");
out.println("<P>&nbsp;</P></FONT></BODY></HTML>");
}
public void destroy() {
}
}
}

```

c) Buka file **web.xml**, tambahkan ampping servletnya:

```

<servlet>
    <servlet-name>TestJNDI</servlet-name>
    <servlet-class>TestJNDIConnection</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>TestJNDI</servlet-name>
    <url-pattern>/TestJNDI</url-pattern>
</servlet-mapping>

```

d) Run → Run Project

## 2) Mengetes koneksi JNDI dengan JSP:

a) Buat file JSP TestJNDIConnection.jsp dengan kode berikut:

```

<%@ page import="java.sql.*" %>
<%@ page import="javax.naming.*" %>
<%@ page import="javax.sql.*" %>

<%
String connectionURL = "jdbc:mysql://localhost:3306/jeni_sql";
DataSource ds = null;
Connection connection = null;
Statement statement = null;
ResultSet rs = null;
%>
<html><body>
<p align=center>
<font face="Helvetica" color="fuchsia" style="background-color:
    white"><big><big>List of Address Book.</big></big></font></p>
<p align=center>
<table align=center border=1 cellPadding=1 cellSpacing=1 width="75%">

```



## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```
<tr align="center">
<td><b>Name</b></td>
<td><b>E-mail</b></td>
<td><b>Website</b></td>
<td><b>Alamat</b></td>
</tr>
<%

/* Used for DataSource JNDI connection
*/

    try {
        Context ctx = new InitialContext();
        Context envCtx ;

        envCtx = (Context) ctx.lookup("java:comp/env");
        ds = (DataSource)envCtx.lookup("jdbc/sql_jdbc");
    } catch (NamingException ne) {
        System.out.println("Naming Exception" + ne);
    }
    if (ds == null){
        System.out.println("Data Source Null");
    }

    connection = ds.getConnection();

    statement = connection.createStatement();
    rs = statement.executeQuery("SELECT * FROM friends");
    while (rs.next()) {
        out.println("<tr>");
        out.println("<td>" + rs.getString("nama") + "</td>");
        out.println("<td>" + rs.getString("email") + "</td>");

        String s=rs.getString("website");

        out.println("<td><a href=" + s + ">" + s + "</a></td>");

        out.println("<td>" + rs.getString("alamat") + "</TD>");
        out.println("</tr>");
    }

//add more function

rs.close();
statement.close();
connection.close();
%>
</table>
</body>
</html>
```

b) Run → Run Project

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

### Percobaan 8 – Operasi INSERT, UPDATE, DELETE dan SELECT :



Tujuan section ini adalah menunjukkan penggunaan operasi SQL INSERT, UPDATE, DELETE dan SELECT dalam aplikasi.

#### Langkah-langkahnya:

1) Buat link dalam file index.jsp

```
<h4>Operasi - INSERT, UPDATE, DELETE, SELECT</h4>
<a href="OperasiSQL">Test Operasi SQL dengan koneksi JNDI (Servlet)</a><br/>
```

2) Buat class Java dengan nama **OperasiSQL**, dan berikan kode berikut:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.sql.*;
import javax.naming.*;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class OperasiSQL extends HttpServlet{
    Connection theConnection = null;
    DataSource ds = null;
    private ServletConfig config;

    public void init(ServletConfig config)
    throws ServletException{
        this.config=config;
    }

    public void service(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
        String sql = "";
        Statement theStatement =null;
        ResultSet theResult = null;
        HttpSession session = req.getSession(true);
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        try {
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```

        Context ctx = new InitialContext();
        Context envCtx ;

        envCtx = (Context) ctx.lookup("java:comp/env");
        ds = (DataSource)envCtx.lookup("jdbc/sql_jdbc");
    } catch (NamingException ne) {
        System.out.println("Naming Exception" + ne);
    }

    out.println("<HTML><HEAD><TITLE>Emai List.</TITLE>");
    out.println("</HEAD>");
    out.println("<BODY bgColor=blanchedalmond text=#008000
topMargin=0>");
    out.println("<P align=center><FONT face=Helvetica><FONT color=fuchsia
style=\"BACKGROUND-COLOR: white\"><BIG><BIG>List of Address
Book.</BIG></BIG></FONT></P>");
    out.println("<P align=center>");
    out.println("<TABLE align=center border=1 cellPadding=1 cellSpacing=1
width=\"75%\">");

    if (ds == null){
        System.out.println("Data Source Null");
    }
    try {
        sql = "INSERT INTO friends (Nama, Email, Website, Alamat) VALUES
('Narita S', 'narita@kampos.ac.id', 'http://narita.kampos.ac.id',
'Surabaya');";
        theConnection = ds.getConnection();
        theStatement = theConnection.createStatement();
        System.out.println(sql);
        if (theStatement.executeUpdate(sql) != 0){
            out.println("<TR>");
            out.println("<TD colspan='4'>"+sql+" inserted
successfully.</TD>");
            out.println("</TR>");
        }else{
            out.println("<TR>");
            out.println("<TD colspan='4'>"+sql+" <font
color='red'>failed</font>.</TD>");
            out.println("</TR>");
        }

        theConnection = ds.getConnection();
        theStatement = theConnection.createStatement();
        sql = "UPDATE friends SET email='aku@rumahku.net',
alamat='Banjarmasin' WHERE nama='Wicaksono';";
        if (theStatement.executeUpdate(sql) != 0){
            out.println("<TR>");
            out.println("<TD colspan='4'>"+sql+" updated
successfully.</TD>");
            out.println("</TR>");
        }else{

```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```
        out.println("<TR>");
        out.println("<TD colspan='4'>" + sql + " <font
color='red'>failed</font>.</TD>");
        out.println("</TR>");
    }
    theConnection = ds.getConnection();
    theStatement = theConnection.createStatement();
    //sql = "DELETE FROM friends WHERE nama = 'Narita S'";
    sql = "DELETE FROM friends WHERE id = 3";
    if (theStatement.executeUpdate(sql) != 0) {
        out.println("<TR>");
        out.println("<TD colspan='4'>" + sql + " deleted
successfully.</TD>");
        out.println("</TR>");
    } else {
        out.println("<TR>");
        out.println("<TD colspan='4'>" + sql + " <font
color='red'>failed</font>.</TD>");
        out.println("</TR>");
    }

    } catch (SQLException ex) {
        ex.printStackTrace();
    }

    sql = "SELECT * FROM friends";
    out.println("<TR>");
    out.println("<TD colspan='4'>" + sql + "</TD>");
    out.println("</TR>");

    out.println("<TR>");
    out.println("<TD>Name</TD>");
    out.println("<TD>E-mail</TD>");
    out.println("<TD>Website</TD>");
    out.println("<TD>Alamat</TD>");
    out.println("</TR>");

    try {
        theResult = theStatement.executeQuery(sql); //Select all records
from friends table.
        while(theResult.next()) //Fetch all the records and print in
table
        {
            out.println();
            out.println("<TR>");
            out.println("<TD>" + theResult.getString("nama") + "</TD>");
            out.println("<TD>" + theResult.getString("email") + "</TD>");
            String s = theResult.getString("website");
            out.println("<TD><a href=" + s + ">" + s + "</a></TD>");
            out.println("<TD>" + theResult.getString("alamat") +
"</TD>");
            out.println("</TR>");
        }
    }
```

## JENI 3 – 05 Modul Praktikum SQL dan JDBC

```
    }

    theResult.close();//Close the result set
    theStatement.close();//Close statement
    theConnection.close(); //Close database Connection
} catch (Exception e) {
    out.println(e.getMessage()); //Print trapped error.
}
out.println("</TABLE></P>");
out.println("<P>&nbsp;</P></FONT></BODY></HTML>");
}
public void destroy(){
}
}
```

3) Buat mapping servlet dalam file **web.xml** seperti berikut:

```
<servlet>
  <servlet-name>OperasiSQL</servlet-name>
  <servlet-class>OperasiSQL</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>OperasiSQL</servlet-name>
  <url-pattern>/OperasiSQL</url-pattern>
</servlet-mapping>
```

4) Run → Run Project