Bab 10

Mengakses Database Lanjut

POKOK BAHASAN:

- Metadata
- DatabaseMetadata
- ResultSetMetaData
- Memproses result

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

- 1. Menggunakan metadata pada resultset.
- 2. Mengetahui cara menggunakan statement dan method pada statement.
- 3. Mengetahui cara melihat data.
- 4. Mengetahui cara mengupdate data.
- 5. Mengetahui cara menghapus data.

11.1 Metadata

Java menyediakan class DatabaseMetaData dan ResultSetMetaData untuk memberikan informasi berupa scema database, jumlah kolom, nama kolom, nama produk, nama driver, dll.

11.2 DatabaseMetaData

Kita dapat menampilkan informasi database dengan cara memanggil method getMetaData() dari Connection. Method getMetaData() akan

mengembalikan object DatabaseMetaData. Berikut ini beberapa contoh method yang ada pada DatabaseMetaData:

- getDataBaseProductName()
- getDataBaseProductVersion()
- getDriverName()
- getDriverVersion()

Contoh penggunan :

DatabaseMetadata dbMetaData = connection.getMetaData();

String productName = dbMetaData.getDatabaseProductName();

String productVersion = dbMetaData.getDatabaseProductVersion();

11.3 ResultSetMetaData

ResultSetMetaData dapat digunakan untuk melihat informasi pada hasil akses ke tabel database. Dari obyek ResultSet, kita dapat mengakses ResultSetMetaData untuk mendapatkan informasi mengenai nama kolom, tipe data kolom, jumlah kolom, dsb. Beriku ini adalah beberapa method dari obyek ResultSetMetaData:

- getColumnCount() untuk mendapatkan banyaknya kolom pada suatu query, index kolom dimulai dari kolom ke 1.
- getColumnName() untuk mendapatkan nana kolom.
- getColumnType() untuk mendapatkan tipe data kolom.
- isSearchable() untuk digunakan pada klausa WHERE
- isNullable() untuk mengetahui apakah nilai null diperbolehkan pada suatu kolom.

ResultSetMetaData tidak mempunya method yang digunakan untuk mendapatkan informasi jumlah baris dalam suatu tabel. Sehingga sebagai solusi kita bisa memanggil method next() pada ResultSet yang mengembalikan nilai true jika isi tabel masih

ada dan akam mengembalikan nilai false bila isi tabel sudah diambil semua. Berikut ini adalah contoh penggunaan ResultSetMetaData :

```
ResutSet rs = statement.executeQuery(query);
ResultSetMetaData rsData = rs.getMetaData();
int columnCount = rsData.getColumnCount()
```

11.4 Memproses result

Untuk memproses result kita bisa menggunakan method next() pada objek ResultSet untuk mendapatkan result per satu baris. Selama data masih ada method next() akan mengembalikan nilai true dan jika sudah tidak ada akan mengembalikan nilai false.

Contoh:

```
String nrp;
String nama;
while (rs.next()){
    nrp=rs.getString(1);
    nama=rs.getString(2);
    System.out.println("NRP : " +nrp);
    System.out.println("NAMA : " +nama);
    System.out.println("-----");
}
```

Kolom pertama mempunyai index 1 bukan 0. Objek ResultSet otomatis akan ditutup bila ada objek ResultSet baru. ResultSet memberikan bermacam-macam method getXxx() dengan parameter indek kolom atau nama kolom dan mengembalikan data. Beberapa method yang ada pada ResultSet adalah sebagai berikut:

- findColumn() untuk mendapatkan index (integer value) berdasarkan nama kolom.
- getMetaData() untuk meretrieve informasi mengenai ResultSet, returns object
 ResultSetMetaData.
- wasNull() untuk mengetahui apakah getXxx() menghasilkan SQL null.

11.5 Percobaan

1. Buat program untuk melihat isi data tabel FILM seperti pada Listing 10.1

```
<%@ page import="java.sql.*" %>
<%
 Connection con=null;
 String dbname="jdbc:odbc:DSFilm";
 String status="";
 ResultSet rs = null;
 try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   con=DriverManager.getConnection(dbname,"","");
   if (con==null)
     status = "gagal";
   else
     status = "berhasil";
 }catch(ClassNotFoundException ex) {
     status = "Driver Error";
 }catch(SQLException ex) {
     status = "gagal";
 }
 Statement st = con.createStatement();
 String kueri = "SELECT * FROM FILM";
 rs = st.executeQuery(kueri);
응>
<html>
<head>
<title>Isi Tabel Film</title>
</head>
<body>
<strong><font size="6">DATA FILM</font>
</strong>
<div align="center">
 ID
    JUDUL
    JUMLAH
    EDIT
    DELETE
   <%
    if (rs != null){
       while(rs.next()){
         String id = rs.getString(1);
         String judul = rs.getString(2);
         String jumlah = rs.getString(3);
   응>
```

```
<
    <
    <a
href="EditData.jsp?ed_id=<%=id%>&ed_judul=<%=judul%>
&ed_jumlah=<%=jumlah%>">EDIT</a>
    <a
href="DeleteData.jsp?ed_id=<%=id%>">DELETE</a>
   <% }
      }
      st.close();
      con.close();
   응>
 <a href="FormMasukanData.htm">TAMBAH DATA</a>
</div>
</body>
</html>
```

Listing 10.1 LihatData.jsp

Jika data sudah dimasukkan semua maka akan keluar tampilan seperti pada Gambar 10.1.

🗿 lsi Tabel Film - Microsoft Internet	Explorer					
Eile Edit View Favorites Tools	<u>H</u> elp			A*		
Address 🥘 http://localhost:8080/proses/LihatData.jsp						
Y - >> → Add Tab X						
DATA FILM						
D	JUDUL	JUMLAH	EDIT DELETE			
10001	Tom and Jerry	4	EDIT DELETE			
10002	SpongeBob and The SquarePants	3	EDIT DELETE			
10003	Teletubbies	2	EDIT DELETE			
20001	The Davinci Code	2	EDIT DELETE			
20002	Casino Royale	3	EDIT DELETE			
TAMBAH DATA						
E Done				Second Intranet		

Gambar 10.1 Tampilan LihatData.jsp

2. Buat program untuk mengedit data yang sudah ada dalam tabel. Misalnya ubahlah jumlah film berjudul Teletubbies dari 2 menjadi 10 dengan cara mengklik kata EDIT yang berada pada baris film berjudul Teletubbies di pada halaman LihatData.jsp. Gunakan form masukan untuk merubah data seperti pada Gambar

10.2. Listing 10.2 EditData.jsp adalah listing untuk membuat form edit Gambar 10.2.

Jika data yang baru sudah dimasukkan, tekan submit sehingga data yang baru diproses oleh Listing 10.3 ProsesEdit.jsp sehingga bila proses edit berhasil akan keluar hasil seperti pada Gambar 10.3.

```
<%@ page import="java.sql.*" %>
<%
  /**
 * Mengambil parameter dari halaman FormMasukanData.html
 */
 String id = request.getParameter("ed id");
 String judul = request.getParameter("ed_judul");
 String jumlah = request.getParameter("ed_jumlah");
응>
<html>
<head>
<title>Edit Data</title>
</head>
<body>
<form name="form1" method="post" action="ProsesEdit.jsp">
 <td
                colspan="2"><strong>FORM
                                             INPUT
DATABASE</strong>
   ID
    <td
         width="78%"><input name="tf_id" type="text"
id="tf_id" value="<%=id%>" size="50">
   JUDUL
    <input name="tf_judul" type="text" id="tf_judul"
value="<%=judul%>" size="50">
   JUMLAH
    <input name="tf_jumlah" type="text" id="tf_jumlah"
value="<%=jumlah%>" size="50">
   <input
                   type="submit"
                                     name="Submit"
value="Submit">
     
   <a href="LihatData.jsp">LIHAT DATA</a>
</form>
```

</body> </html>

Listing 10.2 EditData.jsp

🕘 Edit Data -	- Microsoft Internet Explorer	×
<u> </u>	<u>Vi</u> ew F <u>a</u> vorites <u>I</u> ools <u>H</u> elp	1
Address 🧃 http://localhost:8080/proses/EditData.jsp?ed_id=10003&ed_judul=Teletubbies&ed_jumlah=2		
Y ·	 >> Back ▼ (○) ▼ (▲) (○) ▼ (▲) (○) ♥ Search	»
FORM IN	VPUT DATABASE	^
D	10003	
JUDUL	Teletubbies	
JUMLAH	2	
Submit		
LIHAT DA'	TA	~
E Done	Second intranet	.;;

Gambar 10.2 Tampilan EditData.jsp

```
<%@ page import="java.sql.*" %>
<%
   /**
  * Mengambil parameter dari halaman FormEditData.html
  */
  String id = request.getParameter("tf_id");
 String judul = request.getParameter("tf_judul");
 String jumlah = request.getParameter("tf_jumlah");
 int jml = Integer.parseInt(jumlah);
  /**
  * Menyiapkan variabel untuk mengakses Database
  */
  Connection con=null;
  String dbname="jdbc:odbc:DSFilm";
  String status="";
  Statement st=null;
  /**
  * Melakukan koneksi ke database
  */
  try {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con=DriverManager.getConnection(dbname,"",");
    if (con==null)
       status = "gagal";
    else
       status = "berhasil";
  }catch(ClassNotFoundException ex) {
       status = "Driver Error";
```

```
}catch(SQLException ex) {
       status = "gagal";
  }
  /**
  * Menyiapkan kueri
 */
 String kueri = "UPDATE FILM SET ID='"+id+"',
JUDUL='"+judul+"', JUMLAH="+jml+" WHERE id='"+id+"'";
  //String kueri = "UPDATE FILM SET ID='10', JUDUL='tom',
JUMLAH=12 WHERE ID='10001'";
 st = con.createStatement();
 int editTabel = st.executeUpdate(kueri);
 st.close();
 con.close();
%>
<HTML>
<HEAD>
    <TITLE>Membuat Tabel</TITLE>
</HEAD>
<BODY>
<%
 if (editTabel == 1)
    out.println("Edit data berhasil");
 else
    out.println("Edit data gagal");
응>
<br>
<br>
<a href="FormMasukanData.htm"> INPUT DATA </a>
<br>
<br>
<a href="LihatData.jsp">LIHAT DATA </a>
</BODY>
</HTML>
```

Listing 10.3 ProsesEdit.jsp



Gambar 10.3 Tampilan ProsesEdit.jsp bila proses edit berhasil

3. Buatlah program untuk menghapus data. Penghapusan data dilakukan dengan cara mengklik kata DELETE pada baris yang sesuai di halaman LihatData.jsp. Misal hapuslah data film berjudul Tom and Jerry. Jika DELETE sudah diklik maka proses menghapus akan ditangani oleh Listing 10.4 DeleteData.jsp dan jika berhasil maka akan keluar tampilan seperti pada Gambar 10.4.

```
<%@ page import="java.sql.*" %>
<%
   /**
  * Mengambil parameter dari halaman FormEditData.html
  */
 String id = request.getParameter("ed_id");
 Connection con=null;
 String dbname="jdbc:odbc:DSFilm";
 String status="";
 ResultSet rs = null;
 try {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   con=DriverManager.getConnection(dbname, "", "");
   if (con==null)
       status = "gagal";
   else
       status = "berhasil";
  }catch(ClassNotFoundException ex) {
       status = "Driver Error";
  }catch(SQLException ex) {
      status = "gagal";
```

```
Statement st = con.createStatement();
 //String kueri = "DELETE FROM FILM WHERE FILM.ID="+id;
 String kueri = "DELETE FROM FILM WHERE FILM.ID='1'";
 int deleteData = st.executeUpdate(kueri);
응>
<html>
<head>
<title>Isi Tabel Film</title>
</head>
<body>
 <%
  if (deleteData == 1)
    out.println("Delete data berhasil");
 else
    out.println("Delete data gagal");
응>
  <a href="LihatData.jsp">LIHAT DATA</a>
</div>
</body>
</html>
```

Listing 10.4 DeleteData.jsp

🕘 Isi Tabel Film - Microsoft Internet Explorer	
Eile Edit View Favorites Tools Help	1
Address 🕘 http://localhost:8080/proses/DeleteData.jsp?ed_id=10001	🖌 🄁 🗸
Y · · · · · · · · · · · · · · · · · · ·	×
Delete data berhasil	
LIHAT DATA	
🙆 Done 🧐 Local intrane	et .:

Gambar 10.4 Tampilan DeleteData.jsp jika berhasil menghapus data

11.6 Soal Latihan

- 1. Apa yang dimaksud dengan metadata?
- 2. Apakah yang dimaksud dengan resultset metadata database metadata?
- 3. Bagaimana bahasa SQL untuk melihat isi tabel, mengubah isi dan menghapus isi tabel?
- 4. Buat program dengan JSP untuk menampilkan informasi produk database. Database yang digunakan adalah database yang telah anda buat pada latihan sebelumnya, yaitu database DataFilm. Informasi yang ditampilkan adalah sebagai berikut:
 - Nama Produk
 - Versi Produk
 - Nama Driver
 - Versi Driver
- 5. Buat program dengan JSP untuk menampilkan informasi tabel FILM yang tersimpan dalam database DataFilm. Informasi yang ditampilkan adalah sebagai berikut:
 - Banyaknya kolom
 - Nama tiap-tiap kolom
 - Tipe data tiap-tiap kolom
 - Apakah nilai null diperbolehkan pada tiap-tiap kolom.
- 6. Buat program untuk melihat semua data yang tersimpan pada tabel FILM.
- 7. Buat program untuk mengedit data pada tabel FILM termasuk form untuk mengedit data.
- 8. Buat program untuk menghapus data pada tabel FILM.