

BAB XV

SCROLLING RECORD (NEXT, FIRST, LAST, PREVIOUS)

A. CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat dapat memahami bagaimana Scrolling. menggulir record pada Form terkoneksi database
2. Mahasiswa dapat mengaplikasikan menggulir record ke Awal, Akhir dan maju dan mundur pada Form terkoneksi Database dengan Kode Java
3. Mahasiswa dapat membuat program dengan menggulir record ke Awal, Maju, Akhir dan mundur pada Form yang terkoneksi Database

B. MATERI

1. Scrolling Record Java Database

Kita bisa menampilkan isi record dengan pengguliran di form misalnya maju 1 record (Move Next), ke record akhir (Move Last), mundur 1 record (Move Previous) dan ke reord awal (Move First) atau disebut Scrolling Database .

Selanjutnya yang akan kita lakukan sekarang adalah menambahkan empat tombol ke dalam form. Tombol akan memungkinkan kita untuk bergerak maju (*Move Forward*) melalui Record, mundur (*Move Back*), pindah ke Record terakhir (*Move Last*), dan pindah ke Record pertama (*Move First*).

Tambahkan panel baru ke Form. Perbesar dan kemudian tambahkan untuk tombol/Button ke panel. Ubah nama variabel tombol menjadi berikut ini:

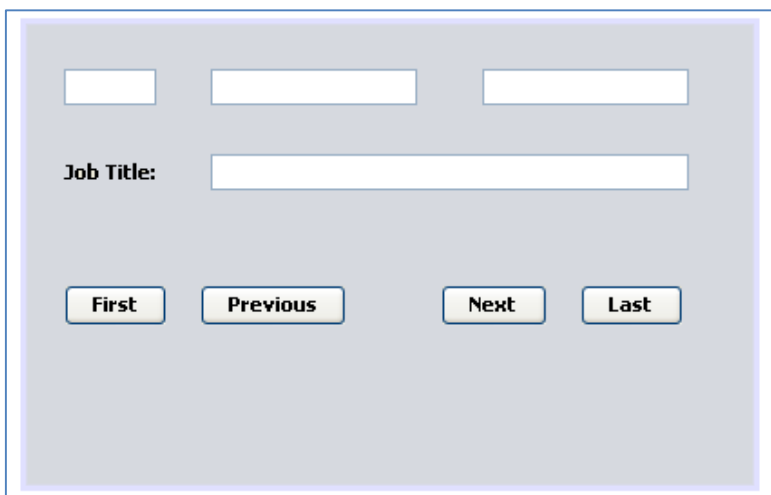
btnNext

btnPrevious

btnLast

btnFirst

Ubah teks pada setiap tombol ***Next***, ***Previous***, ***Last***, ***First*** . Form kita kemudian akan terlihat seperti ini:



Gambar 15. 1 Desain *First*, *Previous*, *Next* , *Last*

2. Scroll ke Next Record

Klik dua kali tombol **Next** untuk membuat kode program atau klik kanan ada tombol **next** pilih menu **Events>Action>ActionPerformed**. Kita perlu melakukan dua hal dengan tombol **Next** : pertama, periksa apakah ada Record berikutnya untuk dipindahkan; dan kedua, jika ada record berikutnya, tampilkan di Text Fields. Kita dapat membuat Pernyataan IF untuk ini. Tapi itu perlu dibungkus block dengan try... catch. Jadi tambahkan yang berikut ini ke rintisan kode tombol **Next** :

```
try {  
    if ( rs.next( ) ) {  
    }  
    else {  
        rs.previous( );  
        JOptionPane.showMessageDialog(Workers.this, "End of File");  
    }  
}  
catch (SQLException err) {  
    JOptionPane.showMessageDialog(Workers.this, err.getMessage());  
}
```

Pernyataan IF memindahkan ResultSet pada satu record pada satu waktu. Jika tidak ada record berikutnya

maka nilai `false` dikembalikan. Bagian `Else` memindahkan `ResultSet` mundur satu record. Itu karena `Kursor` akan bergerak melewati rekord terakhir.

Dalam kurung kurawal untuk Pernyataan `IF` kita dapat menambahkan kode untuk menampilkan Record di `TextField` :

```
int id_col = rs.getInt("ID");
String id = Integer.toString(id_col);
String first = rs.getString("First_Name");
String last = rs.getString("Last_Name");
String job = rs.getString("Job_Title");

textID.setText(id);
textFirstName.setText(first);
textLastName.setText(last);
textJobTitle.setText(job);
```

Ini adalah kode yang sama yang kita miliki dalam metode `DoConnect`. (Kita dapat membuat metode baru, agar tidak menduplikasi kode apa pun, tetapi Kita akan membuatnya tetap sederhana.)

Kode untuk tombol **Next** Anda sekarang akan terlihat seperti ini:

```

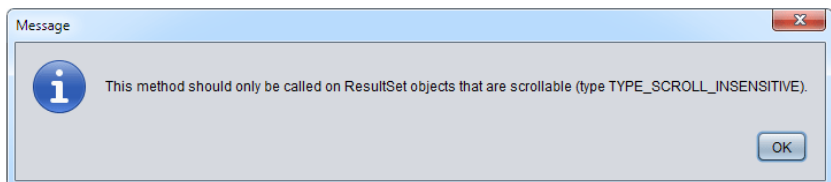
try {
    if (rs.next()) {
        int id_col = rs.getInt("ID");
        String id = Integer.toString( id_col );
        String first_name = rs.getString("First_Name");
        String last_name = rs.getString("Last_Name");
        String job = rs.getString("Job_Title");

        textID.setText(id);
        textFirstName.setText(first_name);
        textLastName.setText(last_name);
        textJobTitle.setText(job);
    }
    else {
        rs.previous( );
        JOptionPane.showMessageDialog(this, "End of File");
    }
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}

```

Ketika kita telah menambahkan kode, jalankan program dan ujilah. Lalu klik tombol **Next** dan akan menggulir semua *Record* dalam tabel

Saat Anda mencapai Record terakhir, kita akan melihat pesan kesalahan muncul:



Gambar 15. 2 Pesan Error

Masalahnya adalah kita telah menambahkan baris `rs.previous`. Namun, telah menggunakan tipe `ResultSet` default. Seperti yang dijelaskan di bagian terakhir, ini memberi Anda `ResultSet` yang hanya bisa bergerak

maju. Kita dapat menggunakan jenis yang disarankan dalam pesan kesalahan.

Hentikan program Anda dan kembali ke jendela pengkodean. Dalam metode DoConnect kita, temukan baris berikut:

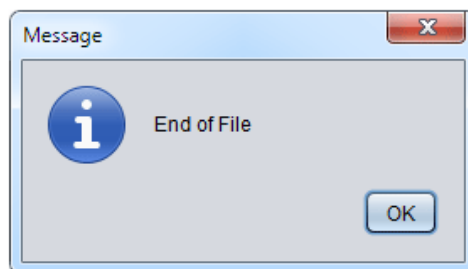
```
stmt = con.createStatement(
```

Ubah ke kode berikut ini :

```
stmt = con.createStatement(  
ResultSet.TYPE_SCROLL_INSENSITIV  
E, ResultSet.CONCUR_UPDATABLE );
```

Jenis ResultSet sekarang akan memungkinkan kita untuk menggulir ke belakang dan juga ke depan.

Jalankan kembali program kita. Klik tombol **Next** sampai mendapatkan *Record* terakhir. Kita akan melihat pesan kesalahan dari bagian try block dari try ... catch yang muncul:



Gambar 15. 3 Pesan End Of File

3. Scroll ke First Record

Pindah ke Record pertama dan terakhir dari database Anda jauh lebih mudah. Klik dua kali tombol Pertama Anda untuk membuat rintisan kode. Sekarang tambahkan kode berikut:

```
try {
    rs.first();
    int id_col = rs.getInt("ID");
    String id = Integer.toString( id_col );
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText( first_name );
    textLastName.setText( last_name );
    textJobTitle.setText( job );
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}
```

Kita tidak membutuhkan Pernyataan IF ... ELSE, sekarang. Satu-satunya hal yang perlu kita lakukan adalah memindahkan kursor ke record pertama dengan `rs.First`, kemudian tampilkan record pertama di Text Fields.

4. Scroll ke Last Record

Demikian pula, tambahkan kode berikut untuk tombol Terakhir Anda (Anda dapat menyalin dan menempelkan kode untuk tombol Pertama):

```

try {
    rs.last();
    int id_col = rs.getInt("ID");
    String id = Integer.toString( id_col );
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}

```

Satu-satunya perubahan yang harus dilakukan adalah penggunaan `rs.Last` pada baris pertama menggantikan `rs.First`.

Setelah Anda menambahkan kode, jalankan kembali program kita. Kita sekarang harus dapat melompat ke record terakhir di database, dan melompat ke record pertama.

5. Scroll ke Previous Record

Kode untuk tombol ***Previous*** mirip dengan tombol ***Next Record***. Tapi alih-alih menggunakan `rs.Next`, Anda menggunakan `rs.Previous`.

Kembali ke jendela Desain dan klik dua kali tombol ***Previous*** untuk membuat kode program.

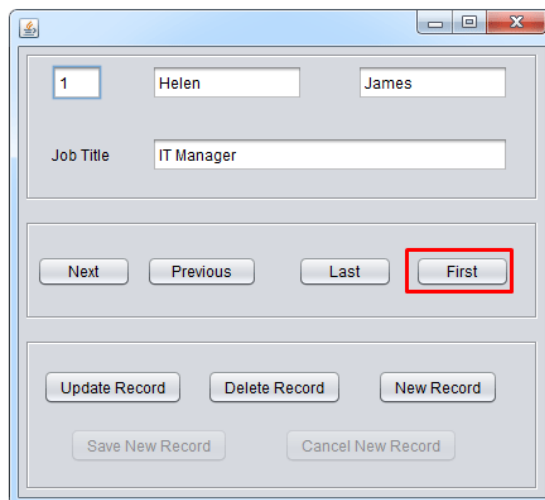
Tidak perlu mengetik semua kode itu lagi, cukup ***Copy*** dan ***Paste*** kode dari tombol ***Next***. Kemudian ubah ***rs.Next***, dalam pernyataan IF menjadi ***rs.Previous***.

Ubah ***rs.Previous*** di bagian ELSE menjadi ***rs.Next***. Anda juga dapat mengubah teks pesan kesalahan dari "End of File" menjadi "Start of File".

```
try {
    if (rs.previous()) {
        int id_col = rs.getInt("ID");
        String id = Integer.toString( id_col );
        String first_name = rs.getString("First_Name");
        String last_name = rs.getString("Last_Name");
        String job = rs.getString("Job_Title");

        textID.setText(id);
        textFirstName.setText(first_name);
        textLastName.setText(last_name);
        textJobTitle.setText(job);
    }
    else {
        rs.next();
        JOptionPane.showMessageDialog(this, "Start of File");
    }
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}
```

Jalankan kembali program kita. jika tidak ada masalah seurnya dapat bergerak mundur dan maju record database dengan mengklik dua tombol tersebut.



Gambar 15. 4 HaSil Run Scrolling

Gambar diatas adalah hasil Run dan jika mengklik tombol **First**

C. LATIHAN

1. Jelaskan Potongan pernyataan kode dibawah ini yang diberi kotak merah:

```
try {  
    if (rs.next()) {  
        int id_col = rs.getInt("ID");  
        String id = Integer.toString( id_col );  
        String first_name = rs.getString("First_Name");  
        String last_name = rs.getString("Last_Name");  
        String job = rs.getString("Job_Title");  
  
        textID.setText(id);  
        textFirstName.setText(first_name);  
        textLastName.setText(last_name);  
        textJobTitle.setText(job);  
    }  
    else {  
        rs.previous( );  
        JOptionPane.showMessageDialog(this, "End of File");  
    }  
}  
catch (SQLException err) {  
    JOptionPane.showMessageDialog(this, err.getMessage());  
}
```

2. Jelaskan potongan kode program dibawah ini yang diberi Kotak Merah :

```

try {
    rs.first();
    int id_col = rs.getInt("ID");
    String id = Integer.toString( id_col );
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}

```

3. Jelaskan potongan kode program dibawah ini yang diberi kotak merah :

```

try {
    if (rs.previous()) {
        int id_col = rs.getInt("ID");
        String id = Integer.toString( id_col );
        String first_name = rs.getString("First_Name");
        String last_name = rs.getString("Last_Name");
        String job = rs.getString("Job_Title");

        textID.setText(id);
        textFirstName.setText(first_name);
        textLastName.setText(last_name);
        textJobTitle.setText(job);
    }
    else {
        rs.next();
        JOptionPane.showMessageDialog(this, "Start of File");
    }
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}

```

D. REFERENSI

Danny Poo Derek Kiong Swarnalatha Ashok, Object-Oriented Second edition Programming and Java, Springer 2008

https://www.homeandlearn.co.uk/java/databases_and_java_forms.html diakses pada tanggal 10 Juli 2022.

<https://www.w3schools.com/sql/>, diakses pada tanggal 30 Juli 2022

Romi satrio Wahono, Java Gui, 2016