

BAB XVI

MENAMBAH RECORD BARU MELALUI FORM JAVA DATABASE

A. CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat dapat memahami Menambah record pada Form terkoneksi database
2. Mahasiswa dapat mengaplikasikan Penambahan record Form terkoneksi Database dengan Kode Java
3. Mahasiswa dapat membuat program dengan Penambahan Record pada Form yang terkoneksi Database

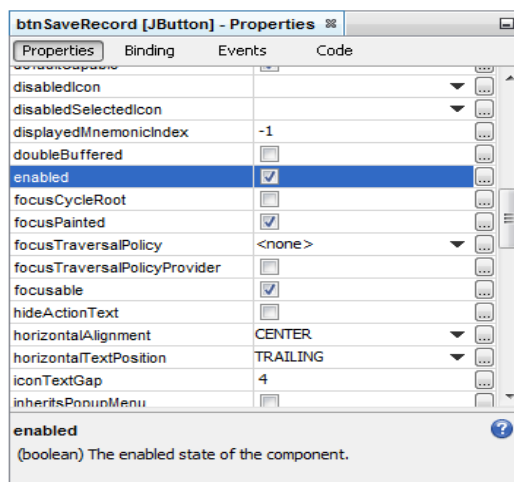
B. MATERI

1. Menambah Record Melalui Form

Kita bisa menambah isi record baru pada form yang terkoneksi database. Pada latihan sebelumnya record yang ditampilkan atau di digulir pengisian nya menggunakan jendela service Java DB. Pada pertemuan ini kita akan mempelajari bagaimana menambahkan record melalui Tombol yang di desain ke dalam form yang sudah terkoneksi dengan database.

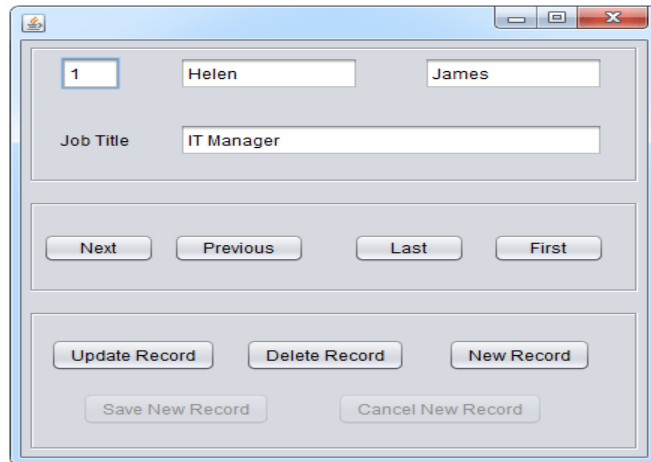
Persiapan untuk itu Pada Desain kita telah memiliki tiga tombol yang merujuk ke **New Record**, **Save New Record**, dan Cance New Record. Tombol **New Record** hanya akan mengosongkan Bidang TeksField, dan menyiapkannya untuk data baru yang akan dimasukkan. Kita juga dapat menonaktifkan /disabled beberapa tombol lain, termasuk tombol **New Record**. Hal lain yang dapat kita lakukan adalah membuat tentang record mana yang sedang dimuat. Jika pengguna berubah pikiran, kita dapat mengaktifkan kembali semua tombol dengan mengklik tombol **Cancel**. Mengklik tombol **Save New Record** akan melakukan penyimpanan data ke database.

Kita Bisa menset Klik tombol **New Record** melalui Properties di sebelah kanan Netbeans, temukan properti Enabled:



Gambar 16. 1 Properties Enabled

Hapus centang pada kotak di sebelah kanan yang diaktifkan untuk **Save New Record** yang akan dinonaktifkan. Lakukan hal yang sama untuk tombol **Cancel New Record** yang akan dinonaktifkan. Sehingga form desain pertama akan terlihat seperti ini:



The screenshot shows a Java Swing window with a light blue title bar. Inside, there's a form with several input fields and buttons. At the top, there are three text boxes: the first contains '1', the second contains 'Helen', and the third contains 'James'. Below these is a 'Job Title' label followed by a text box containing 'IT Manager'. In the middle section, there are four buttons: 'Next', 'Previous', 'Last', and 'First'. At the bottom, there are three buttons: 'Update Record', 'Delete Record', and 'New Record'. Below these are two more buttons: 'Save New Record' and 'Cancel New Record'. All buttons have a disabled appearance, indicated by a light gray color and no focus outline.

Gambar 16. 2 *Save Enabled False*

Bahkan jika kita memiliki kode untuk kedua tombol ini, tidak akan terjadi apa-apa jika kita mengklik salah satunya.

Ketika tombol **New Record** diklik, kita dapat menonaktifkan tombol-tombol berikut:

First

Previous

Next

Last

Update Record

Delete Record

New Record

Tombol **Save** dan **Cancel**, bagaimanapun, dapat diaktifkan. Jika pengguna mengklik **Cancel**, Kita dapat mengaktifkan kembali tombol tersebut.

Klik dua kali tombol **New Record** kita untuk membuat kode program. Tambahkan baris kode Java berikut:

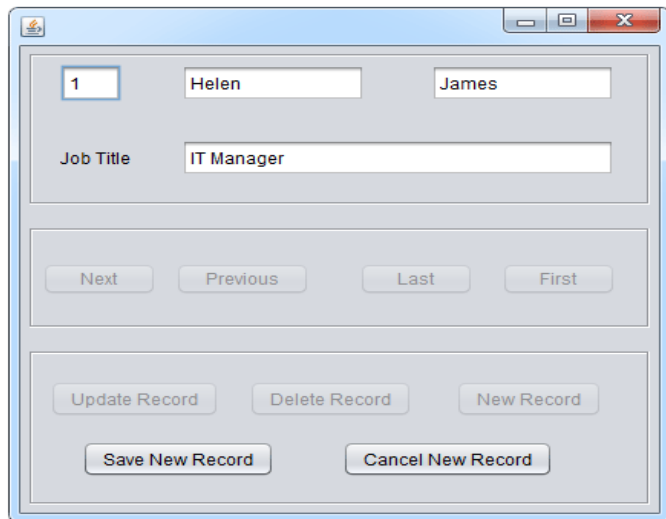
```
btnFirst.setEnabled( false );  
btnPrevious.setEnabled( false );  
btnNext.setEnabled( false );  
btnLast.setEnabled( false );  
btnUpdateRecord.setEnabled( false );  
btnDelete.setEnabled( false );  
btnNewRecord.setEnabled( false );  
btnSaveRecord.setEnabled( true );  
btnCancelNewRecord.setEnabled( true );
```

Jadi tujuh tombol dimatikan menggunakan properti `setEnabled`. Dua tombol dihidupkan yaitu **Save** dan **Cancel**.

Kita bisa melakukan kebalikannya untuk tombol **Cancel**. Beralih kembali ke tampilan Desain. Klik dua kali tombol **Cancel New Record** untuk membuat kode program. Tambahkan berikut ini:

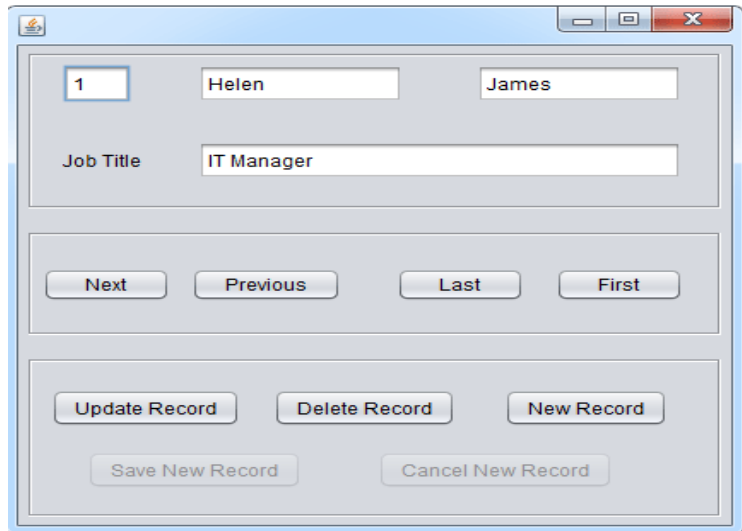
```
btnFirst.setEnabled( true );  
btnPrevious.setEnabled( true );  
btnNext.setEnabled( true );  
btnLast.setEnabled( true );  
btnUpdateRecord.setEnabled( true );  
btnDelete.setEnabled( true );  
btnNewRecord.setEnabled( true );  
btnSaveRecord.setEnabled( false );  
btnCancelNewRecord.setEnabled( false );
```

Sekarang jalankan program kita dan ujilah. Klik tombol New Record dan form akan terlihat seperti ini:



Gambar 16. 3 Save New enabled True

Klik tombol **Cancel New Record** dan form akan terlihat seperti ini:



Gambar 16. 4 Klik *Cancel New Record*

Hal lain yang perlu kita lakukan adalah record pada baris mana yang sedang dimuat. Dengan kata lain, nomor baris mana yang saat ini dimuat di Bidang TeksField. Kita perlu melakukan ini karena Text Fields akan dibersihkan. Jika tombol ***Cancel*** diklik, maka kita dapat memuat ulang data yang terhapus.

Tambahkan variabel Integer berikut ke bagian atas kode kita, tepat di bawah baris Connection, Statement, dan ResultSet Anda:

int curRow = 0;

Bagian atas kode Anda akan terlihat seperti gambar dibawah ini :

```

package Employees;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;

public class Workers extends javax.swing.JFrame {

    Connection con;
    Statement stmt;
    ResultSet rs;
    int curRow = 0;

    public Workers() {...4 lines }

    public void DoConnect( ) {...35 lines }

```

Sekarang kembali ke kode Rekaman Baru Anda.

Untuk mendapatkan baris mana yang ditunjuk oleh Kursor saat ini, ada metode yang disebut `getRow`. Ini memungkinkan kita untuk menyimpan nomor baris tempat kursor berada:

`curRow = rs.getRow();`

Kita akan menggunakan nomor baris ini dalam kode ***Cancel New Record***.

Satu-satunya hal lain yang perlu kita lakukan untuk tombol ***New Record*** adalah mengosongkan Bidang Teksfield. Ini cukup sederhana:

```
textFirstName.setText("");  
textLastName.setText("");  
textJobTitle.setText("");  
textID.setText("");
```

Jadi kita hanya mengatur properti Text ke string kosong. Karena kita telah menggunakan metode dari ResultSet, kita perlu membungkus semuanya dalam blok try ... catch. Berikut tampilan kode untuk tombol New Record kita :

```
package Employees;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.sql.ResultSet;  
import javax.swing.JOptionPane;  
  
public class Workers extends javax.swing.JFrame {  
  
    Connection con;  
    Statement stmt;  
    ResultSet rs;  
    int curRow = 0;  
  
    public Workers() { ...4 lines }  
  
    public void DoConnect( ) { ...35 lines }
```

Untuk tombol **Cancel**, kita perlu mendapatkan baris yang sebelumnya dimuat dan mengembalikan data ke dalam Bidang *Teks Field*.

Untuk memindahkan Kursor kembali ke baris yang sebelumnya ditunjuk, kita dapat menggunakan metode absolut:

```
rs.absolute( curRow );
```

Metode absolut memindahkan Kursor ke posisi tetap di ResultSet. Kita ingin memindahkan nilai yang di simpan di variabel ***curRow***.

Sekarang Kursor menunjuk ke baris yang benar, kita dapat memuat data ke dalam Bidang Teks Fields:

```
textFirstName.setText(  
rs.getString("First_Name" ) );  
textLastName.setText(  
rs.getString("Last_Name" ) );  
textJobTitle.setText( rs.getString("Job_Title" ) );  
textID.setText( Integer.toString( rs.getInt("ID" ) ) );
```

Membungkus semuanya dalam blok *try ... catch* kode kita sebagai berikut:

```

try {
    rs.absolute( curRow );

    textFirstName.setText( rs.getString("First_Name") );
    textLastName.setText( rs.getString("Last_Name") );
    textJobTitle.setText( rs.getString("Job_Title") );
    textID.setText( Integer.toString( rs.getInt("ID") ) );

    btnFirst.setEnabled( true );
    btnPrevious.setEnabled( true );
    btnNext.setEnabled( true );
    btnLast.setEnabled( true );
    btnUpdateRecord.setEnabled( true );
    btnDeleteRecord.setEnabled( true );
    btnNewRecord.setEnabled( true );

    btnSaveRecord.setEnabled( false );
    btnCancelNewRecord.setEnabled( false );
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(this, err.getMessage());
}

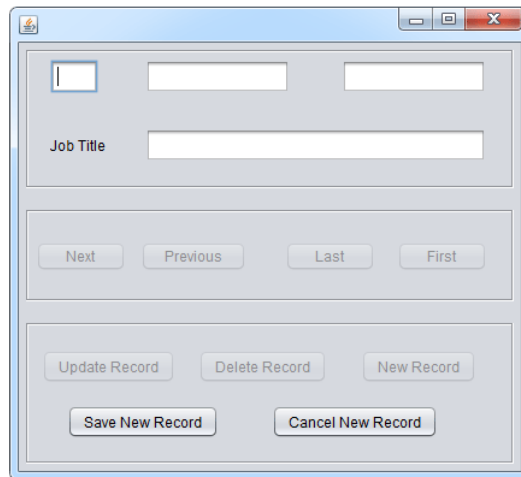
```

Setelah selesai menambahkan kode untuk tombol ***New Record*** dan ***Cancel***, jalankan program Anda dan cobalah. Sebelum mengklik tombol ***New Record***, form akan terlihat seperti ini:

The screenshot shows a Java Swing window titled "Form Record". It features a table with one row containing the values "1", "Helen", and "James". Below the table is a text field labeled "Job Title" with the text "IT Manager". Underneath the text field is a horizontal panel containing four buttons: "Next", "Previous", "Last", and "First". Below this panel is another horizontal panel containing three buttons: "Update Record", "Delete Record", and "New Record". At the bottom of the window is a final horizontal panel containing two buttons: "Save New Record" and "Cancel New Record".

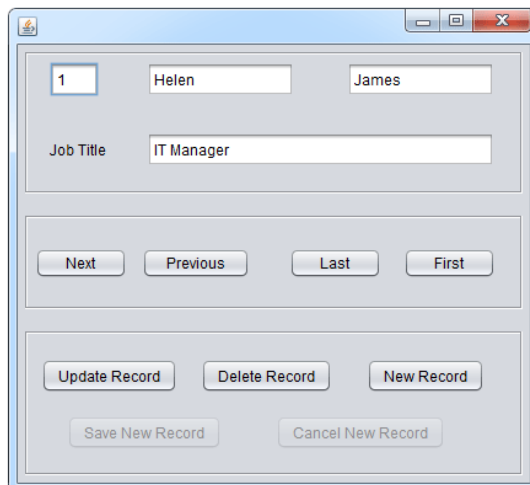
Gambar 16. 5 Form Record

Klik tombol **New Record** untuk melihat Bidang Teks Field dibersihkan:



Gambar 16. 6 Form New Record

Mengklik tombol **Cancel New Record** akan memuat ulang data:



Gambar 16. 7 Klik Tombol *Cancel Reportt*

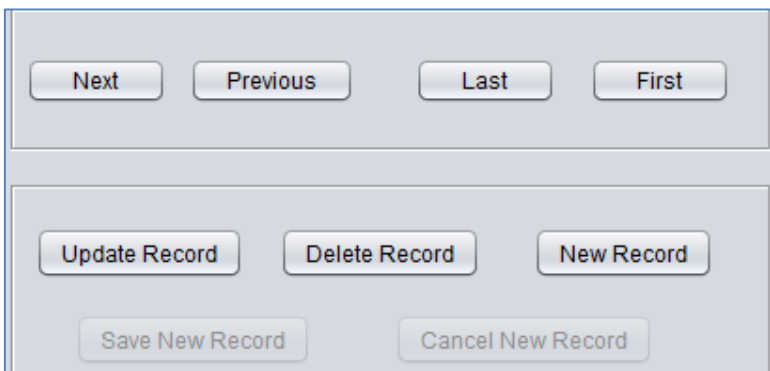
Sekarang setelah tombol **New** dan telah diatur, kita dapat melanjutkan dan menyimpan data baru apa pun yang dimasukkan ke dalam Bidang **Teks Field**.

C. LATIHAN

1. Jelaskan Potongan kode program dibawah setiap barisnya ini digunakan untuk apa? :

```
textFirstName.setText("");  
textLastName.setText("");  
textJobTitle.setText("");  
textID.setText("");
```

2. Terdapat 9 Tombol pada tampilan dibawah ini, Jika nama variabel nya masing masing **BtnNext**, **BtnPrevious**, **BtnLast**, **BtnFirst**, **BtnUpdate**, **BtnDelete**, **BtnNew**, **BtnSave**, **BtnCancel**



Gambar 16. 8 Latihan Tombol *Scrolling*

Bagaimana Kode program agar tampilan nampak seperti gambar?

D. REFERENSI

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