

# BAB XIV

## BEKERJA DENGAN FORM DAN JAVA DATABASE

---

### A. CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat dapat memahami bekerja dengan Form terkoneksi database
2. Mahasiswa dapat meng aplikasikan TextField, Button Form yang terkoneksi Database dengan Kode Java
3. Mahasiswa dapat membuat program dengan Master /Detail Form

### B. MATERI

#### 1. Form Koneksi Database

Pada Pertemuan ini, kita akan membuat Form dengan tombol dan bidang teks atau Textfield. Tombol akan digunakan untuk menggerakkan maju dan mundur record dalam tabel database. Kita juga akan menambahkan tombol untuk melakukan tugas database umum lainnya. Form yang akan kita desain akan terlihat seperti ini:

**Gambar 14. 1 Form Koneksi Database**

Langkah langkah nya :

Mulai Project baru untuk dengan mengklik **File > New Project** dari menu NetBeans. Ketika kotak dialog muncul, pilih **Java > Java Application**. Pada langkah pertama dari kotak dialog, ketik `database_form` sebagai Nama Proyek. Hapus centang pada kotak di bagian bawah untuk Buat Man Class. Klik tombol Finish untuk membuat Project kosong.

Di area Project di sebelah kiri temukan proyek `database_form` Anda, dan klik kanan entri. Dari menu yang muncul pilih **New > JFrame Form**.

Saat kotak dialog muncul, ketik `Pekerja` untuk nama Kelas, dan `Karyawan` sebagai nama paket. Ketika Anda

mengklik Selesai, Anda akan melihat formulir kosong muncul di jendela utama NetBeans.

Tambahkan Panel ke Form Anda. Kemudian tempatkan empat *TextField* pada panel. Hapus teks default di *TextField*, biarkan semuanya kosong. Ubah nama variabel default untuk *TextField* menjadi berikut ini:

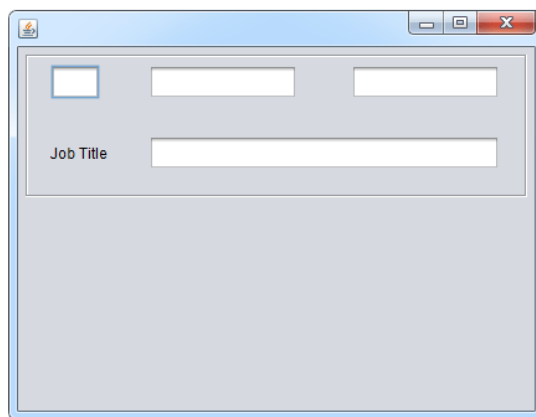
**textID**

**textFirstName**

**textLastName**

**textJobTitle**

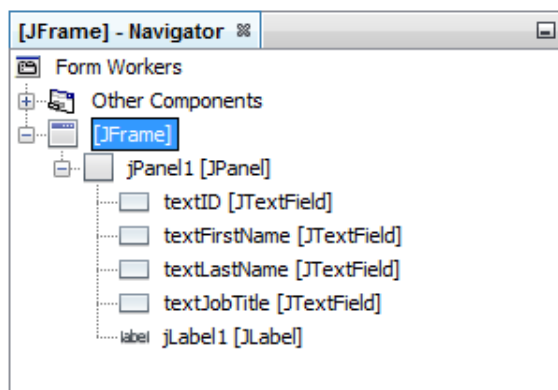
Tambahkan label ke panel. Posisikan tepat di sebelah kiri *TextField JobTitle*. Ketikkan "Job Title" sebagai teks untuk label. Atur *TextField* dan *Label* sehingga form kita terlihat seperti ini:



**Gambar 14. 2 Desain Form**

Sekarang lihat area Navigator di sebelah kiri NetBeans. (Jika tidak terlihat, klik **Window > Navigator**

dari menu NetBeans.) Seharusnya sama dengan gambar dibawah ini :



**Gambar 14. 3 Navigator Desain**

Apa yang ingin kita dilakukan sekarang adalah membuat *record* pertama dari tabel database muncul di Textfield saat form pertama kali run. Untuk melakukan itu, kita dapat memanggil metode dari Konstruktor form.

Namun, pertama-tama, kita dapat menambahkan file JAR Client Driver ke project, seperti terakhir kali. Ini akan mencegah kesalahan "Driver Tidak Ditemukan". Jadi, di area Project, klik kanan entri Library untuk Project kita. Dari menu yang muncul, pilih Tambah JAR/Folder. Ketika kotak dialog muncul, cari file derbyclient.jar. Kemudian klik *Open* untuk menambahkannya ke Project Kita.

Pada jendela NetBeans utama, klik tombol Source di bagian atas untuk mendapatkan kode kita. Sekarang tambahkan pernyataan import berikut di bagian atas:

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

Kita telah mempelajari ini di pertemuan sebelumnya, lima yang pertama dari bagian sebelumnya. Yang terakhir, JOptionPane, adalah agar kita bisa menampilkan pesan kesalahan.

Di dalam Class, tambahkan deklarasi variabel berikut:

```
Connection con;  
Statement stmt;  
ResultSet rs;
```

Tepat di bawah Workers Konstruktor, tambahkan method berikut:

```
public void DoConnect( ) {  
}
```

Sekarang panggil method ini dari Konstruktor:

```
public Workers() {  
    initComponents();  
    DoConnect();  
}
```

Jendela kode kita kemudian akan terlihat seperti 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;

    public Workers() {
        initComponents();
        DoConnect();
    }

    public void DoConnect( ) {
    }
}

```

(Jangan khawatir jika kita memiliki garis bawah untuk pernyataan import. Kecuali jika itu adalah garis bawah merah. Dalam hal ini, kita mungkin telah membuat kesalahan pengetikan)

Apa yang telah kita lakukan di sini adalah menyiapkan objek Connection yang disebut con, objek Statement yang disebut stmt, dan objek ResultSet yang disebut rs. Kita telah mengaturnya di bagian atas karena tombol di form akan membutuhkan akses ke objek ini.

Saat form dimuat, metode DoConnect akan dipanggil. Kita dapat menambahkan kode di sini untuk terhubung ke database, dan menampilkan *record* pertama pada *TextField*.

Kode program yang ditambahkan untuk metode DoConnect hampir identik dengan kode yang kita tulis di bagian sebelumnya. Seperti ini :

```

public void DoConnect( ) {
    try {
        //CONNECT TO THE DATABASE
        String host = "jdbc:derby://localhost:1527/Employees";
        String uName = "Your_Username_Here";
        String uPass= " Your_Password_Here ";
        con = DriverManager.getConnection( host, uName, uPass );

        //EXECUTE SOME SQL AND LOAD THE RECORDS INTO THE RESULTSET
        stmt = con.createStatement();
        String SQL = "SELECT * FROM Workers";
        rs = stmt.executeQuery( SQL );

        //MOVE THE CURSOR TO THE FIRST RECORD AND GET THE DATA
        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");

        //DISPLAY THE FIRST RECORD IN THE TEXT FIELDS
        textID.setText(id);
        textFirstName.setText(first_name);
        textLastName.setText(last_name);
        textJobTitle.setText(job);
    }
    catch ( SQLException err ) {
        JOptionPane.showMessageDialog(this, err.getMessage());
    }
}

```

Satu baris kode yang mungkin belum kita temui adalah yang ini:

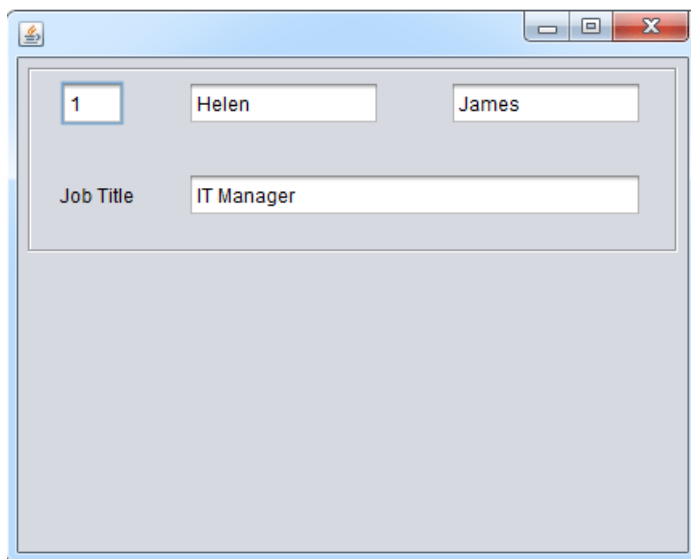
***String id = Integer.toString( id\_col );***

Karena kolom ID adalah Integer, kita perlu mengubahnya/Konversi menjadi String untuk metode setText dari Text Field. Kita perlu melakukan ini karena Text Field tidak menerima nilai Integer secara langsung - Anda perlu mengonversinya menjadi teks.

Semua baris lain dalam kode seharusnya sudah tidak asing lagi bagi kita sekarang. Pelajari kodenya untuk memastikan kita tahu apa yang terjadi. Kemudian tambahkan metode DoConnect Anda sendiri.

Kita dapat Run program sekarang. Namun, pertama-tama, pastikan untuk **Start server Java DB** Anda dari *Services Windows*.

Saat Anda menjalankan program, kita akan melihat Record pertama ditampilkan di *TextField*:



1	Helen	James
---	-------	-------

Job Title IT Manager

**Gambar 14. 4 Run Form**

Sekarang kita memiliki *Record* pertama yang ditampilkan, kita dapat menambahkan beberapa tombol untuk menggulir data tabel lainnya. Kita akan mempelajarinya pada halaman berikut.



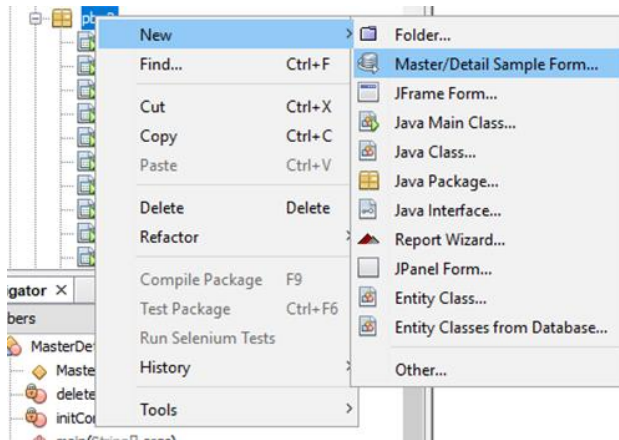
## 2. Koneksi Database dengan Master/Detail Sample Form

Terkadang kita ingin membuat aplikasi secara cepat dalam, Membaca, Memperbarui & Menulis record otomatis menggunakan wizard. fitur Aplikasi Database Desktop Swing seperti itu dapat menggunakan Master/Detail Sample Form

Master/Detail Sample Form dapat menampilkan baris master tabel dan beberapa baris detail dalam satu halaman form. Dengan form ini, pengguna dapat menyisipkan, memperbarui, dan menghapus nilai tabel atau menampilkan Nilai di baris master menentukan baris detail mana yang ditampilkan untuk diperbarui. Misalnya, baris master dapat menampilkan nilai kolom dari tabel Detail. Pengguna akhir formulir detail master bisa memperbarui atau menghapus nilai di baris master. Wizard menyediakan opsi untuk mengizinkan penghapusan, sehingga menghapus baris master juga menghapus baris detail.

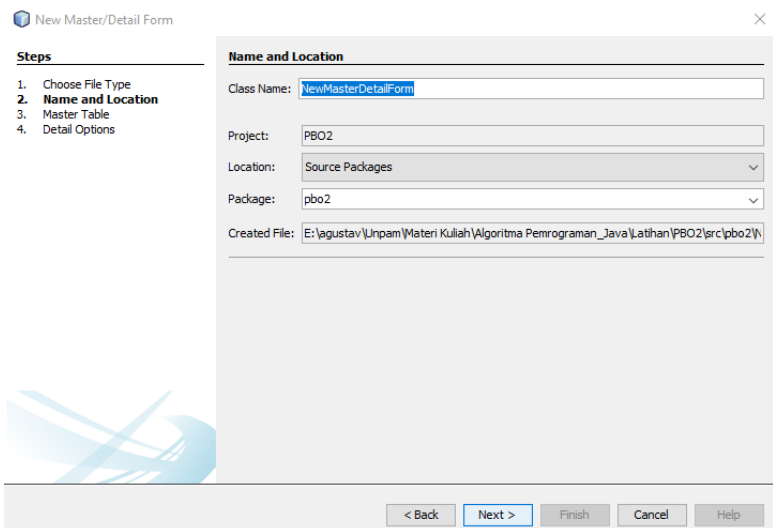
Langkah langkah membuat Form Master/Detail Form :

- a. Jika Project sudah terbuka klik kanan pada package pilih menu ***New>Master/Detail Sample Form***



**Gambar 14. 5 Master Detail Form**

b. Berikutnya adalah tampilan seperti dibawah ini :



**Gambar 14. 6 Step Name and Location**

Kita akan melewati 4 tahapan wizard dalam pembuatan Master Detail Form :

### 1) Choose File Of Type

## 2) Name and Location

## 3) Master Table

## 4) Detail Options

- c. Posisi sekarang pada langkah 2. yaitu menentukan Nama *Class* dan lokasi penyimpanan Form (**Name and Location**)

Isikan nama class contoh Class Name :

MDFormWorker

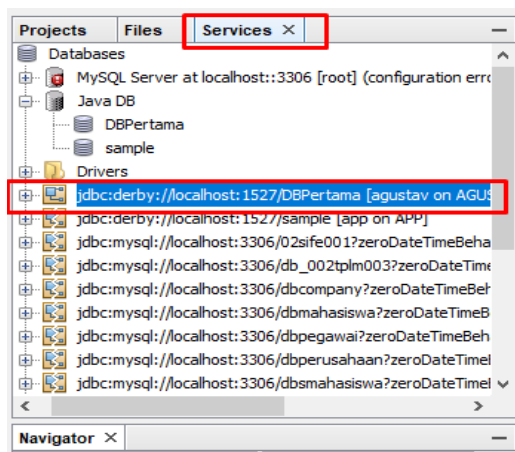
*Package* dan *Create File* isikan sesuai kebutuhan

- d. Klik **Next** untuk Langkah selanjutnya, sehingga masuk ke tahapan **Master Table**, Pada Tahap ini ada 2 kotak isian yang harus diisi

### 1) Database Connection

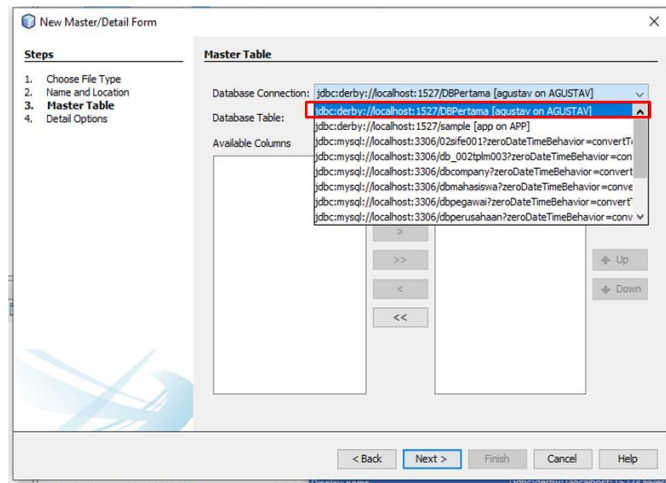
### 2) Database Table

- 1) DataBase Connection adalah koneksi database yang sudah kita bentuk sebelumnya untuk memastikan kita bisa klik menu **services** :



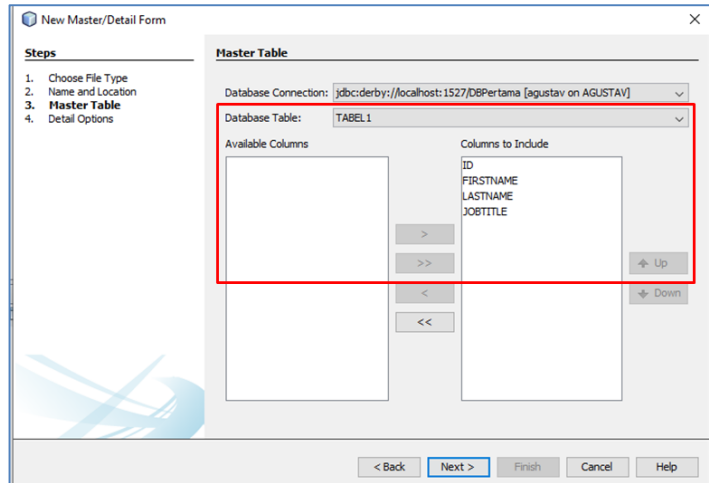
**Gambar 14. 7 Master Table**

contoh isikan DataBase Connection dengan memilih combo seperti dibawah ini :



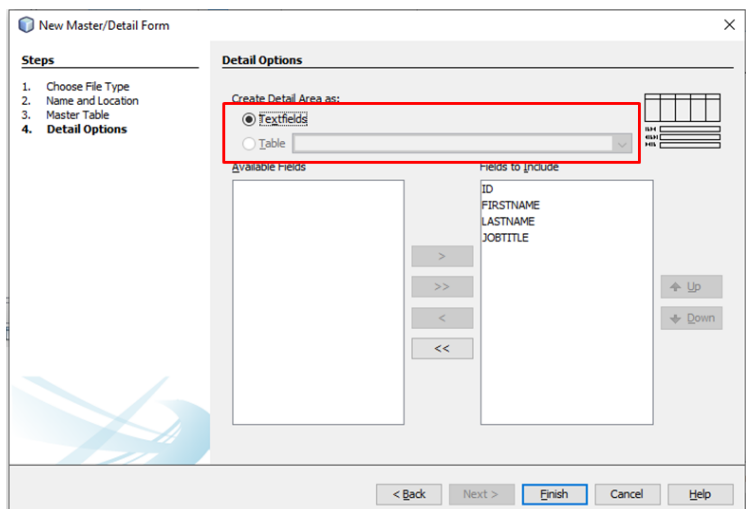
**Gambar 14. 8 Database Connector**

- 2) Database Table : Adalah menentukan table yang sudah dibentuk pada database connection, Database Table akan otomatis muncul ketika database connection kita tentukan . kita tinggal pilih table mana yang akan dimunculkan pada form master detail. Contoh seperti gambar dibawah ini :



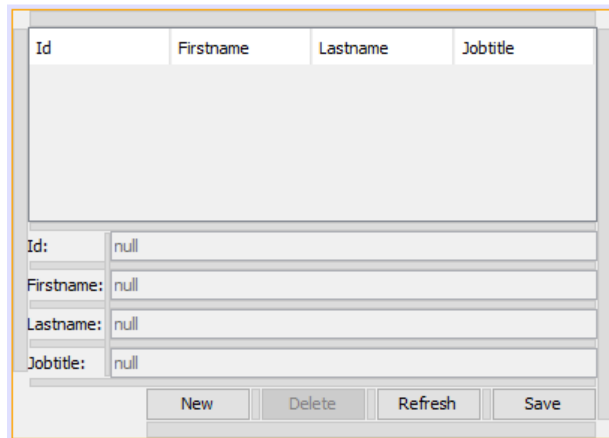
**Gambar 14. 9 Step Master Tabel**

- e. Klik Next untuk masuk tahap selanjutnya yaitu Detail Option, Detail Option menentukan detail yang akan ditampilkan apakah berupa text atau Table



**Gambar 14. 10 Detail Option**

- f. Klik **Finish** jika sudah menentukan setiap tahap demi tahap yang dilalui, tunggu beberapa saat, jika tidak ada masalah maka otomatis frame akan membentuk Master / Detail berikut tombol **New, Delete, Refresh, Save**



Id	Firstname	Lastname	Jobtitle
----	-----------	----------	----------

Id: null

Firstname: null

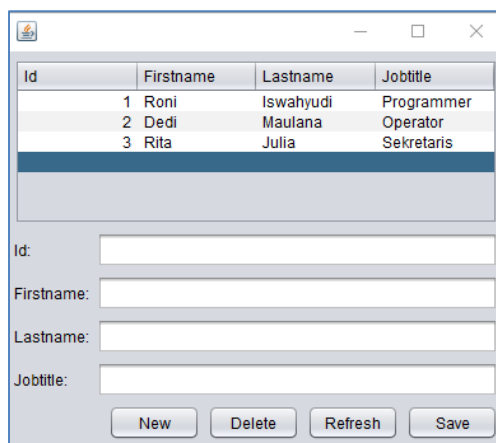
Lastname: null

Jobtitle: null

New Delete Refresh Save

**Gambar 14. 11 Hasil Desain Master Detail**

- g. Run File Form nya jika tidak ada masalah maka hasil nya seperti gambar dibawah ini :



Id	Firstname	Lastname	Jobtitle
1	Roni	Iswahyudi	Programmer
2	Dedi	Maulana	Operator
3	Rita	Julia	Sekretaris

Id:

Firstname:

Lastname:

Jobtitle:

New Delete Refresh Save

**Gambar 14. 12 Hasil Run Master Detail**

### C. LATIHAN

1. Apa yang dimaksud Potongan pernyataan kode dibawah ini :

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

2. Apa yang dimaksud potongan kode program dibawah ini:

```
String id = Integer.toString( id_col );
```

3. Apa yang dimaksud potongan kode program dibawah ini:

```
ResultSet.TYPE_SCROLL_INSENSITIVE,  
ResultSet.CONCUR_UPDATABLE );
```

### 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](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