**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IX**

**JAVA DATABASE**

****

**DOSEN PENGAMPU :**

Agus Priyanto, S. Kom., M.Kom.

**DISUSUN OLEH :**

Filfimo Yulfiz Ahsanul Hulqi

19102143

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**FAKULTAS INFORMATIKA**

**PRODI S1 INFORMATIKA**

**2021**

**BAB I**

**Tujuan Praktikum**

Mahasiswa diharapkan mampu untuk :

1. Memahami konsep *Graphical User Interface* (GUI) di dalam bahasa pemrograman Java dan komponen-komponen didalamnya (Swing atau AWT).
2. Memahami konsep dan praktik koneksi database dengan menggunakan JDBC dan JDBC API.
3. Mampu membuat aplikasi atau program database sesuai dengan studi kasus yang nyata di masyarakat

**BAB II**

**Dasar Teori**

1. **API untuk Aplikasi GUI pada Java**

Ada dua API yang dapat digunakan pada Java

1. AWT (*Abstract Window Toolkit*)

Library dan komponen GUI (java.awt) yang pertama kali diperkenalkan oleh Java, tidak direkomendasikan lagi karena penggunaan komponen GUI dari AWT sudah *outdated*.

1. Swing or JFC (*Java Foundation Class*)

Library dan komponen GUI (javax.swing) terbaru dari Java dan direkomendasikan untuk pemrograman GUI. Komponen Swing sebagian besar adalah turunan dari AWT dan lebih lengkap daripada AWT.

1. **Fitur Swing**
2. Komponen GUI Lengkap 🡪 *button, listbox, combobox, textarea,* dsb.
3. *Pluggable Look-and-Feel* 🡪 Tampilan GUI dapat diubah sesuai dengan keinginan (tidak perlu mengikuti *native* sistem operasi).
4. Data transfer antar komponen 🡪 *drag and drop, copy and paste*.
5. *Internationalization* 🡪 Proses desain aplikasi yang memungkinkan aplikasi berjalan sesuai dengan preferensi tanpa rekompilasi.
6. *Localization* 🡪 Proses translasi teks ke bahasa lokal dan menambahkan komponen lokal.
7. **Komponen Swing**
8. *Top-Level Container*

Kontainer dasar dimana komponen lainnya diletakkan (*jFrame, jDialog,* dan *Applet*).

1. *Intermediate Container*

Kontainer perantara dimana komponen lainnya diletakkan (*jPanel, jScrollPane, jToolbar*, dsb).

1. *Layout Manager*

Mengatur tata letak dan posisi dari komponen dalam kontainer (*BorderLayout, BoxLayout, FlowLayout,* dsb).

1. *Event Handling*

Menangani *event* yang dilakukan oleh *user* (*click, typing*, *zoom-in/out*, dsb).

1. **JDBC dan JDBC API**

**JDBC**adalah standar *Java Database Connectivity*, dan **JDBC API** merupakan *Java Database Connectivity Application Programming Interface*. Seemua komponen dan teknik dari JDBC tertanam dan diimplementasikan dalam JDBC API. Pada dasarnya, JDBC API terdiri dari satu set ***class*** dan ***interface*** yang digunakan untuk berinteraksi dengan database dari aplikasi Java. Umumnya JDBC API melakukan 3 fungisi berikut :

1. Membangun hubungan antara aplikasi Java yang dibuat dengan database terkait.
2. Membangun dan mengeksekusi pernyataan dari SQL.
3. Memproses hasil-hasil.

Beberapa vendor database yang berbeda menyediakan berbagai macam driver JDBC untuk mendukung aplikasi mereka untuk terhubung ke database, Komponen JDBC paling populer terletak di paket berikut :

1. java.sql : terdiri dari komponen JDBC standar.
2. javax.sql : berisi standar pendukung JDBC, yang menyediakan fitur tambahan, seperti Java Naming dan Directory Interface (JNDI) dan Java Transaction Service (JTS).
3. oracle.jdbc : berisi fungsi pendukung yang disediakan oleh java.sql dan javax.sql interface.
4. oracle.sql : berisi *class* dan *interface* yang menyediakan pemetaan Java untuk SQL.

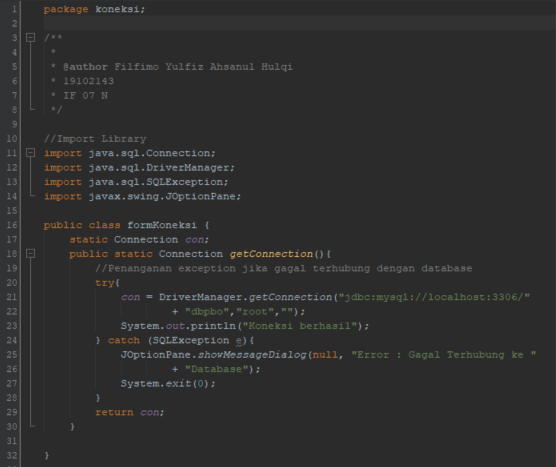
**BAB III**

**PENJELASAN**

**UNGUIDED**

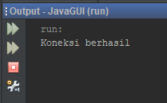
Buat sebuah program implementasi GUI dan Database, seperti pada praktikum, dengan tambahan untuk membuat satu menu lagi. Disini akan membuat sebuah program untuk guru dimana guru dapat melihat data guru atau admin yang terdaftar, input biodata murid, dan melihat nilai dari siswa.

1. Buat file untuk melakukan koneksi ke database terlebih dahulu.



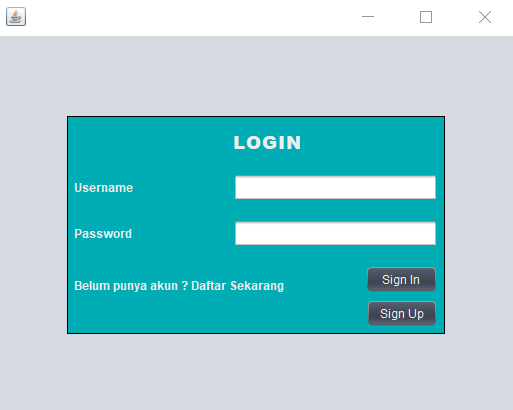
Penjelasan *class* koneksi :

File ini akan dibuat untuk digunakan dalam melakukan koneksi ke database dapat dilihat pada bagian *getConnection* disini terhubung ke *database* dengan nama dbpbo, dan setiap kelas yang akan terhubung dengan database, harus melampirkan kelas koneksi ini.



Jika koneksi terhubung dengan database, maka akan muncul pesan seperti diatas saat melakukan *running program*.

1. Buat file untuk melakukan login, atau sebagai tampilan awalan dari program.

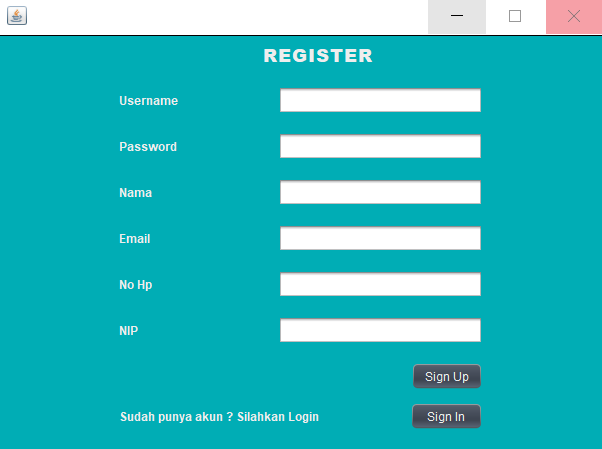


Tampilan dari *Login*

Pejelasan *class* fLogin :

Pada program diatas didalam kode programnya sebenarnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas adalah hanya untuk fungsi dari GUI saja, seperti *button*, *form,* dan *label*. Pada bagian ini *user* nantinyadapat melakukan *login* kedalam program untuk sistem loginnya adalah *usernam* dan *password* dicek apakah cocok dengan yang ada di *database* atau tidak, jika ada datanya maka akan dapat masuk kedalam program tapi jika tidak maka tidak dapat masuk kedalam program atau halaman selanjutnya.

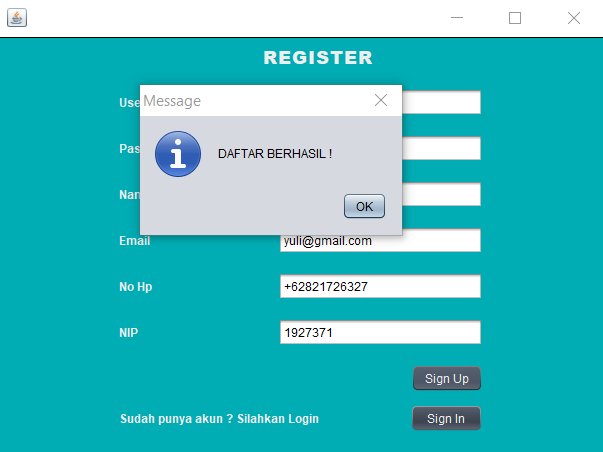
1. Buat file untuk melakukan registrasi, agar *user* tedaftar dan dapat melakukan *sign in*.



Tampilan Registrasi

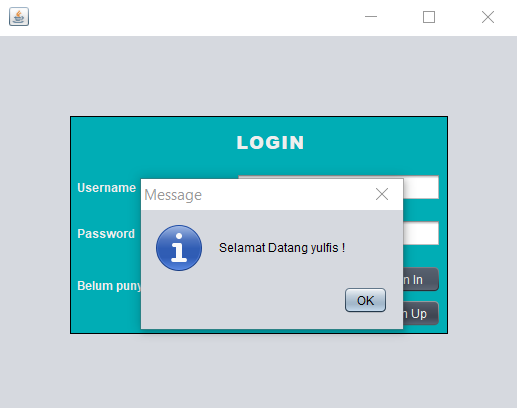
Penjelasan *class* fRegis :

Sama seperti sebelumnya, pada kode program diatas sebenarnya didalamnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas hanya adalah untuk fungsi dari GUI saja, seperti *button*, *form,* dan *label* yang akan digunakan agar *user* dapat melakukan registrasi, yang nantinya registrasi ini akan disimpan ke database dan dapat digunakan untuk melakukan *login* pada program sebelumnya yaitu login.



Tampilan registrasi sukses

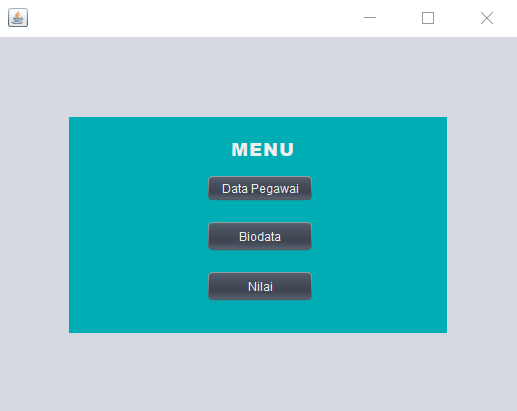
Gambar diatas adalah tampilan jika *user* berhasil melakukan registrasi, data yang sudah di*input* ini nanti akan dikirim ke *database* agar disimpan, dan untuk dapat melakukan *login*.



Tampilan sukses melakukan *login*

Gambar diatas adalah tampilan jika *user* berhasil melakukan *login* karena data yang *user* berikan sama seperti yang tersimpan didalam *database*.

1. Buat file menu, untuk menampilkan menu program yang dapat diakses oleh *user*.

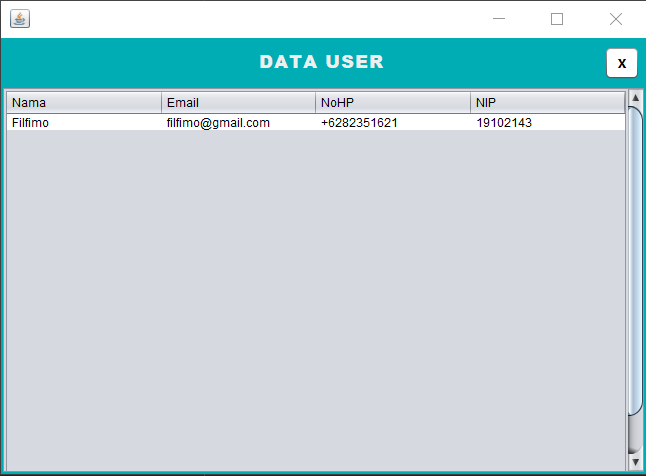


Tampilan menu

Penjelasan *class* fMenu :

Sama seperti sebelumnya, pada kode program diatas sebenarnya didalamnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas hanya adalah untuk fungsi dari GUI saja, seperti *button*, dan *label* yang akan digunakan agar *user* dapat melakukan interaksi dengan program, masing-masing *button* akan menuju ke halaman program yang berbeda. Jadi fungsi lain dari *class* fMenu ini hanya sebagai perantara untuk menampilkan *class* lainnya.

1. Buat file Data, untuk menampilkan data yang terdaftar dalam penggunaan program.

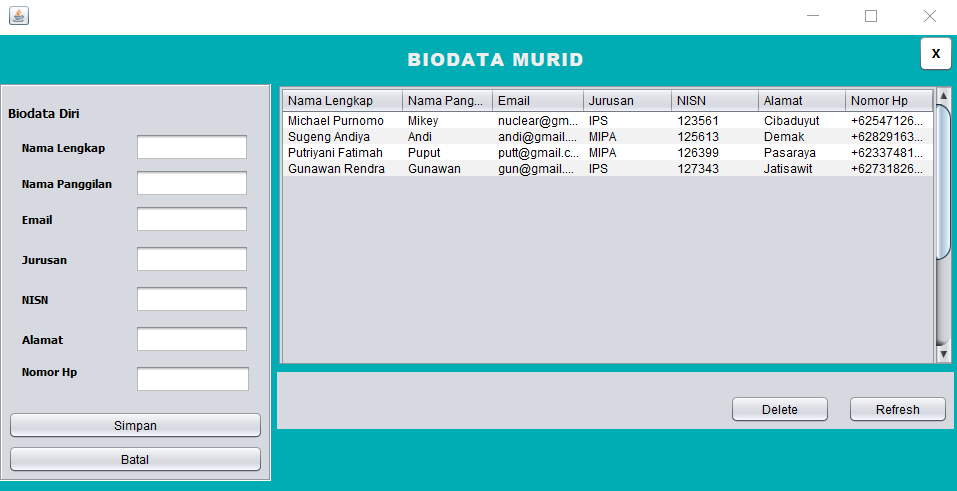


Tampilan menu *data user*

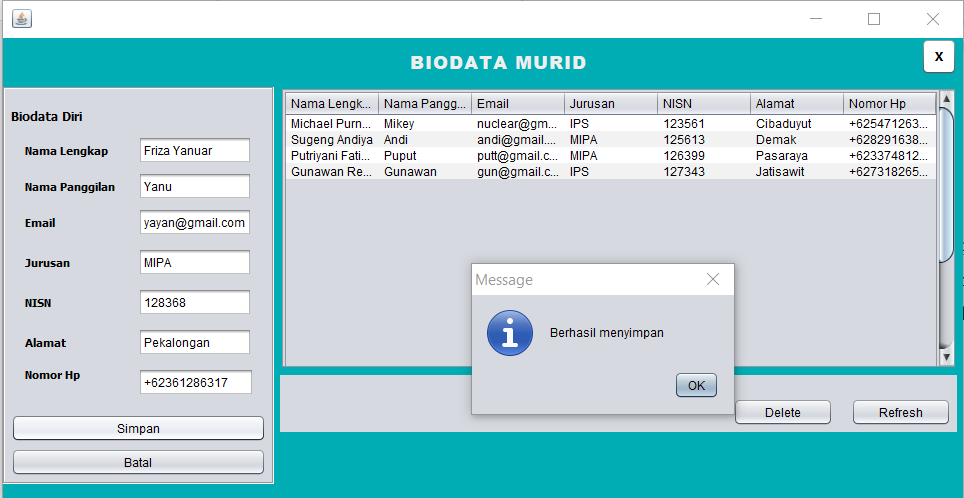
Penjelasan *class* fData :

Sama seperti sebelumnya, pada kode program diatas sebenarnya didalamnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas hanya adalah untuk fungsi dari GUI saja, seperti *tabel*, *close, event* dan *scroll*. Didalam halaman/menu ini *user* dapat melihat data yang terdaftar saat melakukan form *sign up* atau registrasi sebelumnya, data yang ditampilkan berupa nama, email, no hp, dan NIP dan data ini juga adalah data yang tersimpan didalam database. Jika *user* meng-klik tombol X pada pojok kanan maka program akan kembali ke halaman fMenu

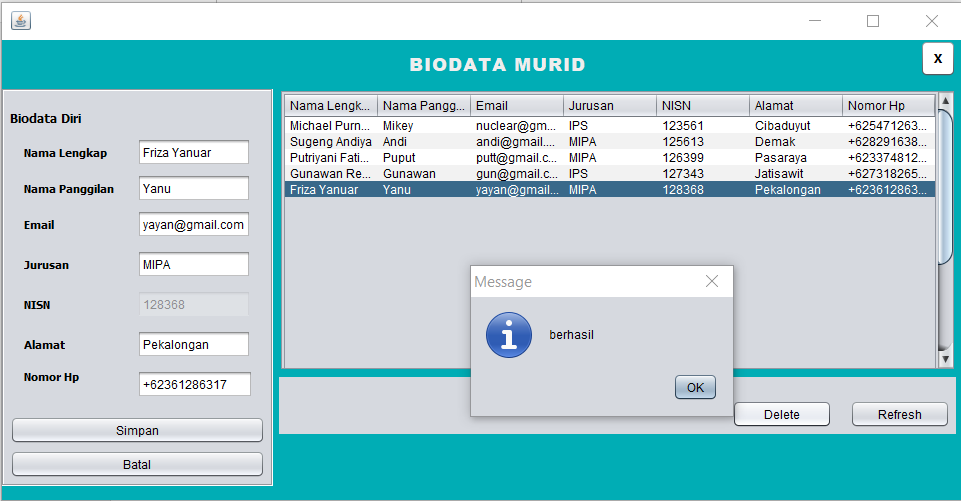
1. Buat file Biodata, untuk melakukan *input* biodata



Tampilan menu biodata



Data yang di*input* berhasil disimpan

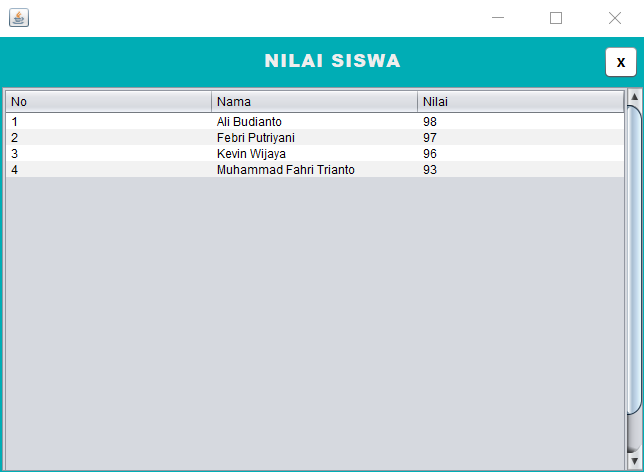


Data yang disimpan berhasil dihapus

Penjelasan *class* fBio :

Sama seperti sebelumnya, pada kode program diatas sebenarnya didalamnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas hanya adalah untuk fungsi dari GUI saja, seperti *tabel*, *close, event,* *scroll, form, split dan button*. Didalam halaman/menu ini *user* dapat melakukan *input* biodata dengan memasukan informasi pada dibagian kiri seperti nama lengkap, nama panggilan, email, jurusan, NISN, alamat, dan nomer hp, kemudian jika sudah klik simpan atau batal jika *user* tidak inginmelakukan *input* biodata, data yang sudah disimpan nanti akan dikirim dan disimpan didalam database, dan dapat ditampilkan pada tabel biodata. Jika data tersebut belum muncul, *user* dapat melakukan *refresh* untuk memperbarui halaman. Dan juga *user* dapat menghapus data yang ada didalam tabel dengan memilih tabel mana yang ingin dihapus dan menekan *button delete*, jika *user* melakukan ini, data yang tersimpan didalam *database* juga ikut terhaapus. Jika *user* meng-klik tombol X pada pojok kanan, maka program akan kembali ke halaman fMenu.

1. Buat file Nilai, untuk menampilkan nilai



Tampilan menu nilai

Penjelasan *class* fNilai :

Sama seperti sebelumnya, pada kode program diatas sebenarnya didalamnya hanya beberapa bagian saja yang dilakukan *coding* oleh Praktikankarena sebenarnya sebagian besar kode program diatas hanya adalah untuk fungsi dari GUI saja, seperti *tabel*, *close, event,* dan *scroll*. Didalam program ini sama seperti pada halaman fData hanya menampilkan data nilai berupa tabel. Disinilah kekurangan pada program ini dimana untuk meng-*input* nilai harus dilakukan langsung didalam *XAMPP* atau *localhost*, tapi mari kita buat sebuah alur atau situasi, anggaplah nilai itu bersifat penting dan perlu perhitungan dari banyak guru karena ini adalah nilai akhir, jadi tidak bisa asal memasukan nilai dan harus menghubungi bagian admin untuk melakukan *upload* nilai kedalam *database*.