**LAPORAN PROYEK TUGAS BESAR**

**MATAKULIAH PEMROGRAMAN BERORIEANTASI OBJEK**



Disusun oleh:

|  |  |
| --- | --- |
| **RIVALDI PRATAMA PUTRA** | **TI19220016** |

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**STMIK LOMBOK**

**2024**

BAB 1. PENDAHULUAN

1.1. Latar Belakang Masalah………………………………………………………… 2

1.2. Struktur Aplikasi……………………………………………………………….. 3

1.3. Pembahasan……………………………………………………………………... 4

# BAB I PENDAHULUAN

## **Latar Belakang Masalah**

Dalam era digital yang terus berkembang pesat, penting bagi pelatih atau manager untuk membuuat aplikasi inputan data pemain futsal yang dapat meningkatkan efisiensi operasional dan meningkatkan pemantauan, ketersediaan, dan keseriuan para pemain. untuk berkonstribusi dalam team. Aplikasi futsal dapat menjadi sumber informasi yang akurat terkait Informasi seperti menginputkan biodata pemain , baik bagi pemain sendiri maupun bagi pihak manajemen lapangan atau penyelenggara turnamen.

Dengan biodata pemain yang tercatat dalam aplikasi, manajemen lapangan atau penyelenggara turnamen dapat dengan mudah berkomunikasi dengan pemain melalui notifikasi atau pesan. Jika ada perubahan dalam biodata pemain seperti nomor telepon atau alamat email, pemain dapat dengan mudah memperbarui informasi tersebut melalui aplikasi tersebut.

Berikut adalah beberapa manfaat yang dapat diperoleh: Aplikasi dapat membantu meminimalkan kesalahan manusia yang seringkali terjadi pada metode manual. Validasi otomatis dan aturan input dapat membantu memastikan bahwa data yang dimasukkan adalah akurat dan sesuai dengan format yang diinginkan. Data yang dimasukkan melalui aplikasi dapat diakses dari berbagai perangkat dan lokasi selama ada koneksi internet.

Hal ini memudahkan pengguna untuk mengakses, memperbarui, atau memantau data di mana saja dan kapan saja. Dengan memanfaatkan otomatisasi dan alat bantu, aplikasi input data dapat meningkatkan produktivitas pengguna, memungkinkan mereka fokus pada tugas-tugas yang lebih kompleks.Dengan adanya Aplikasi tersebut, Team futsal akan menjadi lebih berkembang dan sangat membantu membangun sebuah tim yang lebih kuat menciptakan potensi – potensi yang tak terbatas. karena keberhasilan datang datang dari usaha bersama.

* 1. **Struktur aplikasi**

1. No hp
2. Nama
3. Alamat
4. Umur
5. Id
6. New
7. Delete
8. Add
9. Update
10. View
    1. **Pembahasan**
11. Nomer hp



Inputan nomor handphone (No HP) dalam aplikasi sering digunakan untuk verifikasi identitas pemain. Ini membantu memastikan bahwa pemain yang mendaftar atau menggunakan aplikasi adalah pemilik nomor handphone yang sah.



**String no\_hp = txtno\_hp.getText();**: Baris ini mengambil teks yang dimasukkan ke dalam komponen **txtno\_hp** (kemungkinan berupa JTextField atau sejenisnya) dan menyimpannya dalam variabel **no\_hp**

1. Nama



Inputan nama di aplikasi degan tujuan agar mengetahui nama pemain. Misalnya, sperti menyapa pemain dengan nama mereka. Dan juga dapat memberikan nuansa yang lebih ramah dan membuat pemain merasa dihargai.



**String nama = txtnama.getText();**: Baris ini melakukan hal serupa dengan baris sebelumnya, tetapi untuk komponen **txtnama**

1. Alamat



Menginputkan alamat Pada aplikasi itu agar mempermudah pelatih atau manager ketika pemain mengalami cedera maka pelatih atau manager juga akan lebih mudah dalam mengantar pemain ke alamatnya. Pelatih juga dapat memahami lebih baik kondisi lingkungan atau kehidupan pribadi pemain dengan mengetahui alamat mereka. Ini dapat memungkinkan pendekatan pelatihan yang lebih personal dan sensitif terhadap kebutuhan individu.



**String alamat = txtalamat.getText();**: Baris ini mengambil teks dari komponen **txtalamat** dan menyimpannya dalam variabel **alamat**.

1. Umur



Imputan mur menjadi salah satu informasi penting dalam aplikasi futsal. Umur sangat membantu dalam kategorisasi dan pengelompokan pemain ke dalam kelompok usia tertentu. Pada turnamen atau liga futsal, terdapat pembagian kategori usia untuk memastikan bahwa pertandingan berlangsung dengan adil dan setara dalam hal kemampuan fisik dan teknis. Dalam manajemen tim, informasi umur digunakan untuk mengorganisir dan menyusun



**String umur = txtumur.getText();**: Baris ini mengambil teks dari komponen **txtumur** dan menyimpannya dalam variabel **umur**.

1. Id



Tombol ID sering digunakan untuk merujuk pada catatan atau entitas tertentu dalam data. Dengan menggunakan ID, sistem dapat dengan mudah mencari mengakses data tanpa harus menginput secara manual. Ini juga dapat memastikan bahwa tidak ada duplikat atau tumpang tindih dalam identifikasi.



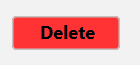
**pst = con.prepareStatement("select id from tbdata");**: Membuat objek **PreparedStatement** dengan pernyataan SQL untuk mengambil semua nilai dari kolom "id" dalam tabel "tbdata".

1. New



proses menginput informasi atau data baru ke dalam suatu sistem atau aplikasi. biasanya New dilakukan untuk memperbarui basis data atau penyimpanan data ke dalam aplikasi dengan informasi terbaru. yang sesuai dengan kebutuhan pemain.

1. Delete



"delete" merupakan tindakan menghapus atau menghilangkan data dari suatu tempat penyimpanan, seperti basis data, sistem file, atau aplikasi.

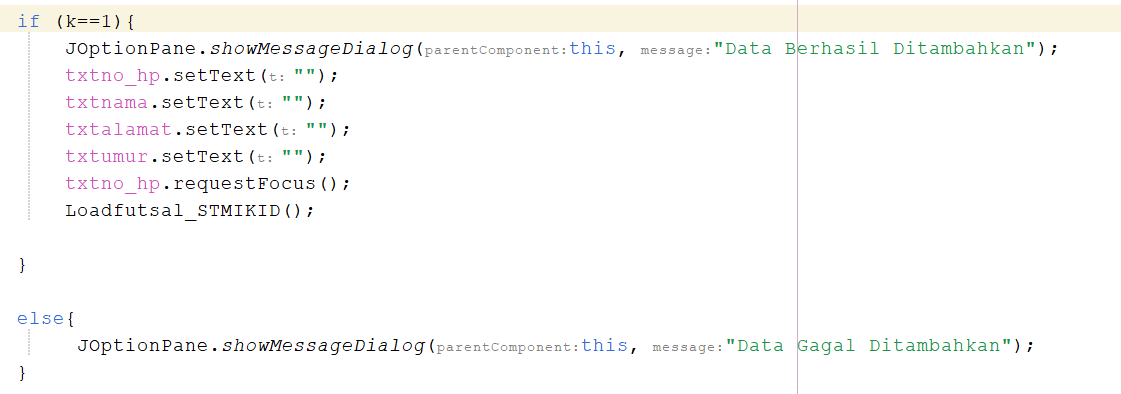


Dengan langkah-langkah ini, code tersebut bersiap untuk menghapus data dari tabel `tbdata` dengan menggunakan `DELETE` SQL statement, di mana kondisi penghapusan adalah berdasarkan nilai `id` yang dipilih dari `JComboBox`.

1. Add



Tombol "add" dapat digunakan untuk menambahkan data ke dalam aplikasi. Ini bisa mencakup menambahkan catatan, kontak, tugas, atau item lainnya sesuai dengan jenis aplikasi tersebut. Dan juga menambahkan anggota baru ke dalam tim atau aplikasi.



Code tersebut adalah bagian dari logika pengkodean yang memberikan umpan balik kepada pengguna setelah menjalankan operasi penambahan data.

**1. \*\*Pengecekan Hasil Penambahan:\*\***

- `if (k == 1)`: Ini adalah kondisi if yang mengevaluasi apakah nilai variabel `k` sama dengan 1.

- Variabel `k` mungkin merupakan hasil dari suatu operasi, seperti `executeUpdate()` pada objek `PreparedStatement` di Java yang berkaitan dengan penggunaan database.

**2. \*\*Dialog Berhasil:\*\***

- `JOptionPane.showMessageDialog(this, "Data Berhasil Ditambahkan")`: Jika kondisi `k == 1` terpenuhi, maka pesan dialog (JOptionPane) akan muncul memberi tahu pengguna bahwa data berhasil ditambahkan.

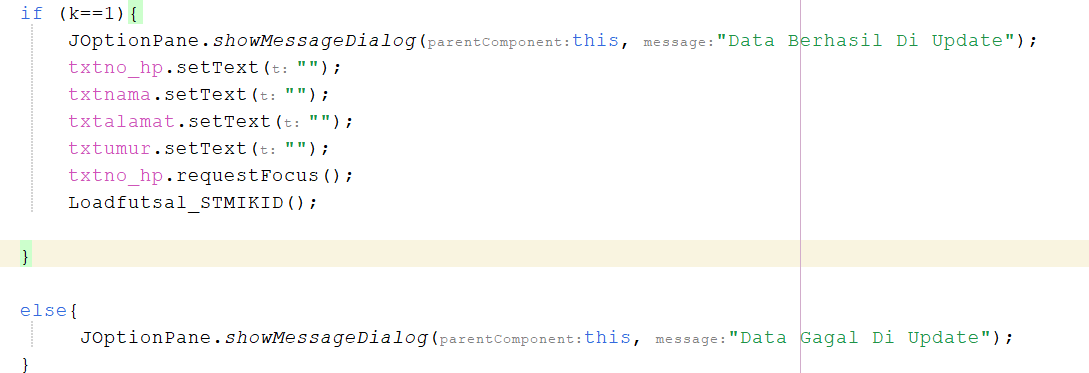
- Parameter pertama (`this`) pada `JOptionPane.showMessageDialog()` menunjukkan bahwa dialog ini terkait dengan komponen atau frame yang sedang aktif (mungkin frame aplikasi atau panel).

Jadi, secara keseluruhan, jika operasi penambahan data berhasil (diindikasikan oleh `k == 1`), maka sebuah pesan dialog akan ditampilkan untuk memberitahu pengguna bahwa data telah berhasil ditambahkan. Umumnya, pesan dialog semacam ini digunakan untuk memberikan umpan balik positif kepada pengguna setelah melakukan tindakan tertentu.

1. Update



Update" digunakan untuk memperbarui atau menyunting informasi yang sudah ada. Update sering digunakan untuk mengubah atau menyunting informasi entitas tertentu untuk menerapkan perubahan atau mengganti data yang sudah diisi sebelumnya pada data pemain.



Code tersebut adalah bagian dari suatu blok pengkodean yang menangani operasi update data pada suatu database.

**1. Pemilihan Data untuk Diupdate**

- `pst = con.prepareStatement("update tbdata set no\_hp=?, nama=?, alamat=?, umur=? where id=?")`: Mendefinisikan suatu pernyataan SQL untuk mengupdate data pada tabel `tbdata`. Pernyataan ini menggunakan placeholder (`?`) yang akan diisi oleh nilai-nilai yang ditentukan nanti.

**2. Penetapan Nilai pada Placeholder.**

- `pst.setString(1, no\_hp)`, `pst.setString(2, nama)`, dst.: Mengisi placeholder dalam pernyataan SQL dengan nilai-nilai yang ingin diupdate. Nilai-nilai ini diambil dari variabel `no\_hp`, `nama`, `alamat`, `umur`, dan `id`. Setiap `setString` sesuai dengan urutan placeholder dalam pernyataan SQL.

**3. Eksekusi Update dan Pengecekan Hasil.**

- `int k = pst.executeUpdate()`: Mengeksekusi pernyataan SQL update. Hasilnya adalah jumlah baris yang terpengaruh oleh operasi update. Nilai ini disimpan dalam variabel `k`.

**4. Pengecekan Hasil dan Umpan Balik kepada Pengguna.**

- `if (k == 1) { ... } else { ... }`: Mengecek apakah operasi update berhasil atau tidak. Jika `k` sama dengan 1, maka pesan dialog (JOptionPane) akan muncul memberitahu pengguna bahwa data berhasil diupdate. Jika tidak, pesan dialog yang memberitahu bahwa operasi update gagal akan muncul.

**5. Reset Input dan Memuat Ulang Data.**

- Jika update berhasil (`k == 1`), beberapa komponen input seperti `txtno\_hp`, `txtnama`, dst., direset menjadi string kosong.

- `txtno\_hp.requestFocus()`: Memindahkan fokus kursor ke komponen `txtno\_hp`.

- `Loadfutsal\_STMIKID()`: Mungkin sebuah panggilan metode atau fungsi yang bertugas untuk memuat ulang data, memastikan tampilan aplikasi selalu terupdate dengan data terbaru.

1. View



"View" dapat digunakan untuk melihat detail lengkap dari suatu entitas atau objek. View dapat membuka data pemain yang sudah menginputkan datanya pada aplikasi tersebut yang berisi mulai dari :

* No\_hp
* Nama
* Alamat
* Umur.

try {

String id = jComboBox1.getSelectedItem().toString();

pst = con.prepareStatement("select \* from tbdata where id=?");

pst.setString(1, id);

rs = pst.executeQuery();

if (rs.next()==true){

txtno\_hp.setText(rs.getString(2));

txtnama.setText(rs.getString(3));

txtalamat.setText(rs.getString(4));

txtumur.setText(rs.getString(5));

}

} catch (SQLException ex) {

Logger.getLogger(futsal\_STMIK.class.getName()).log(Level.SEVERE, null, ex);

}

View terkait dengan objek tertentu yang dipilih oleh pengguna. Misalnya, jika ada tabel atau daftar item, tombol ini mungkin akan menampilkan informasi terkait item yang dipilih.