

UJIAN AKHIR SEMESTER
PEMROGRAMAN II



Disusun Oleh :

NAMA : Yafi Mahadika
NIM : 221011400189
KELAS : 06TPLE004 Reg CS

TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 46 Telp. (021) 7412566
Buaran Serpong – Tangerang Selatan



UNIVERSITAS PAMULANG
KARTU UJIAN AKHIR SEMESTER GENAP 2024/2025
NOMOR UJIAN : 01215272586557

FAKULTAS / PRODI : ILMU KOMPUTER / TEKNIK INFORMATIKA S1

NAMA MAHASISWA : YAFI MAHADIKA

NIM : 221011400189

SHIFT : REGULER C

No	Hari/ Tanggal	Waktu	Ruang	Kelas	Mata Kuliah	Paraf
1	Sabtu, 5 Jul 2025	07.40 - 09.20	V.408	06TPLE004	REKAYASA PERANGKAT LUNAK	1
2	Sabtu, 5 Jul 2025	07.40 - 09.20	V.408	06TPLE004	TEKNOLOGI INTERNET OF THINGS	2
3	Sabtu, 5 Jul 2025	09.20 - 11.00	V.408	06TPLE004	MOBILE PROGRAMMING	3
4	Sabtu, 5 Jul 2025	09.20 - 11.00	V.408	06TPLE004	SISTEM PENDUKUNG KEPUTUSAN	4
5	Sabtu, 5 Jul 2025	11.00 - 13.50	V.408	06TPLE004	KERJA PRAKTEK	5
6	Sabtu, 5 Jul 2025	11.00 - 13.50	V.408	06TPLE004	PEMROGRAMAN II	6
7	Sabtu, 5 Jul 2025	13.50 - 15.30	V.408	06TPLE004	BASIS DATA II	7
8	Sabtu, 5 Jul 2025	13.50 - 15.30	V.408	06TPLE004	TEKNIK KOMPILASI	8

Peraturan dan Tata Tertib Peserta Ujian

1. Peserta ujian harus berpakaian rapi, sopan dan memakai jaket Almamater
2. Peserta ujian sudah berada di ruangan sepuluh menit sebelum ujian dimulai
3. Peserta ujian yang terlambat diperkenankan mengikuti ujian setelah mendapat ijin, tanpa perpanjangan waktu
4. Peserta ujian hanya diperkenankan membawa alat-alat yang ditentukan oleh panitia ujian
5. Peserta ujian dilarang membantu teman, mencontoh dari teman dan tindakan-tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Tangerang Selatan, 1 Juli 2025
Ketua Panitia Ujian

Dr. Ubaid Al Faruq, S.Pd., M.Pd.
NIDN. 0418028702

PENJELASAN

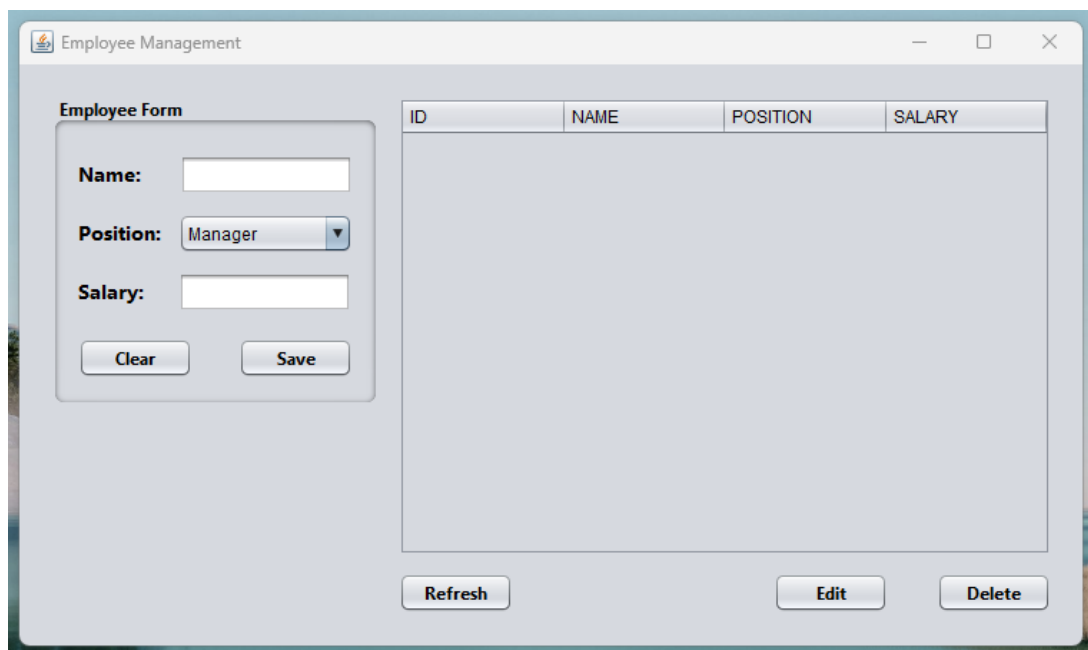
1. Tampilan Awal Aplikasi - Formulir dan Tabel Data

Tampilan awal aplikasi ini menunjukkan form input karyawan dan tabel kosong yang akan menampilkan data karyawan dari database. Aplikasi ini dibangun menggunakan Java Swing dan terhubung dengan MySQL.

Di sisi kiri terdapat **form input** berisi:

- Field **Name** untuk memasukkan nama karyawan.
- Field **Position** berupa dropdown (combo box) berisi pilihan jabatan seperti "Manager", "Developer", dsb.
- Field **Salary** untuk menginputkan gaji (tipe angka).

Di bagian bawah form, terdapat tombol **Clear** untuk menghapus isi form, dan **Save** untuk menyimpan data ke database. Di sisi kanan adalah **tabel data karyawan** (JTable) yang menampilkan kolom ID, Name, Position, dan Salary.

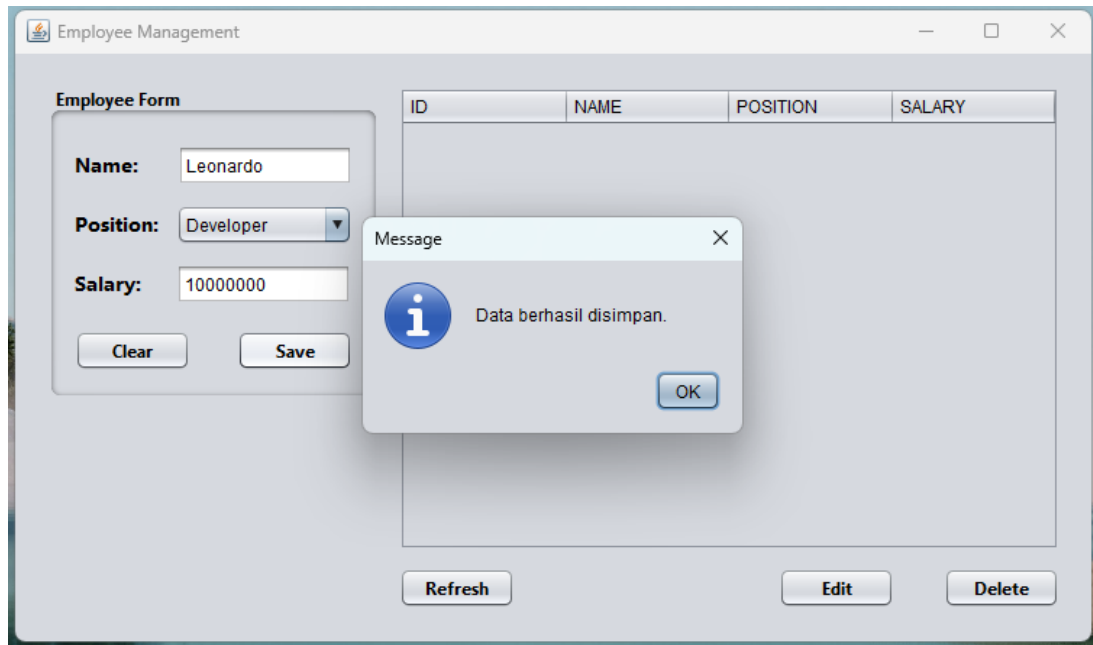


Gambar 1 Tampilan awal aplikasi Employee Management beserta form input dan tabel kosong.

2. Proses Menyimpan Data Baru

Setelah user mengisi field form (contoh: Name: Leonardo, Position: Developer, Salary: 10000000) lalu menekan tombol Save, maka akan muncul notifikasi bahwa data berhasil disimpan.

Fungsi *saveActionPerformed()* akan mengeksekusi proses validasi dan menjalankan perintah *INSERT INTO* ke tabel *karyawan*. Notifikasi ini ditampilkan dengan *JOptionPane.showMessageDialog*.



Gambar 2 Notifikasi setelah berhasil menyimpan data karyawan baru.

3. Data Tersimpan dan Ditampilkan di Tabel

Setelah menyimpan data, sistem secara otomatis memuat ulang (refresh) isi tabel dengan memanggil ulang data dari database. Kita bisa melihat bahwa data **Leonardo** berhasil masuk ke dalam tabel dengan posisi **Developer** dan salary 5000000.0. Kotak merah yang ditandai pada tombol **Edit** menunjukkan bahwa pengguna memiliki opsi untuk **mengubah data** karyawan yang telah tersimpan. Untuk melakukan pengeditan, pengguna harus terlebih dahulu memilih salah satu baris data dari tabel. Setelah baris dipilih dan tombol **Edit** ditekan, maka aplikasi akan membuka jendela atau form baru (EditForm) yang sudah otomatis terisi dengan data karyawan tersebut. Form ini memungkinkan pengguna untuk mengubah nama, posisi, atau gaji dan kemudian menyimpan kembali perubahan ke database.

The screenshot shows a window titled "Employee Management". On the left is an "Employee Form" with fields for Name, Position (a dropdown menu showing "Manager"), and Salary, along with "Clear" and "Save" buttons. On the right is a table with the following data:

ID	NAME	POSITION	SALARY
5	Leonardo	Developer	5000000.0

At the bottom of the window are three buttons: "Refresh", "Edit" (highlighted with a red box), and "Delete".

Gambar 3 Data karyawan yang berhasil ditampilkan pada tabel setelah disimpan.

4. Proses Edit Data - Form Edit

Saat user memilih baris pada tabel dan menekan tombol **Edit**, maka akan terbuka **jendela EditForm**. Form ini memungkinkan pengguna untuk memperbarui data karyawan yang dipilih. Data dari baris yang dipilih akan otomatis ditampilkan dalam field EditForm. Setelah mengubah data (contohnya salary menjadi 7000000.0) dan menekan tombol **Save**, data akan diperbarui di database dengan perintah *UPDATE*.

The screenshot shows the same "Employee Management" window, but with an "EDIT FORM" dialog box open in the foreground. The dialog box contains the following information:

EDIT FORM

Name: Leonardo

Position: Developer

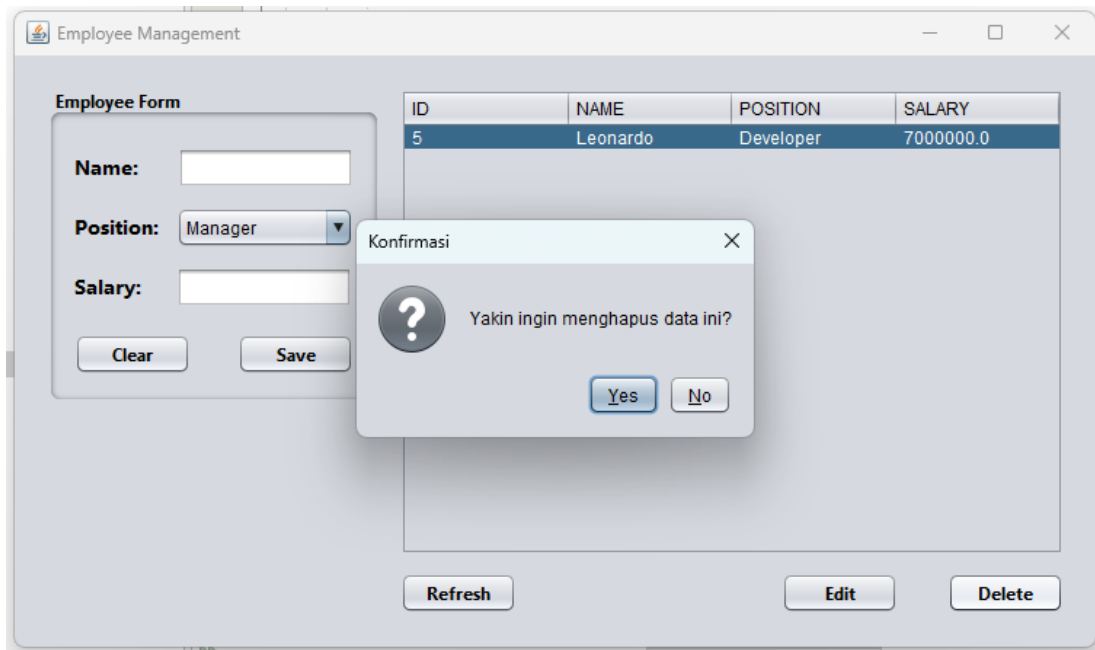
Salary: 7000000.0

At the bottom of the dialog box are two buttons: "SAVE" (highlighted with a red box) and "CANCEL".

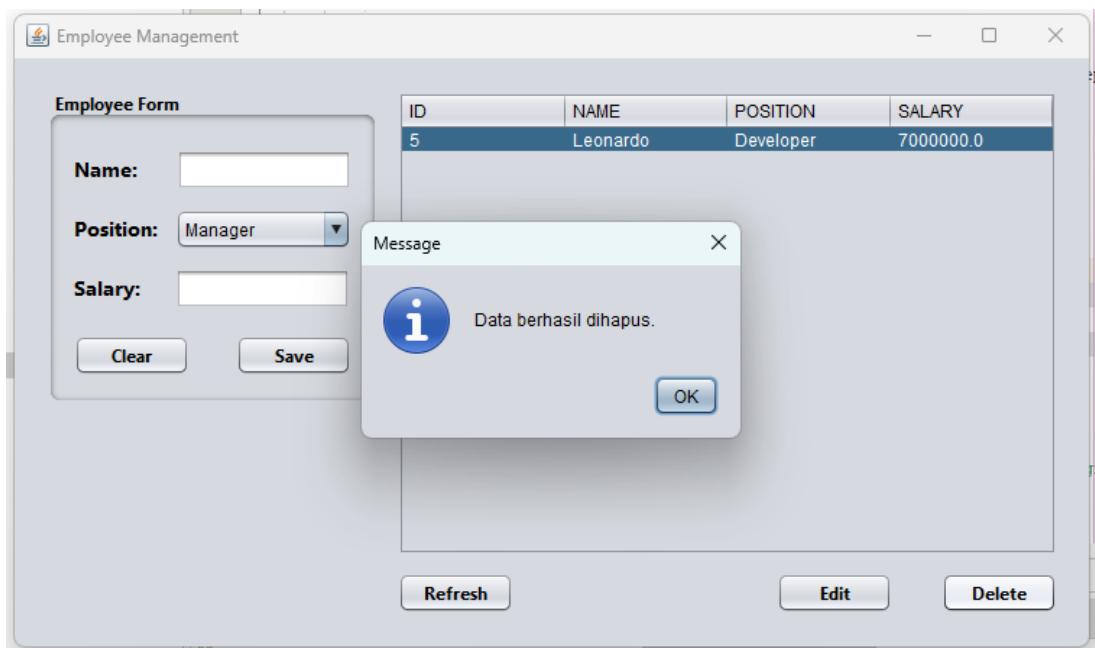
Gambar 4 Tampilan form edit saat melakukan perubahan data karyawan.

5. Proses Hapus Data – Konfirmasi

Saat user memilih baris dan menekan tombol **Delete**, aplikasi akan menampilkan konfirmasi menggunakan `JOptionPane.showConfirmDialog` untuk memastikan bahwa pengguna benar-benar ingin menghapus data. Jika menekan "Yes", maka baris akan dihapus dari database menggunakan query `DELETE FROM karyawan WHERE id=?`.



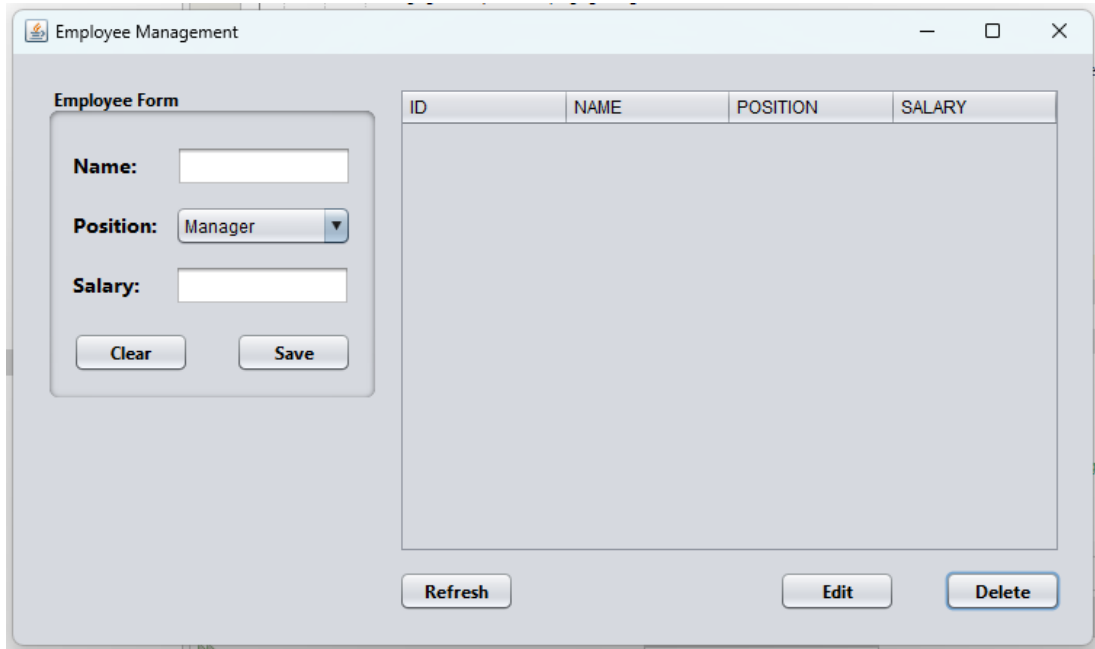
Gambar 5 Konfirmasi penghapusan data sebelum proses delete dijalankan.



Gambar 6 Notifikasi bahwa data berhasil dihapus dari database.

6. Tampilan Setelah Data Dihapus

Setelah proses penghapusan, tabel kembali kosong. Ini menandakan bahwa tidak ada data tersisa di tabel *karyawan*, dan sistem berhasil menyinkronkan data dari database MySQL.



The screenshot shows a web application window titled "Employee Management". On the left is an "Employee Form" with fields for "Name:", "Position:" (a dropdown menu currently showing "Manager"), and "Salary:". Below these fields are "Clear" and "Save" buttons. On the right is a table with four columns: "ID", "NAME", "POSITION", and "SALARY". The table is currently empty. At the bottom of the window are three buttons: "Refresh", "Edit", and "Delete".

Gambar 7 Tabel kembali kosong setelah semua data dihapus.

7. Penjelasan Database

Tabel *karyawan* memiliki empat kolom utama sebagai berikut:

- **id (int)**
Digunakan sebagai ID unik tiap karyawan, otomatis bertambah (AUTO_INCREMENT), dan tidak boleh kosong.
- **nama (varchar 100)**
Menyimpan nama lengkap karyawan, maksimal 100 karakter, dan wajib diisi.
- **posisi (varchar 50)**
Berisi jabatan karyawan seperti "Manager" atau "Developer", maksimal 50 karakter, dan tidak boleh kosong.
- **gaji (double)**
Digunakan untuk menyimpan jumlah gaji dalam bentuk angka desimal, dan wajib diisi.

KESIMPULAN

Program *Employee Management* yang dibangun menggunakan Java Swing dan MySQL ini mampu melakukan proses CRUD (Create, Read, Update, Delete) terhadap data karyawan secara efisien. Melalui antarmuka yang sederhana dan interaktif, pengguna dapat menambahkan data karyawan, mengedit informasi, menghapus data, serta melihat daftar karyawan secara real-time dari database. Sistem ini mempermudah pengelolaan data karyawan dalam skala kecil hingga menengah dan dapat dikembangkan lebih lanjut sesuai kebutuhan. Implementasi koneksi database juga berhasil dilakukan sehingga data yang diolah tersimpan secara permanen.