

LAPORAN UJIAN AKHIR SEMESTER
PERANCANGAN APLIKASI TOKO MAINAN HAPPY
MENGGUNAKAN GUI JAVA



Dosen Pengampu :
I Made Gede Sri Artha, S.T.

Oleh :
Luh Natih Anamaya Santi (2201010156)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN TEKNOLOGI INDONESIA

2024

PEMBAHASAN

Deskripsi Proyek

Aplikasi Toko Mainan adalah sistem yang mengelola data barang di toko mainan. Aplikasi ini dirancang untuk menyederhanakan proses manajemen stok dan data mainan. Teori enkapsulasi digunakan untuk melindungi dan mengatur akses terhadap data penting dalam aplikasi.

Fitur

Manajemen Inventaris: Menambah, menghapus, dan mengubah data mainan.

Transaksi Penjualan: Mencatat dan mengelola transaksi penjualan mainan.

Laporan Penjualan: Membuat laporan penjualan harian, mingguan, dan bulanan.

Penerapan Teori OOP: Enkapsulasi

Private Access Modifier: Digunakan untuk melindungi data dalam kelas Toy dan ToyStore agar tidak dapat diakses secara langsung dari luar kelas.

Public Access Modifier: Digunakan untuk metode yang perlu diakses dari luar kelas, seperti getter dan setter untuk data mainan serta metode untuk menambah, menghapus, menampilkan inventaris mainan, dan melakukan penjualan.

Desain Aplikasi

A. Tampilan

Kode	Nama Mainan	Harga	Stok Mainan
------	-------------	-------	-------------

Di menu tampilan (dashboard) ini terdapat form yang berisikan kode mainan, nama mainan, harga mainan dan stok mainan. Lalu ada button Add, Clear dan Delete.

B. Validasi Input (Button Add)


The screenshot shows the 'Toko Mainan Happy' application interface. On the left, there is a form titled 'Data Mainan' with four input fields: 'Kode Mainan', 'Nama Mainan', 'Harga Mainan', and 'Stok Mainan'. Below these fields are two buttons: '+ Add' and 'Clear'. To the right of the form is a 'Delete' button. A modal dialog box is displayed in the center, with the title 'Coba lagi' and a red 'X' icon. The message inside the dialog is 'Isi semua fields' (Fill all fields). The background table is partially visible, showing headers 'Harga' and 'Stok Mainan'.

The screenshot shows the 'Toko Mainan Happy' application interface after a successful data entry. The 'Data Mainan' form is on the left, and the 'Delete' button is on the right. The background table now displays a single row of data:

Kode	Nama Mainan	Harga	Stok Mainan
M001	Jeep Hot Wheels	200.000	36


Kode yang saya berikan akan memvalidasi input dari pengguna sebelum memprosesnya lebih lanjut. Jika salah satu field kosong, maka pesan kesalahan ditampilkan dan proses penambahan data dihentikan. Ini menunjukkan bagaimana data divalidasi sebelum dimasukkan ke dalam objek, menjaga integritas data dalam aplikasi.

C. Membersihkan Value Pada Field



The screenshot shows a web application window titled "Dashboard" for "Toko Mainan Happy". On the left, there is a "Data Mainan" form with four input fields: "Kode Mainan" (containing "M001"), "Nama Mainan" (containing "Jeep Hot Wheels"), "Harga Mainan" (containing "200.000"), and "Stok Mainan" (containing "36"). Below these fields are two buttons: a light blue "Add" button with a plus icon and a dark blue "Clear" button with a trash icon. To the right of the form is a "Delete" button with a trash icon. Below the buttons is a table with four columns: "Kode", "Nama Mainan", "Harga", and "Stok Mainan". The table body is currently empty.

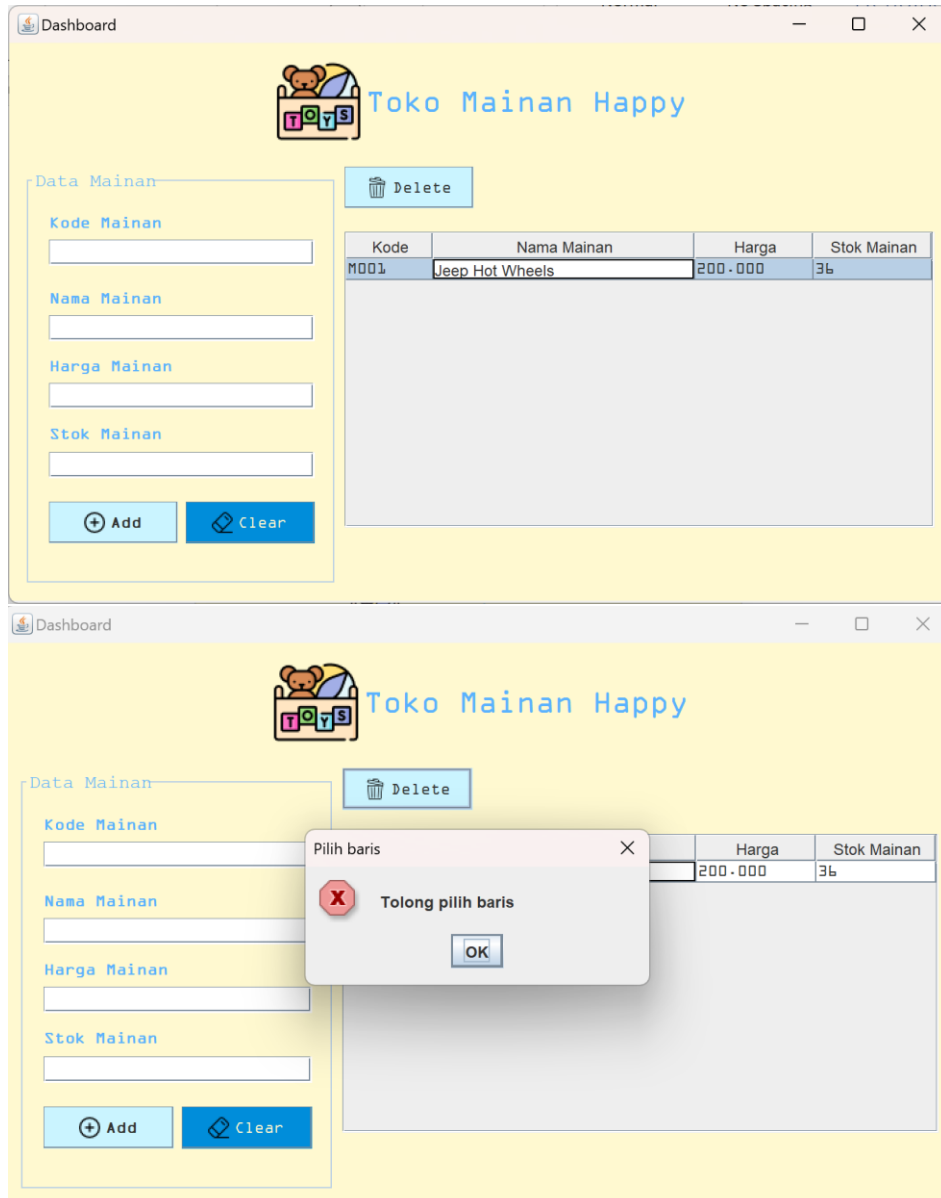
Selanjutnya pada Button clear, jika pengguna menekan tombol clear maka semua value dari field akan dibersihkan



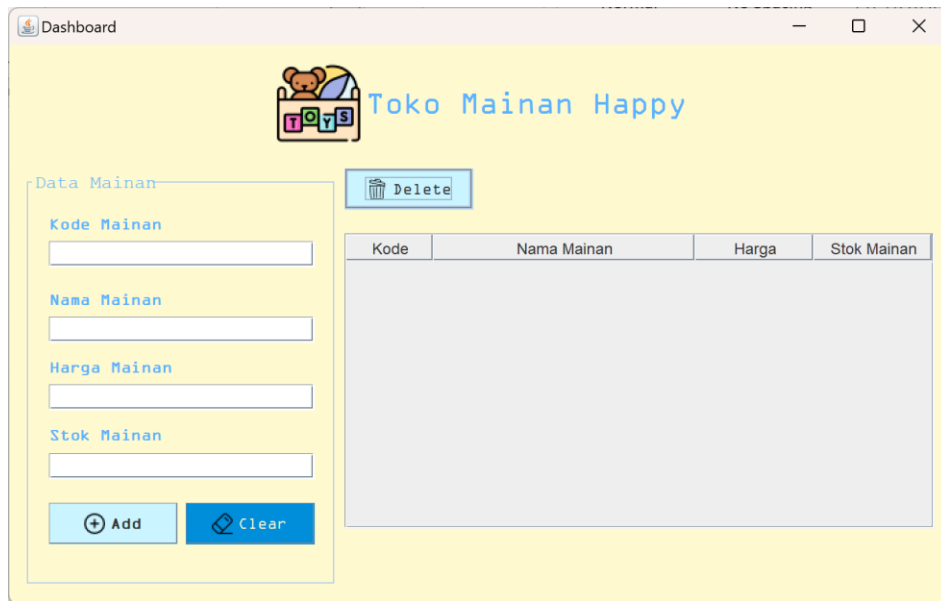
This screenshot shows the same "Toko Mainan Happy" dashboard after the "Clear" button was pressed. The "Data Mainan" form now has four empty input fields for "Kode Mainan", "Nama Mainan", "Harga Mainan", and "Stok Mainan". The "Add" and "Clear" buttons remain at the bottom of the form. The "Delete" button is still present to the right. The table below the form remains empty.

Pada *Button Clear*, saya menggunakan komponen GUI seperti `txKode`, `txNama`, `txHarga`, dan `txStok` yaitu elemen yang hanya bisa diakses dan dimodifikasi melalui metode `setText("")`. Ini menunjukkan bahwa akses dan modifikasi data dikontrol melalui metode, yang merupakan bentuk enkapsulasi.

D. Delete Table



Jika tidak ada baris yang dipilih (nilai row kurang dari nol), maka akan muncul sebuah pop up error message menggunakan JOptionPane. Pop up ini memberitahu pengguna untuk memilih baris sebelum melanjutkan operasi penghapusan.



Jika ada baris yang dipilih (nilai row tidak kurang dari nol), maka baris tersebut dihapus dari model tabel menggunakan `removeRow(row)` dari `DefaultTableModel`. Dengan cara ini, data yang terkait dengan baris yang dipilih akan dihapus dari tampilan tabel.

- Dalam kode Java yang saya buat, terdapat beberapa contoh penerapan konsep enkapsulasi :

1. **Private Access Modifiers:** Beberapa variabel instance seperti `txKode`, `txNama`, `txHarga`, dan `txStok` dideklarasikan sebagai `private`. Hal ini membuat variabel-variabel tersebut hanya dapat diakses dan dimodifikasi langsung oleh kelas `MainanFrame` saja, dan tidak dapat diakses dari kelas lain secara langsung.

```
// Variables declaration - do not modify
private javax.swing.JButton btnHapus;
private javax.swing.JPanel jPanel2;
private javax.swing.JTable tabelMainan;
private javax.swing.JTextField txHarga;
private javax.swing.JTextField txKode;
private javax.swing.JTextField txNama;
private javax.swing.JTextField txStok;
// End of variables declaration
```

2. **Method Encapsulation:** Beberapa method, seperti `btnTambahActionPerformed`, `btnBersihkanActionPerformed`, `formWindowClosing`, `formWindowOpened`, dan `btnHapusActionPerformed`, menerapkan logika tertentu yang memanipulasi data dalam kelas `MainanFrame`. Dengan mengenkapsulasi logika ini dalam method-method tersebut, kontrol atas operasi-operasi tersebut dapat lebih terstruktur dan tidak langsung dapat diakses dari luar kelas.

```
private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
```

3. **Serialization:** Penggunaan `ObjectOutputStream` dan `ObjectInputStream` dalam method `formWindowClosing` dan `formWindowOpened` untuk menyimpan dan memuat data dari file `file.bin` menggambarkan penggunaan enkapsulasi data untuk menyimpan status aplikasi dan menghindari manipulasi langsung terhadap struktur data tabel (`DefaultTableModel`) yang ditampilkan.

```
private void formWindowClosing(java.awt.event.WindowEvent evt) {  
    DefaultTableModel model = (DefaultTableModel) tabelMainan.getModel();  
    Vector<Vector> tableData = model.getDataVector();  
  
    try{  
        FileOutputStream file = new FileOutputStream("file.bin");  
        ObjectOutputStream output = new ObjectOutputStream(file);  
  
        output.writeObject(tableData);  
  
        output.close();  
        file.close();  
    } catch (Exception ex){  
        ex.printStackTrace();  
    }  
}
```