**Laporan Tugas Proyek**

**Pemrograman Berorientasi Objek**

**Aplikasi Minimarket**



**Disusun Oleh :**

|  |  |
| --- | --- |
| 11322025 | Juan Saut Pandapotan Sitorus |
| 11322030 | Mario Andreas Manurung |
| 11322035 | Niko Alvin Simanjuntak |
| 11322040 | Cristian Nicolas Tambunan |

**Institut Teknologi Del**

**Fakultas Vokasi**

**DIII- Teknologi Informasi**

**Laguboti**

**2023**

JUDUL/TOPIK PER KELOMPOK

Minimarket

RUBRIK PROYEK

1. Beri deskripsi secara singkat ada berapa dan uraian mengenai fungsi yang anda buat dalam proyek ini (Minimal 5 fungsi).
2. Jelaskan penerapan konsep OOP yang anda terapkan pada proyek anda kemudian sertakan screenshot dari sistem yang anda bangun berdasarkan point berikut:
   1. Abstraction
   2. Encapsulation
   3. Inheritance
   4. Polymorphism
   5. Interface
   6. Exception handling
3. Penjelasan hasil implementasi keseluruhan
4. Rekam video penjelasan dari No 1 dan No 2. Upload pada Youtube dan masukkan link tersebut kedalam laporan ini.
5. Pengumpulan laporan dan *sourcecode* adalah hari Minggu, 17 Desember 2023, 23.59 WIB di E-Course. Template *filename*:

3digitbelakangNIM1\_3digitbelakangNIM2\_3digitbelakangNIM3\_3digitbelakangNIM4-Proyek PBO-Topik.

Contoh: 005\_010\_015\_020-Proyek PBO-Minimarket.zip  
Keterlambatan pengumpulan akan mendapatkan pengurangan point sebanyak berapa menit anda terlambat.

1. **FUNCTIONAL DESCRIPTION**

a. initializeDatabase():

Purpose: Inisialisasi koneksi database ke MySQL.

Usage: Memastikan koneksi database siap digunakan.

b. fetchDataFromDatabase():

Purpose: Mengambil data dari tabel barang dalam database.

Usage: Mengisi TableView pada tampilan pengguna dengan data barang.

c. handleItemSelection(MouseEvent event):

Purpose: Menangani pemilihan item di TableView.

Usage: Menampilkan informasi terkait barang saat dipilih.

d. handleCompleteTransaction():

Purpose: Menangani penyelesaian transaksi.

Usage: Menyimpan transaksi ke database, termasuk detail transaksi.

e. loadImageFromBlob(Blob blob):

Purpose: Memuat gambar dari tipe data BLOB dalam database.

Usage: Menampilkan gambar barang dalam antarmuka pengguna.

f. handleAddToCart():

Purpose: Menangani logika penambahan item ke keranjang belanja.

Usage: Memeriksa ketersediaan stok, menghitung total harga belanja, dan menambahkan item ke dalam keranjang.

g. handleRemoveFromCart():

Purpose: Menangani penghapusan item dari keranjang belanja.

Usage: Mengurangkan total harga belanja dan memperbarui antarmuka pengguna.

h. updateTotalHargaLabel():

Purpose: Memperbarui label total harga di antarmuka pengguna.

i. showAlert(String title, String content):

Purpose: Menampilkan kotak dialog peringatan.

Usage: Digunakan untuk memberikan pesan peringatan kepada pengguna.

j. initializeUserView():

Purpose: Menginisialisasi tampilan pengguna.

Usage: Melibatkan pengambilan data dari database dan persiapan antarmuka pengguna.

k. isUsernameUnique(String emailText):

Purpose: Memeriksa apakah nama pengguna (email) unik.

Usage: Digunakan dalam logika pendaftaran pengguna.

l. registerUser(String username, String email, String password):

Purpose: Mendaftarkan pengguna baru ke dalam database.

Usage: Memeriksa keunikan nama pengguna dan memasukkan data pengguna ke dalam tabel akun.

m. showAlert(String title, String content):

Purpose: Menampilkan kotak dialog peringatan.

Usage: Digunakan dalam logika pendaftaran pengguna.

n. clearFields():

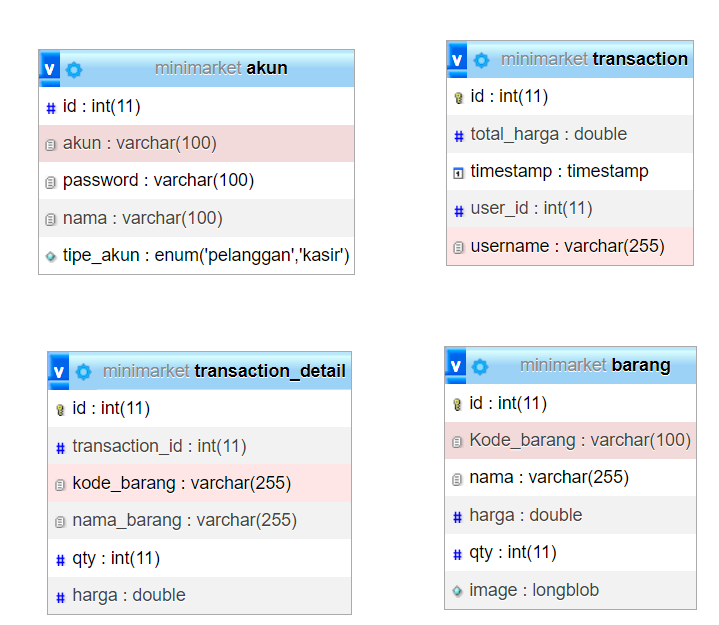
Purpose: Menghapus konten dari semua bidang masukan pada tampilan pendaftaran.

Usage: Dipanggil setelah pendaftaran pengguna berhasil.

o. initialize(URL url, ResourceBundle rb):

Purpose: Inisialisasi komponen saat tampilan dimuat.

1. **DESIGN TABLE STRUCTURE**

****

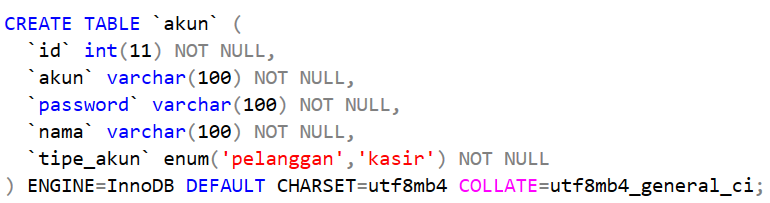
1. **Tabel akun:**
2. Struktur Tabel:

* Kolom: id, akun, password, nama, tipe\_akun
* id sebagai primary key.
* tipe\_akun merupakan ENUM dengan nilai 'pelanggan' atau 'kasir'.

1. Data Tabel:

* Terdapat empat baris data dengan kolom-kolom yang disebutkan.

1. Query Pembuatan:



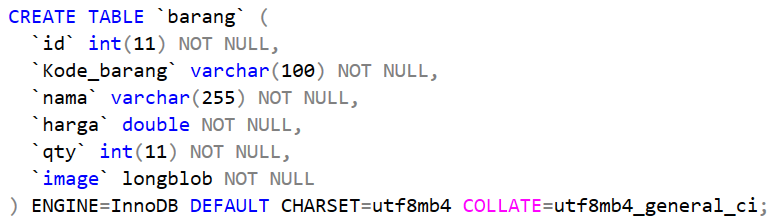
1. **Tabel barang:**
2. Struktur Tabel:

* Kolom: id, Kode\_barang, nama, harga, qty, image
* id sebagai primary key.

1. Data Tabel:

* Terdapat dua puluh tujuh baris data dengan kolom-kolom yang disebutkan.

1. Query Pembuatan:



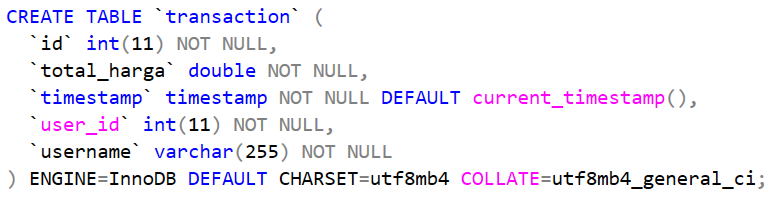
1. **Tabel transaction\_detail::**
2. Struktur Tabel:

* Kolom: id, transaction\_id, kode\_barang, nama\_barang, qty, harga
* id sebagai primary key.
* transaction\_id terhubung sebagai foreign key ke tabel transaction.

1. Data Tabel:

* Data tidak diberikan.

1. Query Pembuatan:



1. **Indeks dan Auto Increment:**

* Setiap tabel memiliki primary key dan indeks yang sesuai.
* Terdapat pengaturan AUTO\_INCREMENT untuk kolom id pada setiap tabel.

1. **Foreign Key Constraints:**

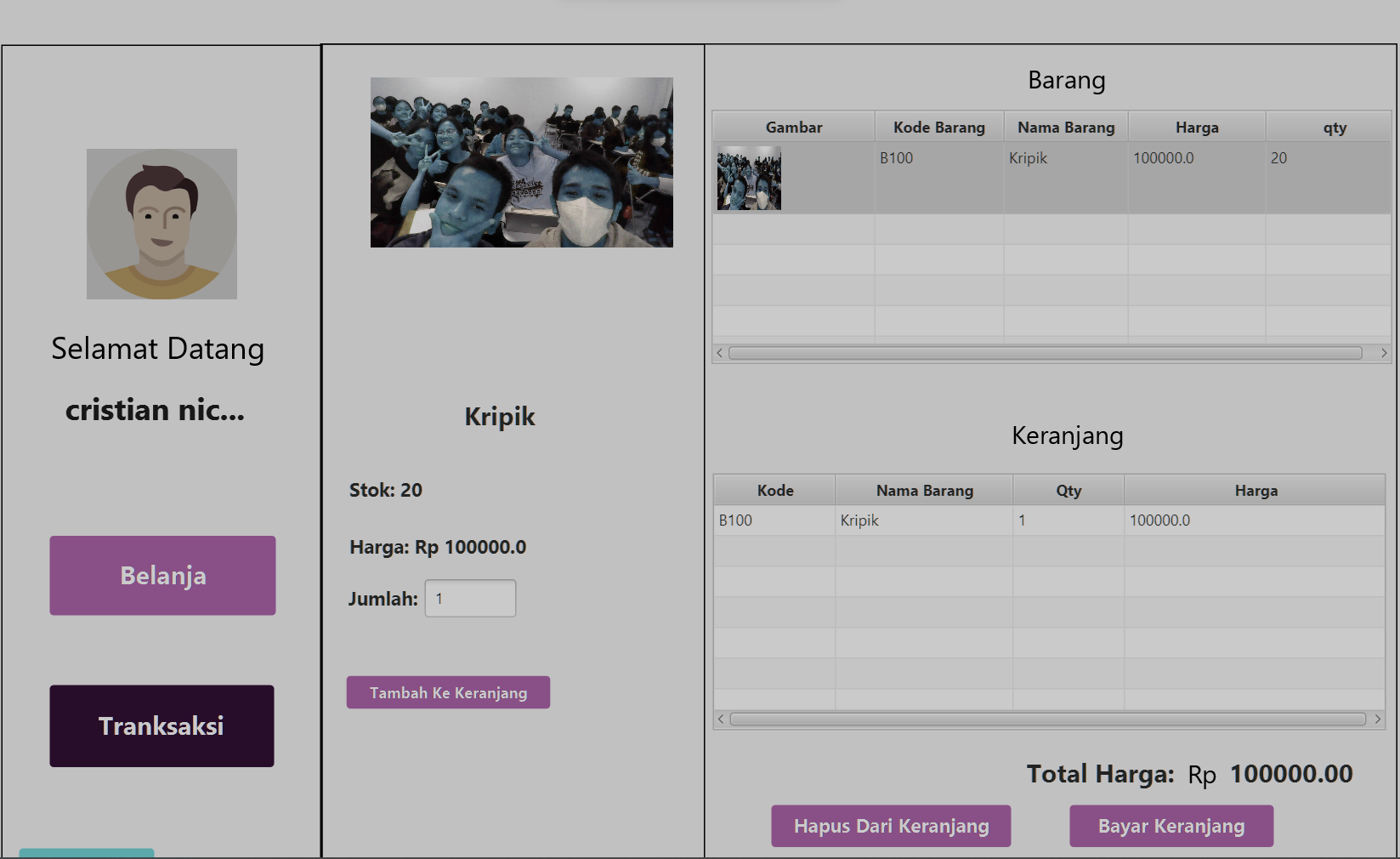
* Tabel transaction memiliki foreign key constraint terhadap kolom user\_id yang terhubung dengan tabel akun.

1. **Catatan:**

* Beberapa baris data ditampilkan untuk tabel akun dan barang sebagai contoh.
* Data untuk tabel transaction dan transaction\_detail tidak diberikan.

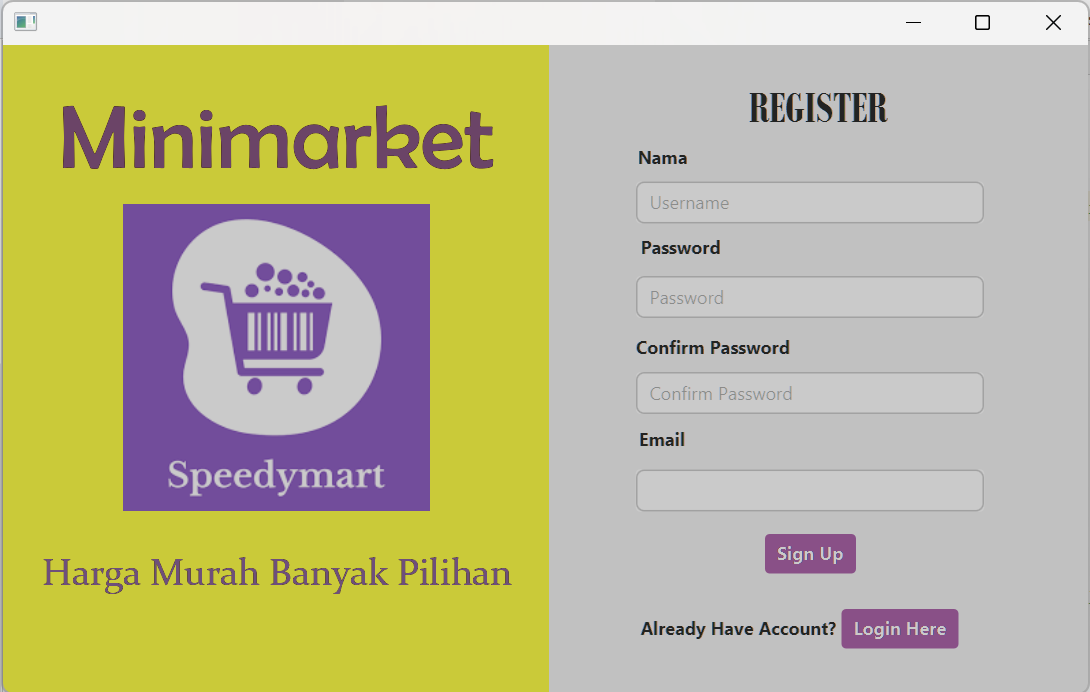
1. **OOP IMPLEMENTATION**
   1. **Abstraction**

Abstraksi terjadi saat kami menggunakan kelas dan objek untuk menyembunyikan detail implementasi dan hanya mengekspos fungsionalitas yang diperlukan. Contoh: Penggunaan kelas Barang dan CartItem sebagai representasi abstrak dari barang dan item dalam keranjang belanja.



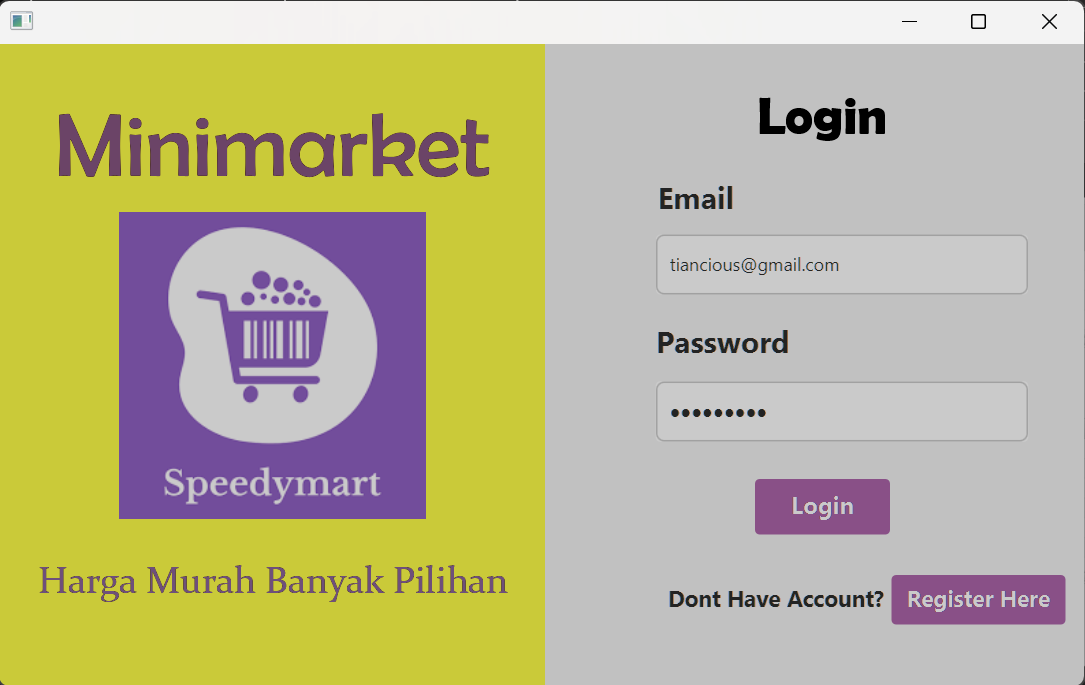
* 1. **Encapsulation**

Encapsulation terjadi saat kami mengelompokkan data dan metode yang beroperasi pada data tersebut dalam satu unit yang disebut kelas. Contoh: Penggunaan kelas registerController dan userController untuk mengelola logika pendaftaran pengguna dan tampilan pengguna.



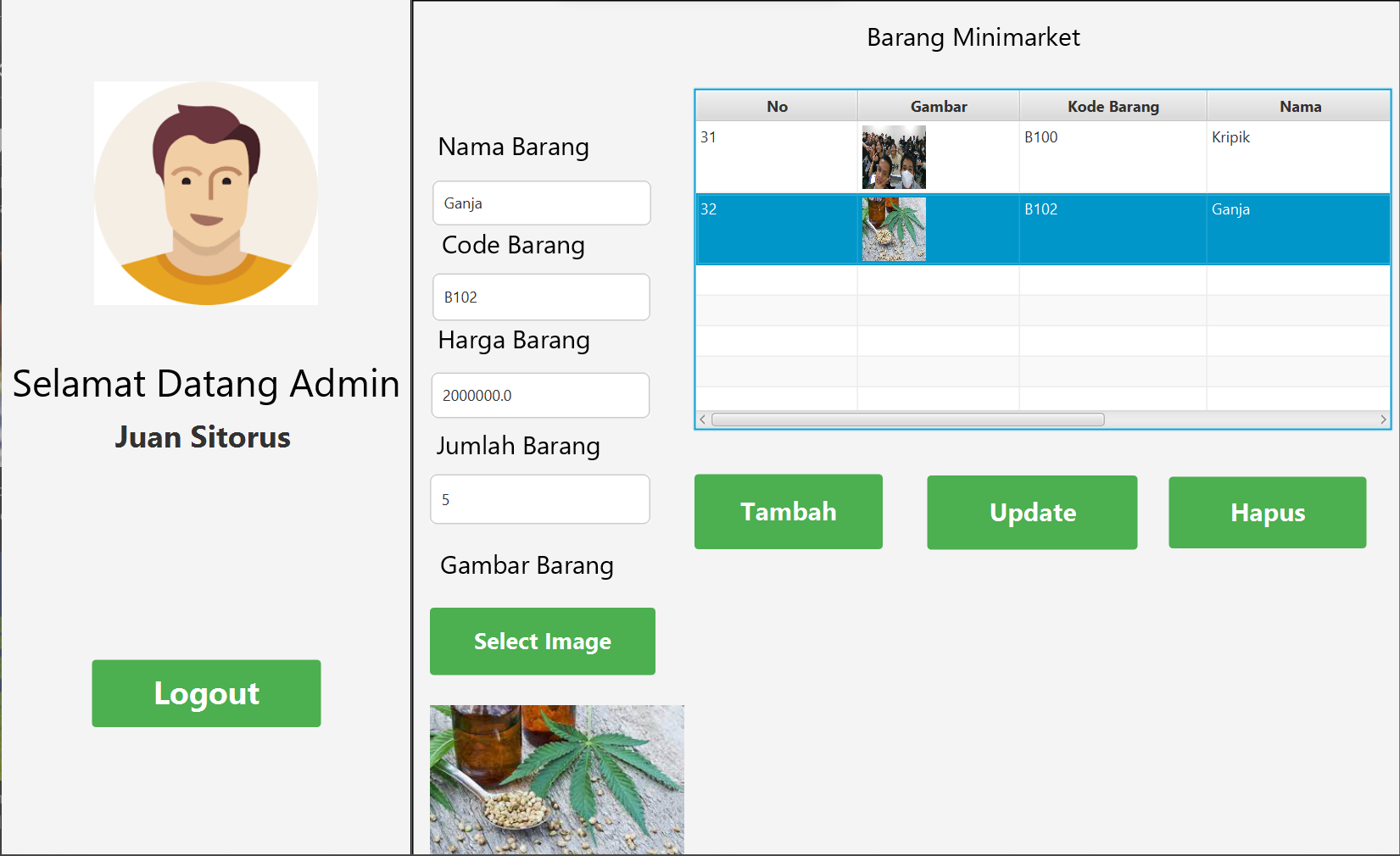
* 1. **Inheritance**

Inheritance terjadi ketika sebuah kelas mewarisi sifat-sifat dan metode dari kelas lain. Contoh: Penggunaan inheritance pada kelas userController yang memperluas atau mewarisi sifat dari BaseController.



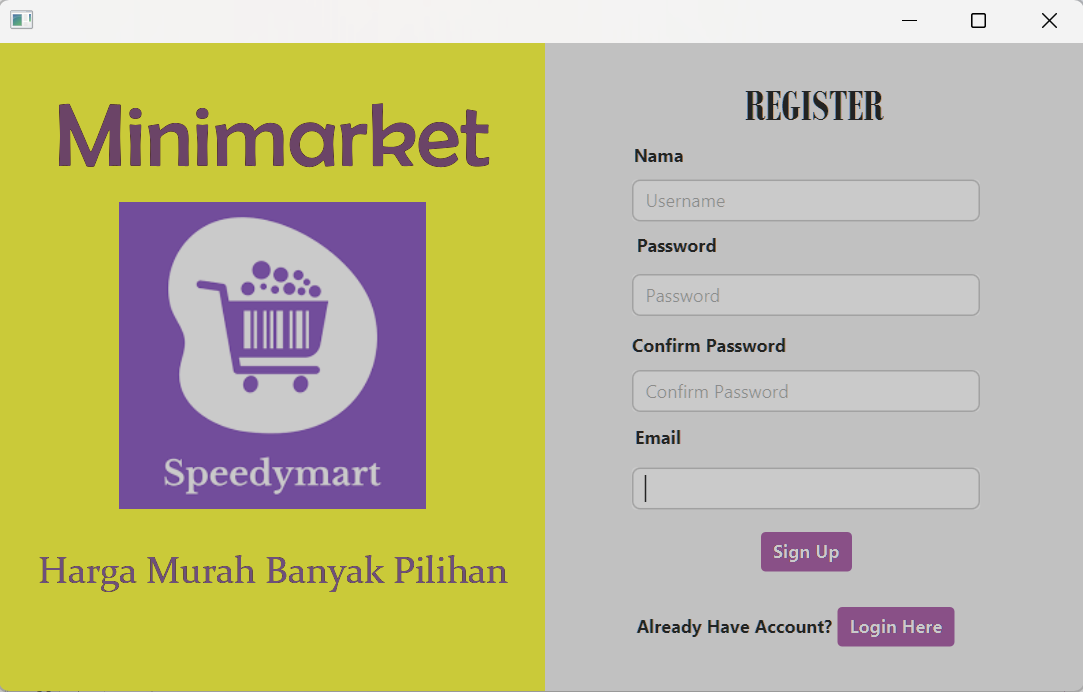
* 1. **Polymorphism**

Polimorfisme terjadi ketika objek dari kelas yang berbeda dapat diproses menggunakan metode yang sama. Contoh: Penggunaan metode handleItemSelection yang dapat memproses pemilihan item baik pada tampilan admin maupun pengguna.



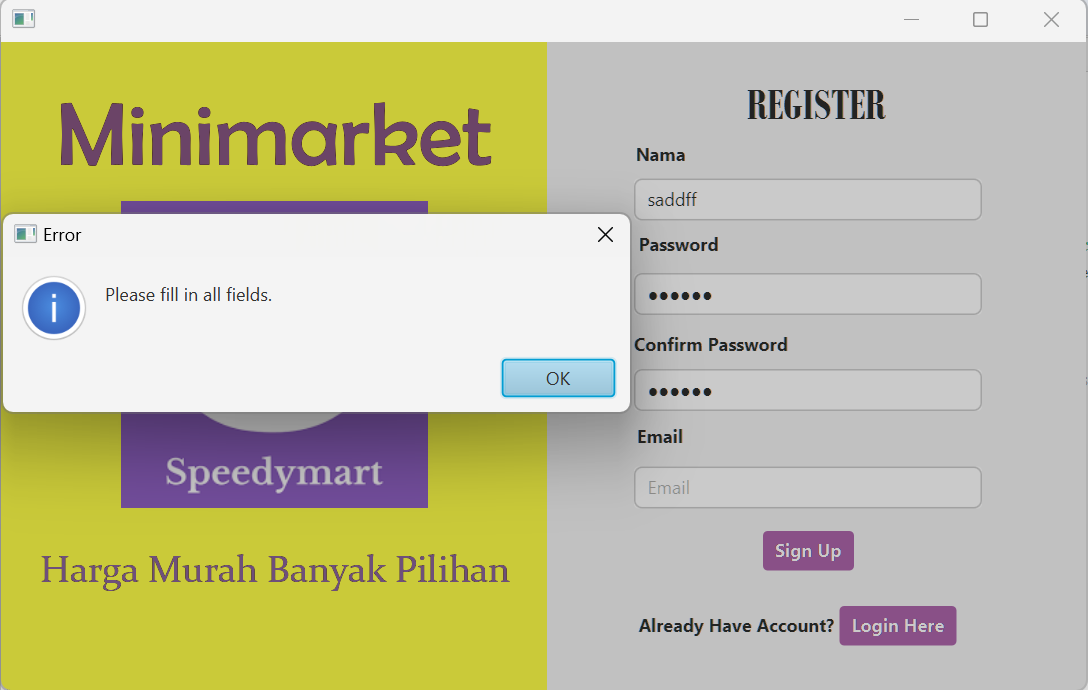
* 1. **Interface**

Interface memberikan kontrak yang harus dipenuhi oleh kelas yang mengimplementasikannya. Contoh: Penggunaan metode initialize dari antarmuka Initializable pada kelas registerController untuk melakukan inisialisasi saat tampilan dimuat.



* 1. **Exception handling**

Exception handling terjadi ketika kesalahan atau kondisi tidak terduga ditangani dengan baik. Contoh: Penggunaan try-catch dalam metode registerUser untuk menangani kesalahan SQL yang mungkin terjadi selama pendaftaran pengguna.



1. **LINK YOUTUBE**