**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.

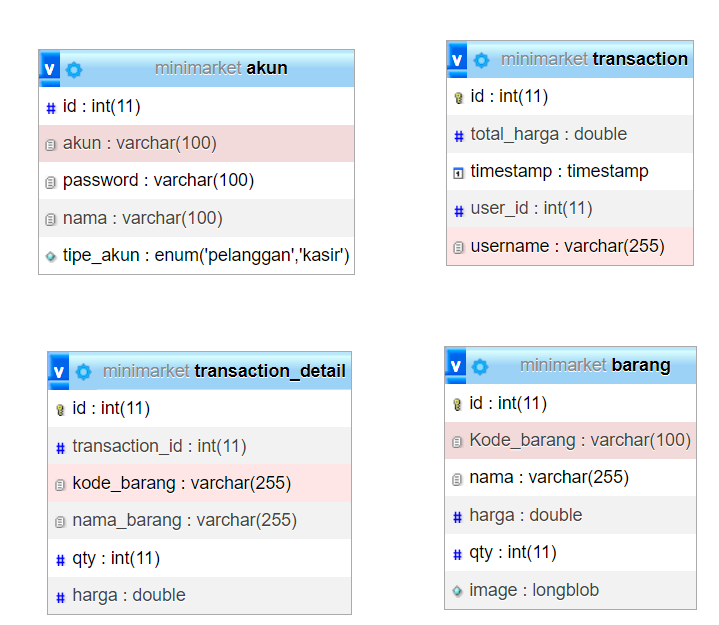
1. **FUNCTIONAL DESCRIPTION**

**User:**

1. **Register:**
   * *Deskripsi:* Fungsi ini memungkinkan pengguna untuk membuat akun baru dengan mengisi informasi yang diperlukan seperti nama, alamat email, dan kata sandi.
   * *Proses:* Pengguna harus mengisi formulir registrasi dengan informasi yang benar. Setelah itu, sistem akan membuat akun baru untuk pengguna.
2. **Login:**
   * *Deskripsi:* Fungsi ini memungkinkan pengguna yang sudah memiliki akun untuk masuk ke dalam sistem dengan menggunakan kombinasi alamat email dan kata sandi yang benar.
   * *Proses:* Pengguna memasukkan alamat email dan kata sandi yang telah didaftarkan. Jika informasi benar, pengguna diarahkan ke dashboard atau halaman utama.
3. **Melihat Barang:**
   * *Deskripsi:* Pengguna dapat melihat daftar barang yang tersedia di toko secara online.
   * *Proses:* Pengguna dapat menjelajahi katalog produk yang ditampilkan dengan informasi seperti nama barang, harga, dan gambar.
4. **Menambah Barang ke Keranjang:**
   * *Deskripsi:* Pengguna dapat menambahkan barang ke dalam keranjang belanja mereka.
   * *Proses:* Pengguna memilih barang yang ingin dibeli, kemudian menekan tombol "Tambah ke Keranjang". Jumlah barang yang dipilih ditambahkan ke keranjang belanja pengguna.
5. **Menghapus Barang dari Keranjang:**
   * *Deskripsi:* Fungsi ini memungkinkan pengguna untuk menghapus barang dari keranjang belanja mereka sebelum melakukan pembayaran.
   * *Proses:* Pengguna masuk ke dalam keranjang belanja, memilih barang yang ingin dihapus, dan menekan tombol "Hapus" atau tanda serupa.
6. **Membayar Keranjang:**
   * *Deskripsi:* Setelah selesai memilih barang, pengguna dapat melakukan pembayaran untuk menyelesaikan transaksi.
   * *Proses:* Pengguna memilih opsi pembayaran, memasukkan informasi pembayaran yang diperlukan, dan menyelesaikan transaksi.
7. **Melihat History Transaksi:**
   * *Deskripsi:* Pengguna dapat melihat riwayat transaksi atau pembelian yang telah dilakukan sebelumnya.
   * *Proses:* Pengguna mengakses halaman riwayat transaksi yang menyajikan informasi seperti tanggal transaksi, daftar barang yang dibeli, dan total pembayaran.

**Admin:**

1. **Login:**
   * *Deskripsi:* Fungsi ini memungkinkan admin untuk masuk ke dalam sistem administrasi toko online.
   * *Proses:* Admin memasukkan kombinasi alamat email dan kata sandi yang sah untuk mengakses panel admin.
2. **Menambah Barang:**
   * *Deskripsi:* Admin dapat menambahkan produk baru ke dalam katalog toko.
   * *Proses:* Admin mengisi formulir tambah barang dengan informasi seperti nama barang, harga, jumlah, dan gambar, lalu menekan tombol "Tambah Barang".
3. **Mengubah Harga Barang:**
   * *Deskripsi:* Admin dapat mengubah harga suatu barang yang sudah ada di katalog.
   * *Proses:* Admin memilih barang yang ingin diubah harganya, mengedit nilai harga, dan menyimpan perubahan.
4. **Mengubah Jumlah Barang:**
   * *Deskripsi:* Admin dapat mengubah jumlah stok barang yang tersedia.
   * *Proses:* Admin memilih barang yang ingin diubah jumlahnya, mengedit nilai jumlah stok, dan menyimpan perubahan.
5. **Mengubah Gambar pada Barang:**
   * *Deskripsi:* Admin dapat mengganti gambar yang terkait dengan suatu barang.
   * *Proses:* Admin memilih barang yang ingin diubah gambarnya, mengunggah gambar baru, dan menyimpan perubahan.
6. **Menghapus Barang:**
   * *Deskripsi:* Admin dapat menghapus barang dari katalog toko.
   * *Proses:* Admin memilih barang yang ingin dihapus, mengonfirmasi penghapusan, dan menghapus barang dari sistem
7. **DESIGN TABLE STRUCTURE**

****

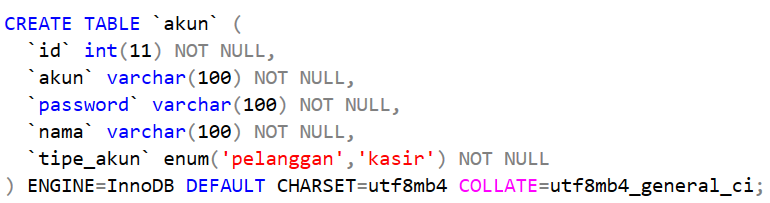
1. **Tabel akun:**
2. Struktur Tabel:

* 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:



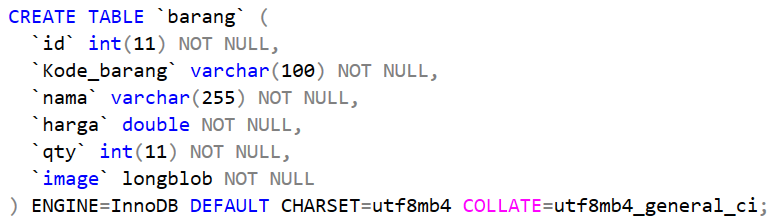
1. **Tabel barang:**
2. Struktur Tabel:

* 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:



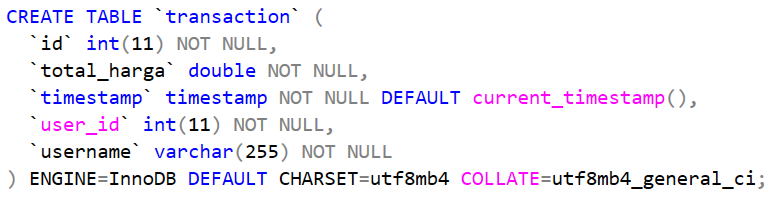
1. **Tabel transaction\_detail::**
2. Struktur Tabel:

* 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:**

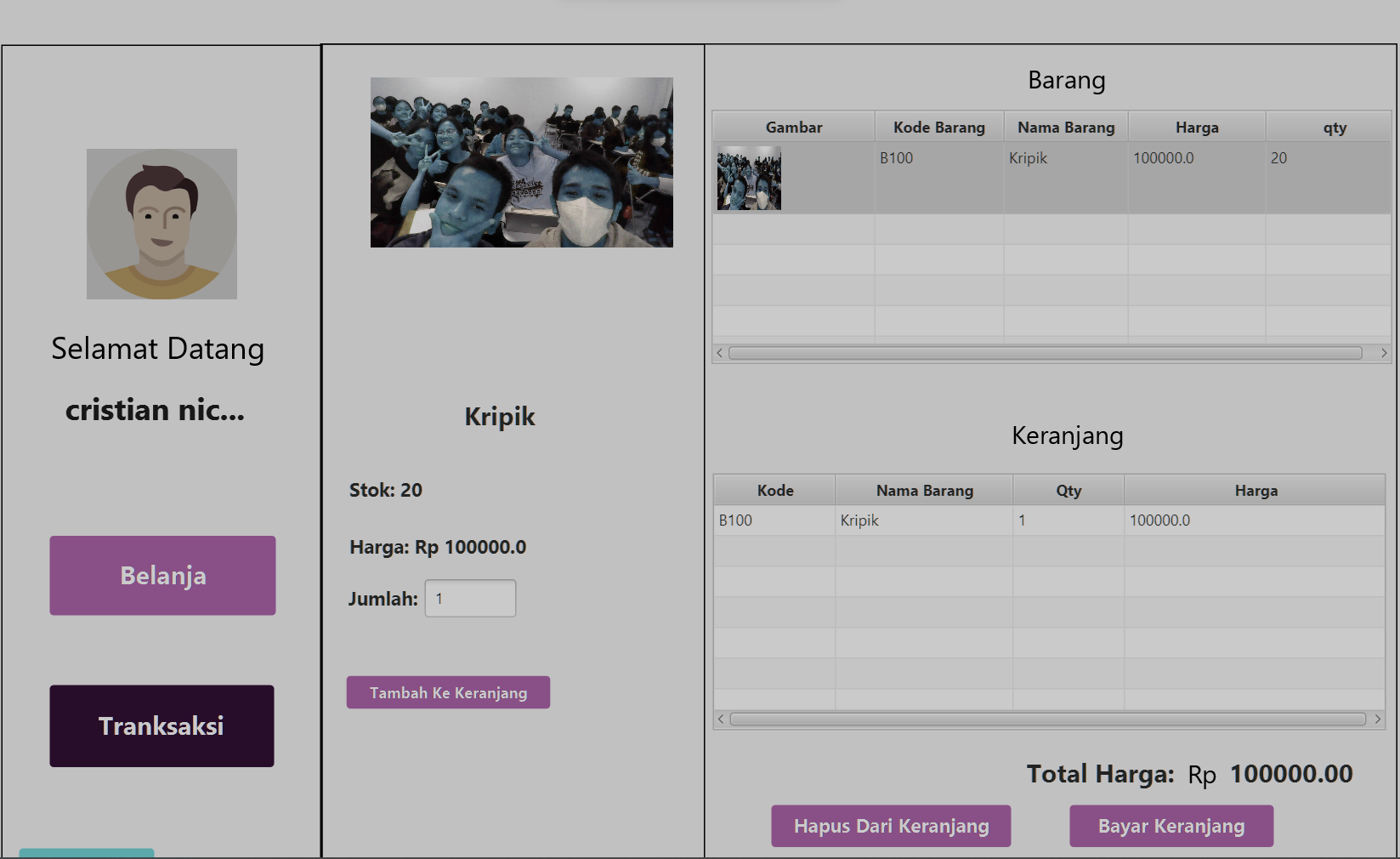
* Tabel transaction memiliki foreign key constraint terhadap kolom user\_id yang terhubung dengan tabel akun.

1. **Catatan:**

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

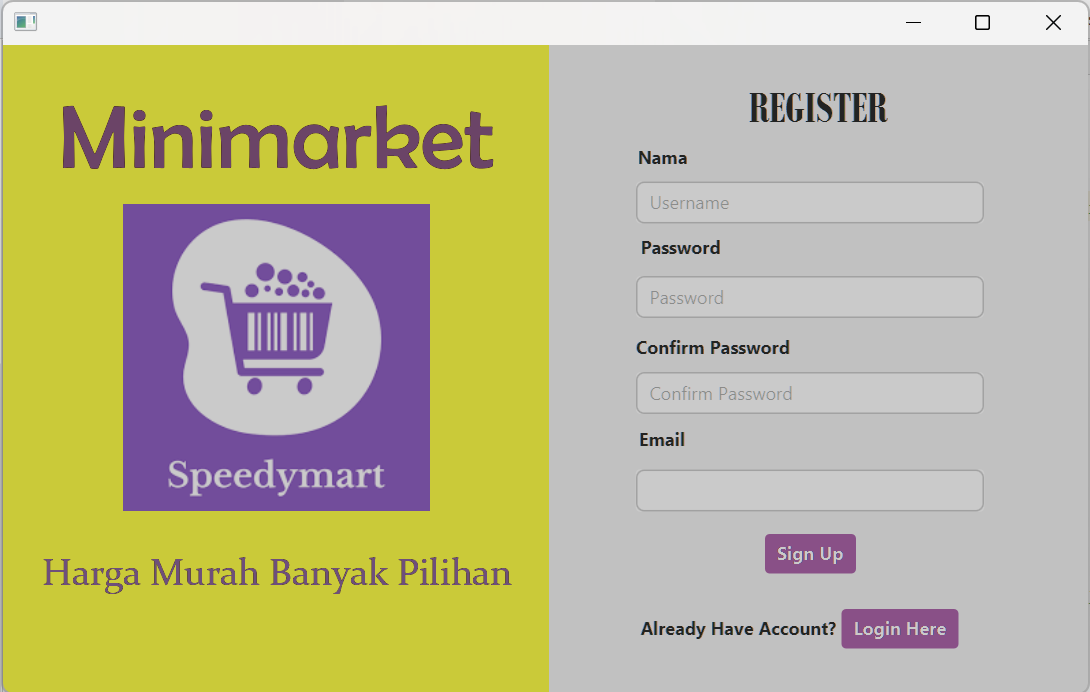
1. **OOP IMPLEMENTATION**
   1. **Abstraction**

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.



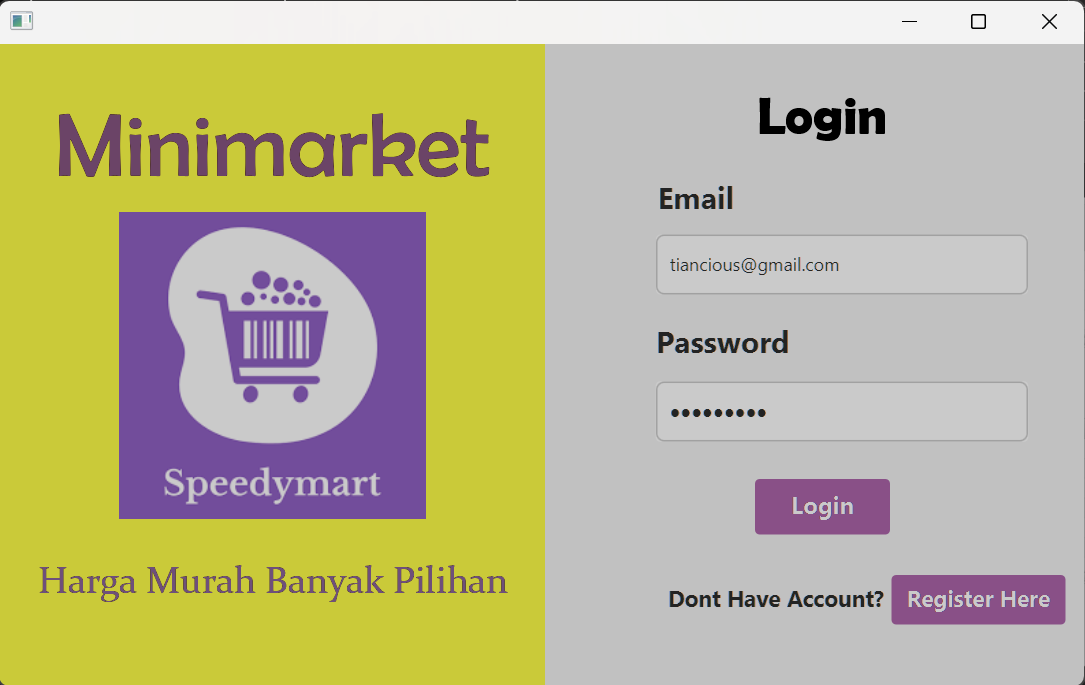
* 1. **Encapsulation**

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.



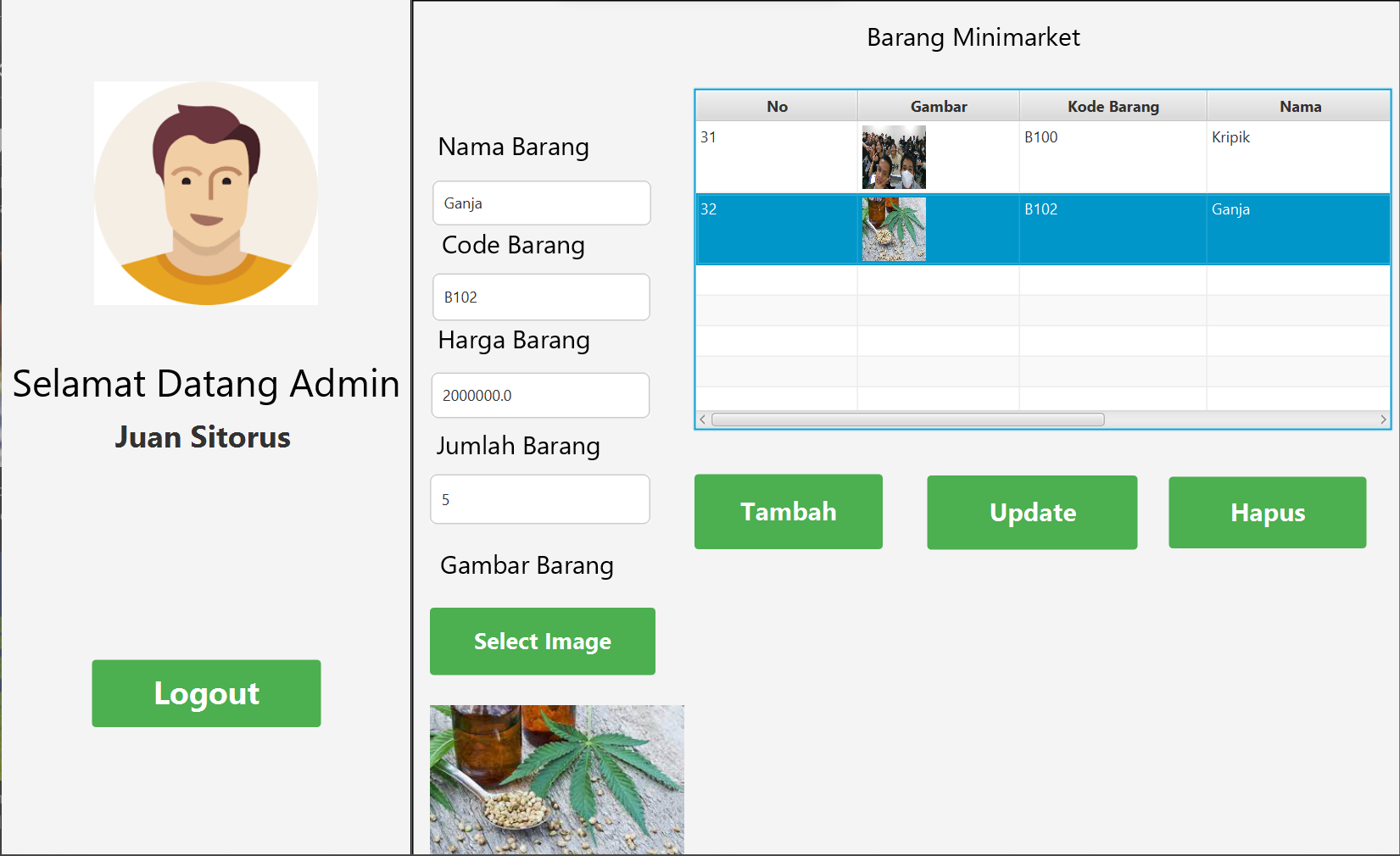
* 1. **Inheritance**

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.



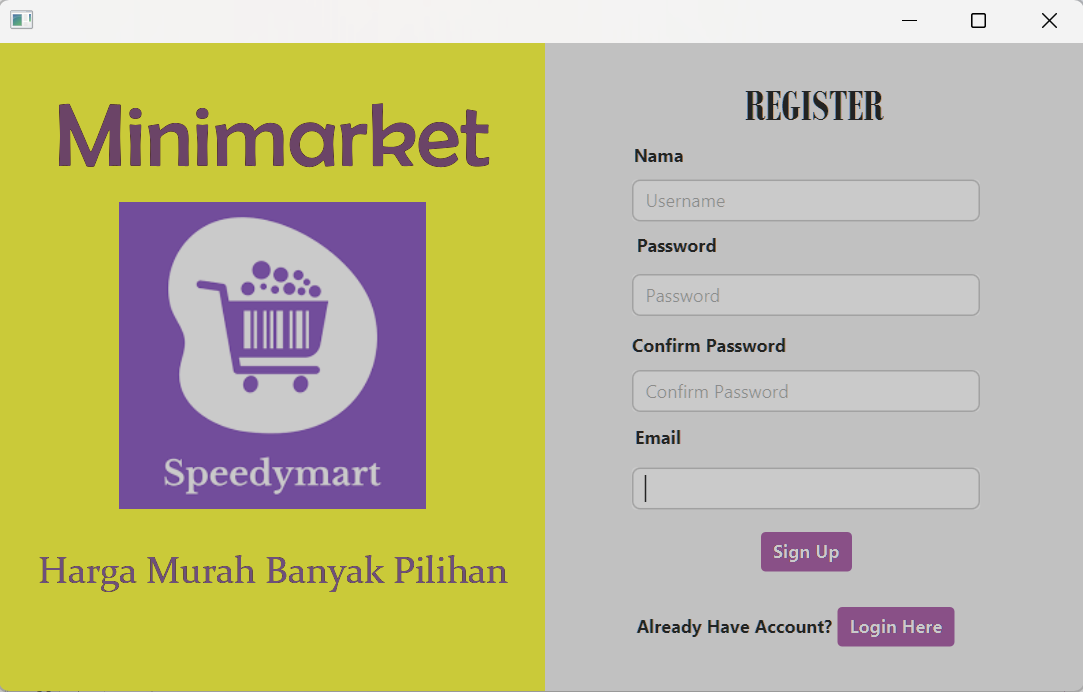
* 1. **Polymorphism**

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.



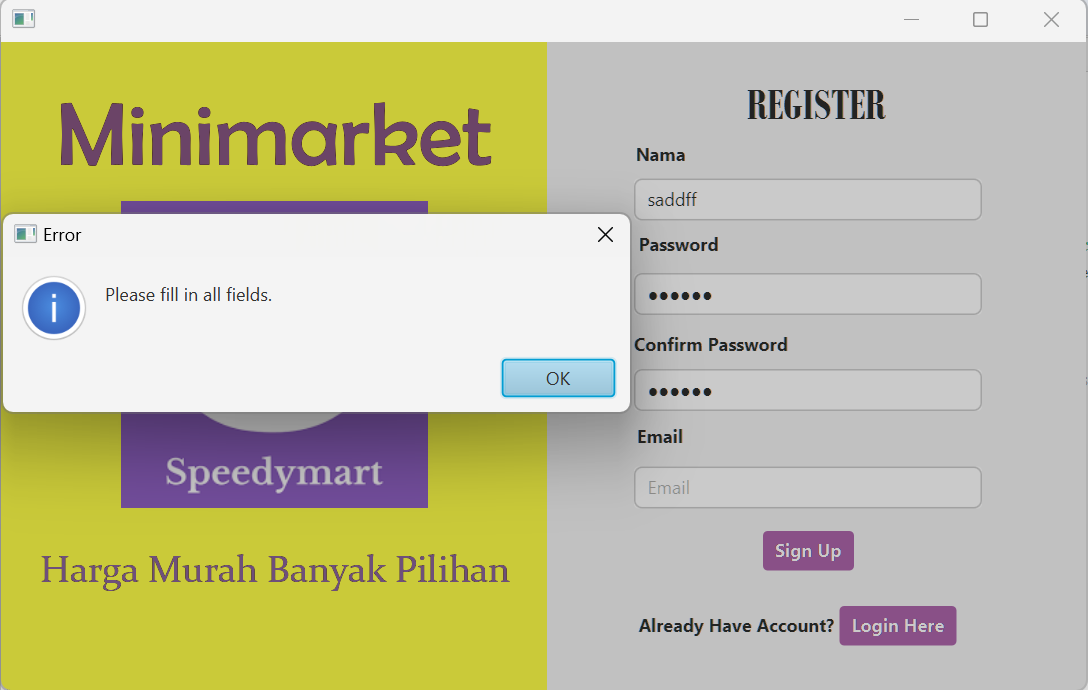
* 1. **Interface**

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**

**https://youtu.be/k1JBMUazz\_s**