**FINAL REPORT PROJECT**

**BUSINESS APPLICATION DEVELOPMENT**

****

Kelompok 1:

2602080623 - Edith Dorothy

2602058056 - Brainard Fibert Setiawan

2602069671 - Axel Sebastian

2602083215 - Willson Antony

2602053502 - Tristan Pranaja Linardo

2602061832 - Yosua Samuelpin

2602066625 - Ervan Fernando Wijaya

2602068510 - Fedry Alvindra

2602060003 - Bryan Limios

2602073574 - Brandon Jhon

2602055092 - Nicholas Edelstein

2602071562 - Philip Antonius Budianto

2602196763 - Jennifer Brigitta Trisulo

Business Application Development

Kelas: LA11

Tahun Ajaran: 2023/2024

**TABLE OF CONTENT**

[**I. CHAPTER I – PRELIMINARY 1**](#_yumhn78rdg5n)

[A. Background and Objective 1](#_eetljcfh7o3o)

[B. Roles and Responsibility 1](#_73cedsr8a9ws)

[C. Timeline 2](#_r27273hqrv9q)

[**II. CHAPTER II – BUSINESS APPLICATION 3**](#_z0ad9nwg5aar)

[A. Use Case and Activity Diagram 3](#_hb5soy95g57z)

[B. GUI Design 5](#_we20cfxu9f1m)

[C. DB Design 8](#_tss7jbhm3nbc)

[D. Coding explanation for DB Connectivity and GUI 9](#_3pte8rlt0sso)

[E. Question and Answer 22](#_npb93a4us96r)

[**III. CHAPTER III – CONCLUSION 23**](#_muc8g7vnqlbd)

# CHAPTER I – PRELIMINARY

## Background and Objective

* Background

Dalam mengelola suatu bisnis atau perusahaan, diperlukan pencatatan keuangan yang jelas agar pengeluaran dan pemasukan dapat dipantau dan dikontrol. Tanpa pencatatan keuangan yang jelas, maka akan sulit bagi perusahaan untuk membuat keputusan keuangan yang tepat. Finance Buddy merupakan aplikasi yang digunakan untuk mencatat transaksi keuangan untuk mengetahui seberapa banyak pemasukan dan pengeluaran yang ada. Aplikasi ini diperuntukkan kepada perusahaan atau individu yang terlibat langsung dalam internal perusahaan dalam mencatat transaksi keuangan perusahaan.

* Objective

Aplikasi Finance Buddy bertujuan untuk membantu perusahaan dalam melakukan beberapa hal sebagai berikut.

* Mencatat pengeluaran dan pemasukan keuangan mereka.
* Mengkategorisasikan jenis transaksi mereka.
* Memeriksa jejak pengeluaran dan pemasukan mereka.
* Melihat total pengeluaran dan pemasukan selama seminggu, sebulan, atau setahun terakhir.

## Roles and Responsibility

Untuk memudahkan pengerjaan *project* kami, kami membagi tugas-tugas yang ada kepada anggota kelompok. Pembagian tugas kelompok kami yaitu sebagai berikut.

* **Team Leader:** Mengatur jadwal pengerjaan project, memantau progress dari setiap tim untuk memastikan pengerjaan project ter-koordinasi, membantu tim-tim yang membutuhkan bantuan tambahan.

2602080623 – Edith Dorothy

* **System Designer:** Merancang bagaimana jalannya sistem, seperti membuat activity diagram, use case diagram, dkan sebagainya.

2602058056 – Brainard Fibert Setiawan

2602069671 – Axel Sebastian

2602083215 – Willson Antony

2602053502 – Tristan Pranaja Linardo

* **Application Developer / Programmer:** Membuat aplikasi dalam bentuk coding.

2602061832 – Yosua Samuelpin

2602066625 – Ervan Fernando Wijaya

2602068510 – Fedry Alvindra

2602060003 – Bryan Limios

* **Tester:** Menguji hasilcodingan programmer dan membuat *test case* sesuai dari hasil uji.

2602073574 – Brandon Jhon

2602055092 – Nicholas Edelstein

2602071562 – Philip Antonius Budianto

* **Presenter:** Mempresentasikan hasil pekerjaan kelompok dan membuat laporan akhir.

2602196763 – Jennifer Brigitta Trisulo

2602080623 – Edith Dorothy

2602060003 – Bryan Limios

## Timeline

| **Phase** | **Activity** |
| --- | --- |
| 15 September 2023 | Role and Job Distribution |
| 22 September 2023 | Discuss and Brainstorm Application Concept |
| 2 Oktober 2023 | Accomplish ERD and UI for Finance Buddy Application |
| 17 Oktober 2023 | Plan and Set Code Production Schedule |
| 19 Oktober 2023 | Sprint Execution, Activity Diagram Ideation by Systems Designer Team and Test Case Template by Tester Team |
| 20 Oktober 2023 | Testing Phase 1 by Tester Team |
| 28 Oktober 2023 | Testing Phase 2 by Tester Team |
| 20 November 2023 | Code Renewal and Update |
| 22 November 2023 | Testing Phase 3 by Tester Team |
| 30 November 2023 | Final Testing by Tester Team and Program Implementation |

# CHAPTER II – BUSINESS APPLICATION

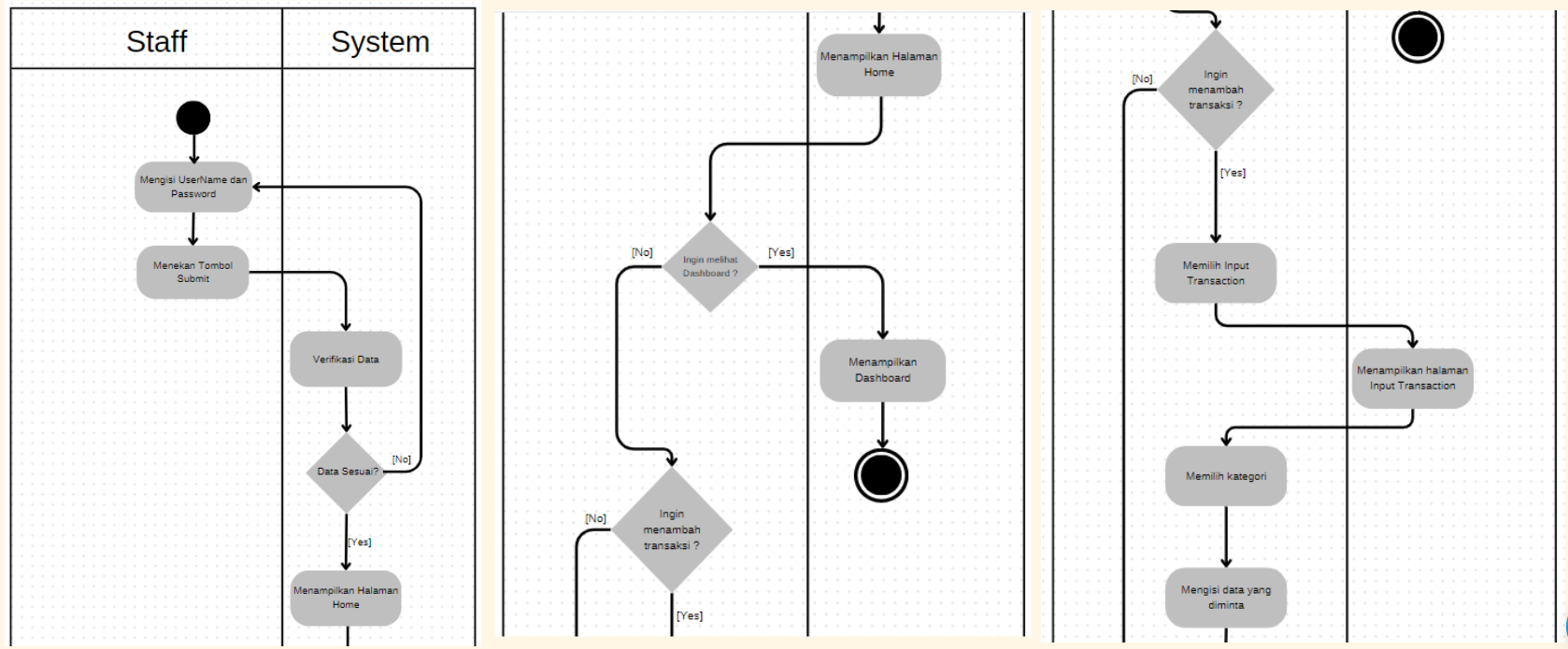
## Use Case and Activity Diagram

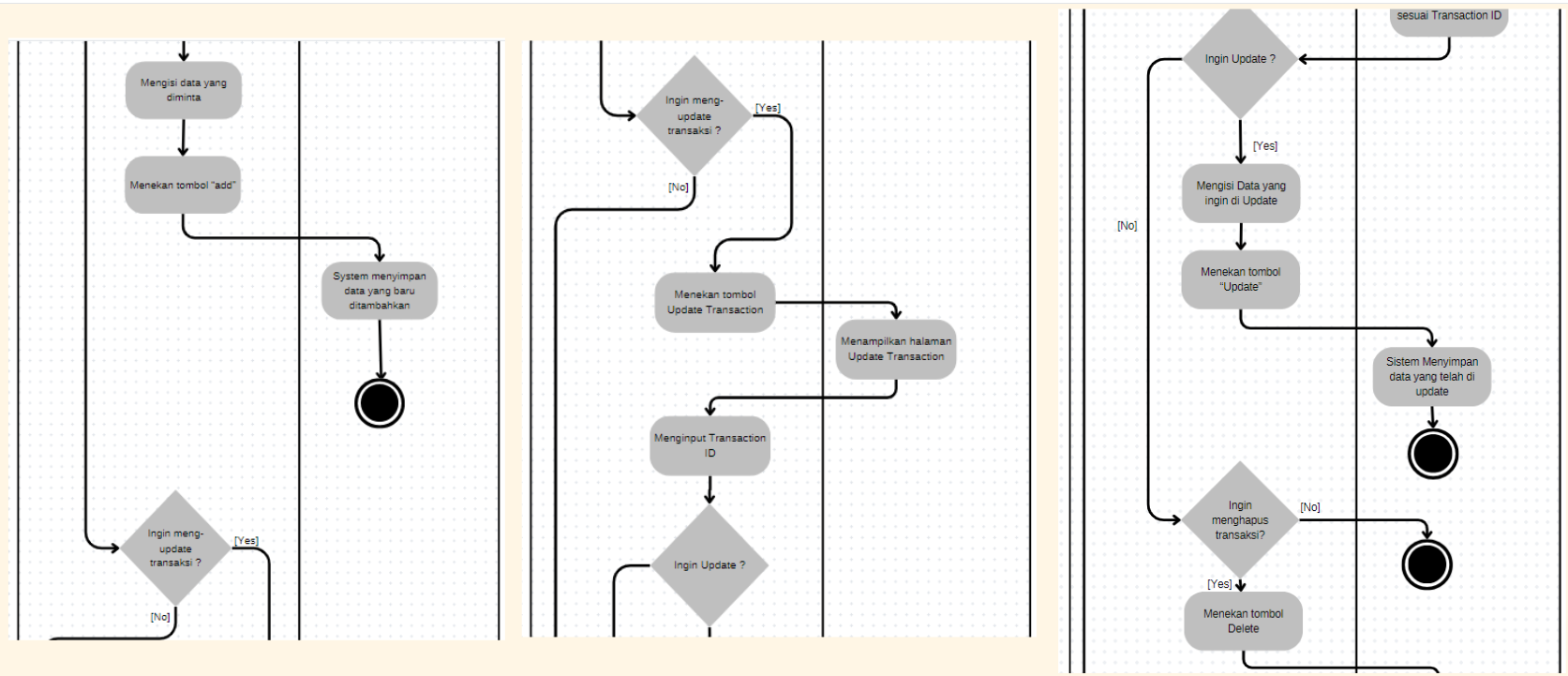
* Use Case

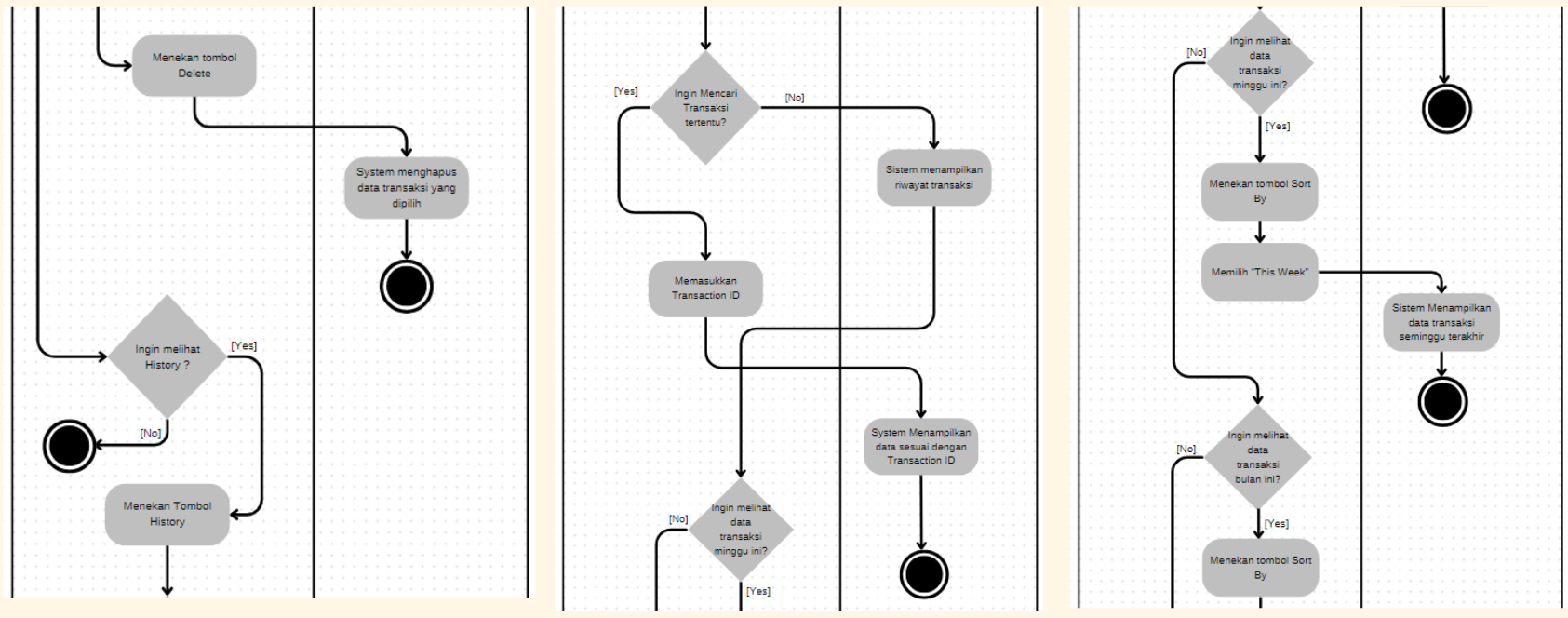


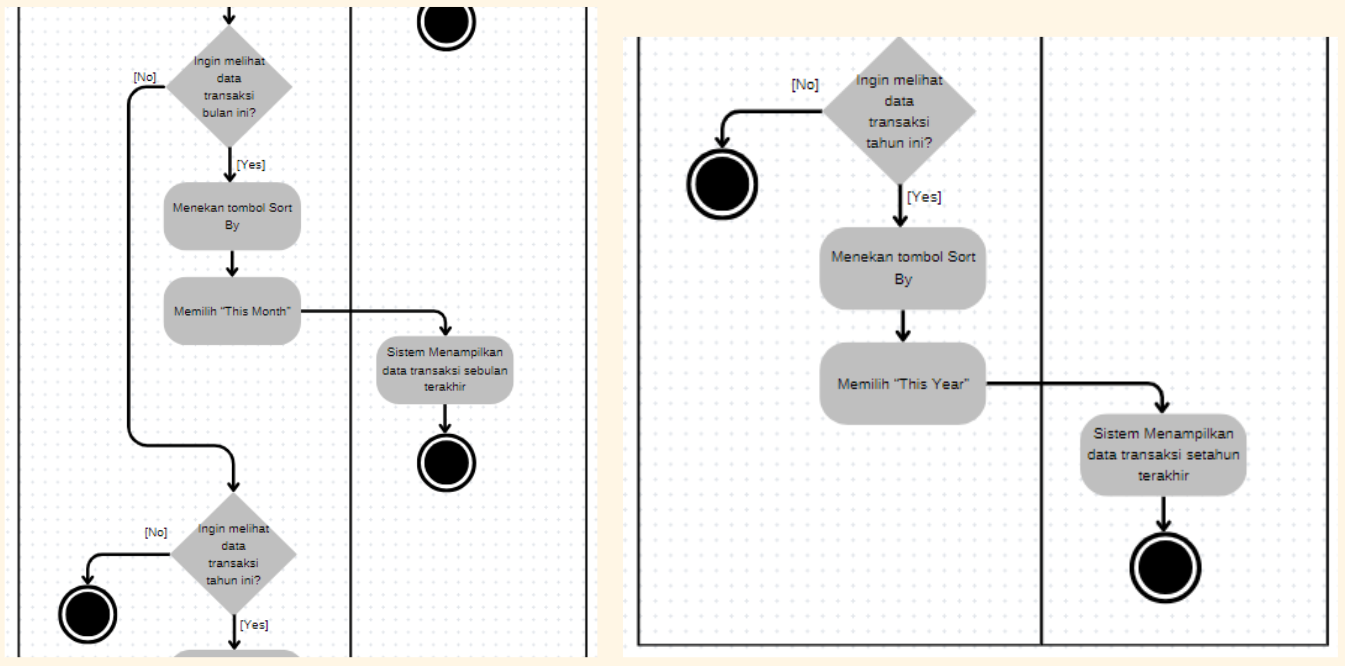
Link: <https://www.canva.com/design/DAFvJiW2NMs/4GGiM__twrPP2WjjOYQfmA/edit?utm_content=DAFvJiW2NMs&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

* Activity Diagram



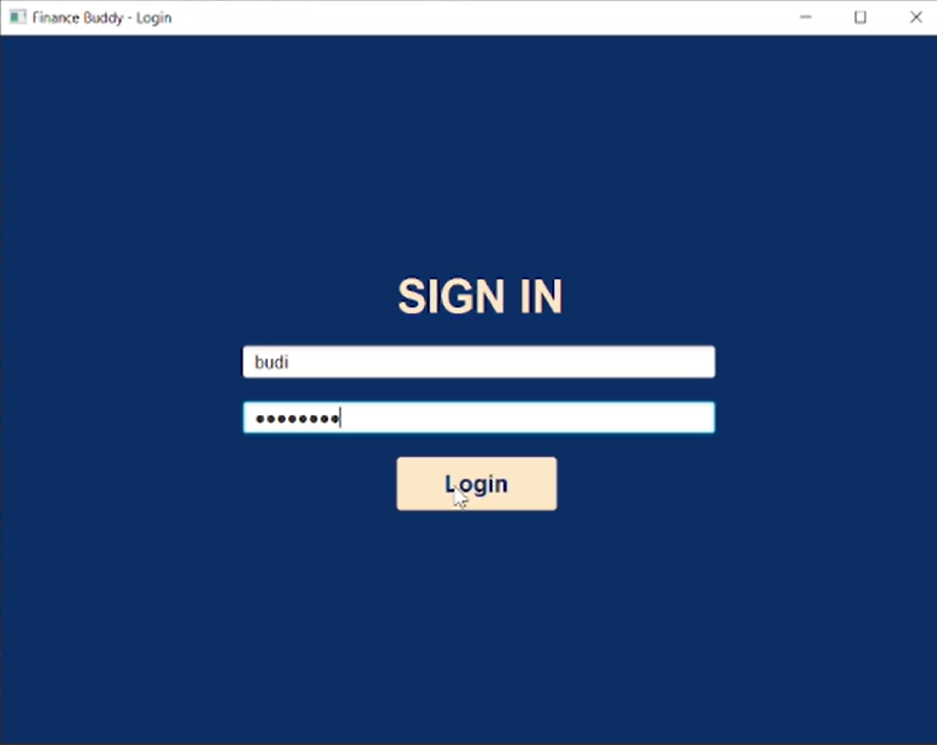


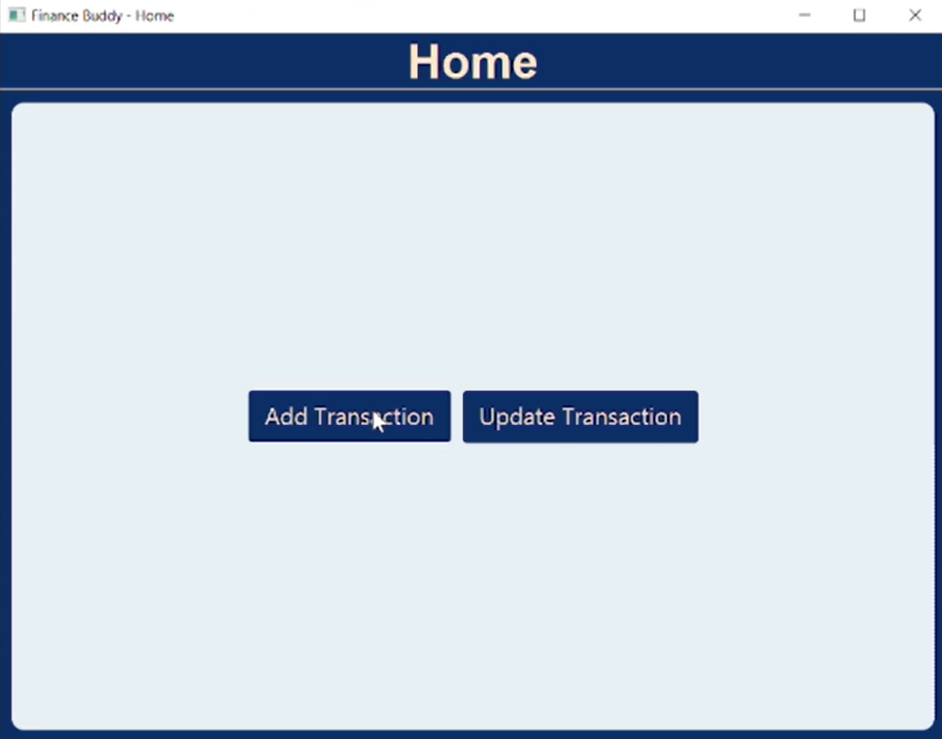


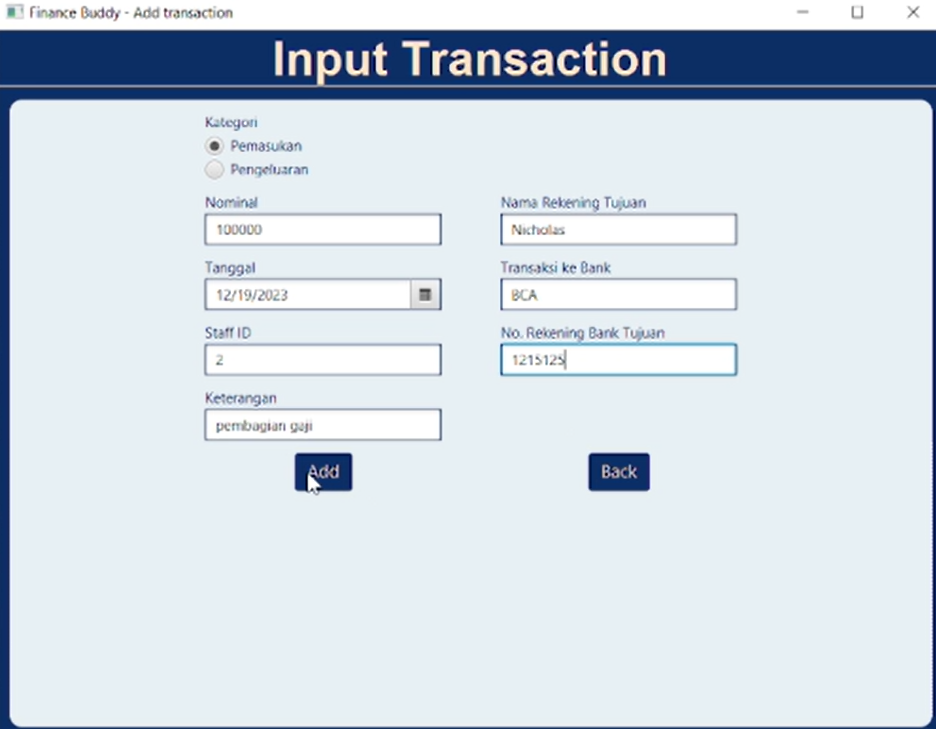


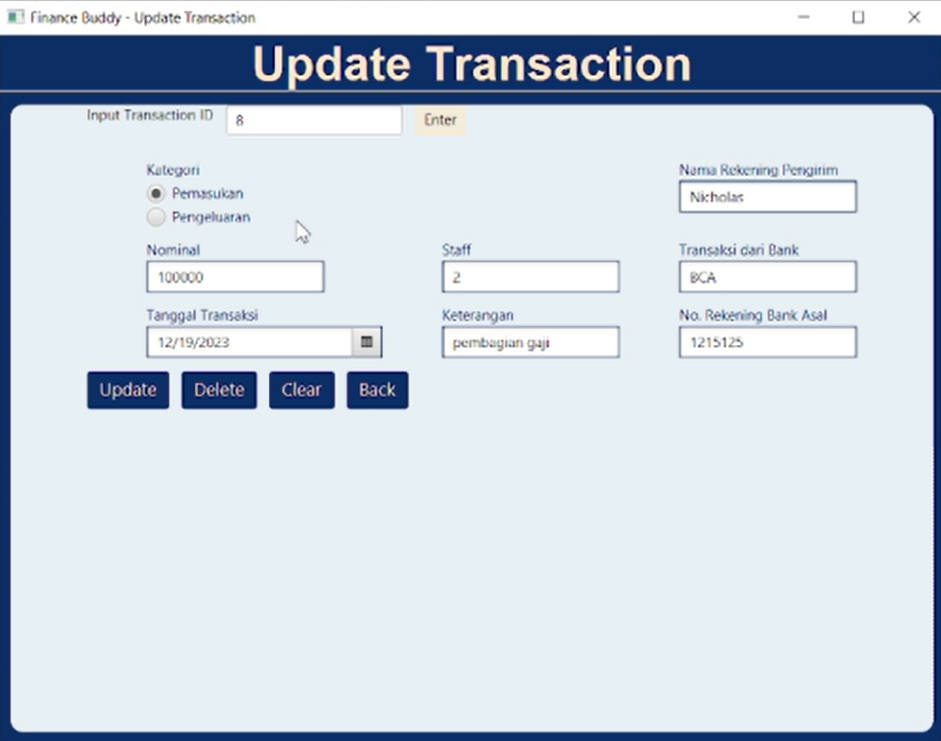
Link: <https://www.canva.com/design/DAFxtHTceAw/xrq6SCx48ppZR5x74m7zhg/edit?utm_content=DAFxtHTceAw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

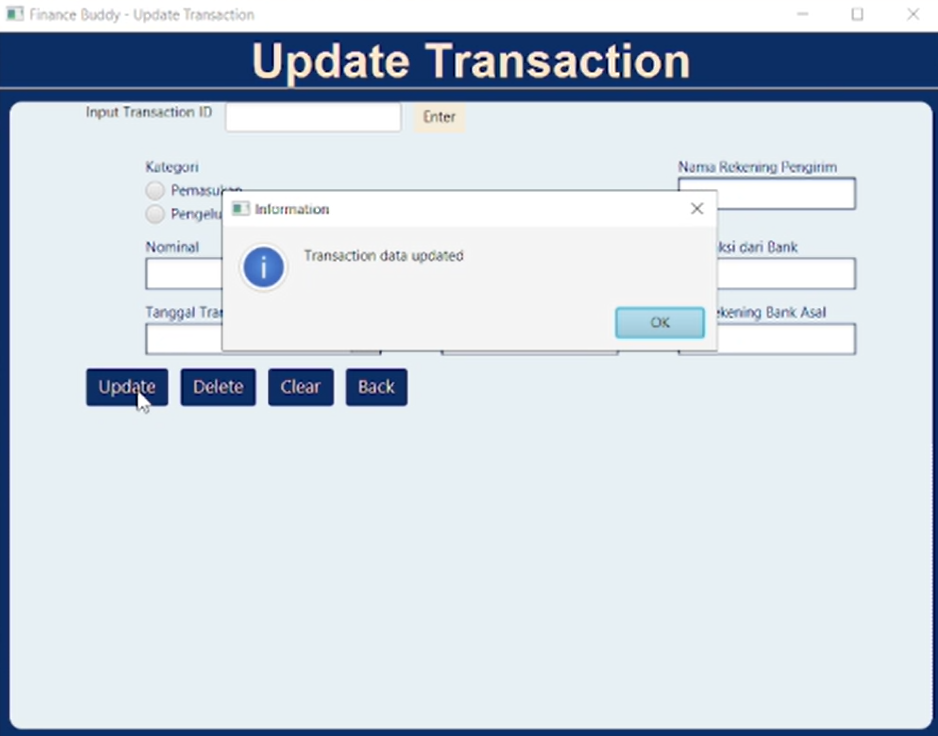
## GUI Design

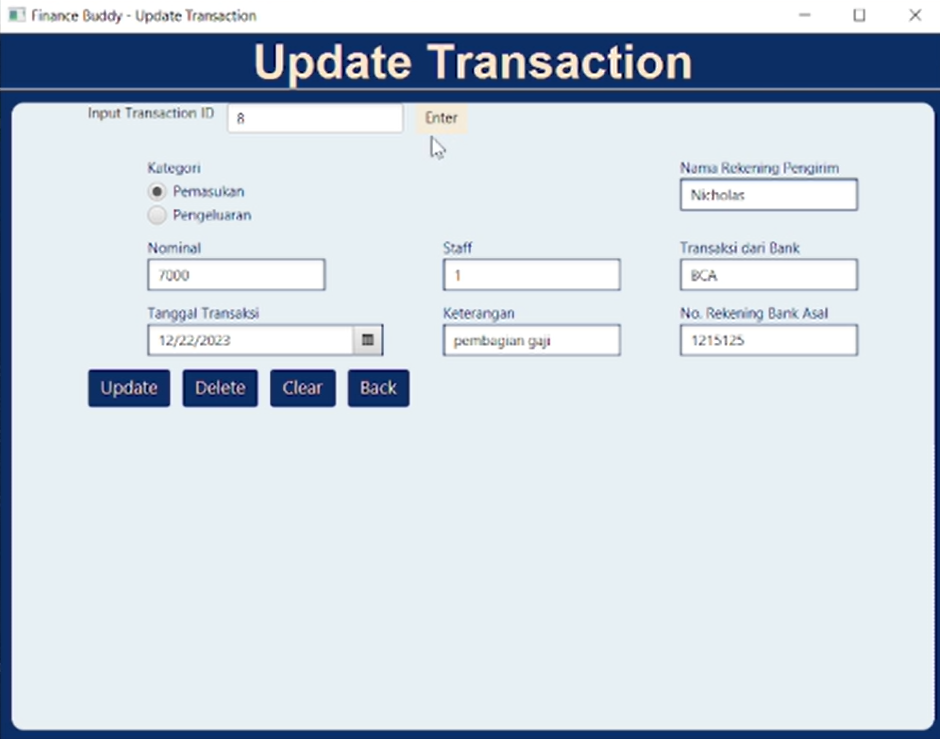


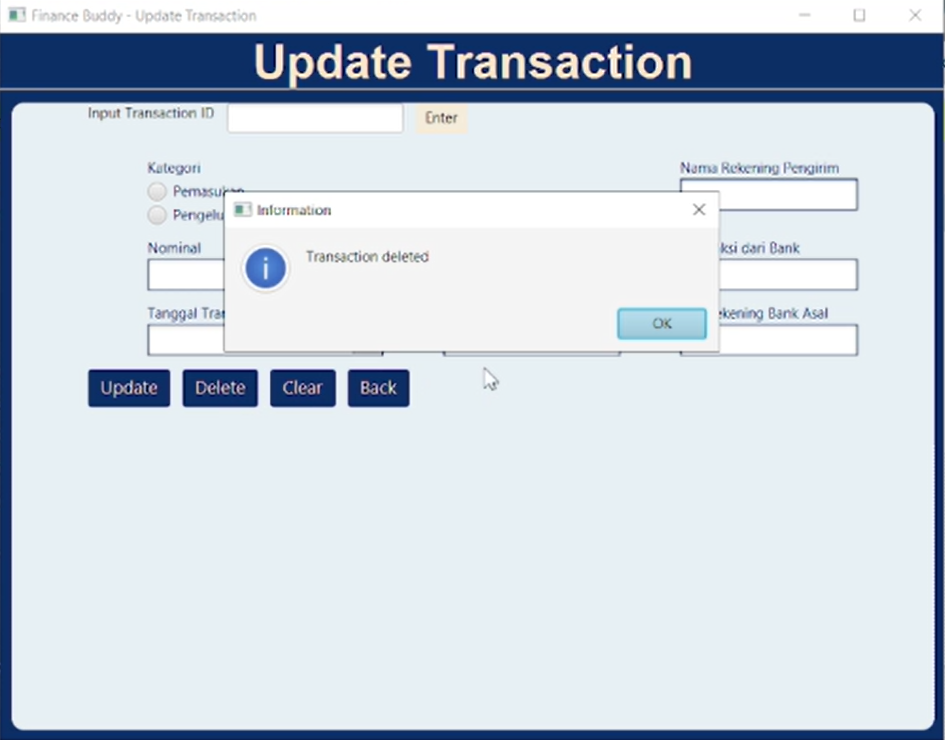






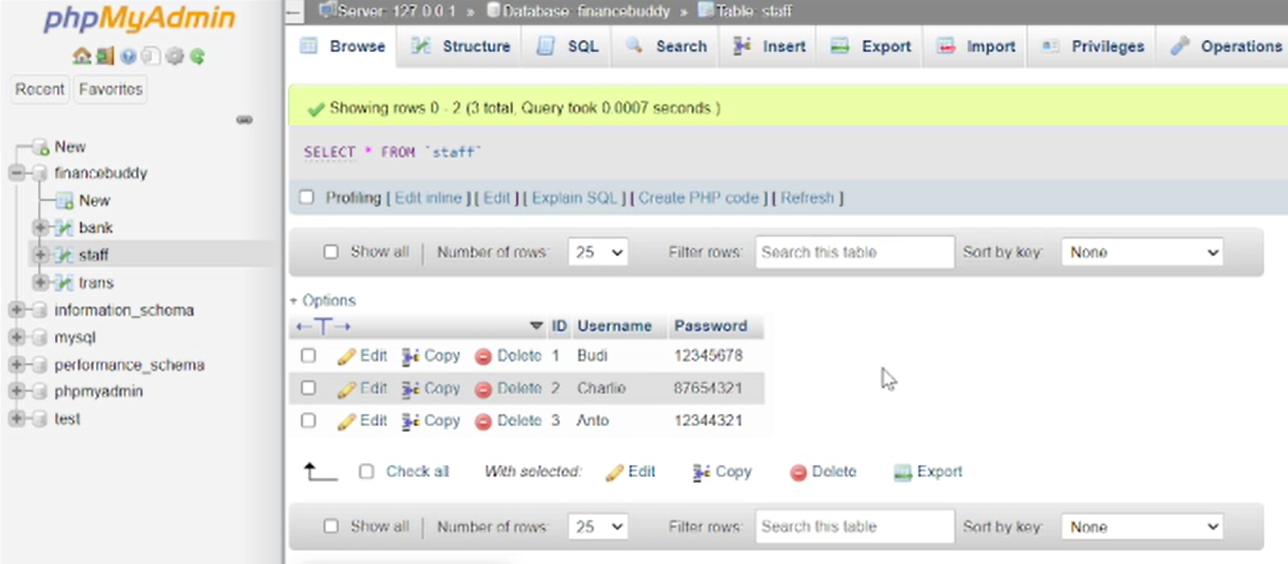




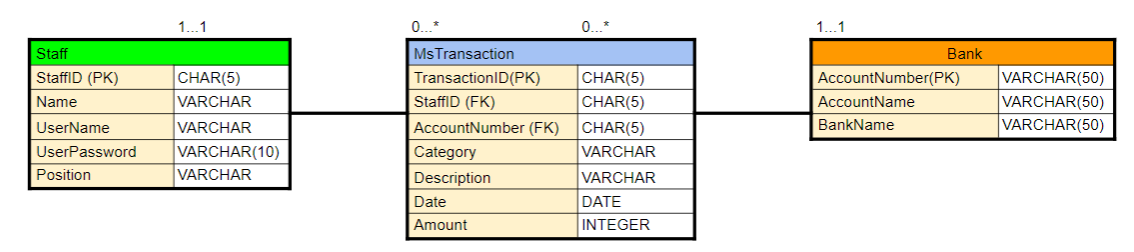


## DB Design

* PhpMyAdmin Database

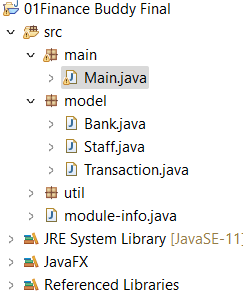


* Entity Relationship Diagram



## Coding explanation for DB Connectivity and GUI

* **Code Project Structure**



Pada project Finance eBuddy kita menggunakan package untuk mengorganisir interaksi dan penyimpanan antar class. Berikut adalah package yang tersedia :

1. Package ‘main’

Bertujuan untuk menyimpan file ‘Main.java’.

1. Package ‘model’

Bertujuan untuk menyimpan model database yaitu ‘Bank.java’, ‘Staff.java’, dan ‘Transaction.java’.

1. Package ‘util

Bertujuan untuk menyimpan class ‘Connect.java’ pada saat menghubungkan sebuah database.

Selain itu, juga terdapat library yang dipakai untuk menunjang kebutuhan project seperti Java SE 11, JavaFX dan referenced library yang berisi sql connector untuk menghubungkan database.

* **Main.java**

1. Package and Import



Package yang diimplementasikan pada class ‘Main.java’ adalah package main. Selain itu, kita juga melakukan impor terhadap module-module javaFX yang digunakan pada class ‘Main.java’.

1. Extends Application



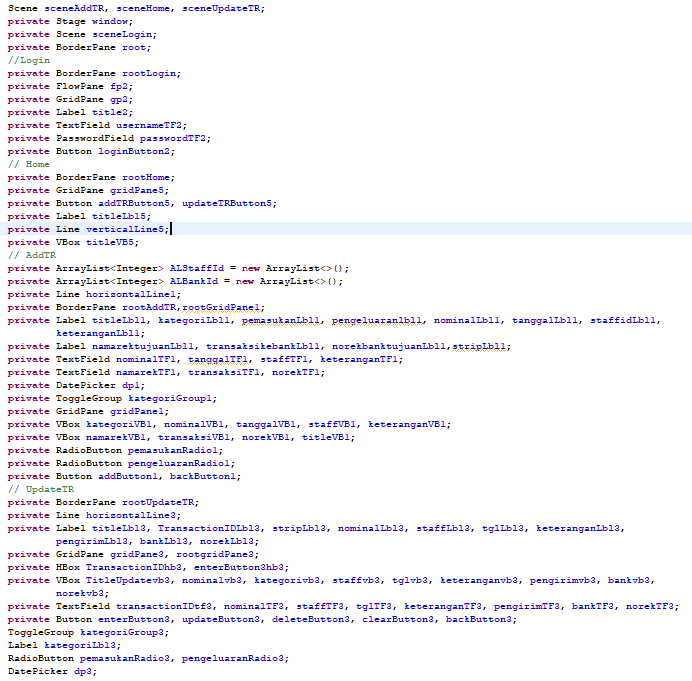
Kita mendeklarasikan class Main sebagai Aplikasi dari javaFX.

1. DB Connection



Database Connection dihubungkan dengan menggunakan kode yang tertera diatas. Kode tersebut berfungsi untuk membuka koneksi pada aplikasi.

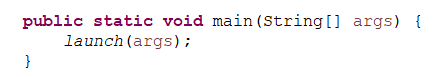
1. JavaFX Variabel



Kode tersebut bertujuan untuk menginisialisasikan variabel yang telah kami impor. Kami juga mengklasifikasikan pengguna variabel berdasarkan tampilan - tampilan scene yang akan kami bangun. Scene tersebut antara lain yaitu:

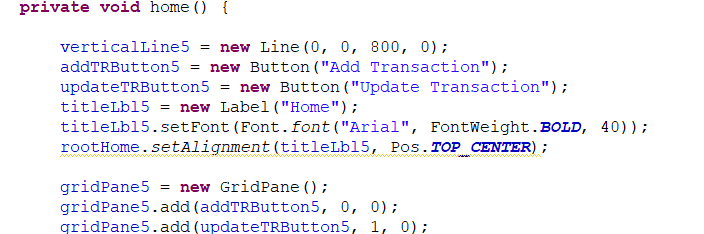
* Login Scene
* Home Scene
* Add Transaction Scene
* Update Transaction Scene

1. Constructor for application launch



Kode yang dideklarasikan dengan menggunakan constructor untuk menjalankan aplikasi pada class “main.java’.

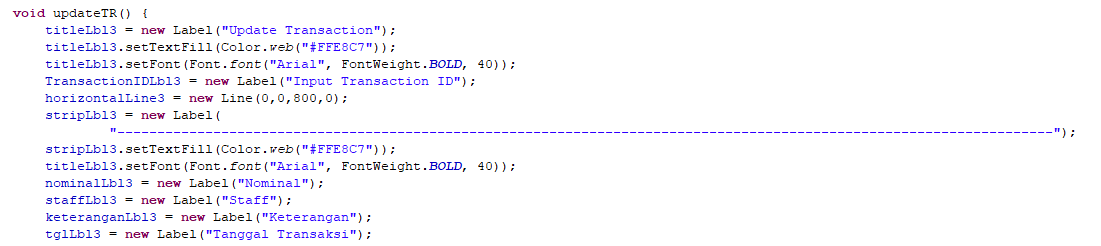
1. Method ‘Home’



*Part of Code*

Method yang digunakan untuk menyusun tampilan pada scene home dengan menggunakan variabel-variabel yang telah ditetapkan. Kami menggunakan CSS untuk membantu mendesain tampilan yang akan dibangun.

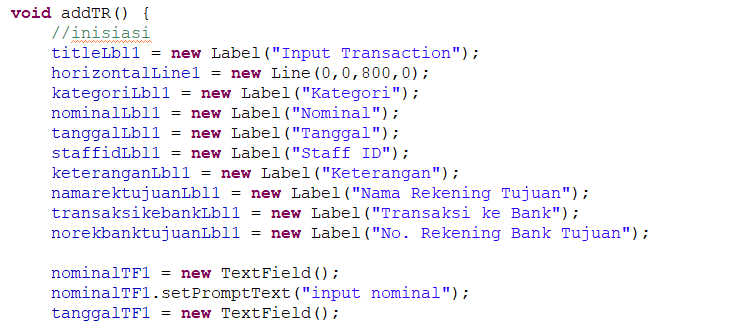
1. Method ‘updateTR’



*Part of Code*

Method yang digunakan untuk menyusun tampilan pada scene update Transaction dengan menggunakan variabel-variabel yang telah ditetapkan. Kami menggunakan CSS untuk membantu mendesain tampilan yang akan dibangun. Selain itu, kami juga menggunakan bantuan style None-CSS untuk melakukan desain terhadap tampilan.

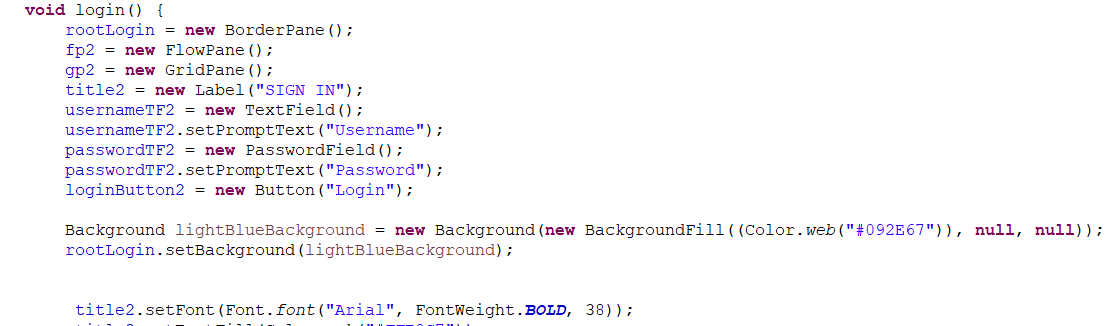
1. Method ‘addTR’



*Part of Code*

Method yang digunakan untuk menyusun tampilan pada scene add Transaction dengan menggunakan variabel-variabel yang telah ditetapkan. Kami menggunakan CSS untuk membantu mendesain tampilan yang akan dibangun. Selain itu, kami juga menggunakan bantuan style None-CSS untuk melakukan desain terhadap tampilan.

1. Method ‘login’



*Part of Code*

Method yang digunakan untuk menyusun tampilan pada scene login dengan menggunakan variabel-variabel yang telah ditetapkan. Kami menggunakan CSS untuk membantu mendesain tampilan yang akan dibangun. Selain itu, kami juga menggunakan bantuan style None-CSS untuk melakukan desain terhadap tampilan.

1. Method ‘eventAction’



Method yang digunakan untuk menyusun interaksi, trigger dan logic

yang ada pada suatu scene.

* 1. Login



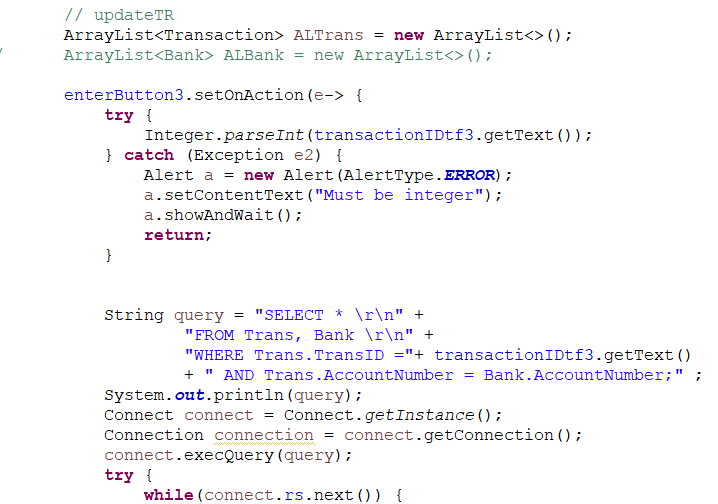
Pada bagian login kita memberikan logic terhadap tombol yang ada pada tampilan tersebut. Pada tampilan login scene terdapat satu tombol login yang berfungsi untuk memvalidasi bahwa username dan password dari staff sesuai dengan data yang ada di database. Jika menekan tombol tersebut pengguna akan mendapatkan sebuah notifikasi baik login berhasil atau gagal.

* 1. Add Transaction

*Part of Code*

Berfungsi untuk memberikan logic pada button add transaction. Transaksi akan tercatat pada database ketika tabel sudah terisi secara tepat. Selain itu, juga terdapat validasi validasi yang ada pada kode tersebut untuk digunakan pada scene add transaction. Dan tersedia juga beberapa exception ketika pengguna gagal dalam melakukan add transaction.

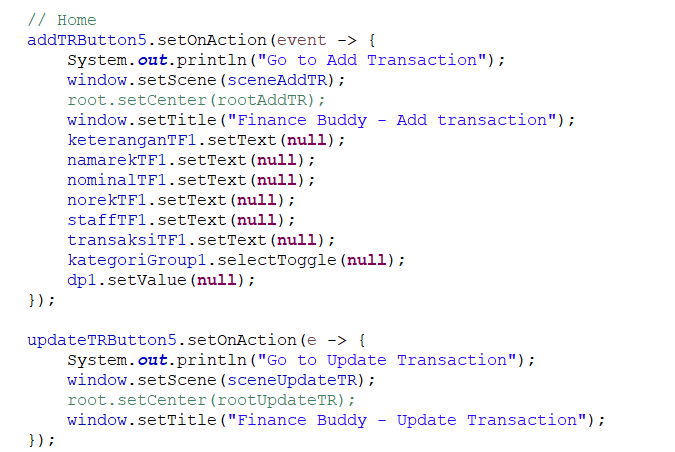
* 1. Update Transaction



*Part of Code*

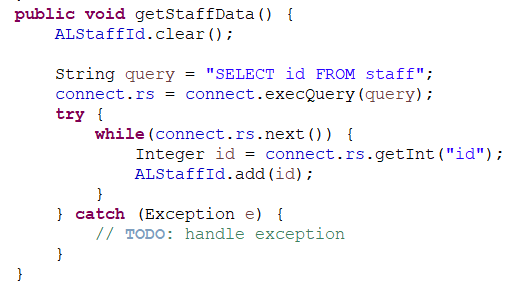
Pada update transaction kita kembali menggunakan logic untuk menyusun fungsi yang ada pada setiap tombol dengan melibatkan query UPDATE. Query dan logic yang terlibat antara lain yaitu query DELETE dan UPDATE. Terakhir kita juga melibatkan fungsionalitas dari back button agar pengguna dapat kembali ke scene yang sebelumnya.

* 1. Home



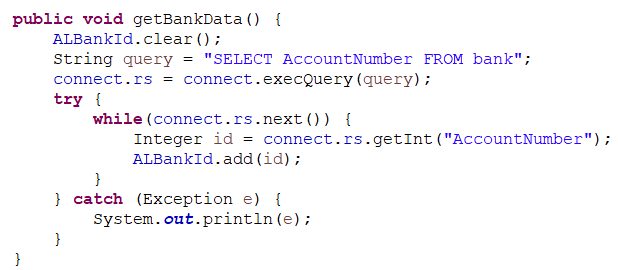
Berikut adalah bagian code yang bertujuan untuk memberikan fungsionalitas pada tombol home yaitu tombol update dan add.

1. Method ‘getStaffData’



Method get staff data bertujuan untuk mengambil data berupa ID dari staff. Sehingga nantinya kita dapat mengambil semua tipe data Staff berdasarkan ID yang ada pada database.

1. Method ‘getBankData’



Method ini bertujuan untuk mengambil data-data bank pada database berdasarkan account number. Kemudian, kita dapat mengambil data-data bank berdasarkan ID yang dimiliki oleh tiap bank.

1. Method ‘validationTF’



Method ini bertujuan untuk memberikan logic validasi pada saat pengguna ingin melakukan add transaction. Yaitu dengan mencegah pengguna dalam melakukan submit ketika salah satu label belum terisi.

1. Method ‘Start’



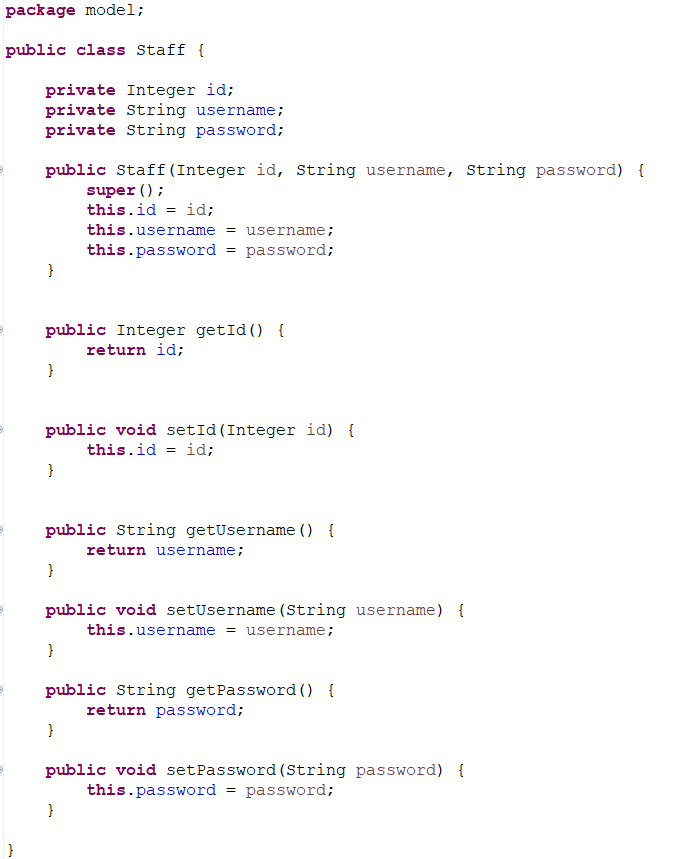
Metode ini bertujuan untuk menetapkan primary stage pada aplikasi tersebut. Serta menetapkan ukuran window yang diinginkan dengan melakukan penyesuaian ukuran pada setiap scene yang ada. Terakhir terdapat method shows yang berguna untuk menjalankan aplikasi tersebut ketika program akan dijalankan.

* **Model for Bank, Staff, and Transaction**
  + Bank



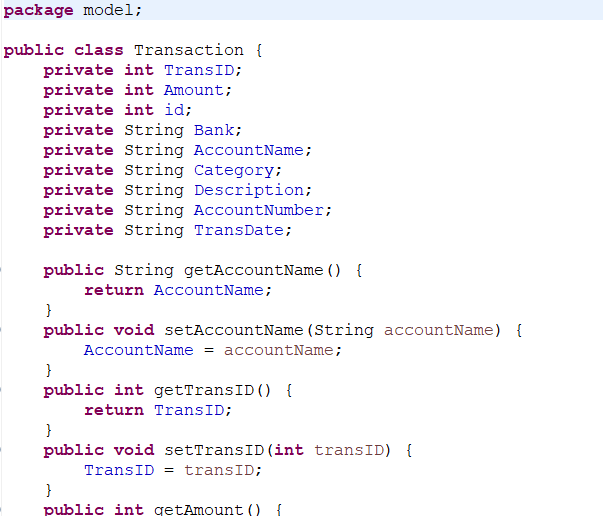
Berikut adalah file model bank yang berfungsi dalam menerima, mengembalikan dan menetapkan data-data pada bank.

* + Staff



Berikut adalah file model Staff yang berfungsi dalam menerima, mengembalikan dan menetapkan data-data pada Staff.

* + Transaction



*Part of Code*

Berikut adalah file model transaction yang berfungsi dalam menerima, mengembalikan dan menetapkan data-data pada transaction.

* **Connect.java**

1. Package and Import



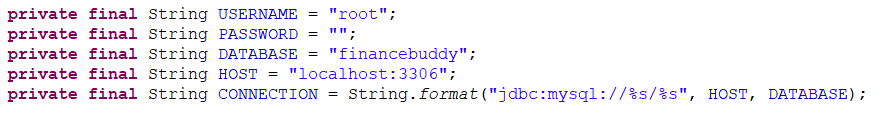
Pada file Connect.java kita menggunakan package yang bernama ‘util’ dan melakukan import module java.sql yang ingin kita pakai untuk melakukan koneksi terhadap database.

1. Deklarasi class Connect



Kode diatas digunakan untuk mendeklarasikan class Connect dari file tersebut.

1. Detail Koneksi



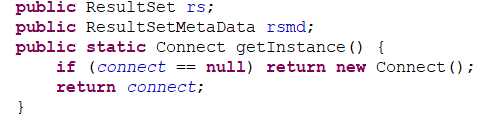
Baris-baris ini menentukan detail koneksi database seperti nama pengguna, kata sandi, nama database, dan host dengan port. Variabel CONNECTION adalah string berformat yang mewakili URL koneksi JDBC.

1. Connection and Statement Object



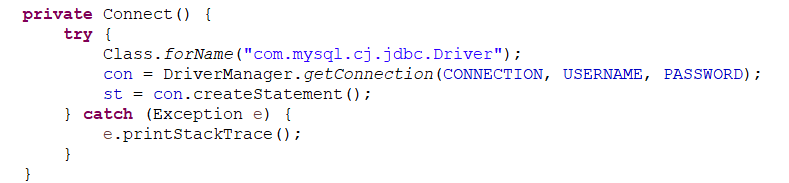
Kode ini mengimplementasikan pola desain Singleton untuk memastikan bahwa hanya satu instance dari kelas Connect yang dibuat. Metode getInstance digunakan untuk mendapatkan instance tunggal.

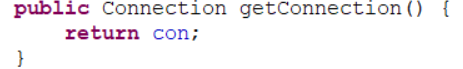
1. Design Singleton



Kode ini mengimplementasikan pola desain Singleton untuk memastikan bahwa hanya satu instance dari kelas Connect yang dibuat. Metode getInstance digunakan untuk mendapatkan instance tunggal.

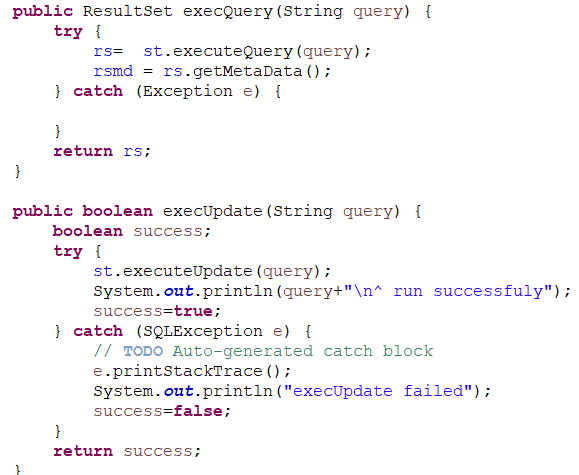
1. Constructor and Connection Access Method





Konstruktor menginisialisasi objek Connect. Ini memuat driver MySQL JDBC, membuat koneksi ke database menggunakan detail koneksi yang disediakan, dan membuat objek Pernyataan.

1. Query Execution Method and Update Method



Metode ini digunakan untuk menjalankan query dan mengembalikan ResultSet. Itu juga mengambil metadata dari kumpulan hasil. Sedangkan metode update digunakan untuk menjalankan query pembaruan (misalnya, INSERT, UPDATE, DELETE) dan mengembalikan boolean yang menunjukkan apakah pembaruan berhasil. Itu juga mencetak kueri ke konsol.

## Question and Answer

Beberapa pertanyaan yang ditanyakan serta jawaban yang kami berikan pada saat sesi Q&A yaitu sebagai berikut.

* Q1: Mengapa kami masih menggunakan atribut *username* padahal sudah ada atribut *name* pada entitas Staff?

A1: *Username* dan *name* pada aplikasi sengaja kami bedakan untuk alasan keamanan. Karena tidak semua staff memiliki wewenang untuk melihat catatan transaksi perusahaan, kami mengharuskan staff yang berwenang untuk melakukan *login* pada aplikasi menggunakan *username* dan *password* yang sudah terdaftar. Hal ini dilakukan untuk membatasi akses pada aplikasi.

* Q2: Bagaimana cari kami mendapatkan *profit* dari aplikasi Finance Buddy?

A2: Kami akan menjual aplikasi kami kepada perusahaan-perusahaan yang membutuhkan. Apabila ada suatu perusahaan yang membutuhkan aplikasi pencatatan keuangan, kami akan menawarkan aplikasi kami. Untuk menggunakannya, perusahaan tersebut harus membayar sejumlah uang. Dari situlah sumber *profit* kami.

* Q3: Apakah kami memiliki rencana untuk mengembangkan aplikasi kami dengan menambahkan fitur laporan keuangan?

A3: Kami menjual aplikasi kami kepada perusahaan sehingga apabila klien kami ingin melakukan penambahan fitur, kami dapat menyesuaikan aplikasi kami dengan permintaan klien. Hal tersebut tentunya akan membutuhkan biaya tambahan.

# CHAPTER III – CONCLUSION

Untuk fitur *Log In, Add Transaction*, dan *Update Transaction* berjalan dengan baik. Pada saat proses coding, terjadi beberapa perubahan dari rencana awal kami. Pada awalnya, fitur *Update Transaction* berada di dalam menu *Add Transaction*. Namun, pada akhirnya kami memutuskan untuk membuat fitur Update Transaction di luar menu Add Transaction agar memudahkan apabila Staff ingin meng-update transaksi tanpa menambahkan transaksi baru. Tampilan UI kami juga mengalami beberapa perubahan, terutama pada menu *Dashboard*, apabila dibandingkan dengan draft awal.