

**TUGAS AKHIR  
PEMROGRAMAN BERBASIS OBJEK  
APLIKASI PENJUALAN TEMBAKAU**



Oleh :

M. INDRA GUNAWAN 152020068

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL BANDUNG  
2022**

## **KATA PENGANTAR**

Dengan menyebut nama Tuhan Yang Maha Esa, Penulis panjatkan puja dan puji syukur atas ke hadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan laporan akhir ini guna memenuhi tugas untuk mata kuliah Pemrograman Berbasis Objek dengan judul : **“APLIKASI PENJUALAN TEMBAKAU”**.

Penulis menyadari dalam penulisan laporan ini tidak terlepas dari bantuan banyak pihak yang dengan tulus memberikan bantuan, saran, dan kritik sehingga laporan akhir ini dapat terselesaikan.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna dikarenakan keterbatasan pengalaman dan waktu pengerjaan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari berbagai pihak yang membaca laporan ini. Akhirnya penulis berharap semoga laporan ini membawa banyak manfaat bagi pembaca.

Bandung 30 Mei 2022

M. Indra Gunawan

## DAFTART ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTART ISI.....</b>	<b>3</b>
<b>BAB I PENDAHULUAN.....</b>	<b>4</b>
1.1 Latar Belakang.....	4
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penulisan .....	4
1.4 Batasan Masalah .....	4
1.5 Sistematika Penulisan Laporan .....	5
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1 Pengertian Tembakau.....	6
2.2 Pemograman Berbasis Object.....	6
2.3 Pengertian Java .....	6
2.4 Pengertian Netbeans .....	7
2.5 Pengertian Database .....	7
<b>BAB III MODEL DESKRIPSI SISTEM.....</b>	<b>8</b>
3.1 Studi Kasus .....	8
3.2 Fitur.....	8
3.3 Flowchart dan Penjelasan .....	8
3.4 Class Diagram .....	12
3.5 Database.....	13
3.6 Peta Aplikasi.....	14
3.7 Rancangan Mock up UI.....	15
<b>BAB IV IMPLEMENTASI .....</b>	<b>17</b>
4.1 Hasil Output .....	17
a) Halaman Login Dan Registrasi.....	17
b) Halaman Customer .....	17
c) Halaman Admin .....	18
4.2 Kode Program Dan Penjelasan.....	19
a) Package com.....	19
b) Package Execute .....	31
c) Package View.....	48
<b>BAB VPENUTUP.....</b>	<b>63</b>
5.1 Kesimpulan.....	63
5.2 Saran .....	63

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dengan perkembangannya teknologi saat ini banyak pengusaha meningkatkan pelayanan demi kualitas yang sangat penting bagi pembisnis, terutama dalam bidang penjualan tembakau dan segala peralatanya. Bidang ini akan sangat membantu apabila di implementasikan dengan penggunaan teknologi yang tepat. Banyak orang berlomba-lomba untuk meningkatkan kualitas pelayanannya sehingga ide bisnis ini sangat berpeluang besar bagi pengusaha yaitu penjualan tembakau yang saat ini banyak varian tembakau dan alat yang sudah modern untuk membuat sebuah rokok.

Pada saat ini banyak orang yang mengeluh soal harga rokok yang bungkus dengan harga pajaknya naik sehingga orang-orang lebih memilih cara untuk menghemat, yaitu membuat rokok sendiri dengan cara yang sudah modern yang dulunya dengan tangan sekarang bisa dengan alat, dengan perkembangan zaman saat ini alat linting sudah di buat untuk memperbuah perokok untuk menghebat dan juga tembakau yang banyak juga variannya oleh karena itu banyak orang yang tertarik untuk menggunakan tembakau dengan harga yang murah dengan membeli sebuah tembakau di toko tembakau.

Bisnis penjualan tembakau ini akan lebih baik mudah dilakukan apabila memasukan unsur teknologi dan informasi di dalamnya yang dimana bisnis ini bisa dapat meningkatkan efisiensi dalam pengelolaan data dan mempermudah pelanggan untuk membeli sebuah tembakau dan alat lainnya. Dengan perkembangan yang sekarang dari sisi teknologi maka penulis memiliki gagasan untuk membuat aplikasi untuk menjual sebuah alat linting dan tembakau dan aplikasi ini di peruntukan untuk yang sudah 18+ dan mempermudah pelanggan agar bisa lebih menghemat pengeluaran.

### **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara pengelolaan aplikasi penjualan tembakau.
2. Bagaimana perancangan sebuah sistem penjualan agar lebih optimal dan membuat pelanggan nyaman dengan harga yang lebih murah.

### **1.3 Tujuan Penulisan**

Laporan ini dibuat dengan tujuan untuk memenuhi tugas akhir dari mata kuliah Pemrograman Berbasis Objek. Sedangkan tujuan dari kegiatan ini adalah untuk membuat, merancang dan mengimplementasikan rancangan sistem agar dapat berjalan sesuai dengan fungsinya.

### **1.4 Batasan Masalah**

Batasan – batasan yang penulis definisikan pada laporan ini antara lain:

1. Sistem menggunakan database sebagai tempat menyimpan data, yang dimana project ini menggunakan penyimpanan data base MySQL.
2. Pada aplikasi ini tidak ada fitur search pada sisi admin dan customer yang digunakan untuk mencari barang yang di perlukan.

3. Pada aplikasi ini tidak ada fitur clear dari sisi admin ketika selesai melakukan proses fitur.
4. Aplikasi penjualan tembakau ini memiliki 2 hak akses, yaitu dari sisi user dan admin tidak untuk bagian kasir.
5. Aplikasi ini belum bisa cetak struk transaksi.
6. Customer tidak bisa menambahkan barang yang ingin di beli, jika ingin membeli selesaikan transaksi terlebih dahulu baru membeli barang yang lain atau istilahnya keranjang belanja.
7. Profil customer ketika selesai di edit customer harus keluar aplikasi terlebih dahulu jika ingin mengetahui perubahan yang di edit.
8. Daftar barang belum memiliki gambar yang sesuai pada daftar barang.
9. Pada aplikasi penjualan ini tidak menampilkan stok barang yang tersisa.

### **1.5 Sistematika Penulisan Laporan**

Sistematika penulis laporan ini mangacu apada subbab berikut ini.

1. **BAB I PENDAHULUAN**  
Pada bab ini menjelaskan tentang latar belakang dan tujuan pembuatan aplikasi, rumusan masalah, tujuan penulisan laporan dan batasan masalah dari penulis buat.
2. **BAB II LANDASAN TEORI**  
Pada bab 2 ini menjelaskan tentang teori-teori penunjang yang digunakan dalam pembuatan laporan ini.
3. **BAB III PERANCANGAN**  
Pada bab 3 ini menjelaskan tentang studi kasus, alur sistem dan database sistem, sitmaps apliaksi dan mcup user interface.
4. **BAB IV IMPLEMENTASI**  
Pada bab 4 ini menampilkan hasil output aplikasi yang berbabis GUI yang sudah di buat dan menampilkan kode program yang sudah di buat dengan menggunakan bahasa pemograman java dan juga penjelasan tentang program tersebut.
5. **BAB V PENUTUP**  
Pada bab 5 ini berisi tentang kesimpulan dan saran dari penulis tentang aplikasi ini

## BAB II

### LANDASAN TEORI

#### 2.1 Pengertian Tembakau

Tembakau adalah tanaman musiman yang tergolong dalam tanaman perkebunan. Pemanfaatan tanaman tembakau terutama pada daunnya yaitu untuk pembuatan rokok. Tanaman tembakau diklasifikasikan sebagai subfamily Nicotianae, *Nicotiana tabacum* dan *Nicotiana rustica* mempunyai perbedaan yang jelas. Pada *Nicotiana tabacum*, daun mahkota bunganya memiliki warna merah muda sampai merah, mahkota bunga berbentuk terompet panjang, daunnya berbentuk lonjong pada ujung runcing, kedudukan daun pada batang tegak, merupakan induk tembakau sigaret dan tingginya sekitar 120 cm. Proporsi kadar nikotin banyak bergantung kepada varietas, tanah tempat tumbuh tanaman, dan kultur teknis serta proses pengolahan daunnya. Kandungan nikotin yang ada dalam tembakau merupakan golongan alkaloid yang terdapat dalam famili Solanaceae. Kadar nikotin berkisar antara 0,6.

#### 2.2 Pemrograman Berbasis Object

Pemrograman berorientasi objek (*Object Oriented Programming* atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. OOP merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. OOP adalah paradigma pemrograman yang diorientasikan pada objek yang berisi atribut dan method. Semua data dan fungsi di dalam paradigma OOP dibungkus dalam *kelas-kelas* atau *objek-objek*..

Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Pemrograman berorientasi objek dalam melakukan pemecahan suatu masalah tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut.

#### 2.3 Pengertian Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh *Sun Microsystems* sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan Smalltalk, namun dirancang agar lebih mudah dipakai dan platform independent, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer. Java digunakan untuk membangun suatu sistem yang berbasis desktop sampai dengan mobile.

Java sebagai sebuah platform dibagi menjadi beberapa bagian, antara lain :

1. Bahasa Pemrograman Java adalah suatu bahasa yang murni Object Oriented Programming. Semua kriteria OOP pada Java antara lain abstraksi data dan enkapsulasi, Inheritance dan Polimorfism.
2. JVM (Java Virtual Machine) Dengan JVM ini maka semboyan Java yaitu "write once run everywhere" dapat direalisasikan dimana dengan JVM, suatu program tidak lagi tergantung pada terhadap platform OS yang digunakan dan berinteraksi dengan OS. Kompilasi terhadap suatu file Java (source) akan menghasilkan suatu file byte code (extension class) dimana byte code Java adalah sama untuk semua

platform, sehingga ketika menjalankan sebuah program Java. JVM akan menangani segala sesuatu yang berhubungan dengan OS dan menjalankan byte code yang telah dihasilkan.

3. Java Basic API (J2SDK) Java Basic API adalah sekumpulan class yang disediakan oleh Java untuk melakukan proses pengembangan terhadap aplikasi Java.

## **2.4 Pengertian Netbeans**

Netbeans adalah aplikasi IDE (*Integrated Development Environment*) yang berbasis Java. Menurut laman web resmi Netbeans, “NetBeans IDE adalah sebuah alat pengembangan untuk menulis program, mengompilasi, mencari kesalahan dan mengembangkan program. Netbeans IDE ditulis dalam Bahasa Pemrograman Java, dan juga dapat mendukung bahasa pemrograman lain”.

## **2.5 Pengertian Database**

Basis Data terdiri dari kata basis dan data. Basis dapat diartikan sebagai markas atau gudang. Sedangkan data adalah catatan atas kumpulan fakta dunia nyata yang mewakili objek. Secara kesatuan, pengertian basis data adalah kumpulan data dalam bentuk file/tabel/arsip yang saling berhubungan dan tersimpan dalam media penyimpanan elektronik, untuk kemudahan dalam pengaturan, pemilahan, pengelompokan dan pengorganisasian data sesuai tujuan.

## BAB III

### MODEL DESKRIPSI SISTEM

#### 3.1 Studi Kasus

Dalam aplikasi penjualan tembakau ini terdapat 2 user yaitu admin dan customer yang dimana customer bisa mendaftar terlebih dahulu agar bisa login di aplikasi penjualan ini dan setelah itu customer bisa memilih barang dan membeli barang yang ada di toko penjualan tembakau ini dan melakukan transaksi sesuai harga yang ada di daftar barang. Setelah selesai melakukan transaksi customer bisa logout dengan menu logout.

Dan untuk admin bisa melakukan penambahan barang yang ada di toko penjualan tembakau ini dan mendelet di daftar barang yang mau di edit dan juga admin bisa menghapus data barang yang sudah ada, admin juga bisa menambahkan admin lain, mengupdate data admin dan menghapus data admin, kemudian admin juga bisa melihat data customer dan juga menghapus data customer, setelah itu admin bisa melihat data history transaksi yang dilakukan oleh customer dan data-data yang sudah dimasukkan akan masuk ke database.

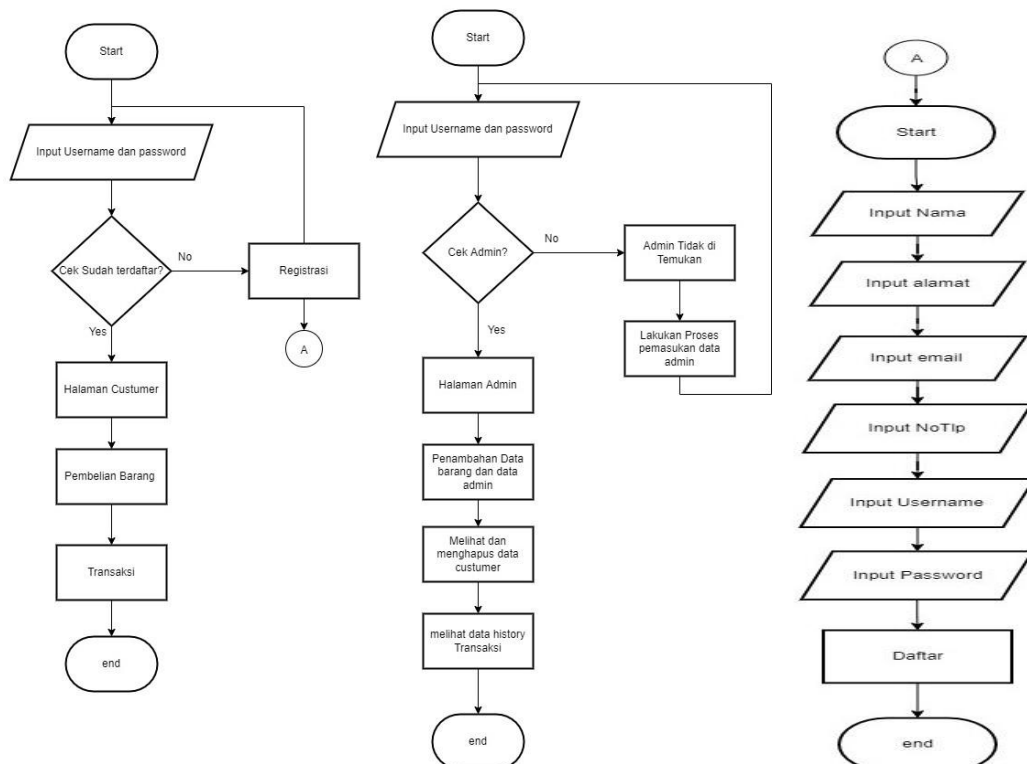
#### 3.2 Fitur

Fitur yang tersedia pada aplikasi penjualan tembakau ini bermacam-macam, dari segi customer bisa mendapatkan fitur daftar barang yang dimana customer bisa memilih dan membeli barang yang diinginkan dan setelah itu fitur selanjutnya yaitu fitur transaksi yang dimana pada fitur pembelian data yang sudah dibeli akan masuk ke bagian transaksi.

Dan dari segi admin mendapatkan fitur create, update, delete, read data pada suatu tabel database yang dimana tidak semua fitur itu bisa digunakan hanya bisa digunakan di bagian data barang, data admin, dan untuk data customer hanya bisa delete, dan juga di bagian transaksi hanya bisa read.

#### 3.3 Flowchart dan Penjelasan

##### 1. Login



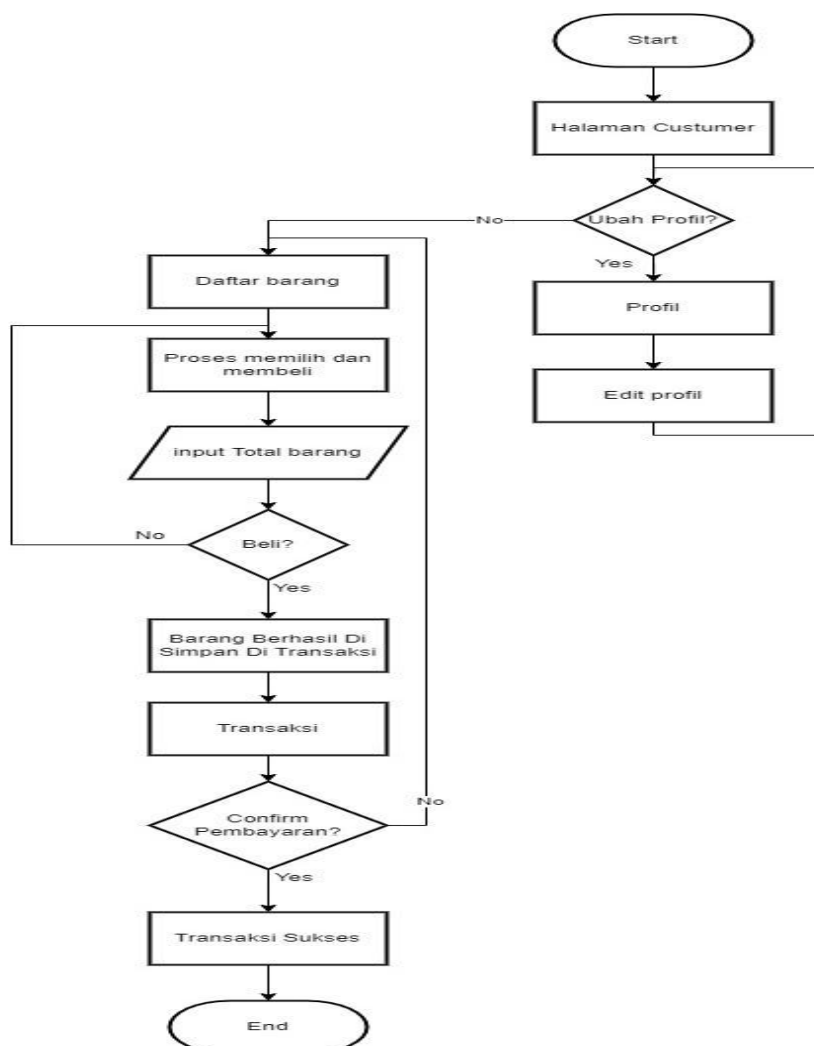


- Penjelasan

Flowchart bagian login memiliki hak akses 2 yaitu satu sebagai user dan satunya laginya sebagai admin, di mulai dari admin memulai dengan start trus input username dan password admin dan ketika pasword dan usernamne itu gak bisa masuk maka akan menampilkan admin tidak ditemuka maka dari itu cek akun jika belum mempunyai akun maka admin yang pertama harus memasukan data admin selanjutnya dan untuk data admin yang pertama di masukan lewat database. Setelah melakukan proses pemasukan data admin dan sudah memiliki username dan password maka akan masuk ke halam admin dan di bagian admin akan ada proses penambahan dan perubahan data barang dan data admin setealh itu proses melihat,menghapus data custumer dan juga melihat data history transaks dan keluar.

Selanjutnya dari sisi user mulai dari star masukan username dan password jika sudah miliki akun akan masuk jika belum lakukan registrasi terlebih dahulu dengan menginputkan data yang di perlukan dan daftar setelah itu masuk ke bagian halaman user yang dimana user bisa membeli dan memilih barang yang di inginkan dan setelah itu melakukan pembayran/proses transaksi.

## 2. User

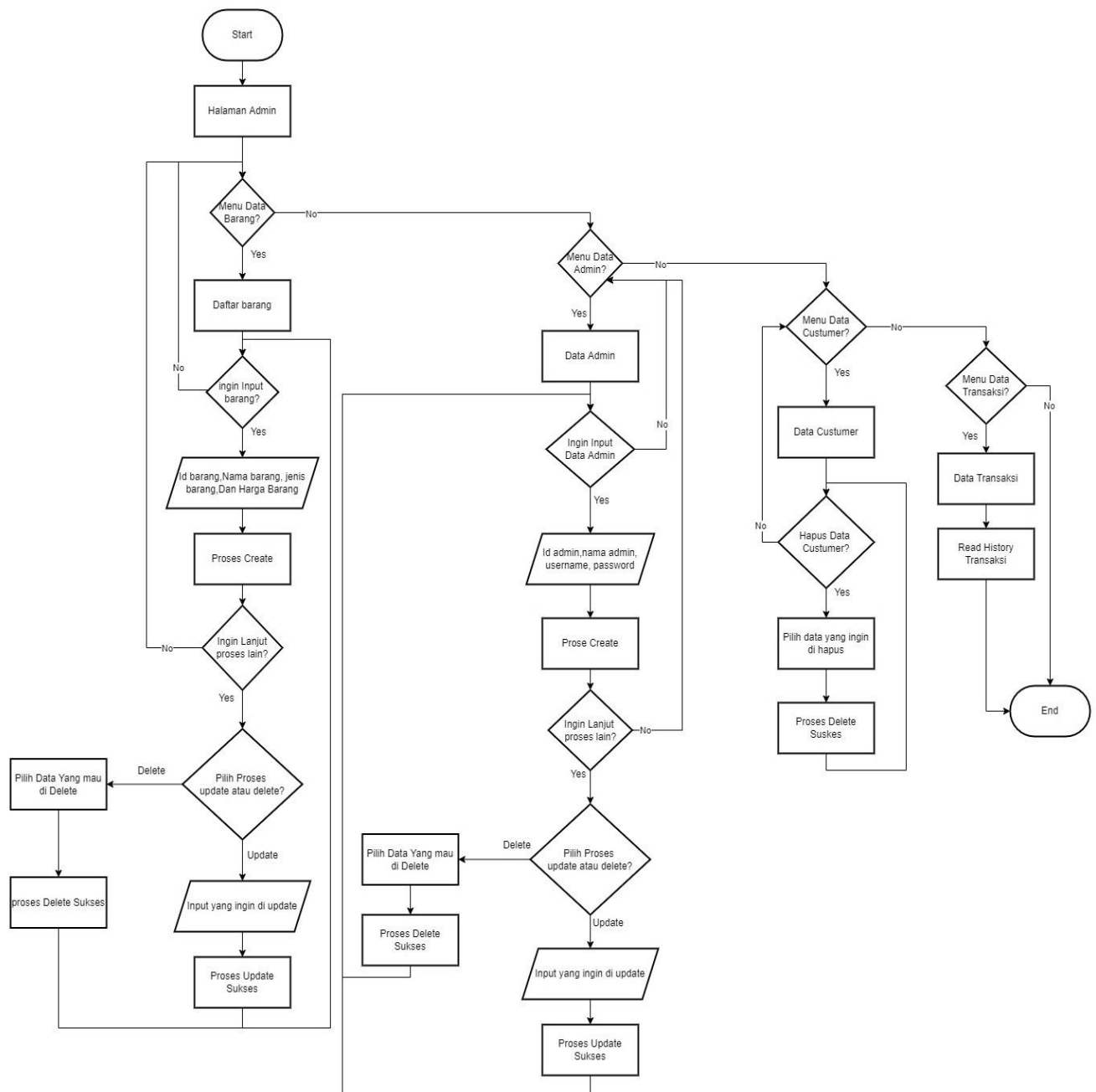


- Penjelasan

Di bagian flowchart user ini setelah user berhasil login dan masuk pada halaman user maka akan diberikan opsi ingin mengubah profil jika no maka di bagian profil dan edit profil. Jika tidak maka akan ke bagian menu daftar barang dan disini user/customer bisa melakukan pembelian dan memilih barang yang ingin di beli.

Setelah di pilih barangnya input kan total barang yang ingin di beli. Jika ingin membeli maka barang akan berhasil di simpan di transaksi di bagian transaksi tinggal proses transaksinya jika ingin confirm pembayaran jika iya maka transaksi sukses dan jika tidak maka akan kembali di pemilihan barang yang ingin di beli

### 3. Admin



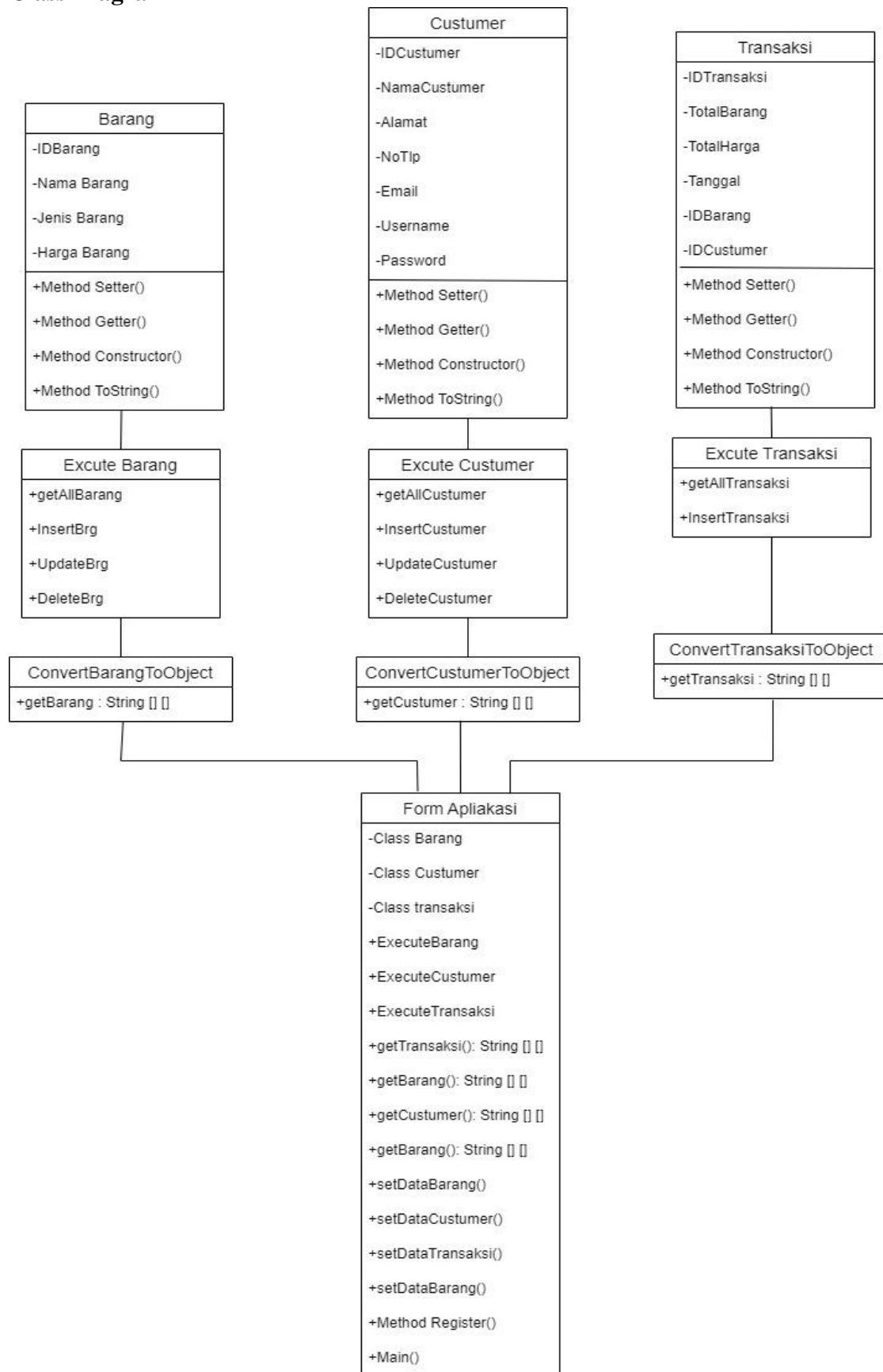
- Penjelasan

Pada flowchar admin di bagian sini setelah selesai melakukan login maka akan masuk ke bagian halaman admin. Jika ke menu barang maka akan masuk ke menu barang jika tidak maka akan masuk ke menu-menu selanjutnya. Jika masuk ke menu barang maka akan di hadapkan dengan daftar barang dan disini admin bisa menambahkan data. Jika ingin input barang maka inputkan bagian-bagian barang seperti nama dan lain sebagainya setelah itu proses create setelah itu ingin lanjut ke proses lain jika tidak maka akan kembali ke menu data barang. Jika iya maka melanjutkan proses ingin melanjutkan proses delete atau update, jika update maka input data yang ingin di update dan proses update dan akan kembali ke input barang jika pilih proses delete maka pilih data yang ingin di delete maka akan di proses delete sukses dan kembali ke input barang. Jika sudah tidak ada yang ingin di inputkan lagi maka pilih bagian menu data admin.

Jika masuk ke menu admin maka akan di hadapkan dengan data admin dan disini admin bisa menambahkan data admin. Jika ingin input data admin maka inputkan data admin yang di perlukan seperti nama dan lain sebagainya setelah itu proses create, setelah itu ingin lanjut ke proses lain, jika tidak maka akan kembali ke menu data admin. Jika iya maka melanjutkan proses ingin melanjutkan proses delete atau update, jika update maka Input data yang ingin di update dan proses update dan akan kembali ke input data admin jika pilih proses delete maka pilih data yang ingin di delete maka akan di proses delete sukses dan kembali ke input data admin. Jika sudah tidak ada yang ingin di inputkan lagi maka pilih bagian menu data customer.

Dibagian menu customer admin akan melihat data customer. Jika admin ingin menghapus data customer maka pilih data yang akan di hapus dan proses hapus sukses jika tidak ada maka kembali ke menu data customer dan memilih menu selanjutnya. Menu selanjutnya menu data transaksi yang dimana di bagian data transaksi admin hanya bisa melihat data history transaksi saja jika sudah selesai maka akan end. Dan kita pada menu transaksi tidak ingin melihat data transaksi maka proses akan end.

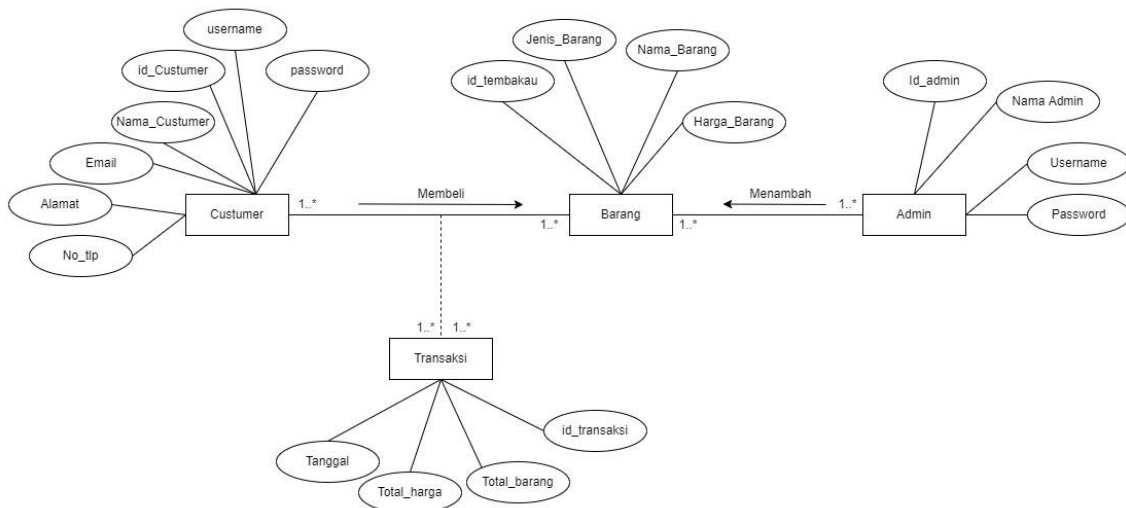
### 3.4 Class Diagram



### 3.5 Database

#### 1. Entity Relationship Diagram

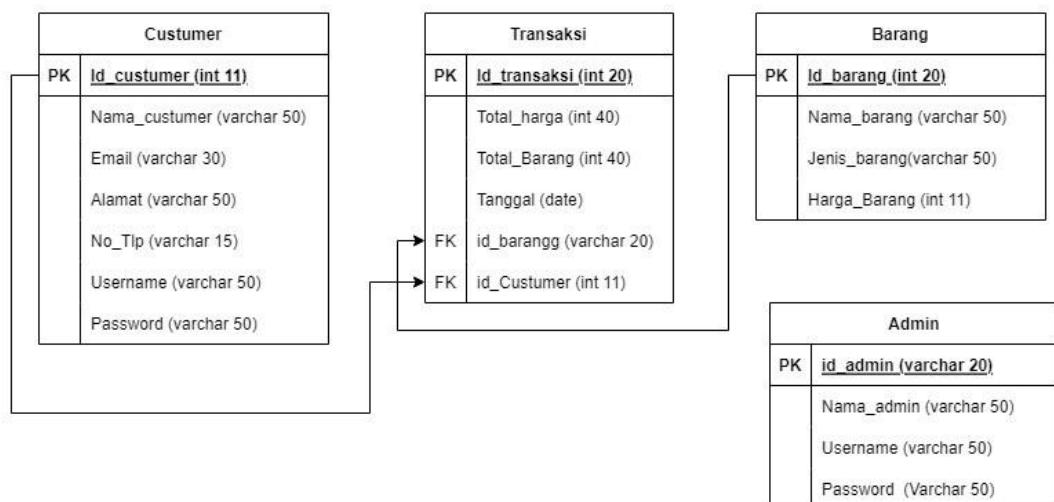
Entity Relationship Diagram adalah bentuk paling awal dalam melakukan perancangan basis data relasional dan juga merupakan teknik penggambaran model jaringan suatu basis data dengan susunan data yang disimpan dalam sistem secara abstrak dengan menggunakan notasi dan symbol.



#### 2. Tabel Relationship Diagram

Tabel Relationship Diagram adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi yang berbentuk tabel dan memiliki atributnya masing-masing yang saling terhubung dengan atribut di tabel yang lain.

#### TRD

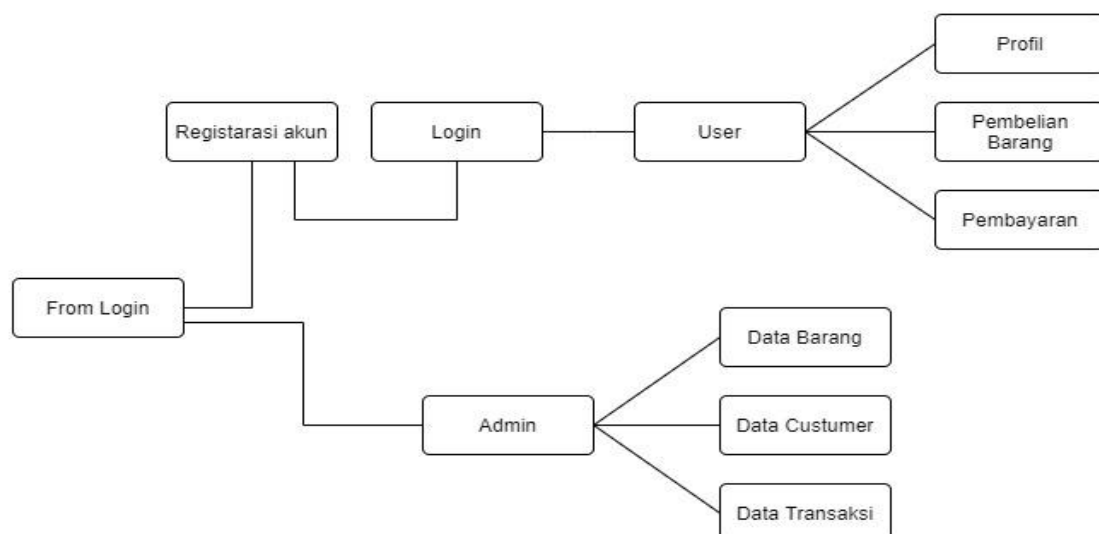


### 3. Data Dictionary System

Data Dictionary System adalah sebuah kamus yang mendefinisikan nama-nama rinci data dan format penyimpanannya sehingga sebuah sistem informasi dapat dipahami secara umum.

1. Barang							
Name Attribute	PK/FK	Datatype	Length/set	Allow Null	Default	Tautan Ke	Komentar
ID_Barang	PK,FK	Varchar	20	Not Null	No Default		id_barang untuk setiap barang memiliki relasi dengan tabel transaksi, Contoh Aht-01
Nama_Barang		Varchar	50	Not Null	No Default		nama Barang dari setiap barang yang ingin di jual, contoh Tbk vanilla
Jenis_Barang		Varchar	50	Not Null	No Default		jenis barang untuk menentukan jenis barang apa saja yang ada, contoh tembakau, alat roll, papir
Harga_Barang		Int	11	Not Null	No Default		harga barang untuk menyimpan harga barang sesuai jenis barang dan nama barangnya, contoh 25.000
2. Customer							
Name Attribute	PK/FK	Datatype	Length/set	Allow Null	Default	Tautan Ke	Komentar
ID_Customer	PK,FK	Int	11	Not Null	Auto_Incremen		id customer sesuai registrasi yang sudah di lakukan contoh 1
Nama_Customer		Varchar	50	Not Null	No Default		nama customer sesuai registrasi yang sudah di lakukan, contoh kajo
Alamat		Varchar	50	Not Null	No Default		alamat sesuai registrasi yang sudah di lakukan, contoh Jl.Gagak
No_Tlp		Varchar	15	Not Null	No Default		no tlp sesuai registrasi yang sudah di lakukan contoh 083631421312
Email		Varchar	50	Not Null	No Default		email digunakan untuk login dan sesuai registrasi yang sudah di lakukan, contoh kajosutarman@gmail.com
Username		Varchar	50	Not Null	No Default		username digunakan untuk login dan sesuai registrasi yang sudah di lakukan contoh kajosut12
Password		Varchar	50	Not Null	No Default		password digunakan untuk login dan sesuai registrasi yang sudah di lakukan, contoh kajo123
3. Transaksi							
Name Attribute	PK/FK	Datatype	Length/set	Allow Null	Default	Tautan Ke	Komentar
ID_Transaksi	PK,FK	Int	20	Not Null	Auto_Incremen		id transaksi, contoh, 01
Total_Barang		Int	40	Not Null	No Default		total barang, digunakan untuk menghitung total barang yang di beli, contoh 3
Total_Harga		Int	40	Not Null	No Default		total harga, hasil kalkulasi dari harga dengan total barang yang di beli, contoh 25000
Tanggal		Varchar	50	Not Null	No Default		tanggal digunakan untuk menandai tanggal transaksi yang di lakukan, contoh 02-06-2022
ID_Barang	FK	Varchar	20	Not Null	No Default	On Update Casade, On Delete Casade	id barang, digunakan untuk mengambil nama barang dan jenis barang dari tabel barang, contoh id barang tbk-03
ID_Customer	FK	Int	11	Not Null	No Default	On Update Casade, On Delete Casade	id customer, yang di ambil dari tabel customer untuk nama dan id nya, contoh kajos id 1
4. Admin							
Name Attribute	PK/FK	Datatype	Length/set	Allow Null	Default	Tautan Ke	Komentar
ID_Admin	PK	Varchar	20	Not Null	No Default		id_admin digunakan untuk admin saja, contoh 001
Nama_Admin		Varchar	50	Not Null	No Default		nama_admin digunakan untuk menyimpan nama admin, contoh Indra gunawan
Username		Varchar	50	Not Null	No Default		username digunakan untuk login sebagai admin, contoh indragun12
Password		Varchar	50	Not Null	No Default		password digunakan untuk login sebagai admin, contoh ind123

### 3.6 Peta Aplikasi



### 3.7 Rancangan Mock up UI

Form Login

Username

Password

Login

Registrasi

Form Registrasi

Nama

Alamat

No Telp

Email

Username

Password

Daftar

#### *Mockup Login dan Register*

Toll-Bar

ProfilDaftarBarangTransaksiLogOut

SELAMAT DATANG DI TOBACCO SHOP

Toll-Bar

DataBarangDataAdminDataCustomerDataTransaksiLogOut

SELAMAT DATANG DI TOBACCO SHOP

Toll-Bar

ProfilDaftarBarangTransaksiLogOut

ID User

Edit Profil

Nama User

Email

Alamat

No Telp

Username

Password

Toll-Bar

DataBarangDataAdminDataCustomerDataTransaksiLogOut

ID Barang

Create

Nama Barang

Update

Jenis Barang

Delete

Harga Barang

Tabel Barang

#### *Mockup Kiri User dan Kanan Admin*

Toll-Bar

Profil

Daftar  
Barang

Transaksi

LogOut

Tabel Barang

ID Barang

Nama Barang

Jenis Barang

Harga Barang

Total Barang

Beli

Toll-Bar

Data  
Barang

Data  
Admin

Data  
Customer

Data  
Transaksi

LogOut

ID Admin

Nama Admin

Username

Password

Create

Update

Delete

Tabel Admin

Toll-Bar

Profil

Daftar  
Barang

Transaksi

LogOut

ID Barang

Nama Barang

Jenis Barang

Harga Barang

Harga Barang

Total Harga

Tanggal

ID Customer

Confirm

Toll-Bar

Data  
Barang

Data  
Admin

Data  
Customer

Data  
Transaksi

LogOut

Nama Customer

Alamat

No Telp

Email

Username

Password

Delete

Tabel Customer

Toll-Bar

Data  
Barang

Data  
Admin

Data  
Customer

Data  
Transaksi

LogOut

Tabel Transaksi

Mockup Kiri User dan Kanan Admin



## BAB IV

### IMPLEMENTASI

#### 4.1 Hasil Output

##### a) Halaman Login Dan Registrasi

The screenshot displays two side-by-side windows from a Java Swing application. The left window, titled 'TOBACO SHOP', contains a login form with fields for 'Username' (filled with 'sadewa12') and 'Password' (filled with 'tobiaz'). Below the password field is a 'Show Password' checkbox which is checked. A 'Login' button is positioned below the fields. At the bottom of the window are an 'Exit' button on the left, a link 'Sudah Memiliki Akun?' in the center, and a 'Register' button on the right. The right window, titled 'Form Registrasi', contains a registration form with fields for 'Nama' (filled with 'TN Sadewa'), 'Alamat' (filled with 'Jl.Raya Grove Street'), 'No Telp' (filled with '085345324512'), 'Email' (filled with 'tnsadewa12@gmail.com'), 'Username' (filled with 'sadewa12'), and 'Password' (filled with 'gshketua'). It also has a checked 'Show Password' checkbox. At the bottom are 'Daftar' and 'Back' buttons.

##### - Penjelasan

Halaman login memasukan username dan password jika sudah ada, jika belum maka lakukan registrasi terlebih dahulu di bagain registrasi-->form registrasi masukan data yang ada form registrasi terus daftar dan masukan username dan password kembali.

##### b) Halaman Customer

The screenshot shows two side-by-side windows of the 'TOBACCA SHOP' application. Both windows feature a top navigation bar with icons and labels for 'Profil', 'Daftar Barang', 'Transaksi', and 'Log Out'. The left window displays a large, stylized logo for 'TOBACCA SHOP'. The right window shows the 'Profil User' section, which contains a form with the following data: ID User: 18, Nama User: Galih, Email: galih@gmail.com, Alamat: Jl Bandung, No Telp: 081264566314, Username: ucok, and Password: ucok. An 'Edit Profil' button is located at the bottom right of the profile form. Both windows also feature the 'TOBACCA SHOP' logo at the bottom.

ID_Barang	Nama_Barang	Jenis_Barang	Harga_Barang
AHL-01	AHL Roler	Alat Linting	10000
AHL-02	Rol becak	Alat Linting	30000
AHL-03	Rol Adjustable	Alat Linting	15000
Pp-01	buffalo bil strip	Papir	8000
Pp-02	Mascotte	Papir	10000
Pp-03	Dinamite	Papir	13000
Tbk-01	Tbk valina	Tembakau	23000
Tbk-02	Tbk Apel	Tembakau	25000
Tbk-03	Tbk Coffie	Tembakau	24000

**Pembelian Barang**

ID Barang :

Nama Barang :

Jenis Barang :

Harga Barang :

Total Barang :

**Transaksi**

ID Barang :  Jenis Barang :

Nama Barang :  Harga Barang :

Total Barang :  Tanggal :

Total Harga :  ID Customer :

## - Penjelasan

Di halaman user/customer akan di melihat tampilan dari aplikasi ada beberapa menu bar yang pertama menu bar untuk profil user/customer terus menu daftar barang dibagian ini bisa memilih dan membeli barang yang ingin di beli dan barang yang sudah di beli akan di tambikan di bagian transaksi dan di bagian transaksi tinggal menekan konfirmasi pembayaran.

## c) Halaman Admin

**TOBACCO SHOP**

**Input Data Barang**

ID Barang :

Nama Barang :

Jenis Barang :

Harga Barang :

**Daftar Barang**

ID_Barang	Nama_Barang	Jenis_Barang	Harga_Barang
AHL-01	AHL Roler Gas L	Alat Linting	10000
AHL-02	Rol becak	Alat Linting	30000
AHL-03	Rol Adjustable	Alat Linting	15000
Pp-01	buffalo bil strip	Papir	8000
Pp-02	Mascotte	Papir	10000
Pp-03	Dinamite	Papir	13000
Tbk-01	Tbk valina	Tembakau	23000
Tbk-02	Tbk Apel	Tembakau	25000
Tbk-03	Tbk Coffie	Tembakau	24000

**Input Data Admin**

ID Admin :

Nama Admin :

Username :

Password :  ☐ Show Password

ID_Admin	Nama_Admin	Username	Password
001	Indra	bismillahdulu	bismillah
002	Jender	jender12	jdr123
003	zaza	zaza123	zaza1

**Data Customer**

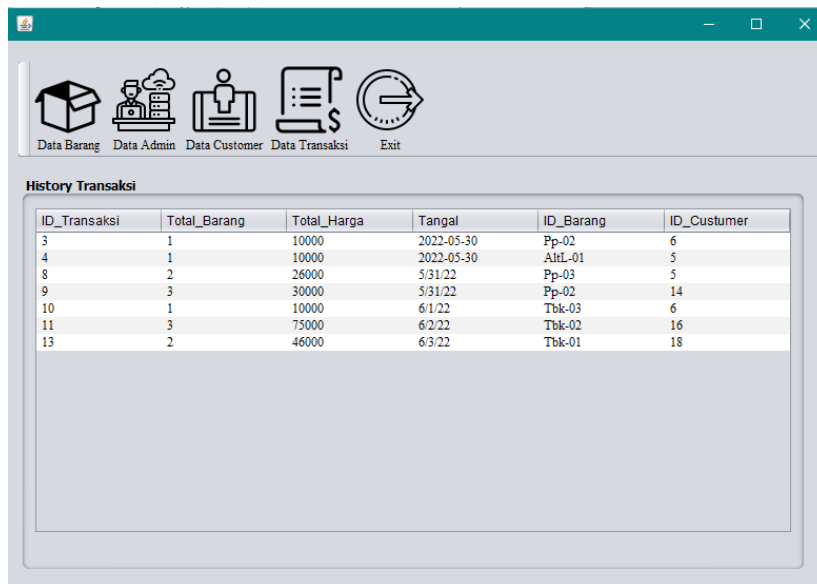
Nama Customer :  Username :

Alamat :  Password :

No\_Telpn :

Email :

ID_Customer	Nama_Custu...	Alamat	No_Tip	Email	Username	Password
3	Wahyu	Jl.Cikutra	081345672389	wahyu1@gmail.c...	wahyu1	wbyu1
5	Ipat	Jl.Katemaso	081345267895	ipat123@gmail.c...	ipat123	ipat1
6	Saka Kajo	Jl.Dago	081527399715	sakakajo15@gmail.com	sakakajo	*****
10	Tobias Sadewa	Jl.Mertani no4	082284895383	tsadewa12@gmail...	sandewa12	tobias
13	FatihHerwanann	Jl.BTN	086345321324	faicher12@gmail...	faiz12	123
14	Yudha san	Jl.UFC	081243243245	yudhasan12@gmail...	yudhasan	123
15	Aiko Morgan	Jl.MegaFall	082735383672	aiko15@gmail.com	aikom	aikomorgan15
16	Kuro Hazman	Jl.Cibaduyut	081534236243	kuro123@gmail....	kurohaz	krhaz



#### - Penjelasan

Pada halaman admin/gui admin ini admin di sambut oleh halaman awal bagian admin. Dan ada beberapa menu yang bisa di pilih oleh admin di menu data barang bisa cread barang yang di inginkan tinggal klik bagian tabel dan akan muncul di bagian text field dan menu selanjutnya menu data admin dibagian ini admin bisa menjalankan fungsi cread yang di inginkan menambahkan,mengedit,menghapus data admin.

Menu selanjutnya data custumer di bagian ini admin hanya bisa melihat data custumer dan juga menghapus data custumer. Dan untuk menu terakhir menu transaksi yang dimana admin bisa melihat history transaksi yang pernah di lakukan.

## 4.2 Kode Program Dan Penjelasan

### a) Package com

#### 1) Class Barang

```

/*
-----
    NAMA      : M. INDRA GUNAWAN
    NRP       : 152020068
    KELAS     : BB Project Tugas Besar
-----
*/
package com;
public class Barang {
    private String ID_Barang;
    private String Nama_Barang;
    private String Jenis_Barang;
    private int Harga_Barang;

    public Barang() {
    }
}

```

```

public Barang(String ID_Barang) {
    this.ID_Barang = ID_Barang;
}

public Barang(String ID_Barang, String Nama_Barang, String
Jenis_Barang, int Harga_Barang) {
    this.ID_Barang = ID_Barang;
    this.Nama_Barang = Nama_Barang;
    this.Jenis_Barang = Jenis_Barang;
    this.Harga_Barang = Harga_Barang;
}

public String getID_Barang() {
    return ID_Barang;
}

public void setID_Barang(String ID_Barang) {
    this.ID_Barang = ID_Barang;
}

public String getNama_Barang() {
    return Nama_Barang;
}

public void setNama_Barang(String Nama_Barang) {
    this.Nama_Barang = Nama_Barang;
}

public String getJenis_Barang() {
    return Jenis_Barang;
}

public void setJenis_Barang(String Jenis_Barang) {
    this.Jenis_Barang = Jenis_Barang;
}

public int getHarga_Barang() {
    return Harga_Barang;
}

public void setHarga_Barang(int Harga_Barang) {
    this.Harga_Barang = Harga_Barang;
}

@Override
public String toString() {
    return "Barang{" + "ID_Barang=" + ID_Barang
        + ", Nama_Barang=" + Nama_Barang
        + ", Jenis_Barang=" + Jenis_Barang
        + ", Harga_Barang=" + Harga_Barang + '}';
}

```

```
}
```

- Penjelsan

Pada class barang pada class ini membuat metod constructor kosong, constructor yang digunakan primary key dan contructor yang digunakan pada variable yang sudah di buat. Dan juga pada class ini membuat method getter setter dan to string.

## 2) Class Customer

```
/*
-----
    NAMA      : M. INDRA GUNAWAN
    NRP       : 152020068
    KELAS     : BB Project Tugas Besar
-----
*/
package com;
public class Customer {
    private int ID_Customer;
    private String Nama_Customer;
    private String Alamat;
    private String No_Tlp;
    private String Email;
    private String Username;
    private String Password;

    public Customer() {
    }
    public Customer(int ID_Customer) {
        this.ID_Customer = ID_Customer;
    }
    public Customer(String Nama_Customer, String Alamat,
String No_Tlp, String Email, String Username, String
Password) {
        this.Nama_Customer = Nama_Customer;
        this.Alatat = Alamat;
        this.No_Tlp = No_Tlp;
        this.Email = Email;
        this.Username = Username;
        this.Password = Password;
    }
    public Customer(int ID_Customer, String Nama_Customer,
String Alamat, String No_Tlp, String Email, String Username,
String Password) {
        this.ID_Customer = ID_Customer;
        this.Nama_Customer = Nama_Customer;
        this.Alatat = Alamat;
        this.No_Tlp = No_Tlp;
        this.Email = Email;
        this.Username = Username;
        this.Password = Password;
    }
    public int getID_Customer() {
```

```

        return ID_Custumer;
    }
    public void setID_Custumer(int ID_Custumer) {
        this.ID_Custumer = ID_Custumer;
    }
    public String getNama_Custumer() {
        return Nama_Custumer;
    }
    public void setNama_Custumer(String Nama_Custumer) {
        this.Nama_Custumer = Nama_Custumer;
    }
    public String getAlamat() {
        return Alamat;
    }
    public void setAlamat(String Alamat) {
        this.Alamat = Alamat;
    }
    public String getNo_Tlp() {
        return No_Tlp;
    }
    public void setNo_Tlp(String No_Tlp) {
        this.No_Tlp = No_Tlp;
    }
    public String getEmail() {
        return Email;
    }
    public void setEmail(String Email) {
        this.Email = Email;
    }
    public String getUsername() {
        return Username;
    }
    public void setUsername(String Username) {
        this.Username = Username;
    }
    public String getPassword() {
        return Password;
    }
    public void setPassword(String Password) {
        this.Password = Password;
    }
    @Override
    public String toString() {
        return "Customer{" + "ID_Custumer=" + ID_Custumer
            + ", Nama_Custumer=" + Nama_Custumer
            + ", Alamat=" + Alamat
            + ", No_Tlp=" + No_Tlp
            + ", Email=" + Email
            + ", Username=" + Username
            + ", Password=" + Password + '}';
    }
}

```

```
}
```

- Penjelasan

Pada class customer pada class ini membuat method constructor kosong, constructor yang digunakan primary key dan constructor yang digunakan pada variable yang sudah di buat. Dan juga pada class ini membuat method getter digunakan untuk mengembalikan nilai pada variable yang di buat dan setter digunakan untuk pemanggilan variable menggunakan this.nama variable dan to string.

### 3) Class Transaksi

```
/*
-----
    NAMA      : M. INDRA GUNAWAN
    NRP       : 152020068
    KELAS     : BB Project Tugas Besar
-----
*/
package com;
public class Transaksi {
    private int ID_Transaksi;
    private int Total_Barang;
    private int Total_Harga;
    private String Tanggal;
    private String ID_Barang;
    private int ID_Customer;

    public Transaksi() {
    }

    public Transaksi(int ID_Transaksi) {
        this.ID_Transaksi = ID_Transaksi;
    }

    public Transaksi(int Total_Barang, int Total_Harga, String
    Tanggal, String ID_Barang, int ID_Customer) {
        this.Total_Barang = Total_Barang;
        this.Total_Harga = Total_Harga;
        this.Tanggal = Tanggal;
        this.ID_Barang = ID_Barang;
        this.ID_Customer = ID_Customer;
    }

    public int getID_Transaksi() {
        return ID_Transaksi;
    }

    public void setID_Transaksi(int ID_Transaksi) {
        this.ID_Transaksi = ID_Transaksi;
    }
}
```

```

    }

    public int getTotal_Barang() {
        return Total_Barang;
    }

    public void setTotal_Barang(int Total_Barang) {
        this.Total_Barang = Total_Barang;
    }

    public int getTotal_Harga() {
        return Total_Harga;
    }

    public void setTotal_Harga(int Total_Harga) {
        this.Total_Harga = Total_Harga;
    }

    public String getTanggal() {
        return Tanggal;
    }

    public void setTanggal(String Tanggal) {
        this.Tanggal = Tanggal;
    }

    public String getID_Barang() {
        return ID_Barang;
    }

    public void setID_Barang(String ID_Barang) {
        this.ID_Barang = ID_Barang;
    }

    public int getID_Customer() {
        return ID_Customer;
    }

    public void setID_Customer(int ID_Customer) {
        this.ID_Customer = ID_Customer;
    }

    @Override
    public String toString() {
        return "Transaksi{" + "ID_Transaksi=" + ID_Transaksi
            + ", Total_Barang=" + Total_Barang
            + ", Total_Harga=" + Total_Harga
            + ", Tanggal=" + Tanggal
            + ", ID_Barang=" + ID_Barang
            + ", ID_Customer=" + ID_Customer + '}';
    }

```



```
}
```

- Penjelasan

Pada class transaksi pada class ini membuat method constructor kosong, constructor yang digunakan primary key dan constructor yang digunakan pada variable yang sudah di buat. Dan juga pada class ini membuat method getter digunakan untuk mengembalikan nilai pada variable yang di buat dan setter digunakan untuk pemanggilan variable menggunakan this.nama variable dan to string.

#### 4) Class Login

```
/*
-----
    NAMA      : M. INDRA GUNAWAN
    NRP       : 152020068
    KELAS     : BB Project Tugas Besar
-----
*/
package com;
public class Login {
    private static int ID_Custumer;
    private static String Nama_Custumer;
    private static String Alamat;
    private static String No_Tlp;
    private static String Email;
    private static String User;
    private static String Pass;

    public Login() {
    }

    public Login(int ID_Custumer){
        this.ID_Custumer = ID_Custumer;
    }

    public Login(String User, String Pass) {
        this.User = User;
        this.Pass = Pass;
    }

    public static int getID_Custumer() {
        return ID_Custumer;
    }

    public static void setID_Custumer(int ID_Custumer) {
        Login.ID_Custumer = ID_Custumer;
    }

    public static String getNama_Custumer() {
        return Nama_Custumer;
    }
}
```

```

    public static void setNama_Custumer(String Nama_Custumer)
    {
        Login.Nama_Custumer = Nama_Custumer;
    }

    public static String getAlamat() {
        return Alamat;
    }

    public static void setAlamat(String Alamat) {
        Login.Alamat = Alamat;
    }

    public static String getNo_Tlp() {
        return No_Tlp;
    }

    public static void setNo_Tlp(String No_Tlp) {
        Login.No_Tlp = No_Tlp;
    }

    public static String getEmail() {
        return Email;
    }

    public static void setEmail(String Email) {
        Login.Email = Email;
    }

    public static String getUser() {
        return User;
    }

    public static void setUser(String User) {
        Login.User = User;
    }

    public static String getPass() {
        return Pass;
    }

    public static void setPass(String Pass) {
        Login.Pass = Pass;
    }

    @Override
    public String toString() {
        return "Login{" + "ID_Custumer=" + ID_Custumer
            + ", Nama_Custumer=" + Nama_Custumer
            + ", Alamat=" + Alamat

```

```

        + ", No_Tlp=" + No_Tlp
        + ", Email=" + Email
        + ", User=" + User
        + ", Pass=" + Pass + '}';
    }
}

```

#### - Penjelasan

Pada class login pada class ini membuat method constructor kosong, constructor yang digunakan primary key dan constructor yang dibutuhkan saja Dan juga pada class ini membuat method getter digunakan untuk mengembalikan nilai pada variable yang di buat dan setter digunakan untuk pemanggilan variable menggunakan this.nama variable dan to string.

#### 5) Class Admin

```

/*
-----
NAMA      : M. INDRA GUNAWAN
NRP       : 152020068
KELAS    : BB Project Tugas Besar
-----
*/
package com;
public class Admin {
    private String ID_Admin;
    private String Nama_Admin;
    private String Username;
    private String Password;
    public Admin() {
    }

    public Admin(String ID_Admin) {
        this.ID_Admin = ID_Admin;
    }

    public Admin(String ID_Admin, String Nama_Admin, String
Username, String Password) {
        this.ID_Admin = ID_Admin;
        this.Nama_Admin = Nama_Admin;
        this.Username = Username;
        this.Password = Password;
    }

    public Admin(String Username, String Password) {
        this.Username = Username;
        this.Password = Password;
    }

    public String getID_Admin() {
        return ID_Admin;
    }
}

```

```

    }

    public void setID_Admin(String ID_Admin) {
        this.ID_Admin = ID_Admin;
    }

    public String getNama_Admin() {
        return Nama_Admin;
    }

    public void setNama_Admin(String Nama_Admin) {
        this.Nama_Admin = Nama_Admin;
    }

    public String getUsername() {
        return Username;
    }

    public void setUsername(String Username) {
        this.Username = Username;
    }

    public String getPassword() {
        return Password;
    }

    public void setPassword(String Password) {
        this.Password = Password;
    }

    @Override
    public String toString() {
        return "Admin{" + "ID_Admin=" + ID_Admin
            + ", Nama_Admin=" + Nama_Admin
            + ", Username=" + Username
            + ", Password=" + Password + '}';
    }
}

```

#### - Penjelasan

Pada class admin pada class ini membuat method constructor kosong, constructor yang digunakan primary key dan constructor yang digunakan pada variable yang sudah di buat. Dan juga pada class ini membuat method getter digunakan untuk mengembalikan nilai pada variable yang di buat dan setter digunakan untuk pemanggilan variable menggunakan this.nama variable dan to string.

#### 6) Class ConvertBarangToObject

```
/*
```

```

-----
NAMA      : M. INDRA GUNAWAN
NRP       : 152020068
KELAS     : BB Project Tugas Besar
-----

```

```
*/
```

```

package com;
import Execute.ExcuteBarang;
import java.util.ArrayList;
import java.util.List;

public class ConvertBarangToObject {
    public String[][] getBarang(){
        List<Barang> myBrg = new ArrayList<Barang>();
        ExcuteBarang eBrg = new ExcuteBarang();
        myBrg = eBrg.getAllBarang();
        String[][] databarang = new String[myBrg.size()][4];
        int i =0;
        for (Barang brg : myBrg){
            databarang[i][0] = brg.getID_Barang();
            databarang[i][1] = brg.getNama_Barang();
            databarang[i][2] = brg.getJenis_Barang();
            databarang[i][3] = ""+brg.getHarga_Barang();
            i++;
        }
        return databarang;
    }
}

```

#### - Penjelasan

Pada class convertBarangToObject digunakan untuk membuat list dari class barang dan pada method getbarang disini menggunakan method getter di gunakan.

#### 7) Class ConvertCustomerToObject

```

/*
-----
    NAMA      : M. INDRA GUNAWAN
    NRP       : 152020068
    KELAS    : BB Project Tugas Besar
-----
*/
package com;
import Execute.ExecutCustomer;
import java.util.ArrayList;
import java.util.List;

public class ConvertCustomerToObject {
    public String[][] getCustomer(){
        List<Customer> myCstm = new ArrayList<Customer>();
        ExecutCustomer eCstm = new ExecutCustomer();
        myCstm = eCstm.getAllCustomer();
        String[][] dataCustomer = new
String[myCstm.size()][7];
        int i =0;
        for (Customer Cstm : myCstm){
            dataCustomer[i][0] = ""+Cstm.getID_Customer();
            dataCustomer[i][1] = Cstm.getNama_Customer();
            dataCustomer[i][2] = Cstm.getAlamat();

```

```

        dataCustomer[i][3] = Cstm.getNo_Tlp();
        dataCustomer[i][4] = Cstm.getEmail();
        dataCustomer[i][5] = Cstm.getUsername();
        dataCustomer[i][6] = Cstm.getPassword();
        i++;
    }
    return dataCustomer;
}
}

```

- **Penjelasan**

Pada class `convertCustomerToObject` digunakan untuk membuat list dari class `Customer` dan pada method `getCustomer` disini menggunakan method `getter` di gunakan.

**8) Class `ConverTransaksiToObject`**

```

/*
-----
NAMA      : M. INDRA GUNAWAN
NRP       : 152020068
KELAS    : BB Project Tugas Besar
-----
*/
package com;
import Execute.ExecutTransaksi;
import java.util.ArrayList;
import java.util.List;
import java.util.Date;

public class ConvertTransaksiToObject {
    public String[][] getTransaksi(){
        List<Transaksi> myTrans = new ArrayList<Transaksi>();
        ExecutTransaksi eTrans = new ExecutTransaksi();
        myTrans = eTrans.getAllTransaksi();
        String[][] dataTransaksi = new
String[myTrans.size()][6];
        int i =0;
        for (Transaksi trns : myTrans){
            dataTransaksi[i][0] = ""+trns.getID_Transaksi();
            dataTransaksi[i][1] = ""+trns.getTotal_Barang();
            dataTransaksi[i][2] = ""+trns.getTotal_Harga();
            dataTransaksi[i][3] = trns.getTanggal();
            dataTransaksi[i][4] = trns.getID_Barang();
            dataTransaksi[i][5] = ""+trns.getID_Customer();
            i++;
        }
        return dataTransaksi;
    }
}
}

```

- **Penjelasan**

Pada class convertTransaksiToObject digunakan untuk membuat list dari class Transaksi dan pada method getTransaksi disini menggunakan method getter di gunakan

#### 9) Class ConvertAdminToObject

```
/*
-----
NAMA      : M. INDRA GUNAWAN
NRP       : 152020068
KELAS     : BB Project Tugas Besar
-----
*/
package com;
import Execute.ExecAdmin;
import java.util.ArrayList;
import java.util.List;

public class ConvertAdminToObject {
    public String[][] getAdmin(){
        List<Admin> myAdm = new ArrayList<Admin>();
        ExecAdmin eAdm = new ExecAdmin();
        myAdm = eAdm.getAllAdmin();
        String[][] dataAdmin= new String[myAdm.size()][4];
        int i =0;
        for (Admin adm : myAdm){
            dataAdmin[i][0] = adm.getID_Admin();
            dataAdmin[i][1] = adm.getNama_Admin();
            dataAdmin[i][2] = adm.getUsername();
            dataAdmin[i][3] = adm.getPassword();
            i++;
        }
        return dataAdmin;
    }
}
```

#### - Penjelasan

Pada class convertAdminToObject digunakan untuk membuat list dari class Admin dan pada method getAdmin disini menggunakan method getter di gunakan

### b) Package Execute

#### 1) Class ExcuteBarang

```
/*
-----
NAMA      : M. INDRA GUNAWAN
NRP       : 152020068
KELAS     : BB Project Tugas Besar
-----
*/
```

```

package Execute;

import com.Barang;
import database.ConnectionManager;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ExcuteBarang {

    public List<Barang> getAllBarang() {
        String query = "select * from tbl_barang";
        ConnectionManager conM = new ConnectionManager();
        List<Barang> lstBarang = new ArrayList<Barang>();
        Connection conect = conM.logOn();
        try{
            Statement stm = conect.createStatement();
            ResultSet rset = stm.executeQuery(query);
            while(rset.next()){
                Barang brg = new Barang();

                brg.setID_Barang(rset.getString("ID_Barang"));

                brg.setNama_Barang(rset.getString("Nama_Barang"));

                brg.setJenis_Barang(rset.getString("Jenis_Barang"));

                brg.setHarga_Barang(rset.getInt("Harga_Barang"));

                lstBarang.add(brg);
            }
        }catch(SQLException ex){

```



```

Logger.getLogger(ExcuteBarang.class.getName()).log(Level.SEVERE,
null, ex);

    }

    conM.logOff();

return lstBarang;

}

//Untuk Menambahkan data atau menginsert data

public int insertBrg(Barang brg){

    int hasil = 0;

    String query = "Insert into
tbl_barang(ID_Barang,Nama_Barang,Jenis_Barang,Harga_Barang)"
        + "value('"+ brg.getID_Barang()+"', '"+
brg.getNama_Barang()+"', '"+brg.getJenis_Barang()+"', '"+brg.getH
arga_Barang()+"' )";

    ConnectionManager conMan = new
ConnectionManager();

    Connection conn = conMan.logOn();

    try {

        Statement stm = conn.createStatement();

        hasil = stm.executeUpdate(query);

    } catch (SQLException ex) {

Logger.getLogger(ExcuteBarang.class.getName()).log(Level.SEVERE,
null, ex);

    }

    conMan.logOff();

    return hasil;

}

//Digunakan untuk menghapus data

public int deleteBrg(String delID_Barang){

    int hasil = 0;

    String query = "delete from tbl_barang where
ID_Barang='"+ delID_Barang+"'";

```

```

        ConnectionManager conMan = new
ConnectionManager();

        Connection conn = conMan.logOn();

        try {

            Statement stm = conn.createStatement();

            hasil = stm.executeUpdate(query);

        } catch (SQLException ex) {

Logger.getLogger(ExcuteBarang.class.getName()).log(Level.SEVERE,
null, ex);

        }

        conMan.logOff();

        return hasil;

    }

    //mengubah data /mengupdate data yang terbaru
    public int updateBrg(Barang newBrg){

        int hasil = 0;

        String query = "update tbl_barang set
Nama_Barang='"+newBrg.getNama_Barang()

+'',Jenis_Barang='"+newBrg.getJenis_Barang()

+'',Harga_Barang = "+
newBrg.getHarga_Barang()

+" where ID_Barang =
 '"+newBrg.getID_Barang()+ "'";

        ConnectionManager conMan = new
ConnectionManager();

        Connection conn = conMan.logOn();

        try {

            Statement stm = conn.createStatement();

            hasil = stm.executeUpdate(query);

        } catch (SQLException ex) {

Logger.getLogger(ExcuteBarang.class.getName()).log(Level.SEVERE,
null, ex);

        }
    }

```

```

        conMan.logOff();

        return hasil;

    }

}

```

#### - Penjelasan

Dibagian executebarang/controller barang ini di bagian ini membuat method getallbarang yang dimana di dalamnya untuk memanggil tabel barang dengan stirng query dan pemanggilan setter yang sudah di buat di bagian class barang, dan selanjutnya membuat method insert yang akan di gunakan untuk menambahkan data barang yang ada pada gui admin dan pada method insert ini pemanggilan tabel database dengan query insert into tabel barang dan pemanggilan method getter. Selanjutnya method delete di method ini digunakan untuk menghapus data yang ada di database dengan fungsi crud di bagian gui admin dengan parameternya dellidbarang dan juga query nya delete from tabel barang where id barang yang dimana di hapus dari idbarang yang merupakan primarykey.

Selanjutnya method update barang dengan parameter class barang dan variable newbrg di bagian method update ini untuk mengupdate atau mengubah data barang yang ada pada gui admin dan juga pada method ini pemanggilan method getter yang ada pada kelas barang dan juga query nya dengan querynya update tabel barang.

## 2) class ExecuteCustumer

```

/*
-----

    NAMA      : M. INDRA GUNAWAN

    NRP       : 152020068

    KELAS     : BB Project Tugas Besar

-----

*/

package Execute;

import com.Customer;

import database.ConnectionManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

```

```

import java.util.logging.Level;
import java.util.logging.Logger;

public class ExecutCustomer {
    public List<Customer> getAllCustomer(){
        String query = "select * from tbl_custumer";
        ConnectionManager conM = new ConnectionManager();
        List<Customer> lstCustomer = new
ArrayList<Customer>();
        Connection conect = conM.logOn();
        try{
            Statement stm = conect.createStatement();
            ResultSet rset = stm.executeQuery(query);
            while(rset.next()){
                Customer Cus = new Customer();

Cus.setID_Customer(rset.getInt("ID_Customer"));

Cus.setNama_Customer(rset.getString("Nama_Customer"));
                Cus.setAlamat(rset.getString("Alamat"));
                Cus.setNo_Tlp(rset.getString("No_Tlp"));
                Cus.setEmail(rset.getString("Email"));
                Cus.setUsername(rset.getString("Username"));
                Cus.setPassword(rset.getString("Password"));

                lstCustomer.add(Cus);
            }
        }catch(SQLException ex){

Logger.getLogger(ExecutCustomer.class.getName()).log(Level.SEVERE, null, ex);

        }

        conM.logOff();
        return lstCustomer;
    }
}

```

```

    }

    //Untuk Menambahkan data atau menginsert data
    public int insertCustmr(Customer Cstm){
        int hasil = 0;

        String query = "Insert into
tbl_custumer>Nama_Custumer,Alamat, No_Tlp, Email
,Username>Password)values"+ "("'+ Cstm.getNama_Custumer()+
        "','"+
Cstm.getAlamat()+"','"+Cstm.getNo_Tlp()+"','"+Cstm.getEmail()+
        "','"+Cstm.getUsername()+"','"+Cstm.getPassword()+"'");

        ConnectionManager conMan = new ConnectionManager();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            hasil = stm.executeUpdate(query);
        } catch (SQLException ex) {

Logger.getLogger(ExecutCustomer.class.getName()).log(Level.SEVERE, null, ex);

        }

        conMan.logOff();
        return hasil;
    }

    //Digunakan untuk menghapus data
    public int deleteCustomer(String delID_cus){
        int hasil = 0;

        String query = "delete from tbl_custumer where
ID_Custumer="+ delID_cus+"";

        ConnectionManager conMan = new ConnectionManager();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            hasil = stm.executeUpdate(query);

```

```

        } catch (SQLException ex) {

Logger.getLogger(ExecutCustomer.class.getName()).log(Level.SEVERE, null, ex);

        }

        conMan.logOff();

        return hasil;

    }

    //Digunakan untuk mengedit Data/mengupdate data yang terbaru

    public int updateCstmr(int id_cus, String Nama_Cus,String
alamat, String noTlp, String email, String Username, String
Password ){

        int hasil = 0;

        String query = "Update tbl_custumer SET
Nama_Customer='"+Nama_Cus+"',Alamat='" +alamat+

                "','No_Tlp='"+noTlp+"', Email='"+email+"',
Username='"+Username+

                "',' Password='"+Password+"'' WHERE
ID_Customer='"+id_cus+"'";

        ConnectionManager conMan = new ConnectionManager();

        Connection conn = conMan.logOn();

        try {

            Statement stm = conn.createStatement();

            hasil = stm.executeUpdate(query);

        } catch (SQLException ex) {

Logger.getLogger(ExecutCustomer.class.getName()).log(Level.SEVERE, null, ex);

        }

        conMan.logOff();

        return hasil;

    }

}

```

- Penjelasan

Dibagian executeCustomer/controller customer ini di bagian ini membuat method getallcustomer yang dimana di dalamnya untuk memanggil tabel customer dengan stirng query dan pemanggilan setter yang sudah di buat di bagian class customer, dan selanjutnya membuat method insert yang akan di gunakan untuk menambahkan data customer yang ada pada gui register dan pada method insert ini pemanggilan tabel database dengan query insert into tabel customer dan pemanggilan method getter. Selanjutnya method delete di method ini digunakan untuk menghapus data yang ada di database dengan fungsi crud di bagian gui admin dengan parameternya dellidcus dan juga query nya delete form tabel Customer where id Customer yang dimana di hapus dari id Customer yang merupakan primarykey.

Selanjutnya method update Customer dengan parameter class Customer dan variable yang di butuhkan di bagian method update ini untuk mengupdate atau mengubah data Customer yang ada pada gui Customer dan juga pada method ini pemanggilan method getter yang ada pada kelas Customer dan juga query nya dengan querynya update tabel Customer.

### 3). Class ExecuteTransaksi

```
/*  
-----  
    NAMA      : M. INDRA GUNAWAN  
    NRP       : 152020068  
    KELAS     : BB Project Tugas Besar  
-----  
*/  
  
package Execute;  
import com.Transaksi;  
import database.ConnectionManager;  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.logging.Level;
```

```

import java.util.logging.Logger;

public class ExecutTransaksi {

    public List<Transaksi> getAllTransaksi() {

        String query = "select * from tbl_transaksi";

        ConnectionManager conM = new ConnectionManager();

        List<Transaksi> lstTransaksi = new
        ArrayList<Transaksi>();

        Connection conect = conM.logOn();

        try{

            Statement stm = conect.createStatement();

            ResultSet rset = stm.executeQuery(query);

            while(rset.next()){

                Transaksi Tran = new Transaksi();

                Tran.setID_Transaksi(rset.getInt("ID_Transaksi"));

                Tran.setTotal_Barang(rset.getInt("Total_Barang"));

                Tran.setTotal_Harga(rset.getInt("Total_Harga"));

                Tran.setTanggal(rset.getString("Tanggal"));

                Tran.setID_Barang(rset.getString("ID_Barang"));

                Tran.setID_Custumer(rset.getInt("ID_Custumer"));

                lstTransaksi.add(Tran);

            }

        } catch(SQLException ex){

            Logger.getLogger(ExecutTransaksi.class.getName()).log(Level.SEVERE, null, ex);

        }

        conM.logOff();

        return lstTransaksi;

    }
}

```



```

//Untuk Menambahkan data atau menginsert data

public int insertTrans( Transaksi Tran){

    int hasil = 0;

    String query = "Insert into
tbl_transaksi(Total_Barang,Total_Harga,Tanggal,ID_Barang,
ID_Customer)values"+ "("+ Tran.getTotal_Barang()+

", "+Tran.getTotal_Harga()+", '"+Tran.getTanggal()+"', '"+Tr
an.getID_Barang()+"', "+Tran.getID_Customer()+")";

    ConnectionManager conMan = new
ConnectionManager();

    Connection conn = conMan.logOn();

    try {

        Statement stm = conn.createStatement();

        hasil = stm.executeUpdate(query);

    } catch (SQLException ex) {

        Logger.getLogger(ExecutTransaksi.class.getName()).log(Lev
el.SEVERE, null, ex);

    }

    conMan.logOff();

    return hasil;

}
}

```

- Penjelasan

Dibagian executeTransaksi/controller transaksi ini di bagian ini membuat method getalltransaksi yang dimana di dalamnya untuk memanggil tabel transaksi dengan stirng query dan pemanggilan setter yang sudah di buat di bagian class transaksi, dan selanjutnya membuat method insert yang akan di gunakan untuk menambahkan data transaksi yang ada pada gui customer di menu transaksi dan pada method insert ini pemanggilan tabel database dengan query insert into tabel transaksi dan pemanggilan method getter.

#### 4) Class ExecuteLogin

```

/*

```

```

-----

```

```

NAMA      : M. INDRA GUNAWAN

```

```

NRP       : 152020068

```

KELAS : BB Project Tugas Besar

```
-----  
*/  
package Execute;  
import com.Admin;  
import com.Login;  
import database.ConnectionManager;  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
public class ExcuteLogin {  
    //Controller untuk login dari sisi User  
    public int getAllLogin(Login Lgn){  
        ConnectionManager conM = new ConnectionManager();  
        int hasil = 0;  
        Connection conect = conM.logOn();  
        try{  
            String query = "SELECT * FROM tbl_customer  
WHERE Username='"+Lgn.getUser()+"' AND  
Password='"+Lgn.getPass()+"'";  
            Statement stm = conect.createStatement();  
            ResultSet rset = stm.executeQuery(query);  
            while(rset.next()){  
  
Lgn.setID_Customer(rset.getInt("ID_Customer"));  
  
Lgn.setNama_Customer(rset.getString("Nama_Customer"));  
  
Lgn.setAlamat(rset.getString("Alamat"));  
            }  
        }  
    }  
}
```

```

Lgn.setEmail(rset.getString("Email"));

Lgn.setNo_Tlp(rset.getString("No_Tlp"));

Lgn.setUser(rset.getString("Username"));

Lgn.setPass(rset.getString("Password"));

        hasil = 1;
    }
} catch (SQLException ex) {

Logger.getLogger(ExcuteLogin.class.getName()).log(Level.SEVERE
, null, ex);

    }

    conM.logOff();

return hasil;
}

//Controller untuk login dari sisi Admin
public int getAdmin(Admin adm){

    ConnectionManager conM = new ConnectionManager();

    int hasil = 0;

    Connection conect = conM.logOn();

    try{

        String query = "SELECT * FROM tbl_admin WHERE
Username='"+adm.getUsername()+"' AND
Password='"+adm.getPassword()+"'";

        Statement stm = conect.createStatement();

        ResultSet rset = stm.executeQuery(query);

        while(rset.next()){

adm.setID_Admin(rset.getString("ID_Admin"));

adm.setNama_Admin(rset.getString("Nama_Admin"));

adm.setUsername(rset.getString("Username"));

```

```

adm.setPassword(rset.getString("Password"));

        hasil = 1;

    }

    }catch(SQLException ex){

Logger.getLogger(ExcuteLogin.class.getName()).log(Level.SEVERE
, null, ex);

    }

    conM.logOff();

    return hasil;

    }

}

```

#### - Penjelasan

Dibagian executelogin/controller login ini di bagian ini membuat method getalllogin yang dimana di dalamnya untuk memanggil tabel login dengan string query dan pemanggilan setter yang sudah di buat di bagian class login data yang ada pada method getall login ini untuk memanggil username dan password dari customer yang di gunakan di Gui Login. Dan method getadmin dengan parameter class admin dan variable adm dan di method ini menggunakan query select from tabel admin where username dan password yang dimana hanya memanggil username dan password di class admin.

#### 5) Class ExcuteAdmin

```

/*
-----

NAMA      : M. INDRA GUNAWAN

NRP       : 152020068

KELAS     : BB Project Tugas Besar

-----

*/

package Execute;

import com.Admin;

import database.ConnectionManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

```

```

import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ExecAdmin {
    public List<Admin> getAllAdmin() {
        String query = "select * from tbl_admin";
        ConnectionManager conM = new ConnectionManager();
        List<Admin> lstAdmin = new ArrayList<Admin>();
        Connection conect = conM.logOn();
        try{
            Statement stm = conect.createStatement();
            ResultSet rset = stm.executeQuery(query);
            while(rset.next()){
                Admin adm = new Admin();
                adm.setID_Admin(rset.getString("ID_Admin"));

                adm.setNama_Admin(rset.getString("Nama_Admin"));
                adm.setUsername(rset.getString("Username"));
                adm.setPassword(rset.getString("password"));
                lstAdmin.add(adm);
            }
        }catch(SQLException ex){

            Logger.getLogger(ExecAdmin.class.getName()).log(Level.SEVERE,
            null, ex);
        }

        conM.logOff();

        return lstAdmin;
    }

    //Untuk Menambahkan data atau menginsert data

```

```

        public int insertAdm(Admin adm) {

            int hasil = 0;

            String query = "Insert into
tbl_admin(ID_Admin,Nama_Admin,Username>Password)values"+ "("+"+
adm.getID_Admin()+"', '"
                                +
adm.getNama_Admin()+"', '"+adm.getUsername()+"', '"+adm.getPassw
ord()+"'")";

            ConnectionManager conMan = new ConnectionManager();

            Connection conn = conMan.logOn();

            try {

                Statement stm = conn.createStatement();

                hasil = stm.executeUpdate(query);

            } catch (SQLException ex) {

                Logger.getLogger(ExecAdmin.class.getName()).log(Level.SEVERE,
                null, ex);

            }

            conMan.logOff();

            return hasil;

        }

        //Digunakan untuk menghapus data

        public int deleteAdmin(String delID_Admin){

            int hasil = 0;

            String query = "delete from tbl_admin where
ID_Admin='"+ delID_Admin+"'";

            ConnectionManager conMan = new ConnectionManager();

            Connection conn = conMan.logOn();

            try {

                Statement stm = conn.createStatement();

                hasil = stm.executeUpdate(query);

            } catch (SQLException ex) {

                Logger.getLogger(ExecAdmin.class.getName()).log(Level.SEVERE,
                null, ex);

            }

        }

```

```

        conMan.logOff();

        return hasil;
    }

    //Digunakan untuk mengedit Data/mengupdate data yang
    terbaru

    public int updateAdm(Admin newAdm) {

        int hasil = 0;

        String query = "update tbl_admin set
Nama_Admin='"+newAdm.getNama_Admin()

                        +"',Username='"+newAdm.getUsername()

                        +"',Password='"+ newAdm.getPassword()

                        +"'where ID_Admin ='"+newAdm.getID_Admin()+
        ""';

        ConnectionManager conMan = new ConnectionManager();
        Connection conn = conMan.logOn();

        try {

            Statement stm = conn.createStatement();

            hasil = stm.executeUpdate(query);

        } catch (SQLException ex) {

            Logger.getLogger(ExecAdmin.class.getName()).log(Level.SEVERE,
            null, ex);

        }

        conMan.logOff();

        return hasil;
    }
}

```

#### - Penjelasan

Dibagian executeadmin/controller admin ini di bagian ini membuat method getalladmin yang dimana di dalamnya untuk memanggil tabel admin dengan stirng query dan pemanggilan setter yang sudah di buat di bagian class admin, dan selanjutnya membuat method insert yang akan di gunakan untuk menambahkan data admin yang ada pada gui admin dan pada method insert ini pemanggilan tabel database dengan query insert into tabel admin dan pemanggilan method getter. Selanjutnya method delete di method ini digunakan untuk menghapus data yang ada di database dengan fungsi crud di bagian gui admin dengan parameternya dellidadmin dan juga query nya

delete form tabel admin where id admin yang dimana di hapus dari idadmin yang merupakan primarykey.

Selanjutnya method update admin dengan parameter class admin dan variable newAdm di bagian method update ini untuk mengupdate atau mengubah data admin yang ada pada gui admin dan juga pada method ini pemanggilan method getter yang ada pada kelas admin dan juga query nya dengan querynya update tabel admin.

## c) Package View

### 1) GUI Admin

```
1  /*
2
3      NAMA      : M. INDRA GUNAWAN
4      NRP       : 152020068
5      KELAS    : BB Project Tugas Besar
6
7  */
8  package View;
9  import com.Admin;
10 import Execute.ExecAdmin;
11 import com.ConvertAdminToObject;
12 import com.Barang;
13 import com.ConvertBarangToObject;
14 import Execute.ExcuteBarang;
15 import Execute.ExecutCustomer;
16 import com.ConvertCustomerToObject;
17 import com.ConvertTransaksiToObject;
18 import java.awt.CardLayout;
19 import javax.swing.JOptionPane;
20 public class GuiAdmin extends javax.swing.JFrame {
21
22     public GuiAdmin() {
23         initComponents();
24         setDataBarang();
25         setDataAdmin();
26         setDataCustomer();
27         setDataTransaksi();
28
29         this.setLocationRelativeTo(null);
30     }
31     private void setDataBarang() {
```

Activate Windows  
Go to Settings to activate Windows.



```

32 ConvertBarangToObject cbto = new ConvertBarangToObject();
33 String [][] dataBarang = cbto.getBarang();
34 tbl_barang.setModel(new javax.swing.table.DefaultTableModel(
35     dataBarang,
36     new String[]{
37         "ID_Barang", "Nama_Barang", "Jenis_Barang", "Harga_Barang"
38     }
39 ));
40 jScrolBarang.setViewportViewView(tbl_barang);
41 }
42
43
44 private void setDataAdmin(){
45     ConvertAdminToObject cato = new ConvertAdminToObject();
46     String [][] dataAdmin = cato.getAdmin();
47     tbl_admin.setModel(new javax.swing.table.DefaultTableModel(
48         dataAdmin,
49         new String[]{
50             "ID_Admin", "Nama_Admin", "Username", "Password"
51         }
52     ));
53     jScrolAdmin.setViewportViewView(tbl_admin);
54 }
55
56
57 private void setDataCustomer(){
58     ConvertCustomerToObject ccto = new ConvertCustomerToObject();
59     String [][] dataCstmr = ccto.getCustomer();
60     tblCus.setModel(new javax.swing.table.DefaultTableModel(
61         dataCstmr,

```

Activate Windows  
Go to Settings to activate Windows.

```

62     dataCstmr,
63     new String[]{
64         "ID_Customer", "Nama_Customer", "Alamat", "No_Tlp", "Email", "Username", "Password"
65     }
66 ));
67 jScrollPaneCustomer.setViewportViewView(tblCus);
68 }
69
70 public void UpdateTbl(){
71     ConvertCustomerToObject ccto = new ConvertCustomerToObject();
72     String [][] dataCstmr = ccto.getCustomer();
73     tblCus.setModel(new javax.swing.table.DefaultTableModel(
74         dataCstmr,
75         new String[]{
76             "ID_Customer", "Nama_Customer", "Alamat", "No_Tlp", "Email", "Username", "Password"
77         }
78     ));
79     jScrollPaneCustomer.setViewportViewView(tblCus);
80     JOptionPane.showMessageDialog(null, " Data Sukses Di Update");
81 }
82
83
84 private void setDataTransaksi(){
85     ConvertTransaksiToObject ctto = new ConvertTransaksiToObject();
86     String [][] dataTran = ctto.getTransaksi();
87     tbl_transaksi.setModel(new javax.swing.table.DefaultTableModel(
88         dataTran,
89         new String[]{
90             "ID_Transaksi", "Total_Barang", "Total_Harga", "Tanggal", "ID_Barang", "ID_Customer"
91         }
92     ));

```

```

93     });
94 }
95
96 /**
97  * This method is called from within the constructor to initialize the form.
98  * WARNING: Do NOT modify this code. The content of this method is always
99  * regenerated by the Form Editor.
100  */
101 @SuppressWarnings("unchecked")
102 Generated Code
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

656     txt_Namabarang.setText>Nama_Barang);
657     txt_JnsBarang.setText(Jenis_Barang);
658     txt_HrgBarang.setText(Harga_Barang);
659 }
660
661 private void btn_insertActionPerformed(java.awt.event.ActionEvent evt) {
662     String id_barang = txt_IDBarang.getText();
663     String nama_barang = txt_Namabarang.getText();
664     String jns_barang = txt_JnsBarang.getText();
665     String strHrg_barang = txt_HrgBarang.getText();
666     int Harga_Barang = Integer.parseInt(strHrg_barang);
667
668     Barang brg = new Barang(id_barang, nama_barang, jns_barang, Harga_Barang);
669     Execute.ExcuteBarang ebrg = new Execute.ExcuteBarang();
670     int hasil = ebrg.insertBrng(brg);
671     if (hasil > 0) {
672         JOptionPane.showMessageDialog(null, "Sukses Di Simpan");
673         setDataBarang();
674     }
675     else {
676         JOptionPane.showMessageDialog(null, "Failed Di Simpan");
677     }
678 }
679
680 private void btn_deleteActionPerformed(java.awt.event.ActionEvent evt) {
681     String id_barang = txt_IDBarang.getText();
682     Execute.ExcuteBarang ebrg = new Execute.ExcuteBarang();
683
684     int hasil = ebrg.deleteBrng(id_barang);
685     if (hasil > 0) {

```

```

686         JOptionPane.showMessageDialog(null, "Data Sukses Di Hapus");
687         setDataBarang();
688     }
689     else{
690         JOptionPane.showMessageDialog(null, "Data Failed Di Hapus");
691     }
692
693     // TODO add your handling code here:
694 }
695
696 private void btn_updateActionPerformed(java.awt.event.ActionEvent evt) {
697     String id_barang = txt_IDBarang.getText();
698     String nama_barang = txt_Namabarang.getText();
699     String jns_barang = txt_JnsBarang.getText();
700     String strHrg_barang = txt_HrgBarang.getText();
701     int Harga_Barang = Integer.parseInt(strHrg_barang);
702
703     Barang brg = new Barang(id_barang, nama_barang, jns_barang, Harga_Barang);
704     Execute.ExcuteBarang ebrg = new Execute.ExcuteBarang();
705     int hasil = ebrg.updateBrg(brg);
706     if (hasil > 0){
707         JOptionPane.showMessageDialog(null, "Sukses Di Update");
708         setDataBarang();
709     }
710     else{
711         JOptionPane.showMessageDialog(null, "Failed Di Update");
712     }
713 }
714
715 private void tbl_adminMouseClicked(java.awt.event.MouseEvent evt) {

```

```

716     int row = tbl_admin.getSelectedRow();
717     String ID_admin = tbl_admin.getValueAt(row, 0).toString();
718     String Nama_admin = tbl_admin.getValueAt(row, 1).toString();
719     String Username = tbl_admin.getValueAt(row, 2).toString();
720     String Password = tbl_admin.getValueAt(row, 3).toString();
721
722     txt_idadmin.setText(ID_admin);
723     txt_namaAdmin.setText(Nama_admin);
724     txt_username.setText(Username);
725     Pass.setText(Password);
726 }
727
728 private void btn_createAActionPerformed(java.awt.event.ActionEvent evt) {
729     String id_admin= txt_idadmin.getText();
730     String nama_admin = txt_namaAdmin.getText();
731     String username = txt_username.getText();
732     String password = Pass.getText();
733
734     Admin adm = new Admin(id_admin, nama_admin, username, password);
735     Execute.ExecAdmin eadm = new Execute.ExecAdmin();
736     int hasil = eadm.insertAdm(adm);
737     if (hasil > 0){
738         JOptionPane.showMessageDialog(null, "Sukses Di Simpan");
739         setDataAdmin();
740     }
741     else{
742         JOptionPane.showMessageDialog(null, "Failed Di Simpan");
743     }
744 }
745

```

```

746 private void btn_updateAActionPerformed(java.awt.event.ActionEvent evt) {
747     String id_admin= txt_idadmin.getText();
748     String nama_admin = txt_namaAdmin.getText();
749     String username = txt_username.getText();
750     String password = Pass.getText();
751
752     Admin adm = new Admin(id_admin, nama_admin, username, password);
753     Execute.ExecAdmin eadm = new Execute.ExecAdmin();
754     int hasil = eadm.updateAdm(adm);
755     if (hasil > 0){
756         JOptionPane.showMessageDialog(null, "Sukses Di Update");
757         setDataAdmin();
758     }
759     else{
760         JOptionPane.showMessageDialog(null, "Failed Di Update");
761     }
762     // TODO add your handling code here:
763 }
764
765 private void btn_deleteAActionPerformed(java.awt.event.ActionEvent evt) {
766     String id_admin = txt_idadmin.getText();
767     Execute.ExecAdmin eAdm = new Execute.ExecAdmin();
768
769     int hasil = eAdm.deleteAdmin(id_admin);
770     if (hasil > 0){
771         JOptionPane.showMessageDialog(null, "Data Sukses Di Hapus");
772         setDataAdmin();
773     }
774     else{
775         JOptionPane.showMessageDialog(null, "Data Failed Di Hapus");

```

```

776     }
777
778
779     private void tblCusMouseClicked(java.awt.event.MouseEvent evt) {
780         int row = tblCus.getSelectedRow();
781         String Nama_Costumer = tblCus.getValueAt(row, 1).toString();
782         String Alamat = tblCus.getValueAt(row, 2).toString();
783         String NoTlp = tblCus.getValueAt(row, 3).toString();
784         String Email = tblCus.getValueAt(row, 4).toString();
785         String Username = tblCus.getValueAt(row, 5).toString();
786         String Password = tblCus.getValueAt(row, 6).toString();
787
788         txt_namcus.setText(Nama_Costumer);
789         txt_alamat.setText(Alamat);
790         txt_notelp.setText(NoTlp);
791         txt_email.setText(Email);
792         txt_usercus.setText(Username);
793         PassCus.setText(Password);
794     }
795
796     private void btnCustomerActionPerformed(java.awt.event.ActionEvent evt) {
797         cl = (CardLayout) PanelUtama.getLayout();
798         cl.show(PanelUtama, "cardCostumer");
799     }
800
801     private void btn_transaksiActionPerformed(java.awt.event.ActionEvent evt) {
802         cl = (CardLayout) PanelUtama.getLayout();
803         cl.show(PanelUtama, "cardTransaksi");
804     }
805

```

```

821     private void btn_DeleteCusActionPerformed(java.awt.event.ActionEvent evt) {
822         String id_cus = txtID_cus.getText();
823         Execute.ExecutCustomer eCstm = new Execute.ExecutCustomer();
824
825         int hasil = eCstm.deleteCustomer(id_cus);
826         if (hasil > 0) {
827             JOptionPane.showMessageDialog(null, "Data Sukses Di Hapus");
828             setDataCustomer();
829         }
830         else {
831             JOptionPane.showMessageDialog(null, "Data Failed Di Hapus");
832         }
833     }

```

```

820     private void btn_exitActionPerformed(java.awt.event.ActionEvent evt) {
821         GuiLogin gtbk = new GuiLogin();
822         gtbk.setVisible(true);
823         this.dispose();
824     }
825     //tbl TransaksiMouseClicked TIDAK DIGUNAKAN
826     private void tbl_transaksiMouseClicked(java.awt.event.MouseEvent evt) {...3 lines }
827
828     //txt namacusActionPerformed TIDAK DIGUNAKAN
829     private void txt_namcusActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
830
831
832     private void cbPassAdmActionPerformed(java.awt.event.ActionEvent evt) {
833         if (cbPassAdm.isSelected()) {
834             Pass.setEchoChar((char)0);
835         }
836         else {
837             Pass.setEchoChar('*');
838         }
839     }

```

```

840     }
841 }
842
843 /**
844  * @param args the command line arguments
845  */
846 public static void main(String args[]) {
847     /* Set the Nimbus look and feel */
848     Look and feel setting code (optional)
849     //</editor-fold>
850
851     /* Create and display the form */
852     java.awt.EventQueue.invokeLater(new Runnable() {
853         public void run() {
854             new GuiAdmin().setVisible(true);
855         }
856     });
857 }
858
859 private CardLayout cl;
860 // Variables declaration - do not modify
861 private javax.swing.JButton Btn_databarang;
862 private javax.swing.JLabel Lb_hrg;
863 private javax.swing.JLabel Lb_id;
864 private javax.swing.JLabel Lb_idadmin;
865 private javax.swing.JLabel Lb_jns;
866 private javax.swing.JLabel Lb_nama;
867 private javax.swing.JLabel Lb_namaAdmin;
868 private javax.swing.JLabel Lb_pass;
869 private javax.swing.JLabel Lb_username;

```

```

890 private javax.swing.JPanel PanelCustomer;
891 private javax.swing.JPanel PanelUtama;
892 private javax.swing.JPanel Panelbrg;
893 private javax.swing.JPasswordField Pass;
894 private javax.swing.JPasswordField PassCus;
895 private javax.swing.JPanel PnlAdmin;
896 private javax.swing.JPanel PnlInput;
897 private javax.swing.JPanel PnlInputCus;
898 private javax.swing.JPanel PnlTblCus;
899 private javax.swing.JPanel PnlTransaksi;
900 private javax.swing.JPanel PnltabelAdmin;
901 private javax.swing.JPanel PnltabelTransaksi;
902 private javax.swing.JButton btnCustomer;
903 private javax.swing.JButton btn_DeleteCus;
904 private javax.swing.JButton btn_admin;
905 private javax.swing.JButton btn_createA;
906 private javax.swing.JButton btn_delete;
907 private javax.swing.JButton btn_deleteA;
908 private javax.swing.JButton btn_exit;
909 private javax.swing.JButton btn_insert;
910 private javax.swing.JButton btn_transaksi;
911 private javax.swing.JButton btn_update;
912 private javax.swing.JButton btn_updateA;
913 private javax.swing.JCheckBox cbPassAdm;
914 private javax.swing.JLabel jLabel1;
915 private javax.swing.JLabel jLabel3;
916 private javax.swing.JLabel jLabel4;
917 private javax.swing.JLabel jLabel5;
918 private javax.swing.JLabel jLabel6;
919 private javax.swing.JLabel jLabel7;

```

```

918 private javax.swing.JLabel jLabel6;
919 private javax.swing.JLabel jLabel7;
920 private javax.swing.JLabel jLabel8;
921 private javax.swing.JScrollPane jScrollPaneAdmin;
922 private javax.swing.JScrollPane jScrollPaneBarang;
923 private javax.swing.JScrollPane jScrollPaneCustomer;
924 private javax.swing.JScrollPane jScrollPaneTransaksi;
925 private javax.swing.JToolBar jToolBar1;
926 private javax.swing.JPanel pnlInputdata;
927 private javax.swing.JPanel pnltabelBrq;
928 private javax.swing.JTable tblCus;
929 private javax.swing.JTable tbl_admin;
930 private javax.swing.JTable tbl_barang;
931 private javax.swing.JTable tbl_transaksi;
932 private javax.swing.JTextField txt_HrgBarang;
933 private javax.swing.JTextField txt_IDBarang;
934 private javax.swing.JTextField txt_JnsBarang;
935 private javax.swing.JTextField txt_Namabarang;
936 private javax.swing.JTextField txt_alamat;
937 private javax.swing.JTextField txt_email;
938 private javax.swing.JTextField txt_idadmin;
939 private javax.swing.JTextField txt_namaAdmin;
940 private javax.swing.JTextField txt_namcus;
941 private javax.swing.JTextField txt_notelp;
942 private javax.swing.JTextField txt_usercus;
943 private javax.swing.JTextField txt_username;
944 // End of variables declaration
945
946 }
947

```

- Penjelasan

Pada gui admin ini method setdatabarang, setdataadmin, setdatacustumer, setdatatransaksi yang digunakan untuk memanggil tabel yang ada pada database dan ingin di tambahkan pada tabel gui. Dan method updatetbl digunakan untuk mengupdate data yang ada pada tabel custumer. Btn data barang, btn admin digunakan untuk pindah layout. Tbl barang mouseclicked di gunakan ketika tabel di klik maka data yang ada pada tabel tersebut akan tampil di textfield yang sudah di buat. Btn insert digunakan untuk menambah data barang yang ingin kita tambah. Btn delete di gunakan untuk menghapus data barang dengan pemanggilan controller/executebarang yang sudah di buat. Btn update di gunakan untuk mengubah data yang ingin di rubah/edit

## 2) GUI Customer

```
1  /*
2  -----
3      NAMA      : M. INDRA GUNAWAN
4      NRP       : 152020068
5      KELAS     : BB Project Tugas Besar
6  -----
7  */
8  package View;
9  import com.Login;
10 import com.ConvertBarangToObject;
11 import com.Transaksi;
12 import Execute.ExecutTransaksi;
13 import Execute.ExecutCustomer;
14 import java.awt.CardLayout;
15 import javax.swing.JOptionPane;
16 public class GuiCustumer extends javax.swing.JFrame {
17
18     public GuiCustumer() {
19         initComponents();
20         setDataBarang();
21
22         this.setLocationRelativeTo(null);
23     }
24     private void setDataBarang(){
25         ConvertBarangToObject cbto = new ConvertBarangToObject();
26         String [] [] dataBarang = cbto.getBarang();
27         tbl_BarangC.setModel(new javax.swing.table.DefaultTableModel(
28             dataBarang,
29             new String[]{
```

```

31         "ID_Barang", "Nama_Barang", "Jenis_Barang", "Harga_Barang"
32     }
33     });
34     jScrollPaneBarang.setViewportViewView(tbl_BarangC);
35 }
36
37 /**
38  * This method is called from within the constructor to initialize the form.
39  * WARNING: Do NOT modify this code. The content of this method is always
40  * regenerated by the Form Editor.
41  */
42 @SuppressWarnings("unchecked")
43 Generated Code
560
561 private void btn_dftrbarangActionPerformed(java.awt.event.ActionEvent evt) {
562     cl = (CardLayout) PnlUtama.getLayout();
563     cl.show(PnlUtama, "cardDftrbarang");
564 }
565
566 private void btn_trnsaksiActionPerformed(java.awt.event.ActionEvent evt) {
567     cl = (CardLayout) PnlUtama.getLayout();
568     cl.show(PnlUtama, "cardTransaksi");
569 }
570
571 private void btn_logOutActionPerformed(java.awt.event.ActionEvent evt) {
572     GuiLogin gtbk = new GuiLogin();
573     gtbk.setVisible(true);
574     this.dispose();
575 }
576

```

```

578 private void tbl_BarangCMouseClicked(java.awt.event.MouseEvent evt) {
579
580     int row = tbl_BarangC.getSelectedRow();
581     String ID_Barang = tbl_BarangC.getValueAt(row, 0).toString();
582     String Nama_Barang = tbl_BarangC.getValueAt(row, 1).toString();
583     String Jenis_Barang = tbl_BarangC.getValueAt(row, 2).toString();
584     String Harga_Barang = tbl_BarangC.getValueAt(row, 3).toString();
585
586     txtIdbarang.setText(ID_Barang);
587     txtnamabarang.setText(Nama_Barang);
588     txtJnsBarang.setText(Jenis_Barang);
589     txtHrgBarang.setText(Harga_Barang);
590 }
591
592 private void btn_beliActionPerformed(java.awt.event.ActionEvent evt) {
593     int total = Integer.parseInt(txtHrgBarang.getText()) * Integer.parseInt(txtTtlBarang.getText());
594     txtIDbrg.setText(txtIdbarang.getText());
595     txtNamaBrg.setText(txtnamabarang.getText());
596     txtJnsBrg.setText(txtJnsBarang.getText());
597     txtHrgBrg.setText(txtHrgBarang.getText());
598     txtTtlBrg.setText(txtTtlBarang.getText());
599     txtTtlHrg.setText(String.valueOf(total));
600     JOptionPane.showMessageDialog(null, "Barang Berhasil Di Simpan Di Transaksi");
601 }
602
603 private void btn_ConfirmActionPerformed(java.awt.event.ActionEvent evt) {
604     int totalbrg = Integer.parseInt(txtTtlBrg.getText());
605     int totalhrp = Integer.parseInt(txtTtlHrg.getText());
606     String tanggal = dateChooserTgl.getText();
607     String idbrg = txtIDbrg.getText();

```

```

614         int idCtm = Integer.parseInt(txtIdCus.getText());
615
616         Transaksi tran = new Transaksi(totalbrg, totalhrg, tanggal, idbrg, idCtm);
617         Execute.ExecutTransaksi eTran = new Execute.ExecutTransaksi();
618         int hasil = eTran.insertTrans(tran);
619         if (hasil > 0) {
620             JOptionPane.showMessageDialog(null, "Transaksi Sukses");
621         }
622         else {
623
624             JOptionPane.showMessageDialog(null, "Transaksi Failed");
625         }
626     }
627
628     private void btn_ProfilActionPerformed(java.awt.event.ActionEvent evt) {
629         cl = (CardLayout) PnlUtama.getLayout();
630         cl.show(PnlUtama, "carProfil");
631     }
632
633     private void btn_edtProfilActionPerformed(java.awt.event.ActionEvent evt) {
634         String strid_user = txtIDuser.getText();
635         int IdUser = Integer.parseInt(strid_user);
636         String nama_user = txtNamaUser.getText();
637         String email = txtEmail.getText();
638         String alamat = txtAlamat.getText();

```

```

632     private void btn_edtProfilActionPerformed(java.awt.event.ActionEvent evt) {
633         String strid_user = txtIDuser.getText();
634         int IdUser = Integer.parseInt(strid_user);
635         String nama_user = txtNamaUser.getText();
636         String email = txtEmail.getText();
637         String alamat = txtAlamat.getText();
638         String notlp = txtNotlp.getText();
639         String User = txtUser.getText();
640         String Pass = txtpassP.getText();
641
642         Customer Cstm = new Customer(IdUser, nama_user, alamat, notlp, email, User, Pass);
643         Execute.ExecutCustomer eCstm = new Execute.ExecutCustomer();
644         int hasil = eCstm.updateCstmr(IdUser, nama_user, alamat, notlp, email, User, Pass);
645         if (hasil > 0) {
646             JOptionPane.showMessageDialog(null, "Perubahan Berhasil, Silahkan Keluar dan Masuk Kembali");
647         }
648         else {
649             JOptionPane.showMessageDialog(null, "Perubahan Gagal");
650         }
651     }
652

```

```

649     private void txtpassPActionPerformed(java.awt.event.ActionEvent evt) {
650         // TODO add your handling code here:
651     }
652
653     private void formWindowActivated(java.awt.event.WindowEvent evt) {
654         txtIDuser.setText(String.valueOf(Login.getID_Customer()));
655         txtNamaUser.setText(Login.getNama_Customer());
656         txtEmail.setText(Login.getEmail());
657         txtAlamat.setText(Login.getAlamat());
658         txtNotlp.setText(Login.getNo_Tlp());
659         txtUser.setText(Login.getUser());
660         txtpassP.setText(Login.getPass());
661         txtIdCus.setText(String.valueOf(Login.getID_Customer()));
662     }
663

```

```

664     public static void main(String args[]) {
665         /* Set the Nimbus look and feel */
666         Look and feel setting code (optional)
667         //</editor-fold>
668
669         /* Create and display the form */
670         java.awt.EventQueue.invokeLater(new Runnable() {
671             public void run() {
672                 new GuiCustomer().setVisible(true);
673             }
674         });
675     }
676
677     private CardLayout cl;
678     // Variables declaration - do not modify

```



```

699     private javax.swing.JPanel PnlBeli;
700     private javax.swing.JPanel PnlPembelian;
701     private javax.swing.JPanel PnlProfil;
702     private javax.swing.JPanel PnlTabel;
703     private javax.swing.JPanel PnlUtama;
704     private javax.swing.JPanel Pnlbarang;
705     private javax.swing.JPanel Pnltransaksi;
706     private javax.swing.JButton btn_Confirm;
707     private javax.swing.JButton btn_Profil;
708     private javax.swing.JButton btn_beli;
709     private javax.swing.JButton btn_dfttrbarang;
710     private javax.swing.JButton btn_edtProfil;
711     private javax.swing.JButton btn_logOut;
712     private javax.swing.JButton btn_trnsaksi;
713     private datechooser.beans.DateChooserCombo dateChooserTgl;
714     private javax.swing.JLabel jLabel1;
715     private javax.swing.JLabel jLabel10;
716     private javax.swing.JLabel jLabel11;
717     private javax.swing.JLabel jLabel12;
718     private javax.swing.JLabel jLabel13;
719     private javax.swing.JLabel jLabel14;
720     private javax.swing.JLabel jLabel15;
721     private javax.swing.JLabel jLabel16;
722     private javax.swing.JLabel jLabel17;
723     private javax.swing.JLabel jLabel18;
724     private javax.swing.JLabel jLabel19;
725     private javax.swing.JLabel jLabel2;
726     private javax.swing.JLabel jLabel20;
727     private javax.swing.JLabel jLabel21;
728     private javax.swing.JLabel jLabel3;

```

```

729     private javax.swing.JLabel jLabel4;
730     private javax.swing.JLabel jLabel5;
731     private javax.swing.JLabel jLabel6;
732     private javax.swing.JLabel jLabel7;
733     private javax.swing.JLabel jLabel8;
734     private javax.swing.JLabel jLabel9;
735     private javax.swing.JPanel jPanel2;
736     private javax.swing.JScrollPane jScrollPaneBarang;
737     private javax.swing.JToolBar jToolBar1;
738     private javax.swing.JTable tbl_BarangC;
739     private javax.swing.JTextField txtAlamat;
740     private javax.swing.JTextField txtEmail;
741     private javax.swing.JTextField txtHrgBarang;
742     private javax.swing.JTextField txtHrgBrg;
743     private javax.swing.JTextField txtIDbrg;
744     private javax.swing.JTextField txtIDuser;
745     private javax.swing.JTextField txtIdCus;
746     private javax.swing.JTextField txtIdbarang;
747     private javax.swing.JTextField txtJnsBarang;
748     private javax.swing.JTextField txtJnsBrg;
749     private javax.swing.JTextField txtNamaBrg;
750     private javax.swing.JTextField txtNamaUser;
751     private javax.swing.JTextField txtNotlp;
752     private javax.swing.JTextField txtTtlBarang;
753     private javax.swing.JTextField txtTtlBrg;
754     private javax.swing.JTextField txtTtlHrg;
755     private javax.swing.JTextField txtUser;
756     private javax.swing.JTextField txtnamabarang;
757     private javax.swing.JTextField txtpassP;
758     // End of variables declaration

```

## - Penjelasan

Pada gui customer ini ada beberapa method yang digunakan yaitu method gui customer yang digunakan untuk memanggil method setdata barang yang dimana method setdatabarang

digunakan untuk menampilkan barang pada tabel gui nantinya dengan memanggil class convert barang to object yang dimana berisikan method yang memanggil data yang ada pada database. Btn daftarbarang digunakan untuk berpindah layout/panel, btn transaksi digunakan untuk berpindah layout/panel btn logout digunakan untuk berpindah Gui ke bagian gui login, tblbarang mouseclicked digunakan ketika tabel yang ada pada gui di tekan maka data yang ada pada tabel akan muncul ke bagian text fiel yang sudah di buat.

btn beli digunakan untuk mengkalulasikan harga barang dengan total barang yang ingin di beli dan data akan muncul ke bagian menu transaksi, btn confirm digunakan untuk mengkonfirmasi pembayaran dan juga memanggil method insert pada execute transaksi yang sudah di buat yang artinya menambahkan data transaksi pada database dan juga pemanggilan constructor pada class transaksi, btn profil digunakan untuk berpindah layout/panel, btn edit profil digunakan untuk mengedit data customer yang di edit oleh customer sendiri pada method ini ketika di tekan btn edit maka data customer akan terupdate di databasenya dan di method ini ada pemanggilan class execute customer yang digunakan untuk update data dan constructor customer yang di ambil dari class customer. Form window digunakan untuk memanggil data user yang sudah masuk pada registrasi dan akan di tampilkan pada menu profil dan juga id customer yang akan di tampilkan pada menu transaksi.

### 3) GUI Login

```
1  /*
2  |
3  |     NAMA      : M. INDRA GUNAWAN
4  |     NRP       : 152020068
5  |     KELAS     : BB Project Tugas Besar
6  |
7  | */
8  |
9  | package View;
10 |
11 | import com.Admin;
12 | import com.Login;
13 | import Execute.ExcuteLogin;
14 | import javax.swing.JOptionPane;
15 |
16 | public class GuiLogin extends javax.swing.JFrame {
17 |
18 |     public GuiLogin() {
19 |         initComponents();
20 |
21 |         this.setLocationRelativeTo(null);
22 |     }
23 |
24 |     /**
25 |      * This method is called from within the constructor to initialize the form.
26 |      * WARNING: Do NOT modify this code. The content of this method is always
27 |      * regenerated by the Form Editor.
28 |      */
29 |
30 |     @SuppressWarnings("unchecked")
31 |     Generated Code
32 |
33 |     private void btn_LoginActionPerformed(java.awt.event.ActionEvent evt) {
34 |         String username = txtUsername.getText();
```

```

99      char[] password_char = Password.getPassword();
100      String password = String.valueOf(password_char);
101
102      String user = txtUsername.getText();
103      char[] pass_char = Password.getPassword();
104      String pass = String.valueOf(pass_char);
105
106      Login Lgn = new Login(user, pass);
107      Admin adm = new Admin(username, password);
108      int hasil = 0;
109      Execute.ExcuteLogin eLgn= new Execute.ExcuteLogin();
110      hasil = eLgn.getAdmin(adm);
111      if (hasil==0){
112          hasil= eLgn.getAllLogin(Lgn);
113
114          if (hasil==0){
115              JOptionPane.showMessageDialog(null, "User Tidak Di Temukan");
116          }else{
117              GuiCustumer gcstm = new GuiCustumer();
118              gcstm.setVisible(true);
119              this.dispose();
120
121              JOptionPane.showMessageDialog(null, "Selamat Datang Custumer");
122          }
123      }else{
124          GuiAdmin db = new GuiAdmin();
125          db.setVisible(true);
126          this.dispose();
127
128          JOptionPane.showMessageDialog(null, "Selamat Datang Admin");

```

```

140      private void btnRegisActionPerformed(java.awt.event.ActionEvent evt) {
141          Register Rgstr = new Register();
142          Rgstr.setVisible(true);
143          this.dispose();
144      }
145
146      private void cbPassActionPerformed(java.awt.event.ActionEvent evt) {
147          if (cbPass.isSelected()) {
148              Password.setEchoChar((char)0);
149          }
150          else{
151              Password.setEchoChar('*');
152          }
153      }
154
155      private void btn_exitActionPerformed(java.awt.event.ActionEvent evt) {
156          int quest = JOptionPane.showConfirmDialog(this, "Keluar Aplikasi?", "Pertanyaan", JOptionPane.YES_NO_OPTION);
157
158          if (quest == JOptionPane.YES_OPTION) {
159              System.exit(0);
160          }
161      }
162
163      public static void main(String args[]) {
164          /* Set the Nimbus look and feel */
165          Look and feel setting code (optional)
166          //</editor-fold>
167
168          /* Create and display the form */
169          java.awt.EventQueue.invokeLater(new Runnable() {
170              public void run() {

```

Act  
Go t

```

147      public static void main(String args[]) {
148          /* Set the Nimbus look and feel */
149          Look and feel setting code (optional)
150          //</editor-fold>
151
152          /* Create and display the form */
153          java.awt.EventQueue.invokeLater(new Runnable() {
154              public void run() {
155                  new GuiLogin().setVisible(true);
156              }
157          });
158      }

```

```

143         Password.setEchoChar('*');
144     }
145 }
146
147 public static void main(String args[]) {
148     /* Set the Nimbus look and feel */
149     Look and feel setting code (optional)
150     //</editor-fold>
151
152     /* Create and display the form */
153     java.awt.EventQueue.invokeLater(new Runnable() {
154         public void run() {
155             new GuiLogin().setVisible(true);
156         }
157     });
158 }
159
160 // Variables declaration - do not modify
161 private javax.swing.JPasswordField Password;
162 private javax.swing.JButton btnRegis;
163 private javax.swing.JButton btn_Login;
164 private javax.swing.JCheckBox cbPass;
165 private javax.swing.JLabel jLabel1;
166 private javax.swing.JLabel jLabel2;
167 private javax.swing.JPanel jPanel1;
168 private javax.swing.JLabel tobaccoShop;
169 private javax.swing.JTextField txtUsername;
170 // End of variables declaration
171 }
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191 }
192

```

#### - Penjelasan

Pada gui login ini menggunakan banyak tombol untuk menjalankan proses selanjutnya. Bnt login digunakan untuk login dan pada method ini ada 2 proses login yang dimana ada pemanggilan constructor dari class admin dan juga login yang di ambil hanya bagian username dan password saja dan juga pemanggilan method yang ada di controller executelogin yang dimana di dalam execute login ada 2 method yaitu admin login. Dan disini akan mengecek ketika admin tidak ada maka akan muncul user/admin tidak ditemukan dan ketika customer ada akan masuk pada gui customer dan juga ketika admin ada maka akan masuk ke bagian halaman admin. Btn registrasi digunakan untuk berpindah gui ke gui register. Cbpass digunakan untuk mengecek password agar terlihat dengan menggunakan combo box, btn exit digunakan untuk keluar aplikasi.

#### 4) GUI Registrasi

```

1  /*
2  -----
3      NAMA      : M. INDRA GUNAWAN
4      NRP       : 152020068
5      KELAS    : BB Project Tugas Besar
6  -----
7  */
8  package View;
9  import Execute.ExecutCustomer;
10 import com.Customer;
11 import javax.swing.JOptionPane;
12
13 public class Register extends javax.swing.JFrame {
14
15     public Register() {
16         initComponents();
17
18         this.setLocationRelativeTo(null);
19     }
20     /**
21      * This method is called from within the constructor to initialize the form.
22      * WARNING: Do NOT modify this code. The content of this method is always
23      * regenerated by the Form Editor.
24      */
25     @SuppressWarnings("unchecked")
26     Generated Code
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92 private void btndaftarActionPerformed(java.awt.event.ActionEvent evt) {
93     String nama = txtNama.getText();
94     String alamat = txtAlamat.getText();

```

```

95     String notlp = txtNoTlp.getText();
96     String email = txtEmail.getText();
97     String user = txtUser.getText();
98     String pass = Pass.getText();
99
100     Customer cstm = new Customer(nama,alamat,notlp,email,user,pass);
101     Execute.ExecutCustomer eCstm = new Execute.ExecutCustomer();
102     int hasil = eCstm.insertCustmr(cstm);
103     if (hasil > 0){
104         JOptionPane.showMessageDialog(null, "Daftar Berhasil");
105         GuiLogin gTmbk = new GuiLogin();
106         gTmbk.setVisible(true);
107         this.dispose();
108     }
109     else{
110         JOptionPane.showMessageDialog(null, "Daftar Gagal ");
111     }
112 }
113
114 private void btn_BackActionPerformed(java.awt.event.ActionEvent evt) {
115     GuiLogin gTmbk = new GuiLogin();
116     gTmbk.setVisible(true);
117     this.dispose();
118 }
119
120 public static void main(String args[]) {
121     /* Set the Nimbus look and feel */
122     Look and feel setting code (optional)
123
124
125     /* Create and display the form */

```

```

221      /* Set the Nimbus look and feel */
222      Look and feel setting code (optional)
243
244      /* Create and display the form */
245      java.awt.EventQueue.invokeLater(new Runnable() {
246          public void run() {
247              new Register().setVisible(true);
248          }
249      });
250  }
251
252  // Variables declaration - do not modify
253  private javax.swing.JPasswordField Pass;
254  private javax.swing.JButton btn_Back;
255  private javax.swing.JButton btndaftar;
256  private javax.swing.JLabel jLabel2;
257  private javax.swing.JLabel jLabel3;
258  private javax.swing.JLabel jLabel4;
259  private javax.swing.JLabel jLabel5;
260  private javax.swing.JLabel jLabel6;
261  private javax.swing.JLabel jLabel7;
262  private javax.swing.JLabel jLabel8;
263  private javax.swing.JPanel jPanel1;
264  private javax.swing.JTextField txtAlamat;
265  private javax.swing.JTextField txtEmail;
266  private javax.swing.JTextField txtNama;
267  private javax.swing.JTextField txtNoTlp;
268  private javax.swing.JTextField txtUser;
269  // End of variables declaration
270  }

```

- Penjelasan

Pada gui registrasi ini menggunakan beberapa tombol untuk menjalankan proses selanjutnya. Btn daftar pada method ini digunakan untuk menginputkan data customer/user dan pada method ini melakukan pemanggilan constructor pada class customer dan juga pemanggilan class controller/execute customer untuk di ambil method insert yang sudah di buat pada class execute, btn back digunakan berpindah gui ke gui login, dan untuk cbshowpassword digunakan untuk memperlihatkan password yang tertutup menggunakan combo box.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dalam aplikasi penjualan tembakau ini terdapat 2 user yaitu admin dan customer yang dimana admin bisa mengakses semua data yang di perlukan dan mengubah dan menambahkan data yang di inginkan. Dan untuk user hanya bisa mengubah bagian profil dan memilih dan membeli barang dan melakukan transaksi.

aplikasi penjualan tembakau adalah aplikasi yang dibuat dengan bahasa pemrograman java yang dimana aplikasi ini di buat yang bertujuan untuk memudahkan perokok agar bisa menghemat pengeluaran perokok ketika ingin meroko dan aplikasi penjualan ini menjual beberapa tembakau yang bervariasi dan alat linting dan juga papirnya. Dan aplikasi ini di peruntukan untuk orang yang sudah berusia 18+.

#### **5.2 Saran**

Saran untuk pengembangan lebih lanjut pada aplikasi ini di tambahkan bagian gambar barang-barang yang ingin di jual. Dan berhati-hati ketika membuat controller yang sesuai kebutuhan agar tidak terjadi error dan juga bagian class yang menggunakan contractor yang memang di butuhkan agar bisa menunjang pembuatan aplikasinya.