|  |
| --- |
| PROYEK SDP  ALTERNATECH  APLIKASI PENJUALAN PERALATAN GAMING |
| LOGO ISTTS TRANS PUTIH |
|  |
| Oleh:  219116826 – ERIKO FERDIAN WINARKO  219116832 – JOHN CALVIN STEVEN  219116835 – KEVIN OCTAVIUS  219116842 – RUSSEL JOSHUA CHANDRA |
| PROGRAM SARJANA  PROGRAM STUDI TEKNIK ELEKTRO  FAKULTAS SAINS DAN TEKNOLOGI  INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA  SURABAYA  2021 |

**DAFTAR ISI**

**Halaman**

HALAMAN JUDUL……………………………………………………… i

DAFTAR ISI…………………………………………………………….... ii

[BAB I PENDAHULUAN 1](#_Toc92366686)

[1.1 Latar Belakang 1](#_Toc92366687)

[1.2 Tujuan 1](#_Toc92366688)

[1.3 Teori Penunjang 2](#_Toc92366689)

[1. My SQL 2](#_Toc92366690)

[2. E-commerce API //payment gateway 2](#_Toc92366691)

[3. C# 2](#_Toc92366692)

[4. Visual studio 3](#_Toc92366693)

[1.4 Ruang Lingkup 3](#_Toc92366694)

[1.5 Metodologi 6](#_Toc92366695)

[1.6 Sistematika Pembahasan 6](#_Toc92366696)

[BAB II ANALISA SISTEM 7](#_Toc92366697)

[2.1 Fact Finding 7](#_Toc92366698)

[2.1.1 Pembanding 7](#_Toc92366699)

[2.1.2 Dokumentasi 7](#_Toc92366700)

[2.1.3 Prototype 14](#_Toc92366701)

[2.2 Analisis Kebutuhan 17](#_Toc92366702)

[2.3 Hardware Requirement 18](#_Toc92366703)

[2.4 Batasan Aplikasi 18](#_Toc92366704)

[Bab 3 SISTEM DESAIN 20](#_Toc92366705)

[3.1 Desain Interface 20](#_Toc92366706)

[1. Tampilan Awal 20](#_Toc92366707)

[2. Tampilan Register 21](#_Toc92366708)

[3. Tampilan Admin 22](#_Toc92366709)

[4. Tampilan User 27](#_Toc92366710)

[BAB IV IMPLEMENTASI 42](#_Toc92366711)

[4.1 Koneksi Server 42](#_Toc92366712)

[4.2 App Page 43](#_Toc92366713)

[4.3 Register Page 44](#_Toc92366714)

[4.4 Admin Page 45](#_Toc92366715)

[4.5 User Page 52](#_Toc92366716)

[BAB V PENUTUP 67](#_Toc92366717)

[5.1 Kesimpulan 67](#_Toc92366718)

[5.2 Saran 67](#_Toc92366719)

[DAFTAR PUSTAKA 68](#_Toc92366720)

BAB I

PENDAHULUAN

Topik yang kita bahas kali ini merupakan sebuah aplikasi bertema toko, dimana kita menjual peralatan gaming untuk memenuhi kebutuhan para konsumen yang akan membeli peralatan gaming serta penjual yang ingin menjual produk mereka. Aplikasi tersebut dibuat agar dapat memenuhi kebutuhan konsumen yang berada di luar jangkauan area penjual.

1. Latar Belakang

Pada masa pandemi ini, sudah banyak merugikan beberapa pihak, baik di pihak konsumen maupun penjual. Yang terdampak pada konsumen yaitu pembelian produk ke suatu toko yang dimana pada saat ini sudah tidak bisa dilakukan karena adanya PPKM. Sedangkan produsen mengalami penurunan penjualan karena kurangnya pembeli yang datang untuk membeli barang. Salah satu contohnya adalah penjualan produk berupa aksesoris computer atau gaming yang sangat diminati oleh para pemuda sekarang. Apalagi pada saat ini lah mereka menikmati masa muda mereka dengan bermain game bersama sahabat-sahabat mereka yang dimana mereka pasti membutuhkan beberapa perlengkapan yang mumpuni supaya bisa menikmati secara maksimal tetapi terhalangi oleh pandemic pada saat ini.

Oleh karena itu, hadirlah produk digital yang kami buat yang bertujuan untuk membantu kedua belah pihak. Dengan ini dapat memungkinkan pembeli dan penjual untuk melakukan transaksi tanpa harus melakukan tatap muka. Ekonomi pun tetap berjalan sehingga penjual tidak mengalami kerugian yang banyak dan pembeli tetap bisa mendapat benda yang diinginkan tanpa harus keluar dari rumah. Produk kami juga menyediakan voucher diskon yang memiliki periode waktu tertentu sehingga dapat lebih menghemat pengeluaran pembeli.

1. Tujuan

Tujuan dari aplikasi ini kita buat adalah:

* mempermudah transaksi antara penjual dan pembeli
* Mempermudah player mendapatkan/ menjual produk digital suatu game

1. Teori Penunjang

Berikut merupakan teori-teori yang digunakan untuk menunjang pembuatan aplikasi ini:

1. **My SQL**

Oracle adalah sebuah software untuk sistem manajemen database relasional (RDBMS) dan bukan software untuk sistem manajemen database biasa (DBMS). Oracle adalah software yang pertama kali melibatkan SQL (Structure Query Language) di dalam pengelolaan database. Adanya SQL ini membuat pengguna komputer lain dapat mengakses informasi dari dalam database komputer yang menggunakan sistem RDBMS dengan mudah dan cepat.

Fungsi Oracle hampir sama seperti fungsi software database biasa (DBMS), yaitu sebagai pengelola data yang tersimpan dalam database. Tetapi, Oracle adalah RDBMS yang memiliki perbedaan fungsi dengan software-software DBMS. Perbedaan fungsi tersebut, intinya dalam hal pengelolaan database.

### 2. E-commerce API //payment gateway

E-commerce API membantu pengembang aplikasi mentransfer data dari satu aplikasi e-commerce ke aplikasi lain dan menggunakan data dalam aplikasi web atau mobile. API jenis ini menawarkan fitur yang luas untuk mendukung fungsionalitas bisnis e-commerce mulai dari pencarian barang dan toko hingga filter rating dan popularitas.

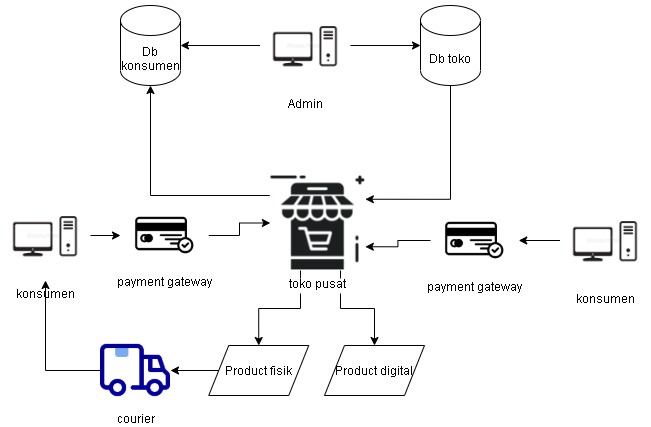
1. **C#**

C# atau yang dibaca C sharp adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman server-side pada website, membangun aplikasi desktop ataupun mobile, pemrograman game dan sebagainya. Selain itu C# juga bahasa [pemrograman yang berorientasi objek](https://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek), jadi C# juga mengusung konsep objek seperti *inheritance, class, polymorphism* dan *encapsulation*.

1. **Visual studio**

Microsoft Visual Studio merupakan sebuah [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup [kompiler](https://id.wikipedia.org/wiki/Kompiler), [SDK](https://id.wikipedia.org/wiki/Software_Development_Kit), [Integrated Development Environment (IDE)](https://id.wikipedia.org/wiki/Integrated_Development_Environment), dan dokumentasi (umumnya berupa [MSDN Library](https://id.wikipedia.org/w/index.php?title=MSDN_Library&action=edit&redlink=1)). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain [Visual C++](https://id.wikipedia.org/wiki/Visual_C%2B%2B), [Visual C#](https://id.wikipedia.org/w/index.php?title=Visual_C&action=edit&redlink=1), [Visual Basic](https://id.wikipedia.org/wiki/Visual_Basic), [Visual Basic .NET](https://id.wikipedia.org/wiki/Visual_Basic_.NET), [Visual InterDev](https://id.wikipedia.org/w/index.php?title=Visual_InterDev&action=edit&redlink=1), [Visual J++](https://id.wikipedia.org/w/index.php?title=Visual_J%2B%2B&action=edit&redlink=1), [Visual J#](https://id.wikipedia.org/w/index.php?title=Visual_J&action=edit&redlink=1), [Visual FoxPro](https://id.wikipedia.org/wiki/Visual_FoxPro), dan [Visual SourceSafe](https://id.wikipedia.org/w/index.php?title=Visual_SourceSafe&action=edit&redlink=1).

1. Ruang Lingkup

Arsitektur Dan gambar

Ketika konsumen ingin melakukan pembelian, konsumen akan diminta untuk melakukan pembayaran. setelah pembayaran, sistem akan lgsg berkomunikasi dengan toko pusat. lalu rincian pembelian akan tercatat ke dalam database. toko pusat pun menyiapkan barang untuk dikirim oleh kurir.

* Master
* Master Barang

Untuk menambahkan Stock serta barang baru.

Proses menambahkan bisa di tambahkan oleh admin serta seller

* Fitur
* Admin

1. Dapat mengecek dan menambah jumlah stok barang / barang baru(Admin)Fitur ini akan tersedia suatu menu barang yang khusus untuk admin dimana admin dapat menambah stok barang yang kurang serta check stock jumlah barang
2. Dapat melihat konfirmasi top up saldo dari customer(admin) dalam Fitur ini admin dapat melihat list konfirmasi topup dari customer beserta button untuk konfirmasi dan batalkan konfirmasi topup
3. Untuk tombol konfirmasi maka proses topup saldo akan diteruskan ke system dan saldo dapat di kirimkan ke customer
4. Untuk tombol batalkan maka proses topup customer tersebut akan dibatalkan

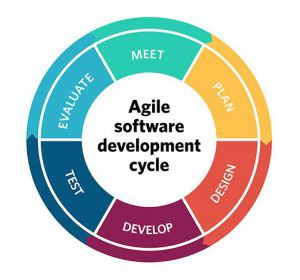
* Customer

1. Dapat melihat daftar harga , merk dan barang(customer) Dalam fitur kali Ini akan tersedia menu barang dimana Fitur ini customer dapat melihat daftar barang yang di sort oleh system berdasarkan merek dan harga yang tersedia
2. Dapat membeli barang (customer) dalam fitur ini akan berhubungan dengan menu barang dimana ketika customer melihat list barang beserta harga dan terdapat tombol untuk memasukan ke shopping cart. Dalam shopping cart user dapat melihat dan akan terdapat button “Beli” yang dimana Ketika ditekan maka customer dapat melakukan transaksi pembelian barang tersebut.
3. Dapat melakukan topup saldo. Fitur ini berguna untuk menambah saldo pengguna
4. Dapat melihat history pembelian barang (customer) Fitur ini Customer dapat melihat history pembelian barang dari bulan dan tahun.
5. Aplikasi kita juga menyediakan tempat untuk menjual atau membeli item in-game bagi pemain. Aplikasi kita layaknya middleman yang menjadi perantara bagi antar pemain untuk melakukan barter/jual beli item in-game
6. Metodologi

* AGILE SOFTWARE DEVELOPMENT

Agile software development adalah Metode dari beberapa kumpulan prinsip untuk pengembangan software di mana persyaratan dan solusi melalui upaya kolaboratif dari antar tim fungsional dan klien

Ini sebagai pendukung perencanaan adaptif, perkembangan evolusi, awal pengiriman, dan perbaikan terus-menerus, dan itu mendorong respon yang cepat dan fleksibel untuk dirubah.



1. Sistematika Pembahasan

* BAB I : PENDAHULUAN
* BAB II : TEORI PENUNJANG
* BAB III : SYSTEM DESAIN
* BAB IV : IMPLEMENTASI
* BAB V : PENUTUP

BAB II

ANALISA SISTEM

Dalam bab ini kami menjelaskan system acuan kami akan sistem apa yg akan dibuat

1. Fact Finding

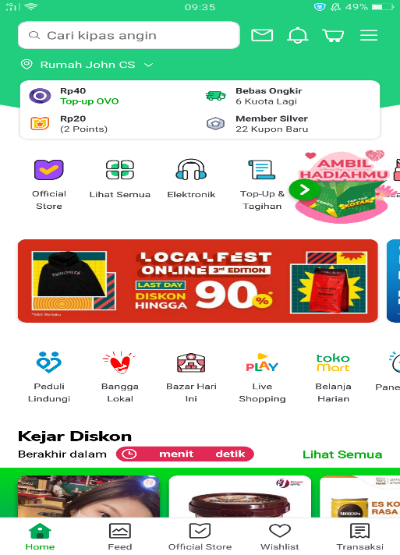
Dalam fact finding, kami menjelaskan seluruh usaha kami dalam menemukan referensi aplikasi yang akan kita kembangkan

1. Pembanding

Sebagai pembanding atau referensi aplikasi yang akan kita kembangkan, kami menggunakan Tokopedia, Itemku.com, Bukalapak dan lapak old shop yang berada di Instagram sebagai referensi dalam pengembangan aplikasi kami yang menjual barang elektronik khusus gaming dan item in game dengan usaha agar interface lebih mudah dipahami.

### 2.1.2 Dokumentasi

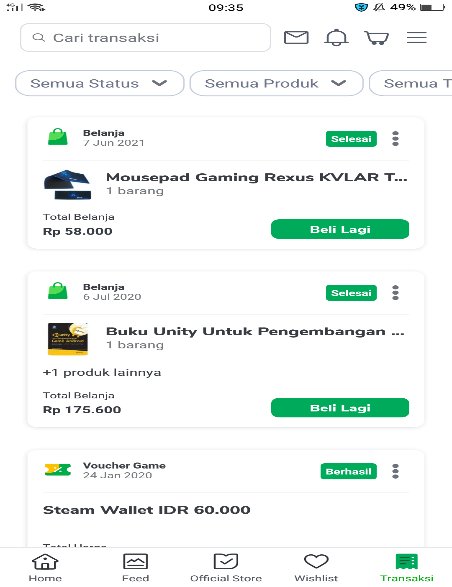
Berikut adalah dokumentasi yang kami temukan dari Tokopedia , Itemku

serta oldshop yang berada di social media lainnya,

Gambar 2.1

Tampilan awal Tokopedia

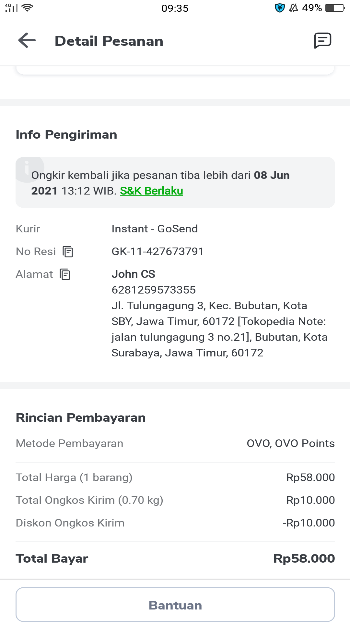
Dari tampilan awal, ditunjukkan saldo user serta kupon atau poin. Dari sini juga kita bisa memilih kategori benda yang ingin kita cari seperti alat gaming dsb.serta terdapat saldo user serta poin yang didapatkan Ketika user membeli barang atau item di E-Commerce tersebut.



Gambar 2.2

Tampilan History User

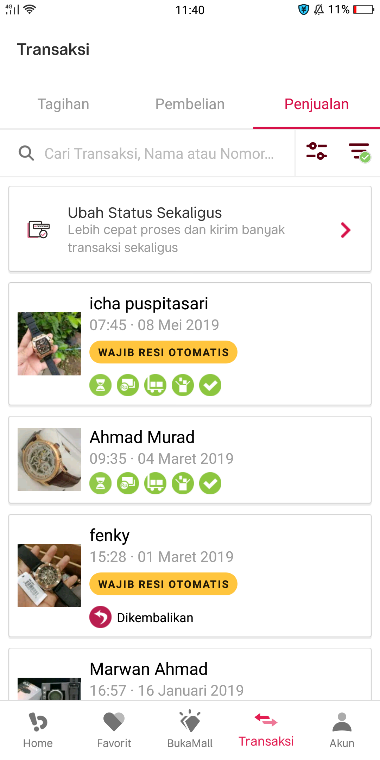
Disini kita bisa melihat riwayat barang yang kita beli serta harga yang tertera pada benda tersebut. Kita bisa melihat rincian pembelian kita setelah menekan salah satu dari produk diatas serta kita bisa membeli barang tersebut lagi dengan menekan tombol beli lagi.



Gambar 2.3

Tampilan detail pesanan

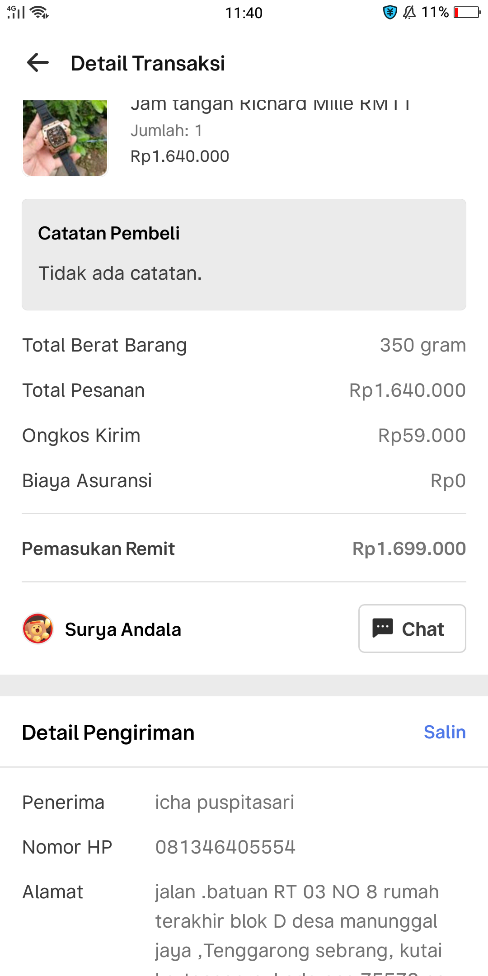
Ini adalah tampilan setelah kita menekan salah satu riwayat pembelian kita, tertera Nomor Resi, total harga dan jumlah total barang atau item, alamat serta nama pembeli.disini juga terdapat tombol bantuan



Gambar 2.4

halaman daftar penjualan barang oldshop

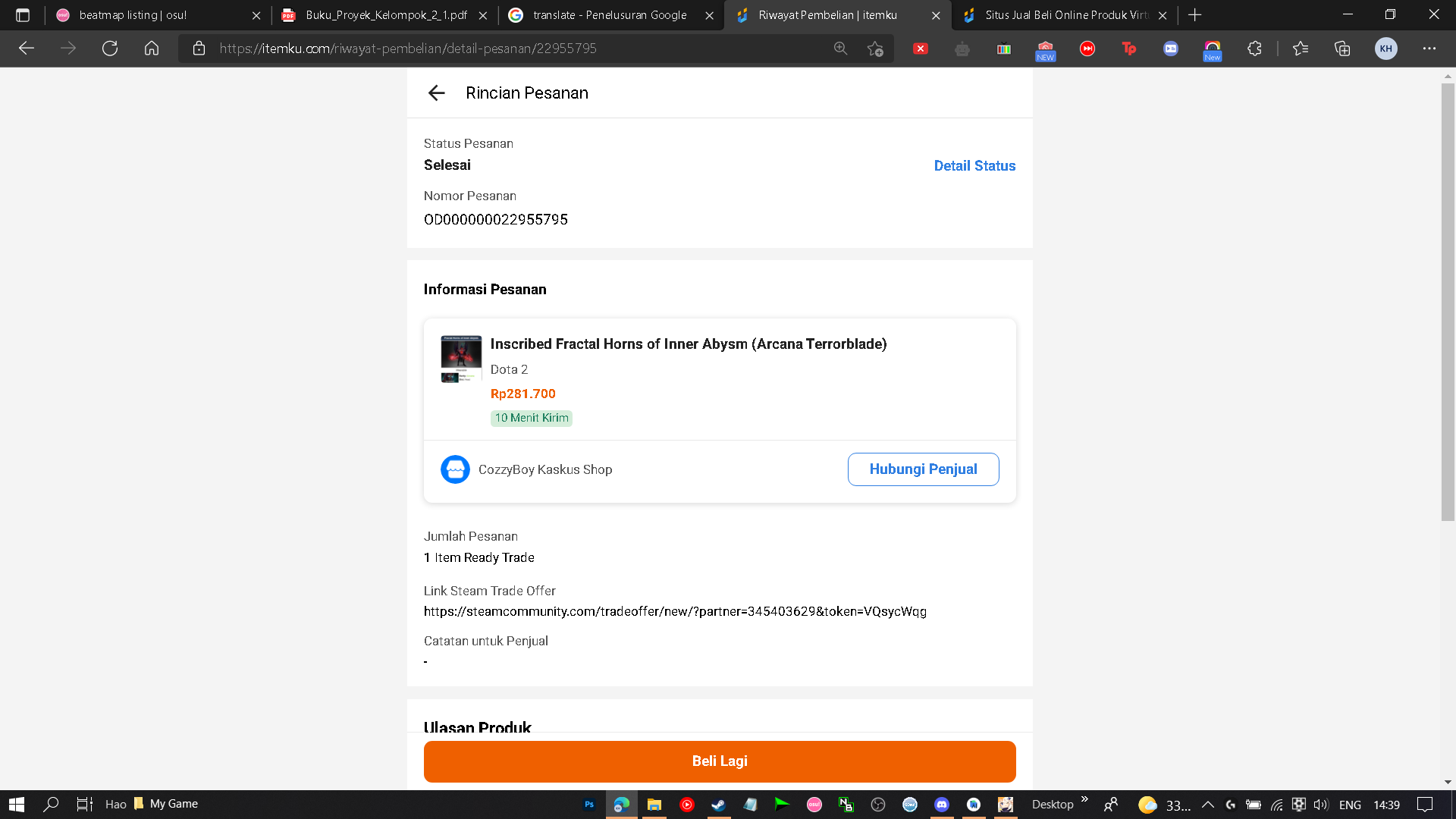
Ini adalah tampilan dari penjual toko untuk melihat daftar penjualanya serta status penjualannya.



Gambar 2.5

halaman detail transaksi yg dijual

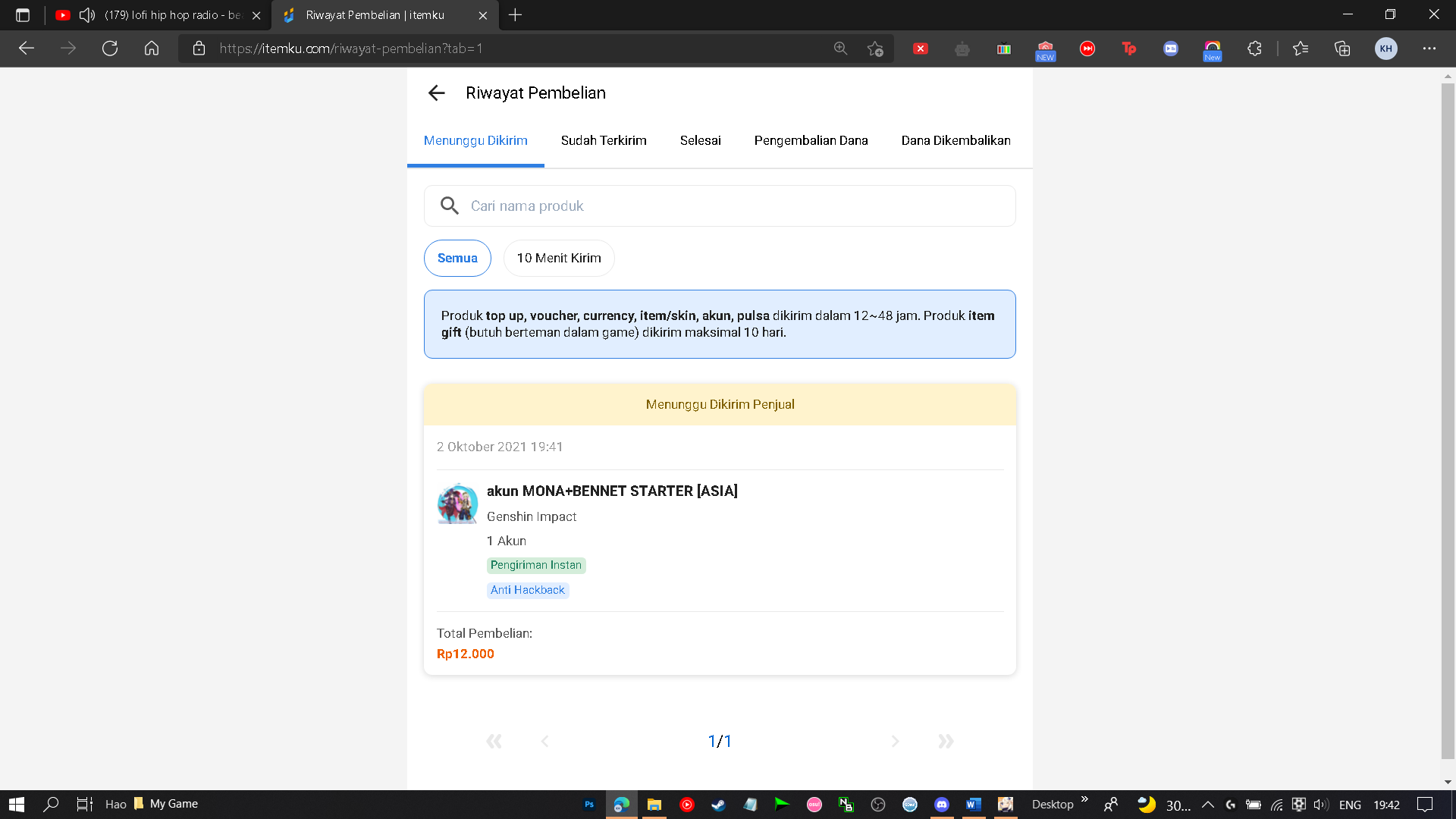
Penjual dapat melihat detail transaksi dari pembeli jika pembeli membeli barang tersebut



Gambar 2.6

halaman Status selesai transaksi

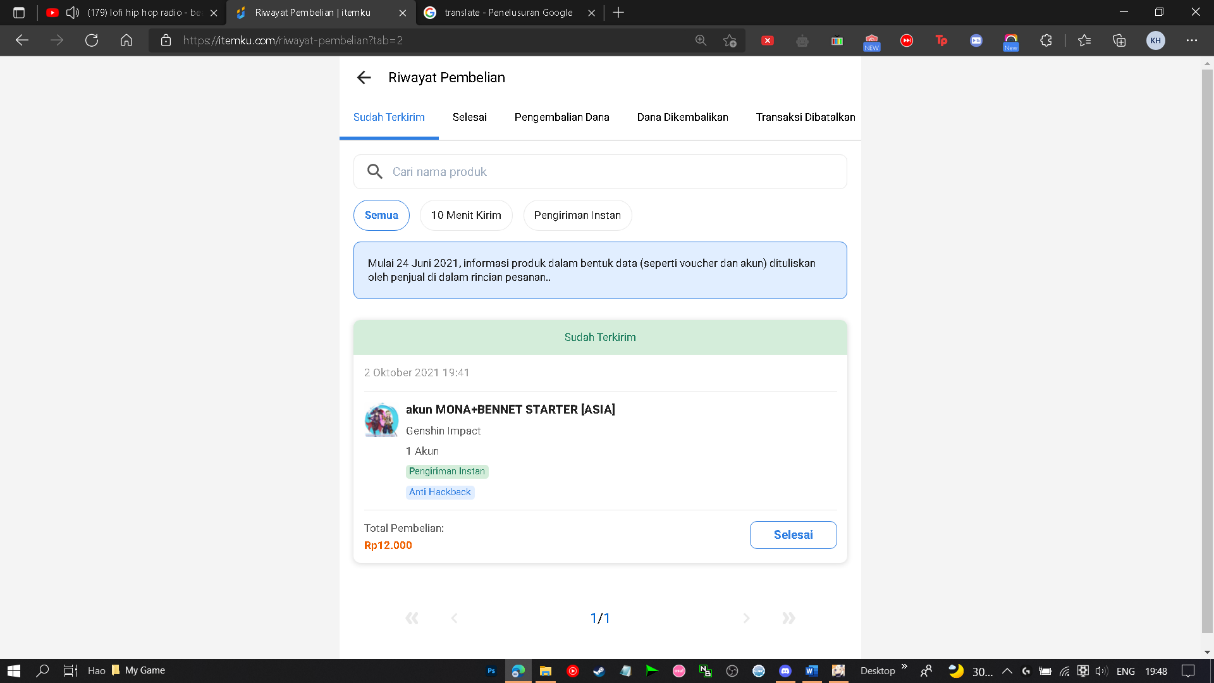
Pembeli dapat melihat status barang yang dibeli saat pembeli sudah menerima item yang diberi oleh seller.disini dijelaskan detail barang atau item serta cara pengiriman barang atau item tersebut serta jumlah pemesanan barang.



Gambar 2.7

halaman Status Menunggu Transaksi

Di halaman ini pembeli menunggu item/barang yang dikirimkan oleh seller dan disini dijelaskan terdapat total pembelian serta berapa stok barang atau item yang dibeli oleh buyer.

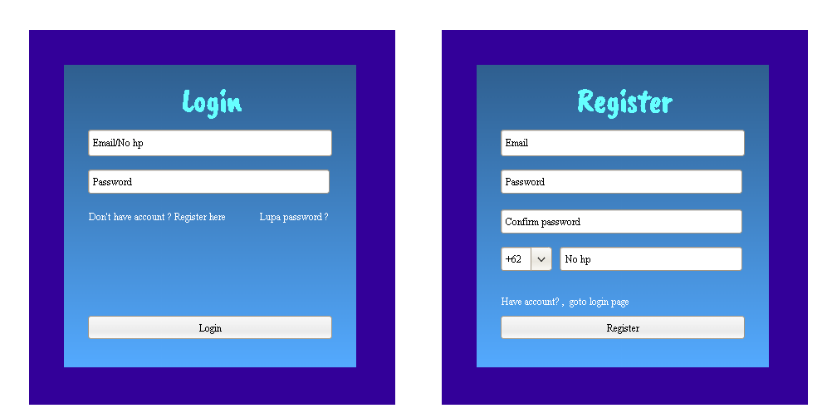


Gambar 2.8

halaman Status Konfirmasi Transaksi

Di halaman ini pembeli dapat mengonfirmasi jika item atau barang sudah dikirim serta terdapat status pembelian sudah terkirim kepada buyer dan terdapat tombol selesai Ketika buyer sudah menerima barang atau item yang telah dikirim oleh seller.

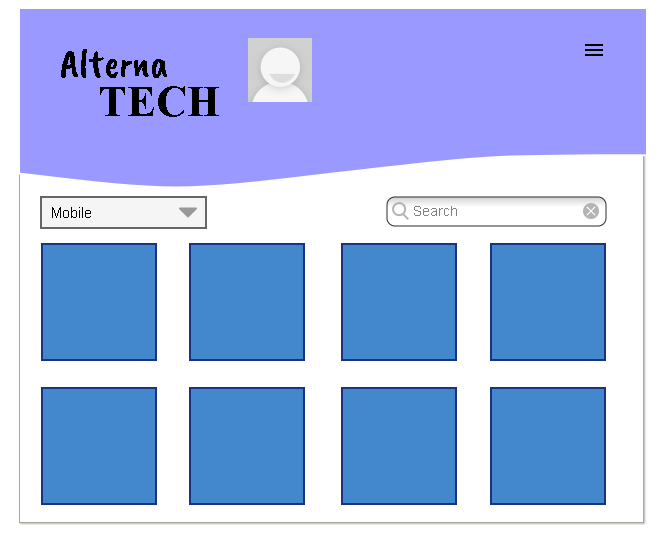
### 2.1.3 Prototype



Gambar 2.9

Login dan Register Page

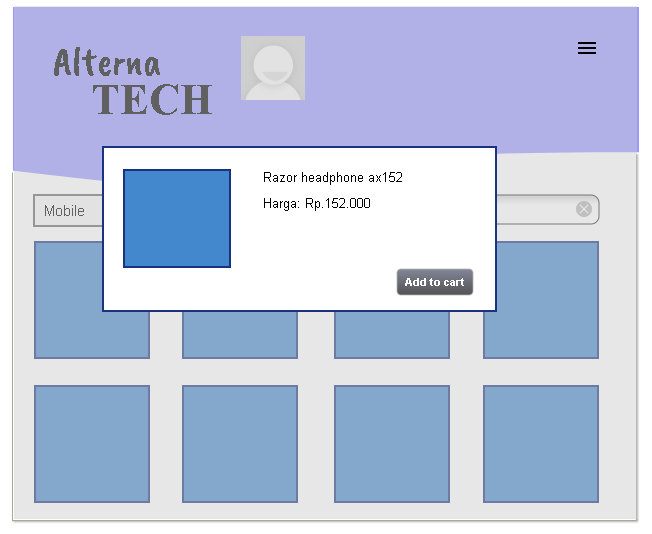
Pada page login, user diminta untuk memasukkan data supaya bisa masuk kedalam homepage. Namun, juka user belum memiliki akun, maka user akan dibawa ke halaman register untuk mengisi data pembuatan akun.



Gambar 2.10

Halaman Home

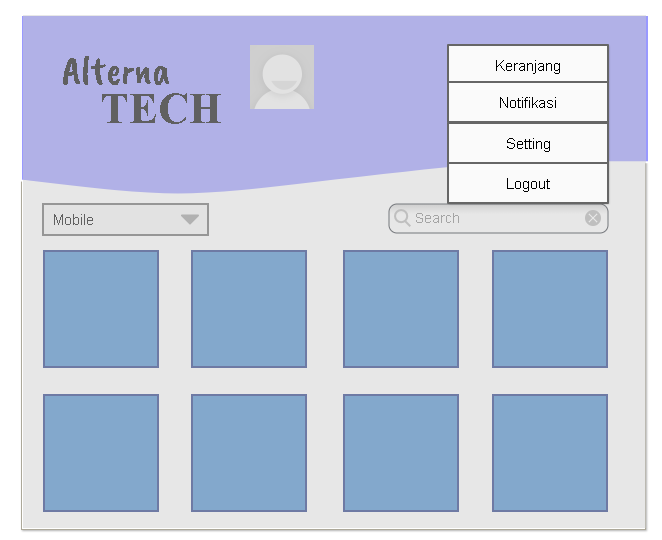
Halaman home berisi perlengkapan gaming yang kita sediakan dalam bentuk grid. Kami jadikan ini sebagai halaman paling depan supaya User bisa dengan mudah melihat produk-produk ketika user baru masuk kedalam aplikasi. Selain itu, ada juga textbox yang berguna untuk melakukan pencarian terhadap suatu barang. Jadi, jika melakukan search maka halaman home akan menunjukan produk sesuai dengan keyword yang ada di textbox yang user isi. Kami juga sediakan sebuah combobox yang berisi kategori produk yang tersedia di toko. Setelah user memilih salah satu kategori di combobox, maka produk yang akan muncul adalah produk dengan kategori yang dipilih.



Gambar 2.11

Home page ketika produk di klik

Ketika salah satu produk di klik, maka akan muncul nama serta harga produk. User akan mudah untuk melihat detail dari suatu produk beserta fotonya secara close-up supaya user bisa memastikan apa yang ingin dia beli. Tombol add to cart yang kami sediakan berfungsi untuk memasukkan item itu kedalam shopping cart user jika user sudah memastikan untuk membeli produk yang dia pilih.



Gambar 2.1

Tampilan Menu User

Ketika menu user di klik, maka akan menampilkan menu keranjang, notifikasi, setting dan logout. Menu Keranjang berguna untuk menunjukkan apa saja yang kita masukkan ke dalam shopping cart. Notifikasi dapat berisi notifikasi persetujuan pembelian, persetujuan pembayaran, kupon, dan lain lain. Setting berguna untuk melakukan update pada detail akun user seperti username, email, password, dan lain lain.



Gambar 2.1

Tampilan Keranjang

Ketika menu Keranjang di klik, maka akan menampilkan barang yang kita masukkan kedalam keranjang. Detail yang diberikan berupa nama barang, harga barang serta jumlah. Kita juga bisa mengganti jumlahnya secara langsung. User bisa melakukan check out semua produk maupun check out secara satu-satu.

* 1. **Analisis Kebutuhan**

Analisis kebutuhan dilakukan untuk menentukan ekspektasi dari software.

berikut adalah Analisa requirement dari use case yang ada pada aplikasi. Requirement dikelompokkan berdasarkan use case masing-masing

* Interface User

Pengguna aplikasi adalah mayoritas gamer, Maka UI akan dibuat secara sederhana guna untuk mudah di operasikan serta digunakan. UI juga disesuaikan supaya user bisa nyaman menggunakan aplikasi ini.

* Pembelian

Produk barang di aplikasi kami terdapat 2 jenis yaitu:

1. Produk digital

Produk Digital adalah produk yang tidak memiliki bentuk fisik maka produk digital dapat berupa voucher,item serta akun

1. Peralatan Gaming

Peralatan gaming berupa Mouse,Keyboard,Headset dsb

* 1. **Hardware Requirement**

Aplikasi yang kita kembangkan merupakan Windows Forms yang tidak membutuhkan resource yang berat sehingga tidak membebani perangkat yang digunakan. Adapun requirement aplikasi kita adalah

- Prosesor = 1ghz single core processor atau lebih

- Ram = >=4gb

* Koneksi internet memadai
* Membutuhkan web browser (Internet Explorer)
  1. **Batasan Aplikasi**

Batasan-batasan dalam sistem aplikasi *Alternatech* merupakan fitur-fitur yang tidak dapat dilakukan dalam sistem ini. Batasan-batasan ini dibuat agar tujuan sistem semakin jelas dan menghindari adanya fitur berlebih ataupun tidak sesuai dengan tujuan awal dibuatnya aplikasi ini. Adapun batasan-batasan tersebut

* Membutuhkan koneksi internet yang baik untuk melakukan transaksi
* Pemilik aplikasi tidak bertanggung jawab dalam sistem transaksi produk in game yang dimana item tersebut merupakan hasil perjanjian pembeli dan penjual item in game
* Sistem transaksi tidak dapat dibatalkan tanpa alasan tertentu

# Bab 3

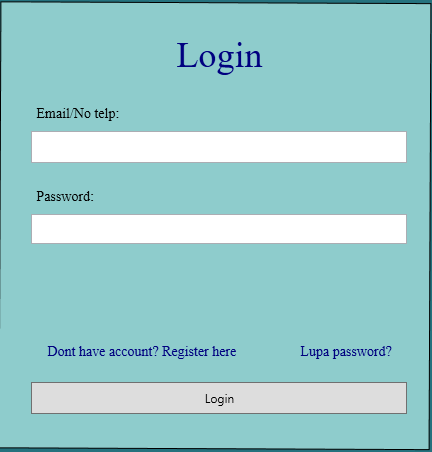
**System Desain**

Di bab ketiga ini, kami membuat berbagai desain Berisi; Sistem desain Database, Entity Relationship Diagram serta desain procedural. Proses mendefinisikan elemen dalam bab ini Sistem, arsitektur, komponen, dan datanya berbasis permintaan Didefinisikan dalam bab sebelumnya. Ini adalah proses definisi, Mengembangkan dan merancang sistem yang memenuhi kebutuhan

## 3.1 Desain Interface

Desain Interface adalah desain awal dari tampilan antarmuka Aplikasi untuk memudahkan pemrosesan lebih lanjut. Desain ini Dibuat dengan aplikasi Microsoft Visual 2019.berikut Ini adalah tampilan prototipe Aplikasi *Alternatech*

1. **Tampilan Awal**



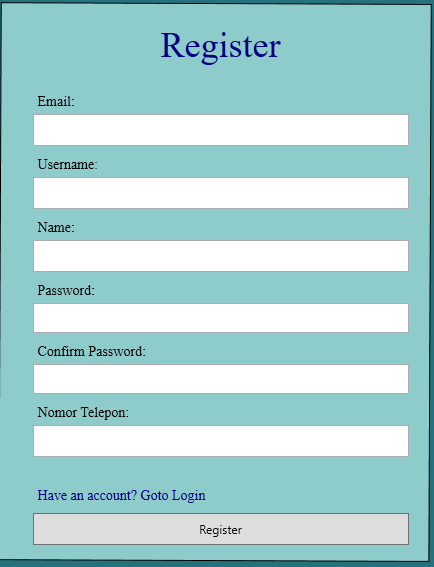
**Gambar 3.1**

**Login**

Gambar 3.1 adalah tampilan yang menunjukkan layar saat memasuki program *Alternatech*.Direncanakan untuk masuk ke bagian user page, harus masuk dengan nama pengguna: <Email> dan password: <password>.

Untuk login sebagai Admin maka pada awal pembuatan client akan kami mengatur username dan password untuk mengakses Admin sesuai keinginan client. Setiap User memiliki role masing-masing yang dapat dibuat ketika membuat akun user itu sendiri pada halaman register.

1. **Tampilan Register**

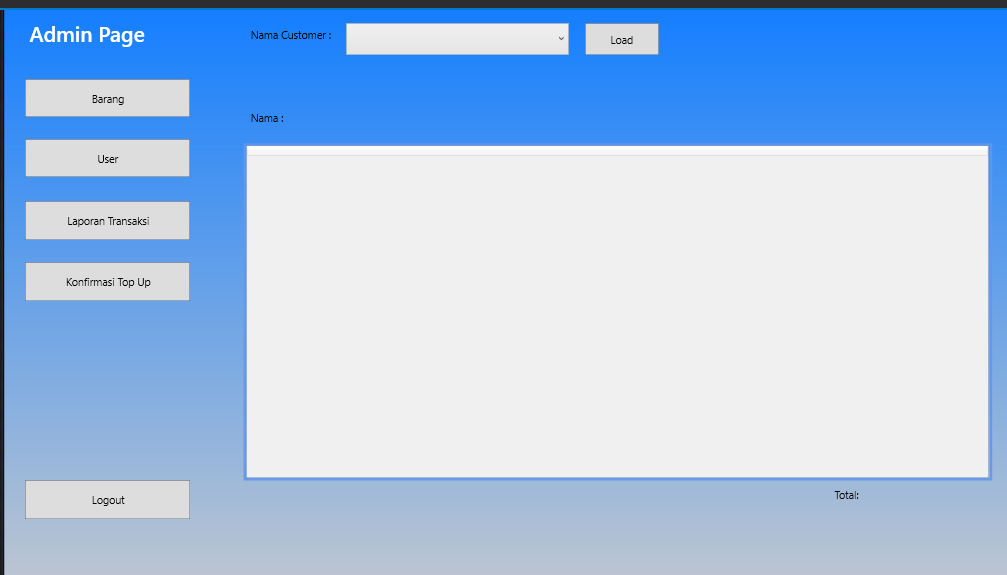


**Gambar 3.2**

**Register**

Gambar 3.2 adalah tampilan yang menunjukkan layer saat memasuki halaman register. Untuk mendaftarkan user agar bisa login dibutuhkan Email ,username ,nama ,password dan no telp untuk data diri. Data diri tersebut akan tercatat dan tersimpan dalam database untuk proses login serta pengubahan data diri user seperti role user,Untuk default role user akan diarahkan ke buyer.

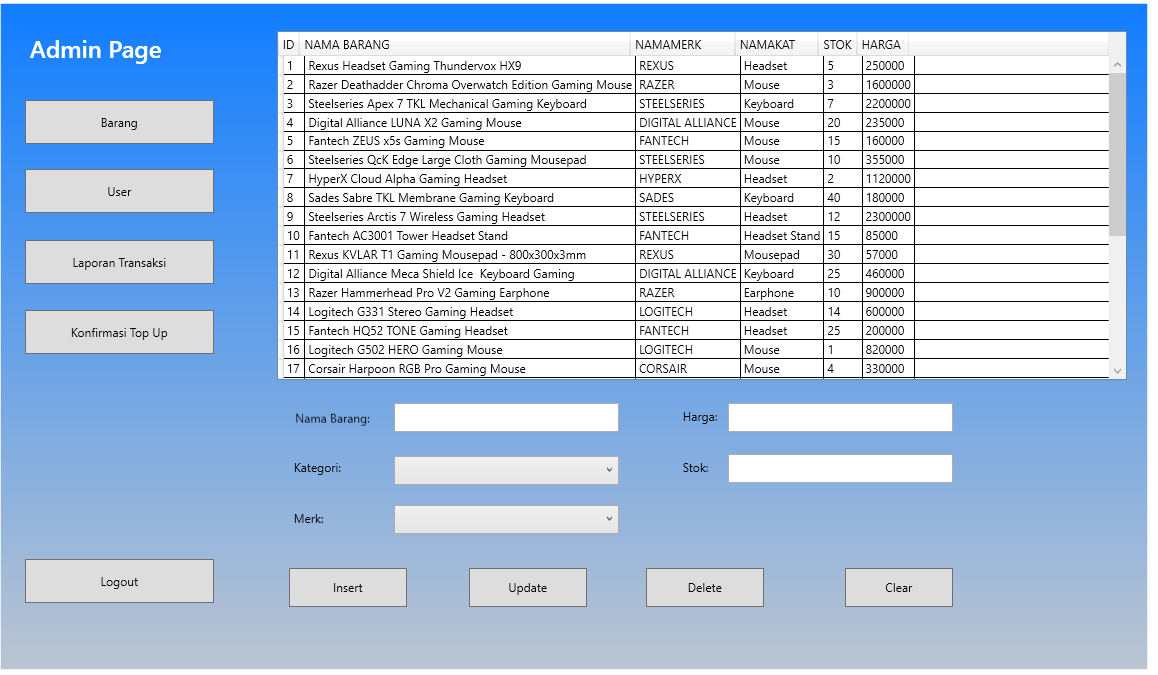
1. **Tampilan Admin**



**Gambar 3.3**

**Tampilan Awal Admin**

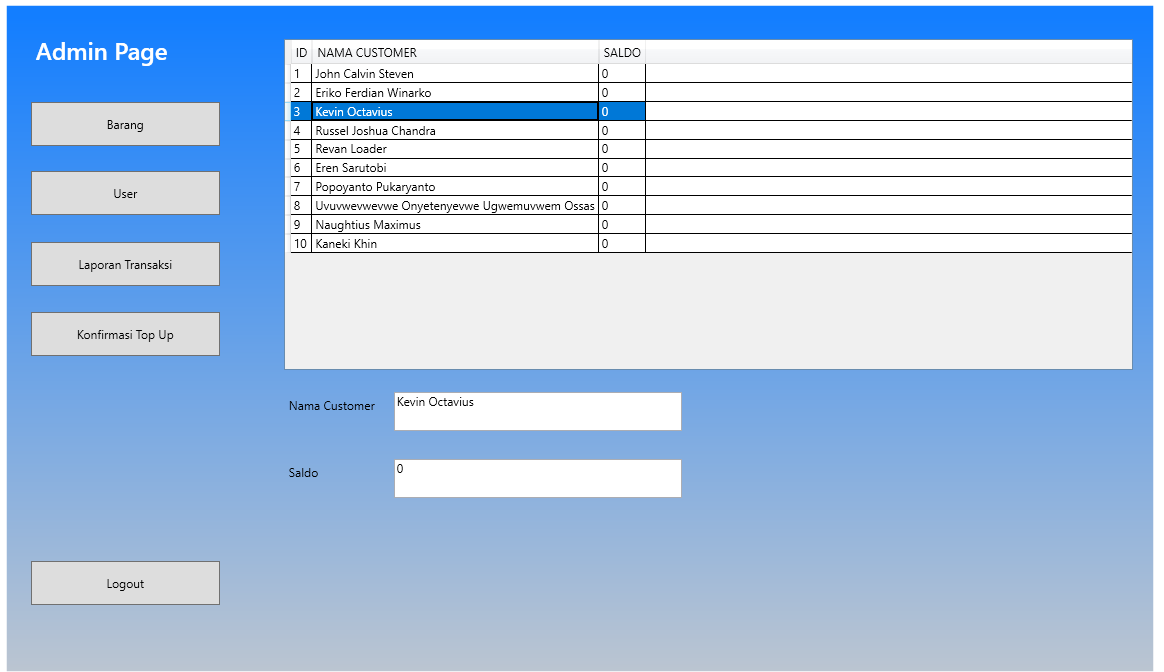
Gambar 3.3 adalah tampilan awal saat client memasuki halaman admin, disini terdapat 5 tombol navigasi yang digunakan untuk melihat list serta laporan user dan barang, list dan laporan tersebut ditampilkan di kotak besar tersebut dan terdapat fitur search yang hanya tampil Ketika client menekan button User. saat button Barang ditekan maka tampilan tersebut menjadi seperti gambar 3.4 dibawah ini



**Gambar 3.4**

**Tampilan Admin Barang**

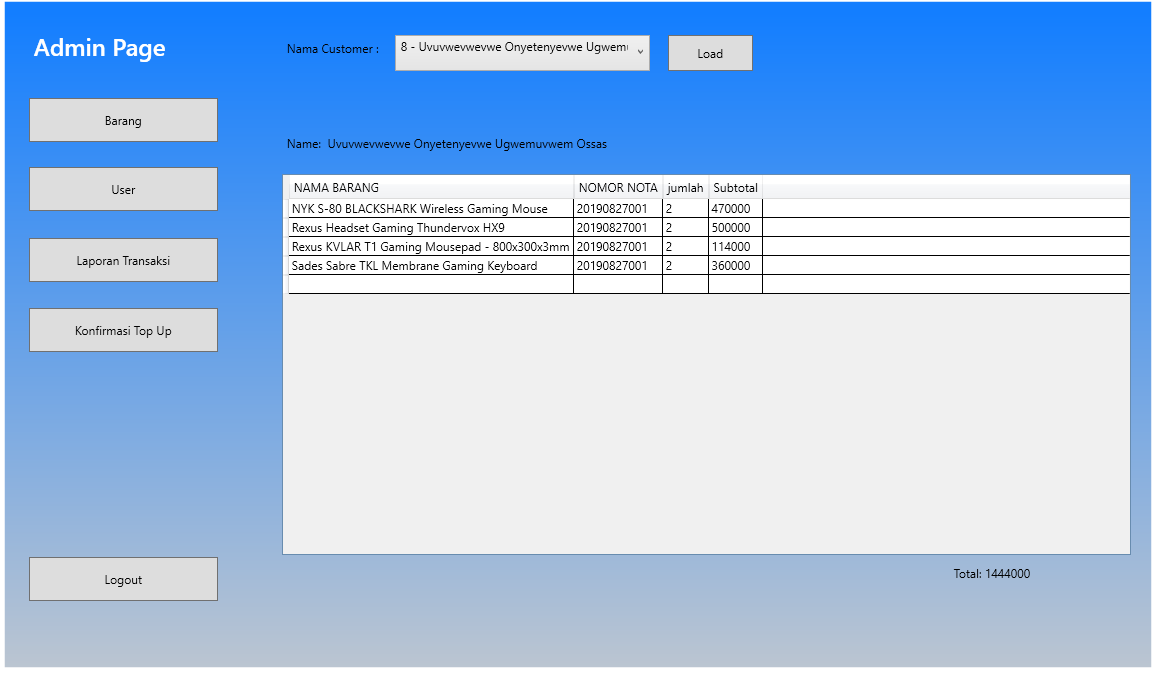
Ditampilan barang ini terdapat fitur Create , update , delete barang yang digunakan untuk mengubah status barang , jika client ingin memasuki barang hanya mengisi field nama barang , kategori , merek serta harganya , jika client ingin mengubah atau update , client harus menekan dua kali ke salah satu barang didalam kotak besar maka akan tertampil secara langsung di dalam field serta tombon insert menjadi disable lalu fitur delete ini juga serupa , lalu client hanya menekan tombol update untuk mengubah data barang serta delete untuk menghapus barang.lalu saat button user ditekan maka akan muncul seperti berikut yang berada pada gambar 3.5



**Gambar 3.5**

**Tampilan Admin User**

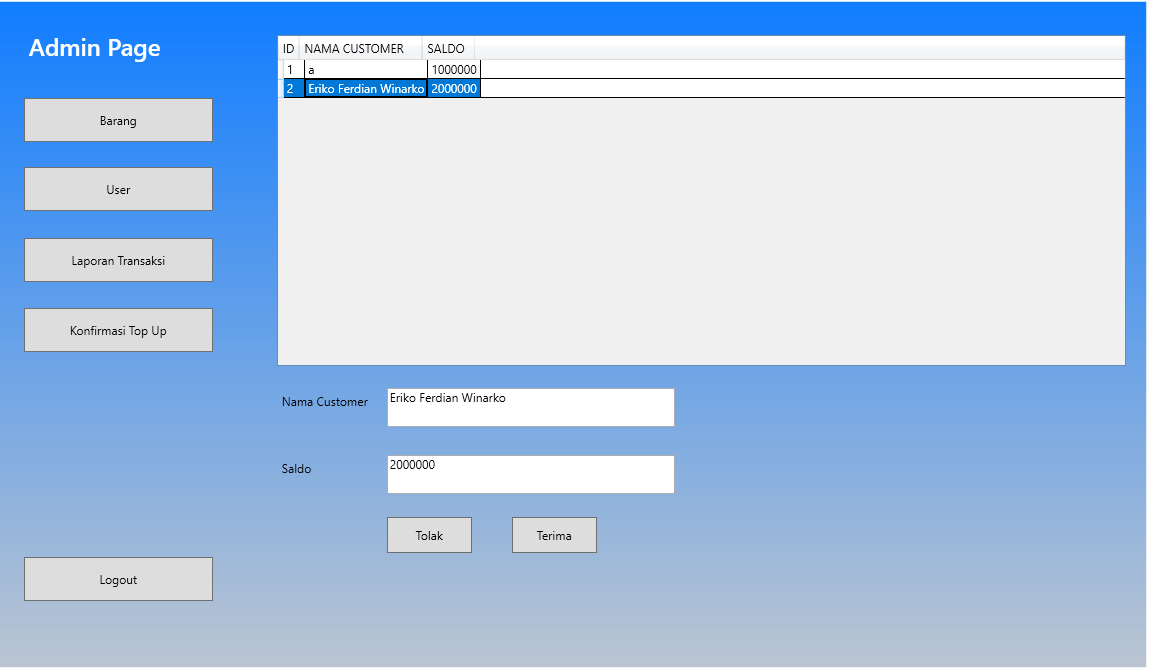
Ditampilan user admin ini terdapat list user yang terdiri dari id, nama customer serta saldonya, lalu Ketika client menekan salah satu text di salah satu colom maka akan ketampil di text box yang berisi nama customer serta saldo, selanjutnya saat client menekan tombol laporan transaksi maka tampilan tersebut menjadi seperti gambar 3.6 dibawah ini



**Gambar 3.6**

**Tampilan Admin Laporan Transaksi**

Ditampilan laporan transaksi ini hanya menampilkan laporan belanja user serta menampilkan total harga dari semua barang yang sudah dibeli, lalu terdapat fitur filter sesuai user di atas kotak grid view dengan menggunakan combo box serta tombol load untuk menampilkan sesuai yang dipilih oleh combobox tersebut, selanjutnya saat client menekan tombol Konfirmasi Top up maka tampilan tersebut menjadi seperti gambar 3.7 dibawah ini

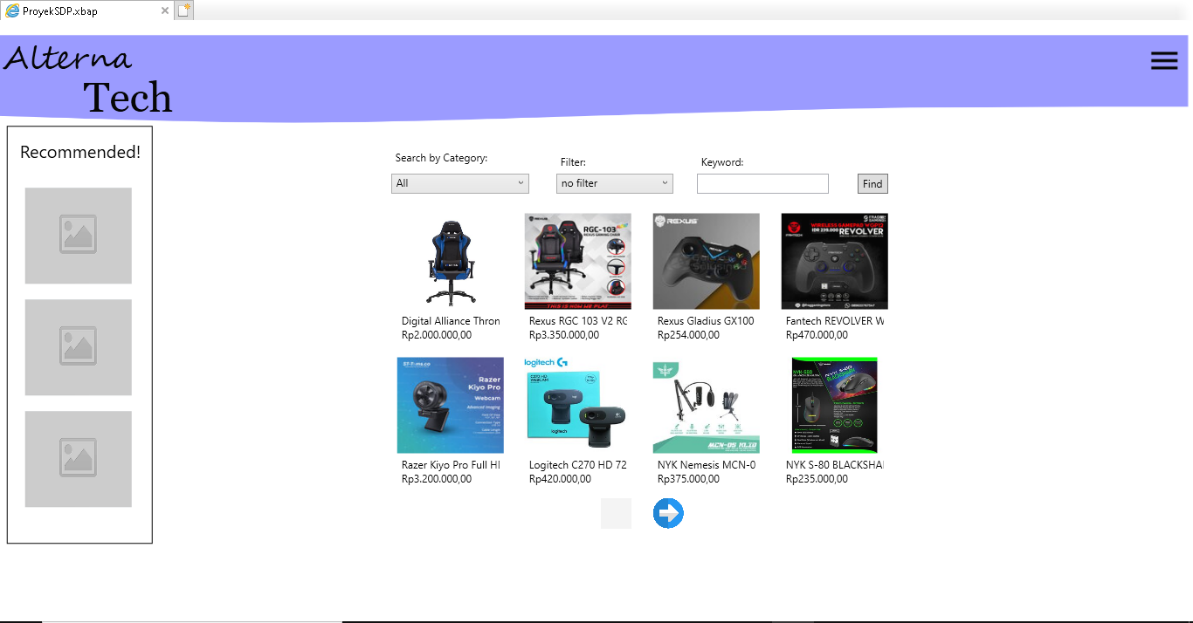


**Gambar 3.7**

**Tampilan Admin Konfirmasi Top Up**

Ditampilan Konfirmasi Top up ini menampilkan request user top up. Client terlebih dahulu menekan text yang terdapat di data grid kotak untuk mengkonfirmasi top up user lalu ada ada 2 tombol untuk menolak request user serta konfirmasi request user , Ketika client sudah mengkonfirmasi request top up user maka saldo di user bertambah dan request terhapus di datagrid kotak tersebut

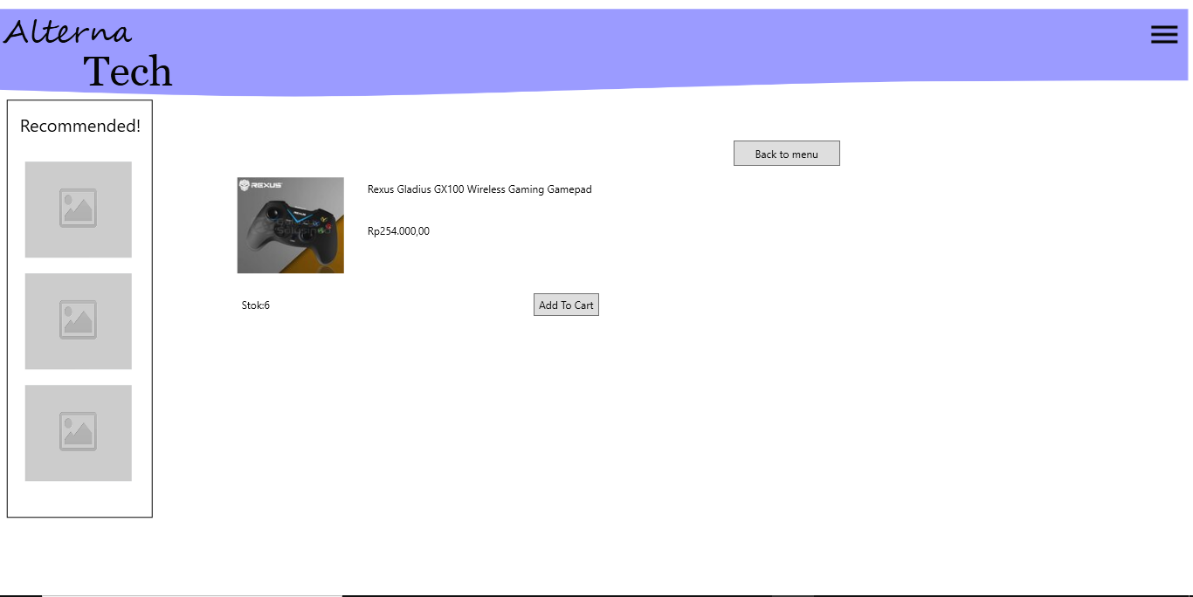
1. **Tampilan User**



**Gambar 3.8**

**Tampilan Home User**

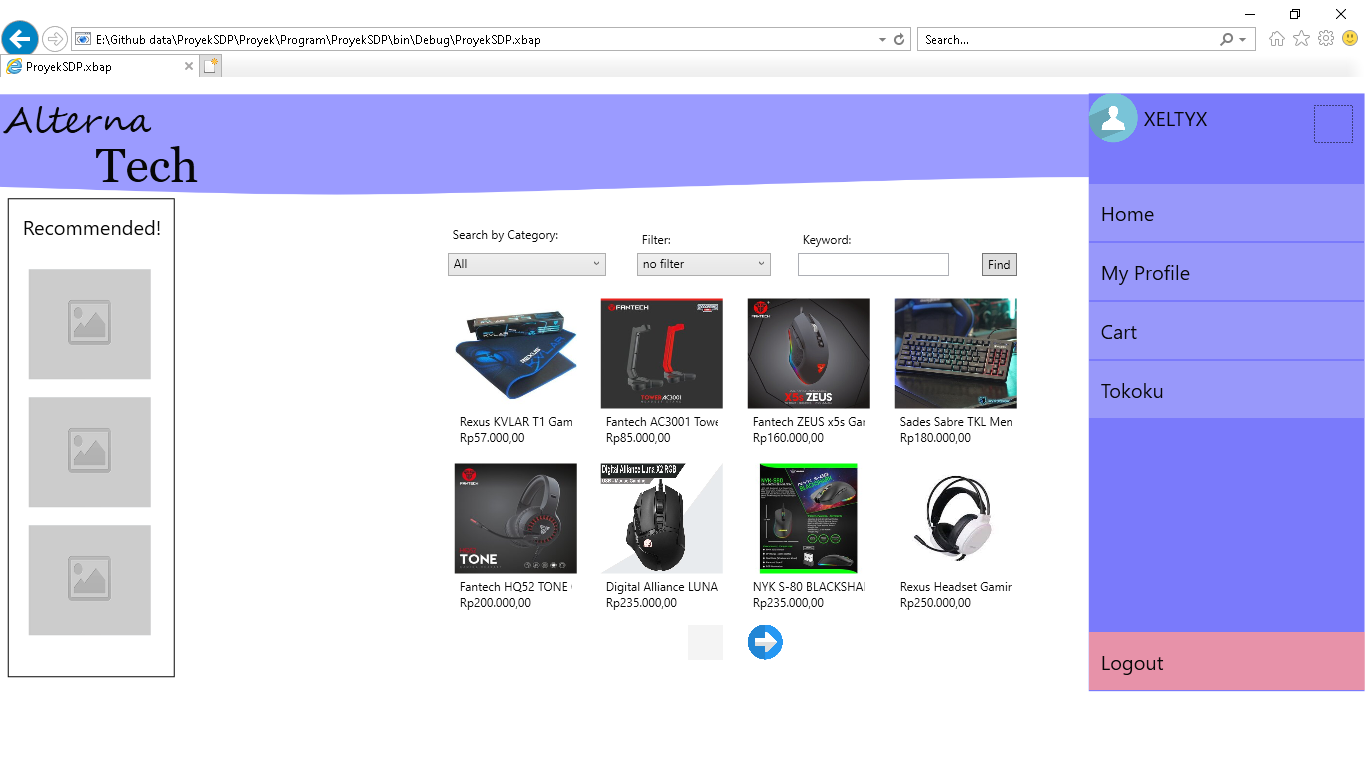
Gambar 3.8 adalah tampilan user pertama kali sesudah login, di home user ini terdapat fitur search barang berdasarkan category misalnya kategori untuk headset ,kursi gaming ,dan sebagainnya, lalu terdapat filter harga di sebelah combo box category yang memiliki filter harga rendah ,tinggi, dan none, lalu terdapat keyword search yang digunakan untuk mencari barang yang sesuai dengan apa yang diketik di textbox sebelah kanan filter harga dan tombol find untuk mencari filter yang ingin di cari.tampilan home ini terdapat slider yang berbentuk arah panah ke kiri dan kanan serta fungsinya untuk arah panah ke kiri dia kembali ke halaman sebelumnya lalu arah panah ke kanan akan pindah ke halaman berikutnya, lalu jika salah satu barang ditekan dengan dua kali maka halaman user home akan dialihkan ke halaman detail barang seperti gambar dibawah ini.



**Gambar 3.9**

**Tampilan Detail Barang User**

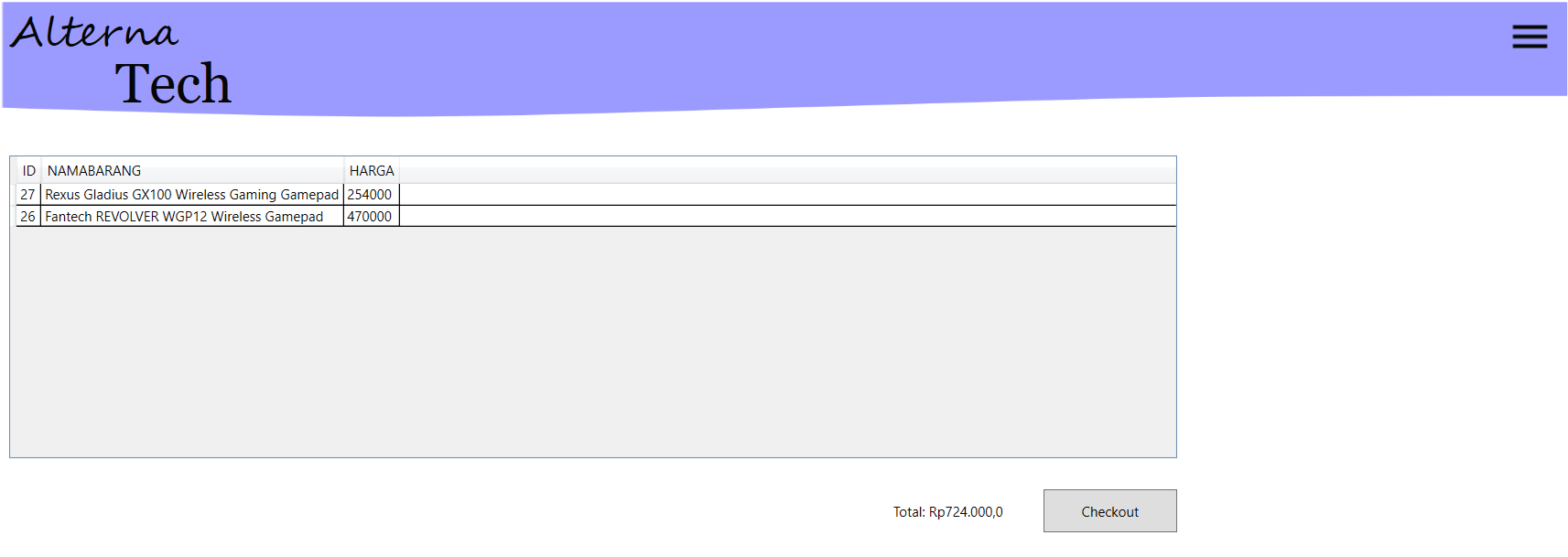
Gambar 3.9 adalah tampilan detail barang , disini user bisa melihat stock barang serta harga barang tersebut. Untuk navigasi nya terdapat 2 tombol yaitu tombol Kembali ke halaman home user serta tombol untuk add barang ke cart user untuk menyimpan produk tertentu tanpa harus mengambilnya kembali sebelum membeli. Lalu untuk membuka profile, user hanya menekan garis tiga disebelah kanan atas seperti gambar dibawah ini



**Gambar 3.10**

**Tampilan Collapse Menu**

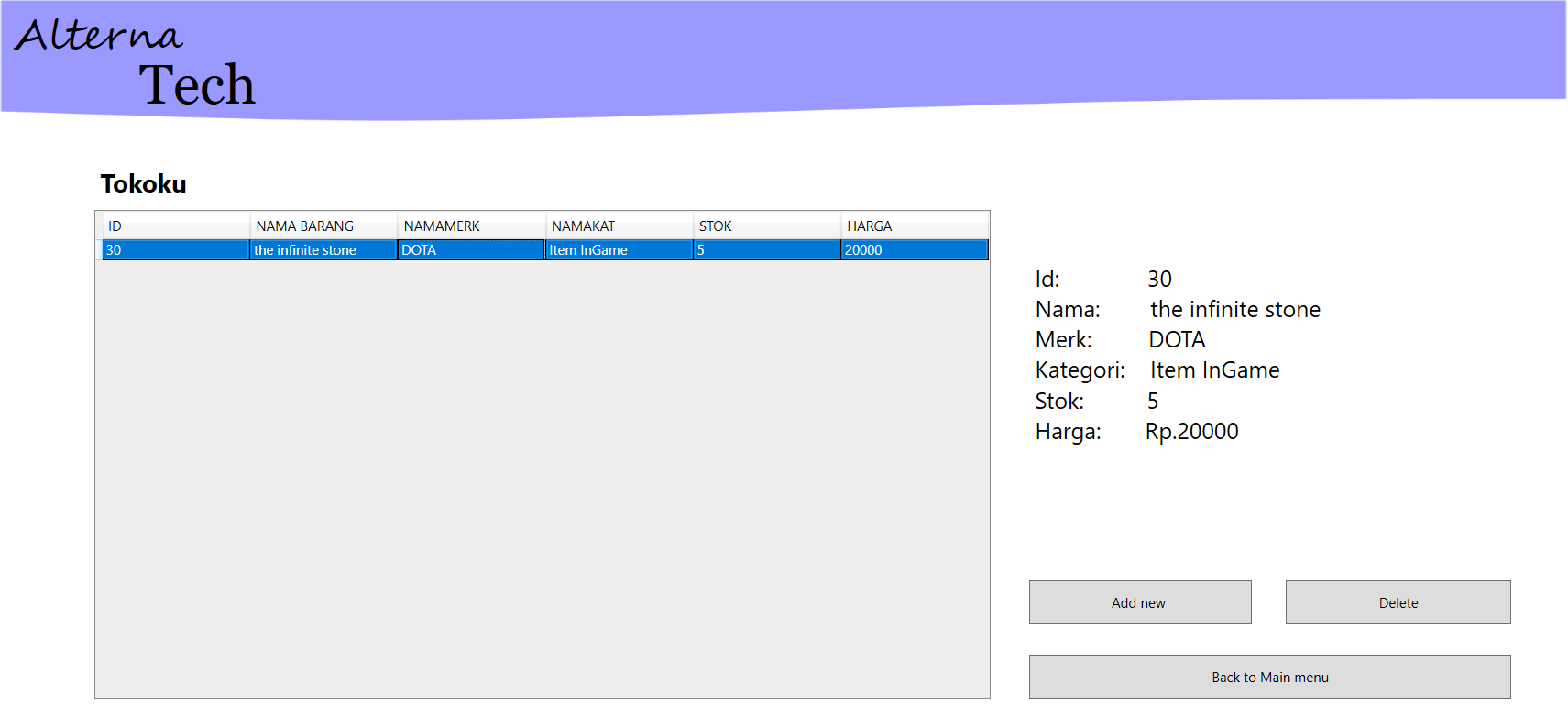
Pada gambar 3.10 tersebut terdapat 4 navigasi yaitu home untuk membuka home user, my profile untuk membuka profile user, cart untuk membuka barang yang sudah disimpan saat user menekan tombol *add to cart*,lalu yang terakhir tokoku untuk menampilkan barang yang dijual oleh user tersebut.



**Gambar 3.11**

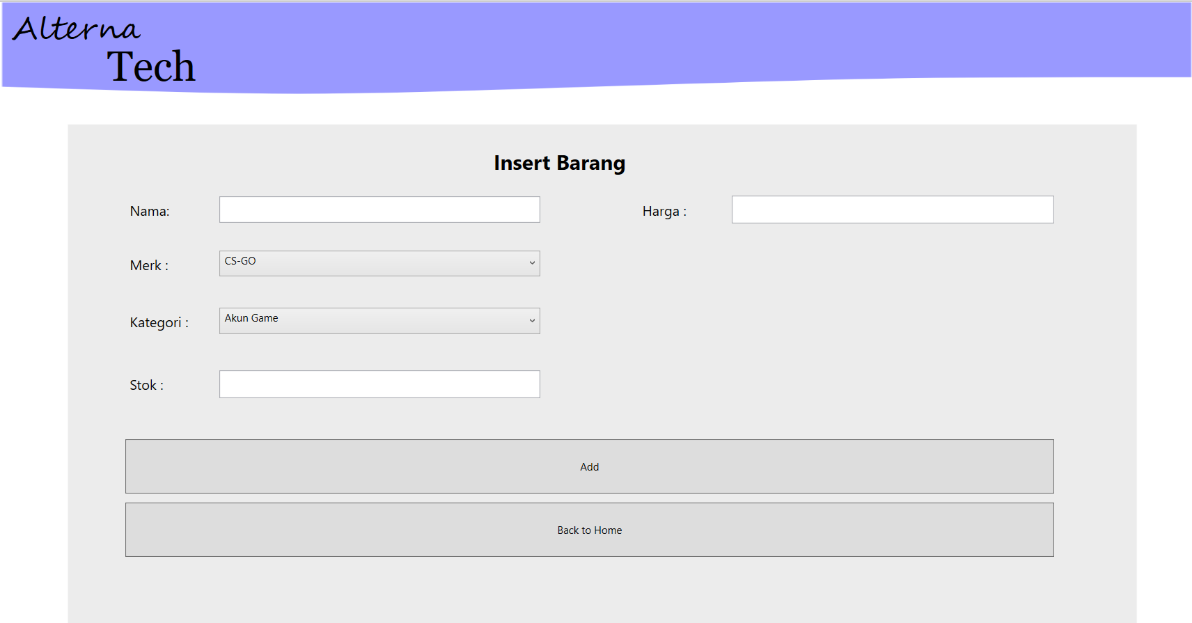
**Tampilan Cart User**

Pada gambar 3.11 tersebut terdapat 1 datagrid yang berisi list barang yang ingin dibeli, 1 tombol *checkout* untuk langsung membeli barang yang ada pada datagrid dan program akan melakukan pengecekan apakah saldo user mencukupi, ,terdapat juga 1 label disebelah kiri checkout untuk menampilkan harga total dari semua barang yang ada pada datagrid, saat salah satu item pada datagrid di double click maka program akan secara otomatis menghapus barang tersebut dari datagrid.

**Gambar 3.12**

**Tampilan User Seller(Toko User)**

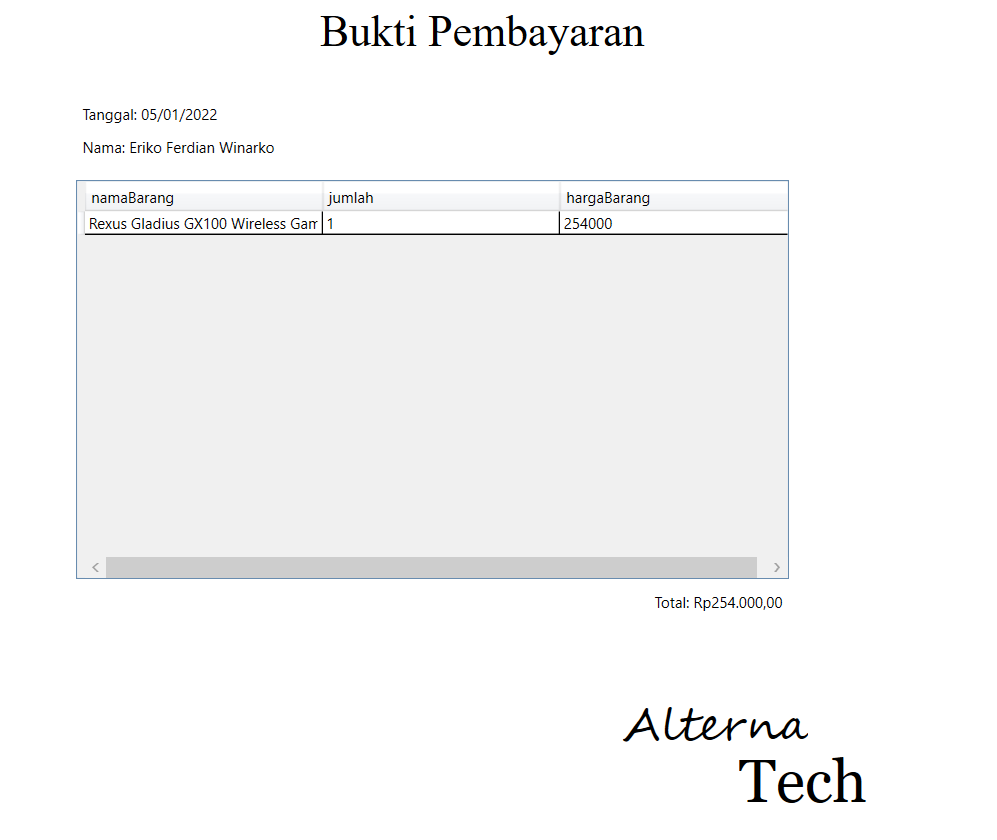
Pada gambar 3.12 tersebut terdapat 1 datagrid yang berisi list barang yang dijual user tersebut, button *add new* untuk mengarahkan ke halaman add barang jualan baru, button *delete* untuk menghapus barang yang dipilih pada datagrid dan button *back to main menu* yang berfungsi untuk mengarahkan tampilan ke halaman home user kembali saat datagrid di double click maka program akan menampilkan detail item di sebelah kanan datagrid



**Gambar 3.13**

**Tampilan Add/Insert Barang User Seller**

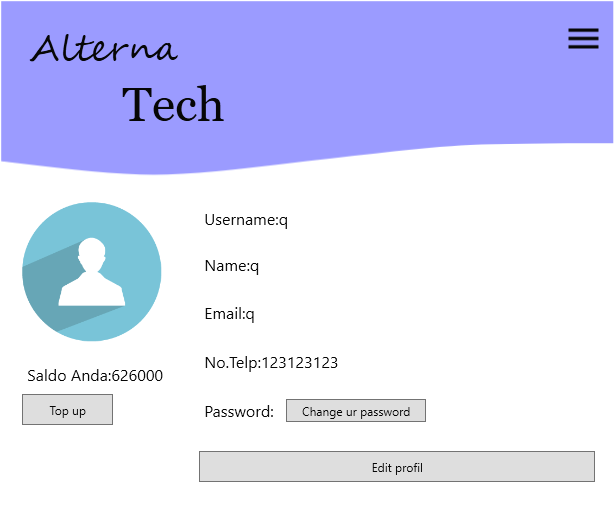
Pada gambar 3.13 tersebut terdapat textbox nama untuk mengisi nama barang, combobox merk untuk menentukan merk barang, combobox kategori untuk menentukan kategori barang, textbox stok untuk memasukan jumlah stok barang, textbox harga berguna untuk menentukan harga barang dan button *Add* yang berguna untuk menambahkan barang jualan dengan keterangan barang sesuai dengan yang diisi diatas ,barang otomatis akan dimasukkan ke database button *back to tokoku* berguna untuk mengarahkan Kembali ke tampilan toko user



**Gambar 3.14**

**Tampilan Bukti Pembayaran**

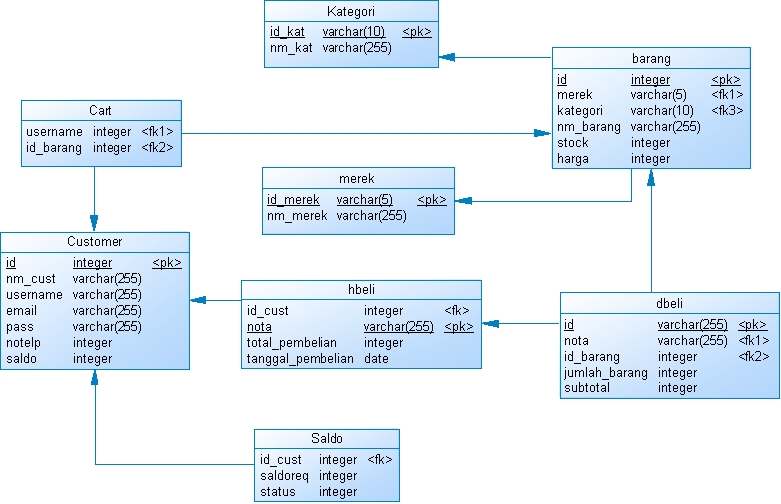
Pada gambar 3.13 tersebut merupakan tampilan saat user berhasil melakukan transaksi. Informasi yang disertakan dalam laporan yaitu, nama barang yang dibeli, jumlah barang yang dibeli, dan harga barang per satuan yang dibeli. Kami juga menampilkan tanggal pembelian, nama pembeli, alamat dari pembeli, serta total keseluruhan barang yang dibeli.

 **Gambar 3.15**

**Tampilan Profil User**

Pada gambar 3.15 tersebut merupakan tampilan saat user menekan tombol MyProfile. Pada menu ini, akan ditampilkan halaman profil yang berisi informasi user. Dihalaman ini user dapat mengubah informasi tersebut beserta top up saldo.

**3.2 Desain Database**

Pada Software *Alternatech*, akan dilakukan pencatatan pemesanan dan pembayaran, daftar menu, dan juga pegawai yang melayani setiap pemesanan customer. Entity pada *Alternatech* ada delapan buah, yakni Customer, Hbeli, Dbeli , Kategori, Merk, Barang, Saldo dan juga Cart. Berikut adalah Diagram ERD *Alternatech*

**Gambar 3.2**

**Entity Relationship Diagram**

Ketika customer melakukan pembelian, pembelian akan dicatat dan ditampung di detail pembelian. Barang yang sudah dibeli oleh customer akan mendapatkan nota. Barang-barang tersebut memiliki banyak kategori dan merek. Berikut adalah detail dari tabel-tabel yang dihasilkan dari ERD tersebut

**Tabel 3.1**

**Customer**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| Id | INT | Primary Key, Auto increment |
| nm\_cust | Varchar(255) |  |
| username | Varchar(255) |  |
| email | Varchar(255) |  |
| pass | Varchar(255) |  |
| no telp | INT |  |
| saldo | INT |  |

Tabel 3.1 Merupakan Tabel Customer yang digunakan untuk menyimpan data customer. Tabel ini berisikan ID yang menjadi Primary Key serta auto increment, username dengan data tipe varchar,email dengan data tipe varchar,password dengan data tipe varchar,notelp dengan data tipe Integer serta saldo dengan data tipe Intenger.

**Tabel 3.2**

**Barang**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id | INT | Primary Key, Auto increment |
| nm\_barang | Varchar(255) |  |
| stock | INT |  |
| harga | INT |  |
| merk(ref.merk) | INT | Foreign Key |
| kategori(ref.kategori) | INT | Foreign Key |

Tabel 3.2 Merupakan Tabel Barang yang digunakan sebagai database dari data-data sebuah barang. Tabel ini berisikan ID yang menjadi Primary Key serta auto increment , nm\_barang dengan data tipe varchar, stock dengan data tipe Integer, harga dengan data tipe Integer,merek yang merupakan foreign key yang digunakan untuk relasi Tabel barang dengan tabel merek dan kategori juga memiliki foreign key yang digunakan untuk relasi Tabel kategori dengan Tabel Barang.

**Tabel 3.3**

**Merek**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode merek | Varchar(5) | Primary Key |
| nm\_merk | Varchar(255) |  |

Tabel 3.3 merupakan tabel merek yang digunakan sebagai database Relasi dari barang dan garansi. Tabel ini berisikan kode merek yang menjadi Primary Key dan pemberian nama kode merek dengah 3 huruf depan dan penomoran serta nm\_merk.

**Tabel 3.4**

**Dbeli**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id | Varchar(255) | Primary Key |
| nota | Varchar(255) | Foreign Key(Hbeli) |
| Id\_barang(fk barang) | INT | Foreign Key(Barang) |
| jumlah barang | INT |  |
| subtotal | INT |  |

Tabel 3.4 merupakan table Dbeli yang digunakan menampilkan detail seperti barang apa saja yang dibeli beserta jumlah dan harganya. Tabel ini berisikan id sebagai primary key , nota merupakan foreign key yang digunakan untuk relasi antara tabel Dbeli dengan tabel hbeli , id\_barang merupakan foreign key yang digunakan untuk relasi antara tabel Dbeli dengan tabel barang , jumlah barang serta sub\_total.

**Tabel 3.5**

**Hbeli**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| nota | Varchar(255) | Primary Key |
| id\_cust(fk customer) | INT | Foreign Key(Customer) |
| total pembelian | INT |  |
| tanggal pembelian | Date |  |

Tabel 3.5 merupakan table Hbeli yang digunakan untuk menampilkan detail total pembelian user dan tanggal pembelian barang tersebut. Tabel ini berisikan nota sebagai primary key dengan format date YYYY/MM/DD + penomoran, id\_cust merupakan foreign key yang digunakan untuk relasi antara tabel Hbeli dengan tabel Customer, total\_pembelian, tanggal\_pembelian.

**Tabel 3.6**

**Kategori**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode\_kat | Varchar(255) | Primary Key |
| nm\_kat | Varchar(255) |  |

Tabel 3.7 merupakan table Kategori yang digunakan untuk menyimpan data kategori serta database Relasi dari tabel kategori dan tabel barang untuk mengkategorikan barang. Tabel ini berisikan kode\_kat sebagai primary key , nm\_kat.

**Tabel 3.7**

**Cart**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| Username | Varchar(255) | Foreign Key(Customer) |
| Id\_barang | Integer | Foreign Key(barang) |
| jumlah | Integer |  |

Tabel 3.7 merupakan table Cart yang digunakan untuk menyimpan data barang ketika user menambahkan kedalam cart dan serta database Relasi dari tabel user dan tabel barang . Tabel ini berisikan *username* sebagai Foreign key yang digunakan untuk relasi saat user memasukkan kedalam cart. Atribut *id\_barang* digunakan untuk mencatat barang yang dimasukkan oleh user kedalam cart. Atribut *Jumlah* berguna untuk menunjukkan jumlah barang yang ingin dibeli oleh user.

**Tabel 3.8**

**Saldo**

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| id | INT | Primary Key, Auto Increment |
| id\_cust(fk customer) | INT | Foreign Key(Customer) |
| saldoreq | INT |  |
| status | INT | (1pending 2accepted 3rejected) |

Tabel 3.8 merupakan table saldo yang digunakan untuk melihat data saldo user serta menyetujui atau menolak request saldo user. tabel ini berisikan id\_cust sebagai foreign key yang digunakan untuk relasi antara tabel Saldo dengan tabel Customer,saldoreq serta status yang dibagi menjadi 3 jenis yaitu 1 = pending , 2 = accepted , 3 = rejected.

**3.3 Desain Procedural**

Pada point ini kita menjelaskan algoritma pada aplikasi yang kita kembangkan pada aplikasi ini. Berikut adalah algoritma yang kami kembangkan pada aplikasi kami

**Algoritma 3.1 Contoh Algoritma Filtering barang**

1. begin
2. If kategori[] is equal “all” then
3. Print filter barang untuk all success
4. Elseif kategori[] is not equal “all”
5. Print filter barang untuk kategori[] success
6. End if
7. end

**Algoritma 3,2 Contoh Algoritma Login**

1. begin
2. If email is not Null and password is not null then
3. If email is equal admin and password is equal admin then
4. Print login sebagai admin
5. Elseif email is equal user and password is equald user then
6. Print login sebagai User
7. End if
8. Else then
9. Print “data fill blank”
10. End if
11. end

**Algoritma 3.3 contoh Algoritma Register**

1. begin
2. If email is not null and username is not null and pass is not null then
3. If pass is equal passConfirm then
4. Print Register Berhasil
5. Else then
6. Print Password not match with Confirm Password
7. End if
8. Else then
9. Print data is fill blank
10. End if
11. end

**Algoritma 3.4 contoh Algoritma CRUD**

Insert barang

1. begin
2. query = “”
3. If kat is not null or merk is not null then
4. Call getKodekat()
5. Call getKodemerk()
6. Query = “insert into BARANG(NAMA\_BARANG, MERK, KATEGORI,STOK,HARGA)”+“values('{nm\_barang}','{kodemerk}','{kodekat}',{stock},{harga})”
7. Print Update barang berhasil
8. Else then
9. Print fill blank !!
10. End if
11. end

Delete barang

1. begin
2. Query = “”
3. If kat is not null or merk is not null then
4. Query = “delete from barang where ID = {index+1}”
5. Print Delete barang berhasil
6. Else then
7. Print fill blank !!
8. End if
9. end

Update barang

1. begin
2. query = “”
3. If kat is not null or merk is not null then
4. Query = “Update Barang SET NAMA\_BARANG = '{ nm\_barang }',MERK = '{ kodemerk }',kategori = '{ kodekat }',STOK = {stock},HARGA = {harga} Where id = " + (index+1)”
5. Print Update barang berhasil
6. Else then
7. Print fill blank !!
8. End if
9. end

**Algoritma 3.4 contoh Algoritma pemilihan detail barang**

1. Begin
2. Query = “”
3. Temp[] = {barang.clicked.name}
4. Windows.Detail.barang = visible
5. Windows.Homepage = Hidden
6. Query = “SELECT harga,stok FROM BARANG WHERE nama\_barang like '%{temp[0]}%'"
7. While(Query read)
8. Print “stock” + query.stock
9. Print “Harga” + query.harga
10. End while
11. end

**Algoritma 3.4 contoh Algoritma memasukkan barang kedalam cart**

1. Begin
2. Query = “”
3. Query = "SELECT ID\_BARANG FROM CART WHERE USERNAME='{user.username}'"
4. While(Query read)
5. isFound = true
6. End while
7. If isFound = false then
8. Query = "INSERT INTO CART VALUES('{user.username}', {idbrg})"
9. Else then
10. Print “Barang sudah didalam cart”
11. isFound = false
12. End if
13. end

**Algoritma 3.4 contoh Algoritma User Melakukan Checkout**

1. Begin
2. query = “SELECT BARANG.ID, BARANG.NAMA\_BARANG, BARANG.HARGA FROM BARANG, CART WHERE CART.ID\_BARANG = BARANG.ID AND CART.USERNAME = '{user.username}'”
3. while (query)
4. total = total + query.total
5. end while
6. saldo = usersaldo
7. if total <= saldo then
8. query = "INSERT INTO H\_BELI VALUES(\"{nota}\", {user.id}, {total}, \"{DateTime.Now.ToString("yyyy-MM-dd")}\")"
9. Query = "UPDATE CUSTOMER SET SALDO = {user.saldo - total} WHERE ID = {user.id}"
10. Query = “DELETE FROM CART WHERE USERNAME = \"{user.username}\"”
11. Query = “"UPDATE BARANG SET STOK = STOK-1 WHERE BARANG.ID = {listidbarang[i]}"”
12. Print Terima kasih sudah berbelanja
13. Else then
14. Print Saldo kurang cukup
15. End if
16. end

**Algoritma 3.4 contoh Algoritma User Melakukan Top Up Saldo**

1. Begin
2. If nominal > 0 then
3. Query = “INSERT INTO REQ\_SALDO(ID\_CUST, SALDOREQ,KONFIRMASI) VALUES('{iduser}','{nominal}','{request}')”
4. Print Sedang Memproses Topup , jika topup tidak masuk dalam kurung 1x24 jam. Hubungi admin
5. End if
6. end

**Algoritma 3.4 contoh Algoritma Konfirmasi Top up User**

1. Begin
2. User\_id = user.id
3. If nominal > 0 then
4. Query ="Update CUSTOMER SET SALDO={Convert.ToInt32 (tbsaldo.Text)}WHERE NAMA\_CUST= '{tbnamacust.Text}'"
5. Query = “Update REQ\_SALDO SET KONFIRMASI = 'accepted' WHERE ID = {user\_id}”
6. Print Top up Berhasil
7. end

**Algoritma 3.4 contoh Algoritma Update profile User**

1. Begin
2. Iduser = user.id
3. Nama = “”
4. Email = “”
5. Notelp = “”
6. Query = "UPDATE customer SET NAMA\_CUST=Nama,EMAIL=Email, NO\_TELP= Notelp WHERE ID={iduser}"
7. end

**Algoritma 3.4 contoh Algoritma Mengubah password User**

1. Begin
2. NewPassword = “”
3. Iduser = user.id
4. Password = “”
5. If password = old.password then
6. If passwordTb = NewPassword then
7. Query = “UPDATE customer SET Password=@password WHERE ID={iduser}”
8. Print password sudah terganti
9. Else then
10. Print Password baru tidak sama dengan password konfirmasi
11. Else then
12. Print password salah
13. End if
14. End if
15. end

BAB IV

IMPLEMENTASI

Pada bab yang keempat ini akan membahas mengenai implementasi desain dari software Alternatech. Implementasi / code dibuat melalui software Microsoft Visual Studio. Software ini menggunakan .NET Framework 4.6.1 serta menggunakan WPF Web Application. Software Alternatech saat ini memiliki 7 form atau window, dan setiap role akan memiliki akses pada window-window yang berbeda. Aplikasi ini sendiri terbagi menjadi 1 halaman admin, 4 window customer, dan juga 1 window login. Software juga membutuhkan koneki dengan server MySql.

## 4.1 Koneksi Server

Segmen Program 4.1 berfungsi untuk membuka koneksi pada server Mysql. Koneksi ini membutuhkan data source, user ID dan juga password untuk dapat terhubung pada server. Selain itu, terdapat juga prosedur *Connect* yang berfungsi untuk membuka koneksi.

**Segment program 4.1 Connection**

1. class Connection
2. {
3. public MySqlConnection conn;
4. public void Connect()
5. {
6. conn= new MySqlConnection("server=localhost;user=root; database=sdp\_db;port=3306;password=");
7. conn.Open();
8. conn.Close();
9. }
10. }

## 4.2 App Page

App page ini berfungsi untuk login (berdasarkan role). Dari window App, user dapat beralih ke window admin, window homepage(user).Window ini memiliki 1 button serta 1 link yang digunakan untuk register , gambar dibawah ini digunakan Ketika user role login sebagai admin atau sebagain user dan link route untuk menuju register

**Segment program 4.2 Login di App Page**

1. if (email.Length > 0 && password.Length > 0)
2. {
3. if (email.ToLower() == "admin" && password.ToLower() == "admin")
4. {
5. var admin = new Admin();
6. this.NavigationService.Navigate(admin);
7. }
8. else
9. {
10. conn.conn.Open();
11. cmd = new MySqlCommand();
12. cmd = new MySqlCommand($"SELECT \* FROM CUSTOMER WHERE EMAIL = '{email}'", conn.conn);
13. MySqlDataReader reader = cmd.ExecuteReader();
14. bool isFound = false;
15. while (reader.Read())
16. {
17. if (reader[3].ToString() == email)
18. {
19. isFound = true;
20. if (reader[5].ToString() == password)
21. {
22. Int id= Convert.ToInt32 (reader[0].ToString());
23. var menu = new MainMenu(id);
24. this.NavigationService.Navigate(menu);
25. break;
26. }
27. else
28. {
29. MessageBox.Show("password salah");

**Lanjutan Segment program** **4.2 Login di App Page**

1. }
2. }
3. }
4. conn.conn.Close();
5. if (!isFound)
6. {
7. MessageBox.Show("User tidak ditemukan");
8. }
9. }
10. }
11. else
12. {
13. MessageBox.Show("data tidak boleh kosong");
14. }

**Segment program 4.2 Route Register di App Page**

1. private void lbregister\_PreviewMouseDown(object sender, MouseButtonEventArgs e)
2. {
3. var reg = new register();
4. this.NavigationService.Navigate(reg);
5. }

# 4.3 Register Page

Register page ini berfungsi untuk mendaftarkan akun user.page ini memiliki 1 button untuk register dan 1 page link untuk menuju login.

**Segment program 4.3 Register**

1. if (email.Length > 0 && username.Length > 0 && pass.Length > 0 && passconf.Length > 0 && nomor.Length > 0)
2. {
3. if(pass == passconf)
4. {
5. try
6. {
7. conn.conn.Open();
8. MySqlCommand cmd = new MySqlCommand($"INSERT INTO CUSTOMER(USERNAME, NAMA\_CUST, EMAIL,NO\_TELP,PASSWORD, SALDO) VALUES('{username}', '{name}', '{email}', '{nomor}', '{pass}', 0)", conn.conn);

**Lanjutan Segment program 4.4 Button Load Laporan Transaksi**

1. cmd.ExecuteNonQuery();
2. conn.conn.Close();
3. }
4. catch(Exception ex)
5. {
6. MessageBox.Show(ex.Message);
7. conn.conn.Close();
8. }
9. }
10. else
11. {
12. MessageBox.Show("Konfirmasi password tidak sesuai");
13. }
14. }
15. else
16. {
17. MessageBox.Show("Data tidak boleh kosong");
18. }

**Segment program 4.3 Route Login**

1. Private void lblogin\_PreviewMouseDown(objectsender, MouseButtonEventArgs e)
2. {
3. var page1 = new Page1();
4. this.NavigationService.Navigate(page1);
5. }

# 4.4 Admin Page

Admin page ini berfungsi untuk CRUD barang , update request saldo user serta dapat meliah report user dan report transaksi. Admin page ini memiliki 5 button dan 1 button untuk logout, button barang digunakan untuk memunculkan data list barang, lalu button user digunakan untuk memunculkan data list user , lalu button laporan transaksi digunakan untuk memunculkan report laporan transaksi, lalu button konfirmasi top up untuk memunculkan list top up yang masih pending ,lalu button load digunakan untuk memunculkan user yang dimana terdapat combo box yang digunakan untuk memunculkan sesuai user yang dipilih Combo Box serta bisa di filter sesuai kegunaannya dengan memilih 4 button yaitu Barang, User, Laporan Transaksi, Konfirmasi Top up.

**Segment program 4.4 Button Load Barang**

1. conn.conn.Open();
2. cmd = new MySqlCommand();
3. cmd = new MySqlCommand($"SELECT BARANG.ID ,BARANG.NAMA\_BARANG AS \"NAMA BARANG\",MERK.NAMA\_MERK ,KATEGORI.NAMA\_KAT,STOK,HARGA FROM BARANG,MERK,KATEGORI WHERE BARANG.MERK = MERK.KODE\_MERK AND BARANG.KATEGORI = KATEGORI.KODE\_KAT ORDER BY id ASC", conn.conn);
4. MySqlDataAdapter sda = new MySqlDataAdapter(cmd);
5. dt = new DataTable();
6. sda.Fill(dt);
7. dgvUser.ItemsSource = dt.DefaultView;
8. conn.conn.Close();

**Segment program 4.4 Button Load User**

1. conn.conn.Open();
2. cmd = new MySqlCommand();
3. cmd = new MySqlCommand($"SELECT CUSTOMER.ID ,CUSTOMER.NAMA\_CUST AS \"NAMA CUSTOMER\",CUSTOMER.SALDO AS \"SALDO\" FROM CUSTOMER ORDER BY id ASC", conn.conn);
4. MySqlDataAdapter sda = new MySqlDataAdapter(cmd);
5. dt = new DataTable();
6. sda.Fill(dt);
7. dgvcustomer.ItemsSource = dt.DefaultView;
8. conn.conn.Close();

pada button load laporan transaksi terdapat history transaksi user, serta admin dapat melihat tiap history user dengan memilih user di combo box lalu menekan button load untuk melihat history user tersebut.

**Segment program 4.4 Button Load Laporan Transaksi**

1. int sub = 0;
2. conn.conn.Open();
3. cmd = new MySqlCommand();
4. cmd = new MySqlCommand($"SELECT BARANG.NAMA\_BARANG AS \"NAMA BARANG\",H\_BELI.NOMOR\_NOTA AS \"NOMOR NOTA\",D\_BELI.jumlah,D\_BELI.SUBTOTAL as \"Subtotal\" from Barang,H\_BELI,D\_BELI,Customer where Customer.id={getiduser} and H\_BELI.ID\_CUSTOMER=CUSTOMER.ID and D\_BELI.NOMOR\_NOTA=H\_BELI.NOMOR\_NOTA and BARANG.ID=D\_BELI.ID\_BARANG ORDER BY BARANG.NAMA\_BARANG", conn.conn);
5. MySqlDataAdapter sda = new MySqlDataAdapter(cmd);
6. dt = new DataTable();

**Lanjutan Segment program 4.4 Button Load Laporan Transaksi**

1. sda.Fill(dt);
2. dgvLaporan.ItemsSource = dt.DefaultView;
3. conn.conn.Close();
4. conn.conn.Open();
5. cmd = new MySqlCommand($"SELECT D\_BELI.SUBTOTAL from Barang,H\_BELI,D\_BELI,Customer where Customer.id= {getiduser} and H\_BELI.ID\_CUSTOMER=CUSTOMER.ID and D\_BELI.NOMOR\_NOTA=H\_BELI.NOMOR\_NOTA and BARANG.ID=D\_BELI.ID\_BARANG ORDER BY BARANG.NAMA\_BARANG", conn.conn);
6. MySqlDataReader reader = cmd.ExecuteReader();
7. while(reader.Read())
8. {
9. sub += Convert.ToInt32(reader[0].ToString());
10. }
11. customerLb.Content = "Name: " + (cblistcustomer.SelectedItem.ToString().Split('-'))[1];
12. totalLb.Content = "Total: " + sub.ToString();
13. conn.conn.Close();

Lalu ketika admin ingin melakukan penambahan data, admin harus memasukkan nama barang, kategori, merek, harga barang dan stock barang dan untuk mengafifkan mode ini user harus menekan tombol load barang terlebih dahulu . Ketika penambahan berhasil dimasukkan, akan muncul notifikasi berhasil. Ketika penambahan gagal, akan terdapat notifikasi juga yang muncul memberitahukan bahwa proses penambahan barang tersebut gagal.

**Segment program 4.4 Insert Barang**

1. if (text\_nmbarang.Text.Length != 0 || text\_harga.Text.Length != 0 || text\_stok.Text.Length != 0)
2. {
3. if (combo\_kategori.SelectedIndex != -1 || combo\_merk.SelectedIndex != -1)
4. {
5. getkodekat();

**Lanjutan Segment program 4.4 insert Barang**

1. getkodemerk();
2. conn.conn.Open();
3. using(MySqlTransaction trans = conn.conn.BeginTransaction())
4. {
5. try
6. {
7. MySqlCommand cmd = new MySqlCommand($"insert into BARANG(NAMA\_BARANG, MERK, KATEGORI,STOK,HARGA) values('{text\_nmbarang.Text}','{kodemerk}','{kodekat}',{Convert.ToInt32(text\_stok.Text)},{Convert.ToInt32(text\_harga.Text)})", conn.conn);
8. cmd.ExecuteNonQuery();
9. trans.Commit();
10. conn.conn.Close();
11. MessageBox.Show("Insert berhasil");
12. }
13. catch (Exception ex)
14. {
15. trans.Rollback();
16. conn.conn.Close();
17. MessageBox.Show(ex.Message);
18. }
19. }
20. conn.conn.Close();
21. loadBarang();
22. normalmode();
23. }
24. else
25. {
26. MessageBox.Show("Pilih combobox terlebih dahulu");
27. }
28. }
29. else
30. {
31. MessageBox.Show("isi textbox terlebih dahulu");
32. }

Lalu Ketika admin ini menghapus salah satu barang harus menekan salah satu dari datagrid terlebih dahulu lalu data tersebut akan tertampil di textbox lalu admin bisa menghapus barang tersebut.

**Segment program 4.4 Delete Barang**

1. conn.conn.Open();
2. using(MySqlTransaction trans = conn.conn.BeginTransaction())
3. {
4. try
5. {
6. MySqlCommand cmd = new MySqlCommand($"delete from barang where ID = {index+1}", conn.conn);
7. cmd.ExecuteNonQuery();
8. trans.Commit();
9. conn.conn.Close();
10. MessageBox.Show("delete berhasil");
11. }
12. catch (Exception ex)
13. {
14. trans.Rollback();
15. conn.conn.Close();
16. MessageBox.Show(ex.Message);
17. }
18. }
19. conn.conn.Close();
20. loadBarang();
21. normalmode();

lalu ketika admin ingin memperbarui isi data barang sama dengan cara untuk menghapus barang yang terdapat pada list , hanya menekan barang yang dipilih serta lalu isi 5 combo box yang mau diubah yang terdiri dari nama barang, stock, harga , kategori, merek.

**Segment program 4.4 Update Barang**

1. if (text\_nmbarang.Text.Length != 0 || text\_harga.Text.Length != 0 || text\_stok.Text.Length != 0)
2. {
3. if (combo\_kategori.SelectedIndex != -1 || combo\_merk.SelectedIndex != -1)
4. {
5. getkodekat();
6. getkodemerk();
7. conn.conn.Open();
8. using (MySqlTransaction trans = conn.conn.BeginTransaction())
9. {
10. try
11. {

**Lanjutan Segment program 4.4 Delete Barang**

1. MySqlCommand cmd =new MySqlCommand($"Update BARANG SET
2. NAMA\_BARANG = '{ text\_nmbarang.Text }',MERK = '{ kodemerk
3. }',kategori='{kodekat}',STOK={Convert.ToInt32
4. (text\_stok.Text)},HARGA = {Convert.ToInt32(text\_harga.Text)}
5. Where id = " + (index+1) ,conn.conn);
6. cmd.ExecuteNonQuery();
7. trans.Commit();
8. conn.conn.Close();
9. MessageBox.Show("Update berhasil");
10. }
11. catch (Exception ex)
12. {
13. trans.Rollback();
14. conn.conn.Close();
15. MessageBox.Show(ex.Message);
16. }
17. }
18. conn.conn.Close();
19. loadBarang();
20. normalmode();
21. }
22. else
23. {
24. MessageBox.Show("Pilih combobox terlebih dahulu");
25. }
26. }
27. else
28. {
29. MessageBox.Show("isi textbox terlebih dahulu");
30. }

pada fitur konfirmasi saldo, pertama program tersebut menampilkan user request untuk mengisi saldo,

**Segment program 4.4 Button Load Konfirmasi Top Up**

1. conn.conn.Open();
2. cmd = new MySqlCommand();
3. cmd = new MySqlCommand($"SELECT REQ\_SALDO.ID AS \"ID\",CUSTOMER.NAMA\_CUST AS \"NAMA CUSTOMER\",REQ\_SALDO.saldoreq AS \"SALDO\" FROM REQ\_SALDO,CUSTOMER WHERE CUSTOMER.ID = REQ\_SALDO.ID\_CUST AND REQ\_SALDO.KONFIRMASI = 'pending' ORDER BY REQ\_SALDO.id ASC", conn.conn);
4. MySqlDataAdapter sda = new MySqlDataAdapter(cmd);
5. dt = new DataTable();
6. sda.Fill(dt);
7. dgvkonfirsaldo.ItemsSource = dt.DefaultView;
8. conn.conn.Close();

Lalu pada fitur konfirmasi saldo, admin dapat menyetujui request saldo atau menolak request saldo user, terlebidahulu admin harus menekan salah satu user yang terdapat pada datagridview, lalu terdapat 2 tombol terima dan tolak.saat admin menekan tombol terima maka saldo akan langsung masuk ke saldo user yang sudah di konfirmasi dan saat admin menekan tombol tolak makan request tersebut dibatalkan.

**Segment program 4.4 menyetejui request saldo**

1. if (tbnamacust.Text.Length != 0 || tbsaldo.Text.Length != 0)
2. {
3. conn.conn.Open();
4. using (MySqlTransaction trans = conn.conn.BeginTransaction())
5. {
6. try
7. {
8. MySqlCommand cmd = new MySqlCommand($"Update CUSTOMER SET SALDO = {Convert.ToInt32(tbsaldo.Text)} WHERE NAMA\_CUST = '{tbnamacust.Text}'" , conn.conn);
9. MySqlCommand cmd2 = new MySqlCommand($"Update REQ\_SALDO SET KONFIRMASI = 'accepted' WHERE ID = {dt.Rows[dgvkonfirsaldo.SelectedIndex][0]}", conn.conn);
10. cmd.ExecuteNonQuery();
11. cmd2.ExecuteNonQuery();
12. trans.Commit();
13. conn.conn.Close();
14. MessageBox.Show("Top Up Berhasil Diterima");
15. LoadSaldo();
16. }
17. catch (Exception ex)
18. {
19. trans.Rollback();
20. conn.conn.Close();
21. MessageBox.Show(ex.Message);
22. }
23. }
24. }
25. else
26. {
27. MessageBox.Show("pilihan tidak boleh kosong");
28. }

**Segment program 4.4 menolak request saldo**

1. if (tbnamacust.Text.Length != 0 || tbsaldo.Text.Length != 0
2. {
3. conn.conn.Open();
4. using (MySqlTransaction trans = conn.conn.BeginTransaction())
5. {
6. try
7. {
8. MySqlCommand cmd = new MySqlCommand($"Update REQ\_SALDO SET KONFIRMASI = 'rejected' WHERE ID = {dt.Rows[dgvkonfirsaldo.SelectedIndex][0]}", conn.conn);
9. cmd.ExecuteNonQuery();
10. trans.Commit();
11. conn.conn.Close();
12. MessageBox.Show("Top Up Berhasil Ditolak");
13. LoadSaldo();
14. }
15. catch (Exception ex)
16. {
17. trans.Rollback();
18. conn.conn.Close();
19. MessageBox.Show(ex.Message);
20. }
21. tbnamacust.Clear();
22. tbsaldo.Clear();
23. }
24. }
25. else
26. {
27. MessageBox.Show("pilihan tidak boleh kosong");
28. }

# 4.5 User Page

Pada page ini , user dapat mengakses nya Ketika sudah login dan diarahkan ke page home user, di page ini terdapat list barang yang terjual beserta harga dan terdapat route unruk melihat profile untuk melihat konfigurasi user, cart untuk melihat barang yang masih ingin di beli (on going) , home untuk Kembali ke halaman home usr , tokoku untuk halaman user bagian penjualan barang user itu sendiri.

**Segment program 4.5 load barang pada user**

1. barangList.Clear();
2. indexingItem = 0;
3. if (kategori[kat] == "All")
4. {
5. conn.conn.Open();
6. if(filtercb.SelectedIndex == 0)
7. {
8. if (tbsearch.Text.Length > 0)
9. {
10. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE UPPER(NAMA\_BARANG) like '%{tbsearch.Text.ToUpper()}%' ORDER BY ID DESC", conn.conn);
11. MySqlDataReader reader = cmd.ExecuteReader();
12. while (reader.Read())
13. {
14. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
15. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
16. }
17. }
18. else
19. {
20. MySqlCommand cmd = new MySqlCommand("SELECT \* FROM BARANG ORDER BY ID DESC", conn.conn);
21. MySqlDataReader reader = cmd.ExecuteReader();
22. int ctr = 1;
23. while (reader.Read())
24. {
25. ctr++;
26. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
27. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
28. }}

**Lanjutan Segment program 4.5 load barang pada user**

1. conn.conn.Close();
3. }else if (filtercb.SelectedIndex == 1)
4. {
5. if (tbsearch.Text.Length > 0)
6. {
7. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE UPPER(NAMA\_BARANG)
8. like '%{tbsearch.Text.ToUpper()}%' ORDER BY harga DESC", conn.conn);
9. MySqlDataReader reader = cmd.ExecuteReader();
10. while (reader.Read())
11. {
12. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
13. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
14. }
15. }
16. else
17. {
18. MySqlCommand cmd = new MySqlCommand("SELECT \* FROM BARANG ORDER BY harga DESC", conn.conn);
19. MySqlDataReader reader = cmd.ExecuteReader();
20. while (reader.Read())
21. {
22. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
23. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
24. }
25. }
26. conn.conn.Close();
27. }else if (filtercb.SelectedIndex == 2)
28. {

**Lanjutan Segment program 4.5 load barang pada user**

1. if (tbsearch.Text.Length > 0)
2. {
3. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE UPPER(NAMA\_BARANG) like '%{tbsearch.Text.ToUpper()}%' ORDER BY harga ASC", conn.conn);
4. MySqlDataReader reader = cmd.ExecuteReader();
5. while (reader.Read())
6. {
7. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
8. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
9. }
10. }
11. Else
12. {
13. MySqlCommand cmd = new MySqlCommand("SELECT \* FROM BARANG ORDER BY harga ASC", conn.conn);
14. MySqlDataReader reader = cmd.ExecuteReader();
15. while (reader.Read())
16. {
17. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
18. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
19. }
20. }
21. conn.conn.Close();
22. }
23. }
24. else
25. {
26. if(filtercb.SelectedIndex == 0)
27. {
28. if (tbsearch.Text.Length > 0)

**Lanjutan Segment program 4.5 load barang pada user**

1. {
2. conn.conn.Open();
3. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE KATEGORI='{kategori[kat]}' AND UPPER(NAMA\_BARANG) like '%{tbsearch.Text.ToUpper()}%' ORDER BY ID DESC", conn.conn);
4. MySqlDataReader reader = cmd.ExecuteReader();
5. while (reader.Read())
6. {
7. byte[] binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
8. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
9. }
10. }
11. else
12. {
13. conn.conn.Open();
14. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE KATEGORI='{kategori[kat]}' ORDER BY ID DESC", conn.conn);
15. MySqlDataReader reader = cmd.ExecuteReader();
16. while (reader.Read())
17. {
18. byte[] binaryData =
19. Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(re
20. ader[6])));
21. barangList.Add(new
22. Barang(Convert.ToInt32(reader[0].ToString()),
23. reader[1].ToString(), reader[2].ToString(),
24. reader[3].ToString(), Convert.ToInt32(reader[4].ToString()),
25. Convert.ToInt32(reader[5].ToString()), binaryData));
26. }
27. }
29. conn.conn.Close();
31. }else if (filtercb.SelectedIndex == 1)
32. {
33. if (tbsearch.Text.Length > 0)
34. {
35. conn.conn.Open();

**Lanjutan Segment program 4.5 load barang pada user**

1. MySqlCommand cmd = new
2. MySqlCommand($"SELECT \* FROM BARANG WHERE
3. KATEGORI='{kategori[kat]}' AND UPPER(NAMA\_BARANG) like
4. '%{tbsearch.Text.ToUpper()}%' ORDER BY harga DESC",
5. conn.conn);
6. MySqlDataReader reader =
7. cmd.ExecuteReader();
8. while (reader.Read())
9. {
10. byte[] binaryData =
11. Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(re
12. ader[6])));
13. barangList.Add(new
14. Barang(Convert.ToInt32(reader[0].ToString()),
15. reader[1].ToString(), reader[2].ToString(),
16. reader[3].ToString(), Convert.ToInt32(reader[4].ToString()),
17. Convert.ToInt32(reader[5].ToString()), binaryData));
18. }
19. }
20. else
21. {
22. conn.conn.Open();
23. MySqlCommand cmd = new
24. MySqlCommand($"SELECT \* FROM BARANG WHERE
25. KATEGORI='{kategori[kat]}' ORDER BY harga DESC", conn.conn);
26. MySqlDataReader reader =
27. cmd.ExecuteReader();
28. while (reader.Read())
29. {
30. byte[] binaryData =
31. Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(re
32. ader[6])));
33. barangList.Add(new
34. Barang(Convert.ToInt32(reader[0].ToString()),
35. reader[1].ToString(), reader[2].ToString(),
36. reader[3].ToString(), Convert.ToInt32(reader[4].ToString()),
37. Convert.ToInt32(reader[5].ToString()), binaryData));
38. }
39. }
40. conn.conn.Close();
42. }else if (filtercb.SelectedIndex == 2)
43. {
44. if (tbsearch.Text.Length > 0)
45. {
46. conn.conn.Open();

**Lanjutan Segment program 4.5 load barang pada user**

1. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE KATEGORI='{kategori[kat]}' AND UPPER(NAMA\_BARANG) like '%{tbsearch.Text.ToUpper()}%' ORDER BY harga ASC", conn.conn);
2. MySqlDataReader reader = cmd.ExecuteReader();
3. while (reader.Read())
4. {
5. byte[] binaryData =
6. Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(re
7. ader[6])));
8. barangList.Add(new
9. Barang(Convert.ToInt32(reader[0].ToString()),
10. reader[1].ToString(), reader[2].ToString(),
11. reader[3].ToString(), Convert.ToInt32(reader[4].ToString()),
12. Convert.ToInt32(reader[5].ToString()), binaryData));
13. }
14. }
15. else
16. {
17. conn.conn.Open();
18. MySqlCommand cmd = new MySqlCommand($"SELECT \* FROM BARANG WHERE KATEGORI='{kategori[kat]}' ORDER BY harga ASC", conn.conn);
19. MySqlDataReader reader = cmd.ExecuteReader();
20. while (reader.Read())
21. {
22. byte[]binaryData = Convert.FromBase64String(Encoding.UTF8.GetString((byte[])(reader[6])));
23. barangList.Add(new Barang(Convert.ToInt32(reader[0].ToString()), reader[1].ToString(), reader[2].ToString(), reader[3].ToString(), Convert.ToInt32(reader[4].ToString()), Convert.ToInt32(reader[5].ToString()), binaryData));
24. }
25. }
26. conn.conn.Close();
27. }
28. }
29. loadBarangScreen();
30. nextbtn.IsEnabled = true;
31. if(indexingItem == Math.Ceiling(Convert.ToDouble(barangList.Count() / 8)))
32. {
33. nextbtn.IsEnabled = false;
34. }

**Lanjutan Segment program 4.5 load barang pada user**

1. if (indexingItem == 0)
2. {
3. prevButton.IsEnabled = false;
4. }

Page home user pada bagian barang Ketika di tekan dua kali maka akan dialihkan ke detail barang sesuai dengan barang yang ditekan, lalu terdapat 2 tombol yaitu add to cart untuk menambahkan barang ke cart serta tombol back to home untuk kembali ke home user. Di page ini user dapat mengetahui informasi barang seperti stock barang.

**Segment program 4.5 menambahkan barang ke cart**

1. string temp = lbldetail.Content.ToString();
2. conn.conn.Open();
3. int idbrg = -1;
4. MySqlCommand cmd = new MySqlCommand($"SELECT ID FROM BARANG WHERE NAMA\_BARANG = '{temp}'", conn.conn);
5. MySqlDataReader reader = cmd.ExecuteReader();
6. while(reader.Read())
7. {
8. idbrg = Convert.ToInt32(reader[0].ToString());
9. }
10. conn.conn.Close();
11. conn.conn.Open();
12. cmd = new MySqlCommand($"SELECT ID\_BARANG FROM CART WHERE USERNAME='{user.username}'", conn.conn);
13. reader = cmd.ExecuteReader();
14. bool isFound = false;
15. while(reader.Read())
16. {
17. if(idbrg == Convert.ToInt32(reader[0].ToString()))
18. {
19. isFound = true;
20. }
21. }
22. conn.conn.Close();

**Lanjutan Segment program 4.5 menambahkan barang ke cart**

1. if(!isFound)
2. {
3. conn.conn.Open();
4. cmd = new MySqlCommand($"INSERT INTO CART VALUES('{user.username}', {idbrg})", conn.conn);
5. cmd.ExecuteNonQuery();
6. conn.conn.Close();
7. idbrg = -1;
9. }
10. else
11. {
12. MessageBox.Show("Barang sudah ada di dalam cart");
13. idbrg = -1;
14. isFound = false;
15. }

Saat user ingin checkout barang, user harus ke cart terlebih dahulu. Lalu di page cart terdapat detail barang yang ingin di checkout dan jumlah harga semua barang. Untuk melakukan pembayaran maka user harus menekan tombol checkout,lalu program akan melakukan pengecekan apakah saldo user mencukupi atau tidak.

**Segment program 4.5 checkout barang**

1. conn.conn.Open();
2. MySqlCommand cmd = new MySqlCommand($"SELECT BARANG.ID, BARANG.NAMA\_BARANG, BARANG.HARGA FROM BARANG, CART WHERE CART.ID\_BARANG = BARANG.ID AND CART.USERNAME = '{user.username}'", conn.conn);
3. int total = 0;
4. MySqlDataReader reader = cmd.ExecuteReader();
5. while (reader.Read())
6. {
7. total += Convert.ToInt32(reader[2].ToString());
8. }
9. conn.conn.Close();
10. if (total <= user.saldo)
11. {
12. List<dBeli> dbeli = new List<dBeli>();
13. string nota = generateNota();

**Lanjutan Segment program 4.5 menambahkan barang ke cart**

1. conn.conn.Open();
2. cmd = new MySqlCommand($"INSERT INTO H\_BELI VALUES(\"{nota}\", {user.id}, {total}, \"{DateTime.Now.ToString("yyyy-MM-dd")}\")",conn.conn);
3. cmd.ExecuteNonQuery();
4. cmd = new MySqlCommand($"UPDATE CUSTOMER SET SALDO = {user.saldo - total} WHERE ID = {user.id}", conn.conn);
5. cmd.ExecuteNonQuery();
6. conn.conn.Close();
7. List<int> listidbarang = new List<int>();
8. conn.conn.Open();
9. cmd = new MySqlCommand($"SELECT BARANG.ID, BARANG.NAMA\_BARANG, BARANG.HARGA FROM BARANG, CART WHERE CART.ID\_BARANG = BARANG.ID AND CART.USERNAME = '{user.username}'", conn.conn);
10. reader = cmd.ExecuteReader();
11. while(reader.Read())
12. {
13. dbeli.Add(new dBeli(Convert.ToInt32(reader[0].ToString()),reader[1].ToString(), Convert.ToInt32(reader[2].ToString())));
14. listidbarang.Add(Convert.ToInt32(reader[0].ToString()));
15. }
16. conn.conn.Close();
17. conn.conn.Open();
18. for (int i = 0; i < dbeli.Count; i++)
19. {
20. cmd = new MySqlCommand($"UPDATE BARANG SET STOK = STOK-1 WHERE BARANG.ID = {listidbarang[i]}", conn.conn);
21. cmd.ExecuteNonQuery();
22. }
23. conn.conn.Close();
24. listidbarang.Clear();
25. conn.conn.Open();
26. foreach(dBeli data in dbeli)
27. {
28. cmd = new MySqlCommand($"INSERT INTO D\_BELI VALUES(\"{nota}\", {Convert.ToInt32(data.idBarang)}, 1, {Convert.ToInt32(data.hargaBarang)})", conn.conn);
29. cmd.ExecuteNonQuery();
30. }
31. conn.conn.Close();

**Lanjutan segment program 4.5 checkout barang**

1. conn.conn.Open();
2. cmd = new MySqlCommand($"DELETE FROM CART WHERE USERNAME = \"{user.username}\"", conn.conn);
3. cmd.ExecuteNonQuery();
4. conn.conn.Close();
5. conn.conn.Open();
6. cmd = new MySqlCommand($"DELETE FROM BARANG WHERE STOK = 0", conn.conn);
7. cmd.ExecuteNonQuery();
8. conn.conn.Close();
9. MessageBox.Show("Terima kasih sudah berbelanja");
10. loadData();
11. loadCart();
12. var report = new ReportPembelian(user.id, dbeli);
13. this.NavigationService.Navigate(report);
14. }
15. else
16. {
17. MessageBox.Show("Uang tidak cukup");
18. }

ketika user sudah melakukan pembayaran makan akan mendapatkan invoice atau bukti pembayaran.

**Segment program 4.5 pembuatan nota barang**

1. conn.conn.Open();
2. string temp = $"SELECT CONCAT(\"{
3. DateTime.Now.ToString("yyyyMMdd")}\" ,
4. LPAD((COUNT(NOMOR\_NOTA) + 1), 3, '0')) FROM H\_BELI WHERE
5. NOMOR\_NOTA LIKE '%{DateTime.Now.ToString("yyyyMMdd")}%'";
6. MySqlCommand cmd = new MySqlCommand(temp,
7. conn.conn);
8. MySqlDataReader reader = cmd.ExecuteReader();
9. string nota = "";
10. while(reader.Read())
11. {
12. nota = reader[0].ToString();
13. }
14. conn.conn.Close();
15. return nota;

saat user menekan tombol my profile pada menu disamping kanan maka akan mengarahkan user pada halaman profil yang berisi informasi user, segment berikut merupakan bagian untuk melakukan update profil

**Segment program 4.5 update profil pada user**

1. conn.conn.Open();
2. MySqlCommand cmd = new MySqlCommand($"UPDATE customer SET NAMA\_CUST=@name,EMAIL=@email, NO\_TELP=@notelp WHERE ID={iduser}", conn.conn);
3. cmd.Parameters.AddWithValue("@name", name);
4. cmd.Parameters.AddWithValue("@email", email);
5. cmd.Parameters.AddWithValue("@notelp", notelp);
6. cmd.ExecuteNonQuery();
7. conn.conn.Close();
8. loaduser(user.id);
9. MessageBox.Show("Update berhasil");

Pada saat user memilih opsi ganti password maka user akan dibawa menuju halaman ganti password dan konfirmasi password

**Segment program 4.5 ganti password pada user**

1. if(tbupdate\_oldpass.Password == user.password)
2. {
3. if(tbupdate\_newpass.Password == tbupdate\_confirm.Password)
4. {
5. conn.conn.Open();
6. MySqlCommand cmd = new MySqlCommand($"UPDATE customer SET Password=@password WHERE ID={iduser}", conn.conn);
7. cmd.Parameters.AddWithValue("@password", password);
8. cmd.ExecuteNonQuery();
9. conn.conn.Close();
10. loaduser(user.id);
11. MessageBox.Show("Change Password Success");
12. }
13. else
14. {
15. MessageBox.Show("Password baru dan password confirmation tidak sama");
16. }
17. }

**Lanjutan Segment program 4.5 ganti password pada user**

1. else
2. {
3. MessageBox.Show("Password salah");}

Saat user memilih opsi tokoku pada menu di samping kanan maka user akan diarahkan ke halaman toko user yang berisi barang-barang yang ingin dijual oleh user,salah satu fiturnya yaitu menambahkan atau mendaftarkan item baru untuk dijual oleh user yang login dengan menekan tombol add new maka akan terarah kan ke halaman add barang seller

**Segment program 4.5 tambah barang di user seller**

1. string namabarang = tb\_seller\_nama.Text.ToString();
2. int stok = int.Parse(tb\_seller\_stok.Text.ToString());
3. int harga =
4. int.Parse(tb\_seller\_harga.Text.ToString());
5. getid();
7. try
8. {
9. conn.conn.Open();
10. MySqlCommand cmd = new MySqlCommand($"INSERT
11. INTO BARANG(NAMA\_BARANG,MERK,KATEGORI,STOK,HARGA,IMG)
12. VALUES('{namabarang}','{kodemerk}','{kodekat}','{stok}','{ha
13. rga}','iVBORw0KGgoAAAANSUhEUgAAAPoAAAD6BAMAAAB6wkcOAAAAG1BMV
14. EXMzMwAAAB/f39MTEwZGRmysrKZmZlmZmYzMzNjTEbvAAAACXBIWXMAAA7EA
15. AAOxAGVKw4bAAACeklEQVR4nO3WMU/bQBTA8cPh7IwYHKdjIoGUkaq0YmxIp
16. WR0WkpXpxJRR6gQzUigLXzsvnc+F0ckqK09dPj/pDiPy9M9v8udgzEAAAAAA
17. AAAAAAAAAAAAAAAAAAAAADNCI7iUSbvgwfRkyD6nrx5Ji/QvFsdWQxHec3iY
18. fJ1PtiV4PibkNnsu+vPy4vNea17ybuW4HQ8H3TrVu/J5VKq3vmB7R/S3+7mv
19. PahH5nISvR7Ncu7ijLjvY8vdXmPNue1/bpEHbm03jZQPZJJxj5279PexrxtX
20. 92926eL9G/VT4owTPVaLK/N3Ui2Wt3f2Mx9eGfq01l9F8FOOSCxu5PZxUrel
21. q/ed9erBqprq6mda5PRzeM9uF0WdlfztvKzc40XlXuoR1bRdpdxIl22XPWw4
22. 8ajdKV1t9rTRRzr3jguRjJT24N8x8mBeT+R6oeV6tJ8tXXNM9OTzO7LLb5qq
23. nqgtXJ5nV74/VZWj9Jq6y4vlJcdl/utgeqzng9kV62svDTfXZenm62plbejM
24. pKivvpOWTBdl6dFffW6xc1p+ew0NvVnrbjqhtdt/yTPTHN/1mqfODt5DDt+z
25. YsV0A0fpWvyzDTza35ct3qlJT3m7qnjH6e64X83X8nTY77V02Bs6gkfv03Xs
26. vuB6RebSTd82Xw1T1t2PzBBx9RTPVL9XBZV/rbFTi/Oum9+5eilPmf69B+Bv
27. xLsfRSZCW4z80EnDPcys39TVHdzR3k1zwwOjH2pH8xem2CyftI/1o5Vx9ire
28. PjC1fkSD9d8m2We+TRM4p/u3pZJUrP1irN5GZw/m2fneRk08JAHAAAAAAAAA
29. AAAAAAAAAAAAAAAAAD/mV+CPmI6PVs6jQAAAABJRU5ErkJggg==')",
30. conn.conn);
31. cmd.ExecuteNonQuery();
32. conn.conn.Close();
33. }
34. catch (Exception ex)
35. {
36. MessageBox.Show(ex.Message);
37. conn.conn.Close();

**Lanjutan Segment program 4.5 tambah barang di user seller**

1. }
2. conn.conn.Open();
3. cmd = new MySqlCommand("SELECT MAX(ID) FROM
4. BARANG", conn.conn);
5. reader = cmd.ExecuteReader();
6. while (reader.Read())
7. {
8. barangID = int.Parse(reader[0].ToString());
9. }
10. conn.conn.Close();
11. try
12. {
13. conn.conn.Open();
14. MySqlCommand cmd = new MySqlCommand($"INSERT
15. INTO DIGITAL\_SELLER(ID\_USER, ID\_BARANG)
16. VALUES('{userID}','{barangID}')", conn.conn);
17. cmd.ExecuteNonQuery();
18. conn.conn.Close();
19. }
20. catch (Exception ex)
21. {
22. MessageBox.Show(ex.Message);
23. conn.conn.Close();
24. }
25. MessageBox.Show("Berhasil insert");

Saat user menekan tombol delete pada tampilan tokoku dalam keadaan salah satu barang pada datagrid terpilih salah satu

**Segment program 4.5 delete barang di user seller**

1. Try
2. {
3. conn.conn.Open();
4. MySqlCommand cmd = new MySqlCommand($"DELETE
5. FROM BARANG WHERE id = {barangID}", conn.conn);
6. cmd.ExecuteNonQuery();
7. conn.conn.Close();
8. }
9. catch (Exception ex)
10. {
11. MessageBox.Show(ex.Message);
12. conn.conn.Close();
13. }
14. try

**Lanjutan Segment program 4.5 delete barang di user seller**

1. {
2. conn.conn.Open();
3. MySqlCommand cmd = new MySqlCommand($"DELETE FROM DIGITAL\_SELLER WHERE id\_user = {userID} and id\_barang = {barangID}", conn.conn);
4. cmd.ExecuteNonQuery();
5. conn.conn.Close();
6. }
7. catch(Exception ex)
8. {
9. MessageBox.Show(ex.Message);
10. conn.conn.Close();
11. }
12. loaddatagrid();

# BAB V

**PENUTUP**

Pada bab ini akan dijelaskan kesimpulan dari aplikasi pembelian peralatan gaming ALTERNATECH yang telah kami buat. Pada bab ini juga akan disertakan saran – saran untuk software ALTERNATECH. Pada bagian akhir bab disertakan daftar pustaka dan sumber-sumber teori dari buku.

1. **Kesimpulan**

Pembuatan aplikasi Alternatech ini bertujuan membantu proses pembelian peralatan gaming serta penjualan produk digital game. Aplikasi ini dibuat dalam jangka waktu 1 semester (5 – 6 bulan).Berdasarkan hasil perancangan dan pembuatan aplikasi ini, dapat disimpulkan hal-hal sebagai berikut:

* Program ini bertujuan mempermudah pembelian produk di masa pandemic ini.
* Program ini dikembangkan dengan Bahasa C# berbentuk WPF Browser Application dan dibuat pada Visual Studio beserta dengan databasenya MySQL
* Aplikasi kami masih sulit dalam pengisian saldo karena membutuhkan konfirmasi dari admin terlebih dahulu

1. **Saran**

ini akan dijelaskan kesimpulan dari aplikasi pemesanan peralatan gaming ALTERNATECH yang telah kami buat. Pada bab ini juga akan disertakan saran – saran untuk software ALTERNATECH. Pada bagian akhir bab disertakan daftar pustaka dan sumber-sumber teori dari buku.

* Memperindah tampilan aplikasi supaya enak dilihat
* Optimisasi fungsi yang terdapat dalam tiap event supaya bisa memaksimalkan kinerja software pada system.

DAFTAR PUSTAKA

Wikipedia: *Agile Software Development*

<https://en.m.wikipedia.org/wiki/Agile_software_development>

Github: *SSH.NET*

<https://github.com/sshnet/SSH.NET>

Net-information.com, *How To Connect to MySQL Database to C#:*

<http://net-informations.com/q/faq/mysql.html#:~:text=Add%20Reference,Library%20in%20your%20C%23%20project>

Bina Nusantara University. (n.d.). *Teknik Dalam White-box dan Black-box Testing*.

Retrieved from Bina Nusantara University:

<https://socs.binus.ac.id/2020/07/02/teknik-dalam-white-box-dan-black-boxtesting/>

Filipova, O., & Vilão, R. (2018*). Software Development From A to Z*. Apress.

Visual Paradigm. (n.d.). *Requirement Analysis Technique*. Retrieved from Visual Paradigm:

<https://www.visual-paradigm.com/guide/requirementsgathering/requirement-analysis-techniques/>

Visual Paradigm. (n.d.). *What is Activity Diagram?* Retrieved from Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-isactivity-diagram/>

Visual Paradigm. (n.d.). *What is Entity Relationship Diagram (ERD)?* Retrieved from

Visual Paradigm[: https://www.visual-paradigm.com/guide/data-modeling/whatis-entity-relationship-diagram/](https://www.visual-paradigm.com/guide/data-modeling/whatis-entity-)