|  |
| --- |
| 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 |

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.

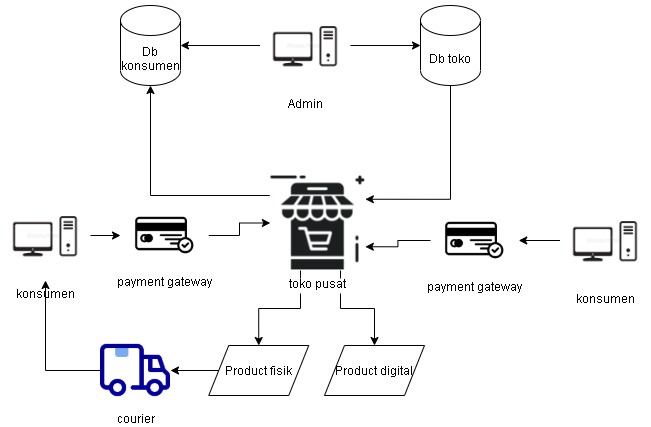
1. **C#**

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

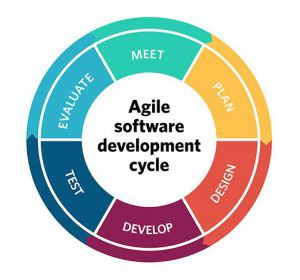
1. Dapat mengecek dan menambah jumlah stok barang / barang baru(Admin)Fitur ini akan tersedia suatu menu barang yang khusus untuk admin dimana admin dapat menambah stok barang yang kurang serta check stock jumlah barang
2. Dapat melihat konfirmasi top up saldo dari customer(admin) dalam Fitur ini admin 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 II

ANALISA SISTEM

Dalam bab ini kami menjelaskan system acuan kami…

System apa yg akan dibuat

1. Fact Finding

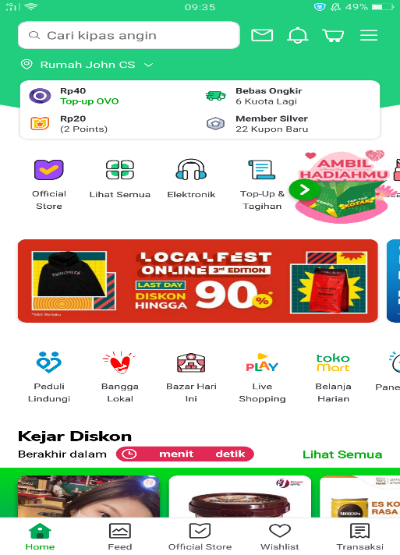
Dalam fact finding, kami menjelaskan seluruh usaha kami dalam menemukan referensi aplikasi yang akan kita kembangkan

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.

**2.1.2 Dokumentasi**

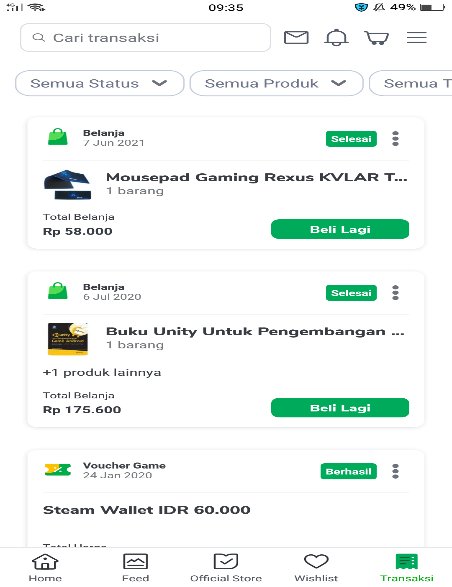
Berikut adalah dokumentasi yang kami temukan dari Tokopedia , Itemku

serta oldshop yang berada di social media lainnya,

Gambar 2.1

Tampilan awal Tokopedia

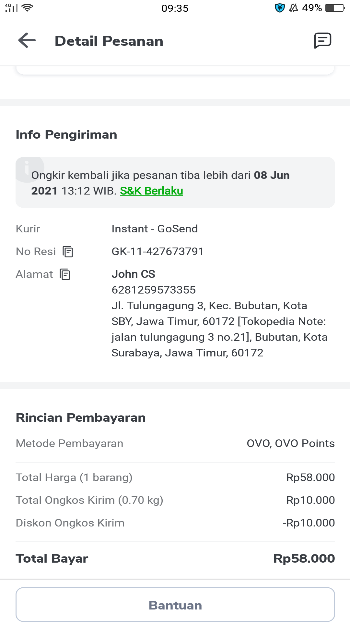
Dari tampilan awal, ditunjukkan saldo user serta kupon atau poin. Dari sini juga kita bisa memilih kategori benda yang ingin kita cari..



Gambar 2.2

Tampilan History User

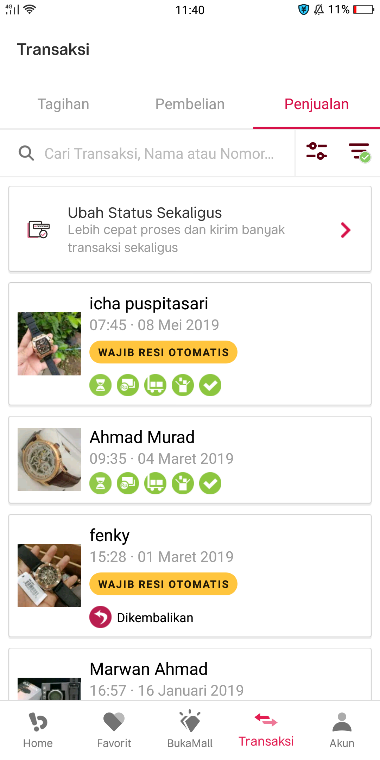
Disini kita bisa melihat riwayat barang yang kita beli serta harga yang tertera pada benda tersebut. Kita bisa melihat rincian pembelian kita setelah menekan salah satu dari produk diatas.



Gambar 2.3

Tampilan detail pesanan

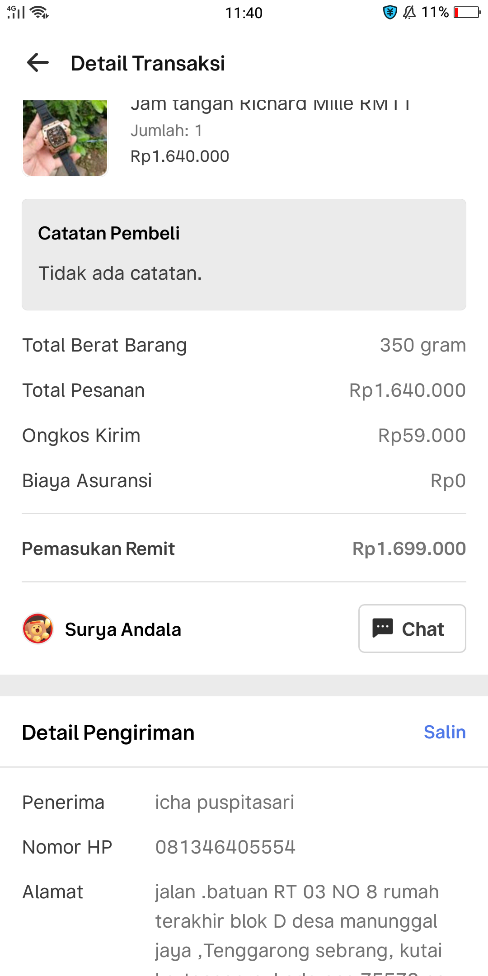
Ini adalah tampilan setelah kita menekan salah satu riwayat pembelian kita, tertera Nomor Resi, harga dan jumlah total, alamat serta nama pembeli.



Gambar 2.4

halaman daftar penjualan barang oldshop

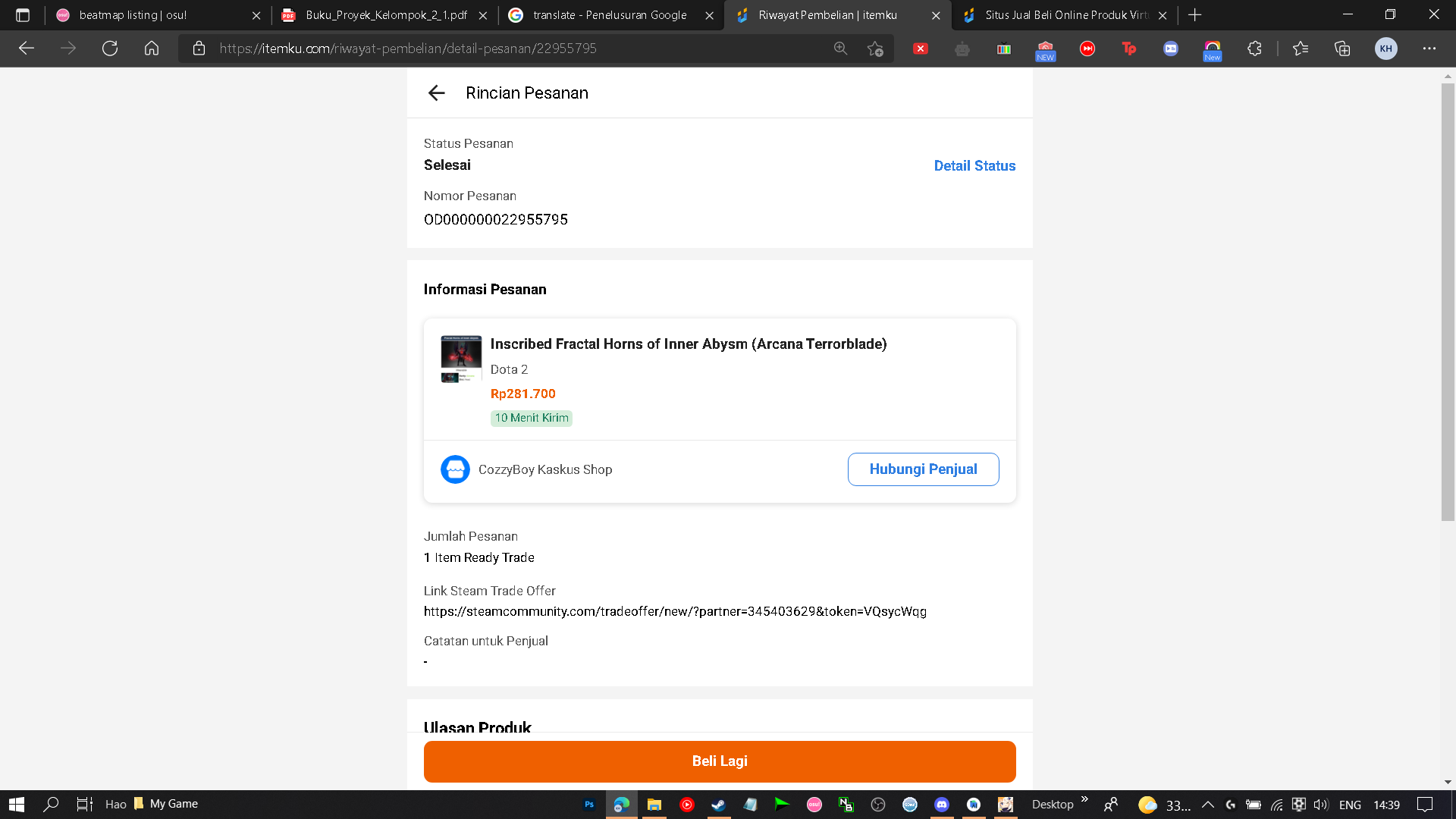
Ini adalah tampilan dari penjual toko untuk melihat daftar penjualanya serta status penjualannya



Gambar 2.5

halaman detail transaksi yg dijual

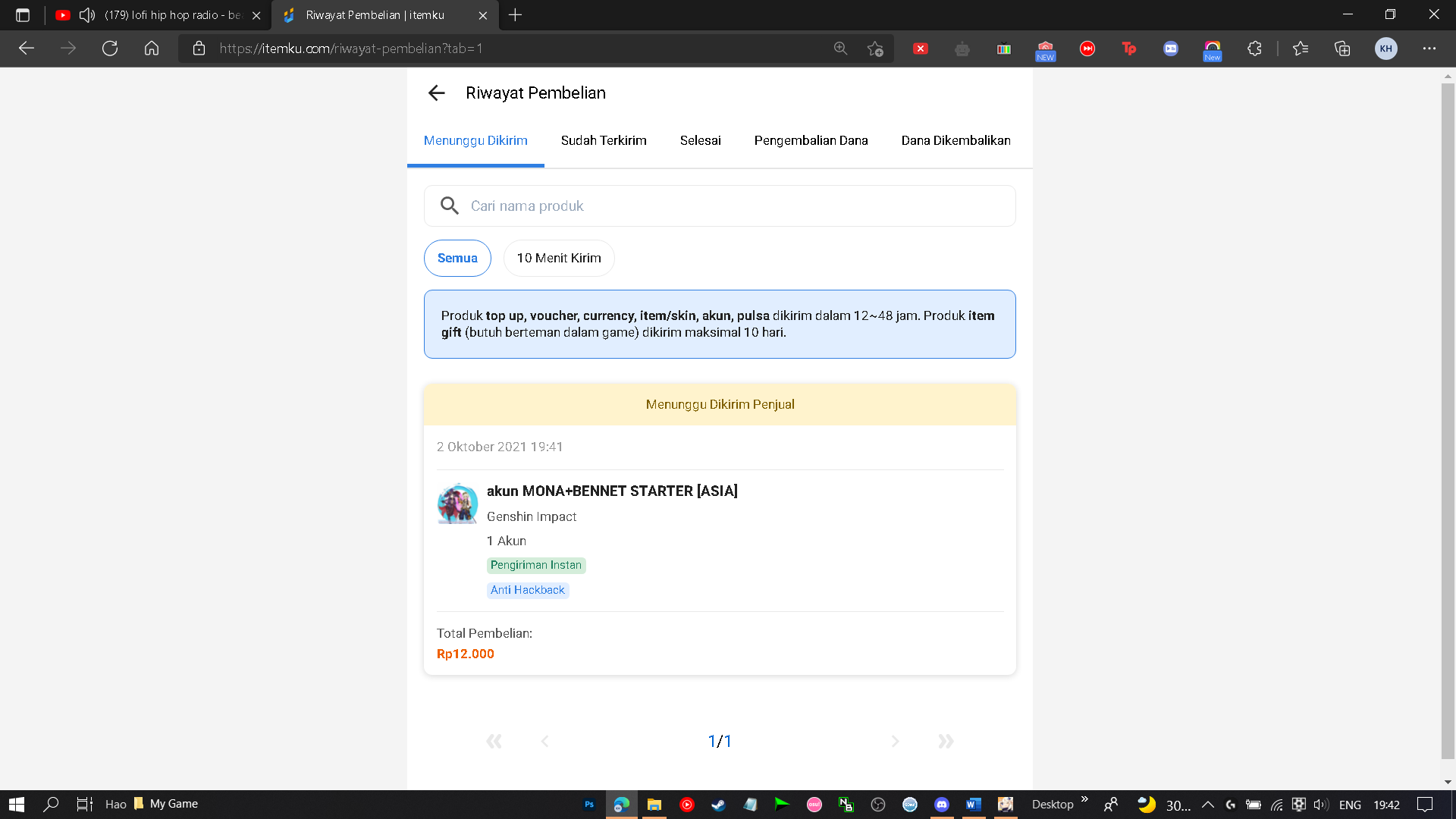
Penjual dapat melihat detail transaksi dari pembeli jika pembeli membeli barang tersebut



Gambar 2.6

halaman Status selesai transaksi

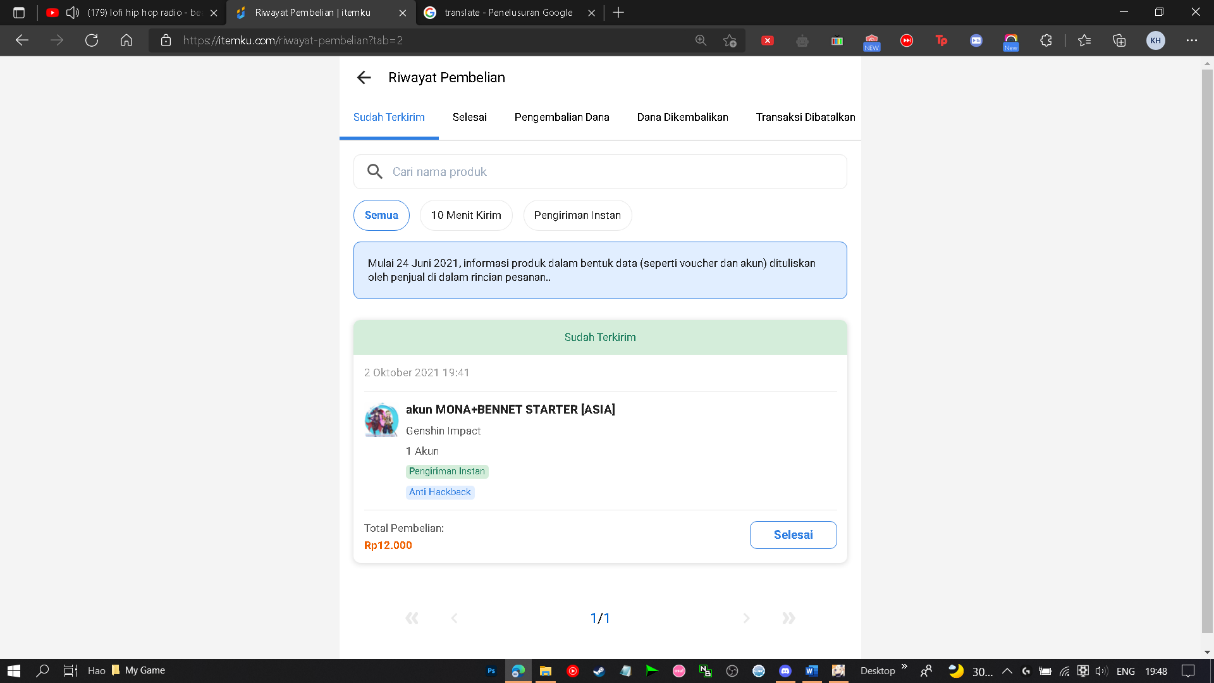
Pembeli dapat melihat status barang yang dibeli saat pembeli sudah menerima item yang diberi oleh seller.



Gambar 2.7

halaman Status Menunggu Transaksi

Di halaman ini pembeli menunggu item/barang yang dikirimkan oleh seller

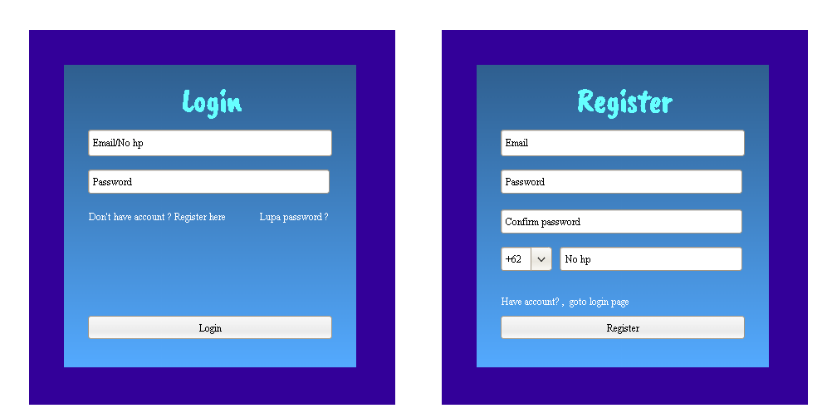


Gambar 2.8

halaman Status Konfirmasi Transaksi

Di halaman ini pembeli dapat mengonfirmasi jika item / barang sudah dikirim

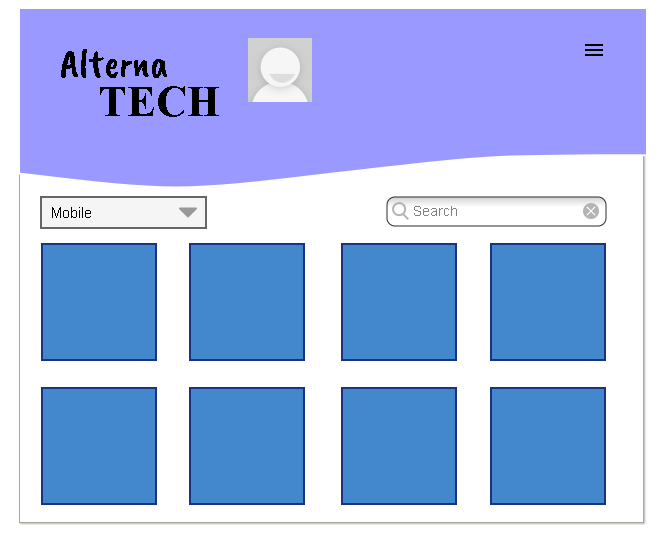
**2.1.3 Prototype**



Gambar 2.9

Login dan Register Page

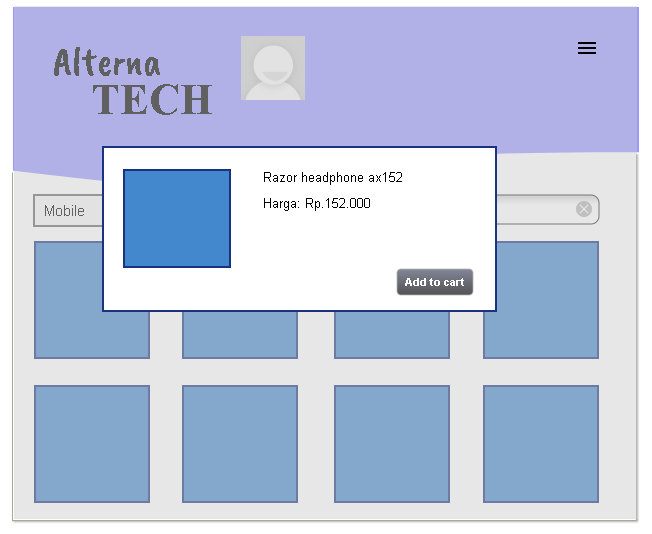
Pada page login, user diminta untuk memasukkan data supaya bisa masuk kedalam homepage. Namun, juka user belum memiliki akun, maka user akan dibawa ke halaman register untuk mengisi data pembuatan akun.



Gambar 2.10

Halaman Home

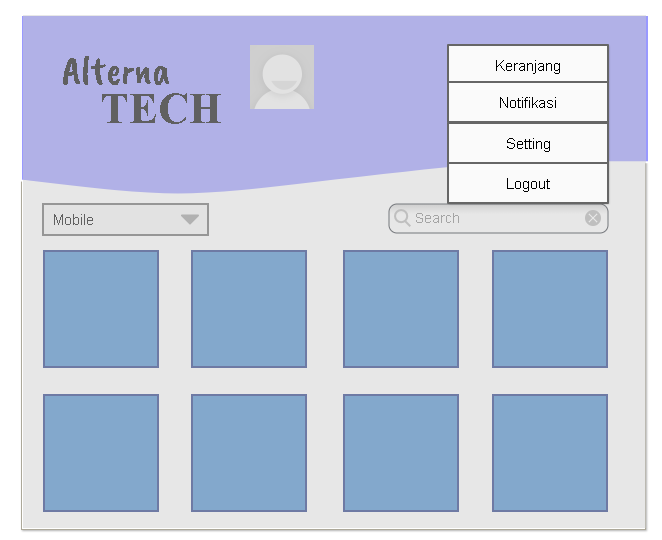
Halaman home berisi perlengkapan gaming yang kita sediakan dalam bentuk grid. Kami jadikan ini sebagai halaman paling depan supaya User bisa dengan mudah melihat produk-produk ketika user baru masuk kedalam aplikasi. Selain itu, ada juga textbox yang berguna untuk melakukan pencarian terhadap suatu barang. 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 2.12

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 2.13

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

* Bahasa

Bahasa yang digunakan pada aplikasi adalah Bahasa inggris

* Pembelian

Pembelian barang di aplikasi kami terdapat 2 jenis

* Produk digital

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

* 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 = >=1gb

* Koneksi internet memadai
  1. **Batasan Aplikasi**

Batasan-batasan dalam sistem aplikasi ELREORS 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

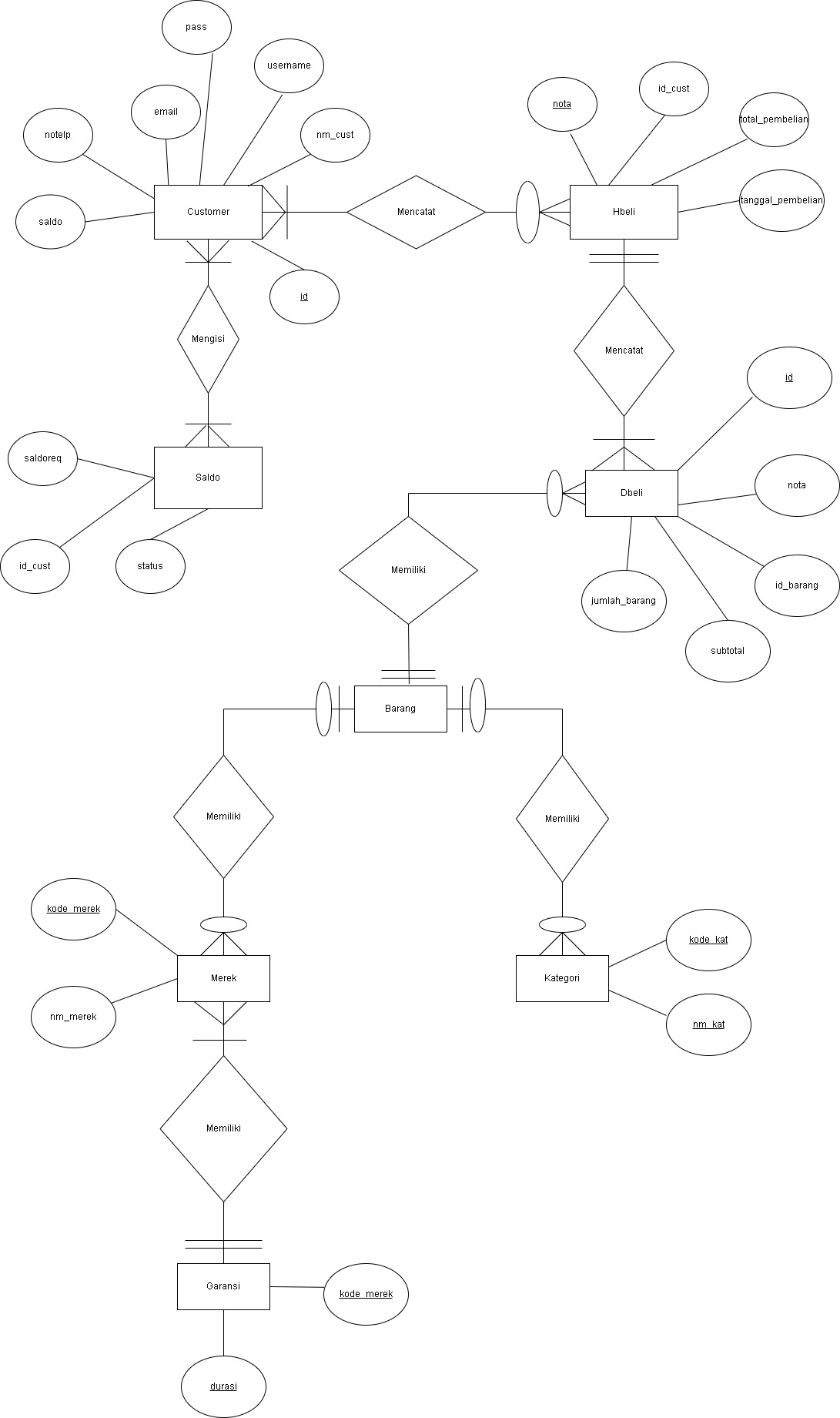
**BAB III**

**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

**3.2 Desain Database**

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



Gambar 3.3

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,email,password,notelp serta saldo

Tabel 3.2

Barang

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

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, stock, harga,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 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 , id\_cust merupakan foreign key yang digunakan untuk relasi antara tabel Hbeli dengan tabel Customer, total\_pembelian, tanggal\_pembelian

Tabel 3.6

Garansi

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| kode merk(fk merk) | Varchar(255) | Foreign Key(Merek) |
| durasi(hari) | INT |  |

Tabel 3.6 merupakan table Garansi yang digunakan untuk

Tabel 3.7

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 3.8

Saldo

|  |  |  |
| --- | --- | --- |
| Nama | Data Type | Keterangan |
| 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. If kategori[] is equal “all” then
2. Print filter barang untuk all success
3. Elseif kategori[] is not equal “all”
4. Print filter barang untuk kategori[] success
5. End if

Algoritma 3,2 Contoh Algoritma Login

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

Algoritma 3.3 contoh Algoritma Register

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

Algoritma 3.4 contoh Algoritma CRUD

Insert

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

Delete

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

Update

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