# LAPORAN PENGERJAAN APLIKASI

**16TIN2043 – Proyek 1**

**APLIKASI *E-COMMERCE* BERBASIS JAVA DEKTOP “CARTRITY”**

****

**Oleh :**

**Kelompok 1**

**Ketua : Farhan Rahmat Abdillah 171524010**

**Anggota : Afif Rana Muhammad 171524001**

**Annazar Darul Ismawan 171524005**

**Hamzah Prasetio Utomo 171524011**

**Ilham Gibran Achmad M 171524012**

**Program Studi D-4 Teknik Informatika**

**Politeknik Negeri Bandung**

**2018**

# DAFTAR ISI

[LAPORAN PENGERJAAN APLIKASI i](#_Toc518358002)

[DAFTAR ISI ii](#_Toc518358003)

[DAFTAR GAMBAR iv](#_Toc518358004)

[BAB I PENDAHULUAN 1](#_Toc518358005)

[1.1 Latar Belakang 1](#_Toc518358006)

[1.2 Rumusan Masalah 2](#_Toc518358007)

[1.3 Solusi 2](#_Toc518358008)

[BAB II RANCANGAN APLIKASI 3](#_Toc518358009)

[2.1 Rancangan Fitur 3](#_Toc518358010)

[2.2 Desain Aplikasi 4](#_Toc518358011)

[2.2.1 *Frame* Transaksi 4](#_Toc518358012)

[2.2.2 *Frame* Daftar Barang 5](#_Toc518358013)

[2.2.3 *Frame* *Login* 6](#_Toc518358014)

[2.2.4 *Frame Home* 7](#_Toc518358015)

[2.2.5 *Frame* *Admininstrator* 7](#_Toc518358016)

[2.2.6 *Frame* Donatur 8](#_Toc518358017)

[BAB III IMPLEMENTASI 9](#_Toc518358018)

[3.1 Implementasi Tampilan 9](#_Toc518358019)

[3.1.1 Implementasi *Frame* Home 10](#_Toc518358020)

[3.1.2 Implementasi *Frame* Transaksi 10](#_Toc518358021)

[3.1.3 Implementasi *Frame* Barang 11](#_Toc518358022)

[3.1.4 Implementasi *Frame* Lokasi 12](#_Toc518358023)

[3.1.5 Implementasi *Frame* Pelanggan 12](#_Toc518358024)

[3.1.6 Implementasi *Frame* *Login* 13](#_Toc518358025)

[3.1.7 Implementasi *Frame* *Shopping* 14](#_Toc518358026)

[3.1.8 Implementasi *Frame* Keranjang 15](#_Toc518358027)

[3.1.9 Implementasi *Frame* Transaksi 16](#_Toc518358028)

[3.1.10 Implementasi *Frame* *Report* 17](#_Toc518358029)

[3.2 Implementasi Fitur 17](#_Toc518358030)

[3.2.1 Implementasi Fitur Keranjang 17](#_Toc518358031)

[3.2.2 Implementasi Fitur *Login* 19](#_Toc518358032)

[3.2.3 Implementasi Fitur Transaksi dan *Report* Transaksi 21](#_Toc518358033)

[3.2.4 Implementasi Fitur Visualisasi Data 21](#_Toc518358034)

[3.2.5 Implementasi Fitur *Searching* dan *Filtering* 22](#_Toc518358035)

[BAB IV PENGUJIAN 24](#_Toc518358036)

[BAB V PENUTUP 25](#_Toc518358037)

[5.1 Kesimpulan 25](#_Toc518358038)

[5.2 Saran Pengembangan 25](#_Toc518358039)

[LAMPIRAN 27](#_Toc518358040)

[DAFTAR PUSTAKA 31](#_Toc518358041)

# DAFTAR GAMBAR

[Gambar 2.1 Tampilan frame transaksi 4](#_Toc518358042)

[Gambar 2.2 Tampilan frame daftar barang 5](#_Toc518358043)

[Gambar 2.3 Tampilan detail barang 5](#_Toc518358044)

[Gambar 2.4 Tampilan frame login 6](#_Toc518358045)

[Gambar 2.5 Tampilan frame home 7](#_Toc518358046)

[Gambar 2.6 Tampilan frame administrator 7](#_Toc518358047)

[Gambar 2.7 Tampilan frame donatur 8](#_Toc518358048)

[Gambar 3.1 JFrame utama yang mewadahi JInternalFrame 9](#_Toc518358049)

[Gambar 3.2 Visualisasi Data pada Home 10](#_Toc518358050)

[Gambar 3.3 JFrame Detail isi Table 10](#_Toc518358051)

[Gambar 3.4 Tombol Edit dan Delete hanya ditampilkan untuk admin 11](#_Toc518358052)

[Gambar 3.5 Tampilan frame lokasi 12](#_Toc518358053)

[Gambar 3.6 Tampilan Frame Pelanggan 13](#_Toc518358054)

[Gambar 3.7 JFrame Login 13](#_Toc518358055)

[Gambar 3.8 Tampilan frame shopping yang telah di filtering 14](#_Toc518358056)

[Gambar 3.9 Tampilan frame keranjang 15](#_Toc518358057)

[Gambar 3.10 Halaman pembayaran transaksi (logged in) 16](#_Toc518358058)

[Gambar 3.11 Halaman report proses transaksi selama sebulan 17](#_Toc518358059)

# BAB I PENDAHULUAN

## Latar Belakang

Jual beli merupakan kegiatan yang sering terjadi dalam kehidupan sehari-hari. Kegiatan ini merupakan salah satu upaya manusia untuk memenuhi hajat hidupnya dengan menukar apa yang ia miliki dengan apa yang ia butuhkan. Menurut KBBI, jual beli adalah persetujuan yang saling mengikat antara penjual, yakni pihak yang menyerahkan barang, dan pembeli sebagai pihak yang membayar harga yang dijual. Di dunia saat ini alat tukar untuk membayar harga dari sebuah barang umumnya berupa mata uang, baik itu secara fisik maupun digital.

Dunia modern menuntut berbagai aktivitas yang menguras tenaga dan menghabiskan waktu lebih banyak demi sebuah produktivitas. Efisiensi dan efektivitas menjadi suatu isu penting dalam menuntaskan pekerjaan atau aktivitas mereka. Termasuk di antaranya adalah aktivitas umum berupa belanja kebutuhan pribadi dan rumah tangga.

Di sisi lain, teknologi berkembang pesat dengan segala kemajuannya. Salah satu bidang yang menjadi primadona saat ini adalah bidang teknologi informasi. Di mana pada saat yang sama teknologi informasi ini memancing pertumbuhan situs belanja online nasional, atau yang kita kenal sebagai *e-commerce*.

Ketua Umum *Indonesian E-Commerce Association* (idEA), Aulia E. Marinto mengungkapkan, "Data Sensus Ekonomi 2016 dari Badan Pusat Statistik (BPS) menyebutkan, industri *e-Commerce* Indonesia dalam sepuluh tahun terakhir tumbuh sekitar tujuh belas persen dengan total jumlah usaha *e-Commerce* mencapai 26,2 juta unit." (Liputan6.com).

Sementara itu, riset global dari Bloomberg menyatakan, pada 2020 lebih dari separuh penduduk Indonesia akan terlibat di aktivitas e-Commerce. (Liputan6.com). Tentu hal ini membuktikan bahwa industri *e-commerce* sangat berpotensial untuk tumbuh subur di Indonesia.

Namun, dalam sudut pandang lain, negeri ini sedang mengalami ketimpangan sosial yang luar biasa. Dalam artikel Heyder Affan, wartawan BBC Indonesia ada sekitar 10 ranah yang disebut sebagai sumber ketimpangan sosial di Indonesia, yaitu, pertama, penghasilan (71,1%), lalu pekerjaan (62,6%), rumah/tempat tinggal (61,2%), harta benda (59,4%), kesejahteraan keluarga (56,6%), pendidikan (54%), lingkungan tempat tinggal (52%), terlibat dalam politik (48%), hukum (45%), serta kesehatan (42,3%).

Walaupun sebenarnya sudah banyak solusi yang berusaha diterapkan untuk mengatasi fenomena ketimpangan sosial ini, tetap saja tidak cukup untuk dapat mengatasi permasalahan tersebut secara keseluruhan. Salah satu solusi yang mulai marak saat ini adalah *crowd funding* yang menjadi alternatif penggalangan dana sosial berbasis *web aplication*.

Pada tahun 2017, salah satu situs crowd funding Indonesia, kitabisa.com, menyebutkan telah mengumpulkan dana sebanyak Rp. 193 miliyar melalui 12 ribu kampanye penggalangan dana dengan berbagai topik dan isu. Pertumbuhan kampanye pada tahun 2017 mencapai 300% dibandingkan tahun 2016. Sementara, pertumbuhan donasi tahun 2017 mencapai 230% dibanding tahun sebelumnya.

Hal ini menjadi menarik karena di saat situs belanja *online* berkembang pesat, kesasadaran masyarakat Indonesia dalam dunia crowd funding juga semakin meningkat. Bila dihubungkan fenomena tumbuh suburnya *e-commerce* dan situs *crowd funding* ini dengan fenomena ketimpangan sosial yang ada di Indonesia, maka bisa dimungkinkan untuk menerapkan solusi baru terhadap ketimpangan sosial di Indonesia dengan memanfaatkan berkembangnya *e-commerce* dan *crowd funding* di Indonesia.

## Rumusan Masalah

1. Bagaimana cara memanfaatkan perkembangan teknologi informasi untuk membantu menyelesaikan fenomena ketimpangan sosial di Indonesia?
2. Bagaimana cara memanfaatkan fenomena pertumbuhan *e-commerce* dan *crowd funding* di Indonesia untuk membantu menyelesaikan fenomena ketimpangan sosial di Indonesia?

## Solusi

Dari fenomena pertumbuhan *e-commerce* dan *crowd funding* di Indonesia serta fenomena meningkatnya ketimpangan sosial tersebut, terlintaslah sebuah gagasan yang mencoba menghubungkan ketiganya. Di satu sisi, pertumbuhan *e-commerce* makin subur di Indonesia. Dan di sisi lain, pertumbuhan subur ini dapat diikuti dengan upaya penurunan persentase ketimpangan sosial di Indonesia dengan memanfaatkan konsep *crowd funding* pada *e-commerce*. Gagasan yang terbentuk ialah mencoba membuat sebuah produk situs belanja *online* yang memiliki sistem donasi. Sistem donasi ini nantinya akan menjadi sarana untuk membantu masyarakat Indonesia yang membutuhkan.

# BAB II RANCANGAN APLIKASI

## Rancangan Fitur

Aplikasi dibuat pada basis *desktop* menggunakan bahasa pemrograman Java, sehingga fitur-fitur yang tersedia dan dapat ditangani oleh aplikasi ini dibuat dengan pertimbangan cocok diterapkan pada aplikasi berbasis *desktop*. Beberapa fitur yang terdapat pada aplikasi yang dibuat adalah :

1. Keranjang Belanjaan

Fitur keranjang belanjaan memungkinkan aplikasi yang dibuat untuk menangani proses pembelian lebih dari satu barang. Fitur keranjang memungkinkan pembeli untuk menyimpan data barang belanjaan di dalam keranjang, sehingga barang dapat diproses dalam satu transaksi yang sama.

1. Login

Fitur login memungkinkan aplikasi untuk membagi dan membatasi akses dari pihak tertentu, fitur login dibuat untuk memastikan bahwa pengguna (pembeli) yang akan melakukan transaksi adalah pengguna yang telah terdaftar di dalam aplikasi. Fitur ini juga dapat melakukan pembatasan *frame* (halaman) yang dapat diakses oleh *user* tertentu, sehingga dapat meningkatkan ke amanan dari aplikasi.

1. Transaksi

Fitur ini memungkinkan aplikasi untuk melakukan pengelolaan transaksi pembelian barang yang dilakukan di dalam aplikasi yang dibuat. Fitur ini memungkinkan aplikasi untuk dapat menghitung total biaya yang harus dibayar dalam suatu transaksi, meliputi biaya pembelian barang dan biaya pengiriman.

1. Pelaporan Transaksi

Fitur ini memungkin aplikasi dapat melaporkan hasil transaksi yang terjadi di dalam aplikasi yang dibuat. Laporan transaksi dibuat dalam bentuk *spreadsheet* untuk mempermudah pengelolaan transaksi selanjutnya yang mungkin terjadi.

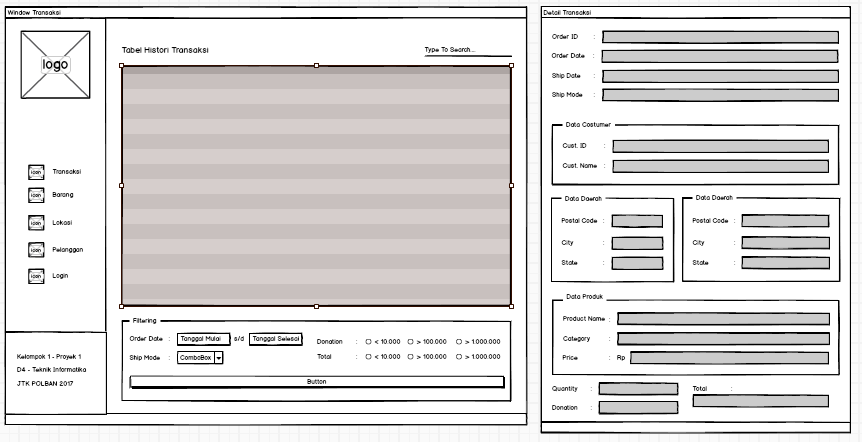
1. Visualisasi Data

Aplikasi yang dibuat dapat melakukan visualisasi terhadap data transaksi yang terjadi pada aplikasi, hal ini dilakukan untuk mempermudah pengguna untuk memahami representasi data yang tersedia di aplikasi. Visualisasi data yang dilakukan di dalam aplikasi dibuat menggunakan media diagram grafik baik itu *pie chart*, *line chart* maupun *bar chart*.

## Desain Aplikasi

Desain aplikasi dibuat menggunakan *software* Balsamiq MockUp 3. Berikut adalah desain tampilan dari aplikasi yang dibuat.

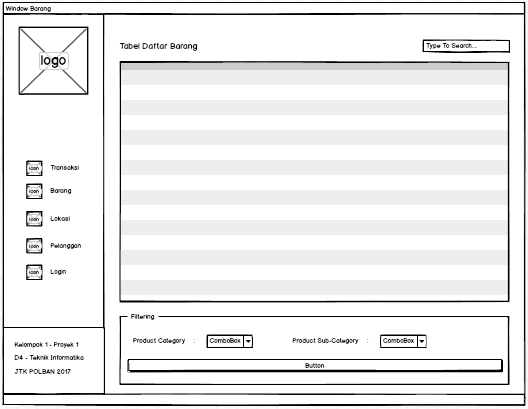
### *Frame* Transaksi



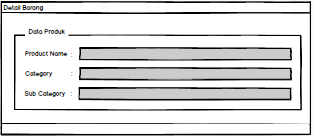
Gambar 2.1 Tampilan frame transaksi

Gambar 2.1 menunjukan desain tampilan dari *frame* yang akan menampilkan daftar transaksi yang telah dilakukan melalui aplikasi Cartrity. Data transaksi ditampilkan dalam bentuk tabel dan dapat dilakukan proses pencarian, serta *filtering*. Pada tampilan sebelah kanan terdapat tampilan detail dari *frame* transaksi, tampilan detail ini menampilkan detail dari transaksi yang ditampilkan pada tabel. *Frame* detail transaksi akan muncul, saat salah satu dari *record* transaksi di *frame* transaksi di-klik.

### *Frame* Daftar Barang



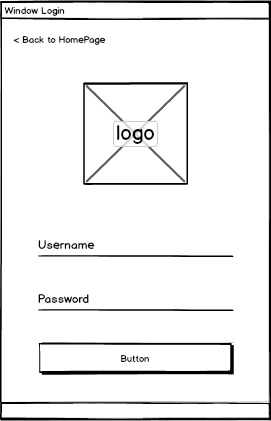
Gambar 2.2 Tampilan frame daftar barang



Gambar 2.3 Tampilan detail barang

*Frame* daftar barang memiliki fungsi utama untuk menampilkan daftar barang yang tersedia beserta harga dari barang tersebut. *Frame* ini direncanakan akan ditampilkan sebelum pengguna *login* ke dalam aplikasi hal ini bertujuan untuk mempermudah pengguna untuk memeriksa barang yang hendak ia beli apakah tersedia atau tidak, tanpa perlu melakukan *login* terlebih dahulu. Sama halnya seperti *frame* transaksi, *frame* daftar barang juga memiliki *frame* detail yang akan menampilkan data barang secara lebih mendetail yang dapat dilihat pada Gambar 2.3 Tampilan detail barang.

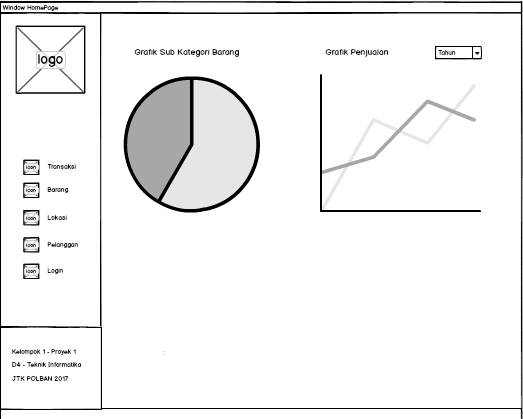
### *Frame* *Login*



Gambar 2.4 Tampilan frame login

*Frame* *login* adalah *frame* yang akan digunakan oleh pengguna untuk dapat masuk kedalam fitur transaksi pada aplikasi Cartrity. *Frame* ini berfungsi untuk membatasi akses pengguna, serta mengarahkan pengguna menuju *frame* yang menjadi hak akses mereka. Sebagai contoh, seorang yang memiliki hak akses anggota atau donatur ketika melakukan proses login akan diarahkan menuju *frame* donatur, yang fungsi utamanya adalah untuk melakukan transaksi pembelian, sedangkan seorang admin akan diarahkan menuju *frame* admin, yang berfungsi untuk melakukan pengelolaan data yang terdapat di dalam aplikasi Cartrity.

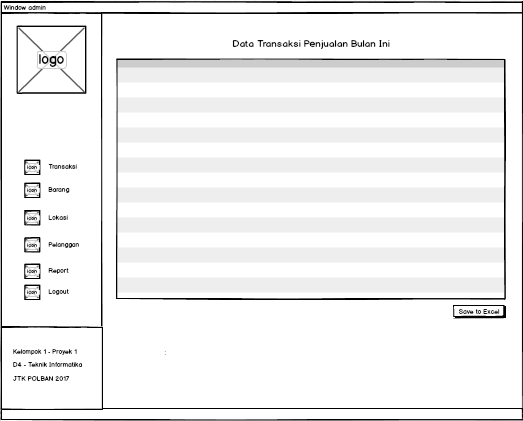
### *Frame Home*



Gambar 2.5 Tampilan frame home

*Frame* ini adalah *frame* yang pertama kali akan tampil ketika program dijalankan. *Frame* ini menangani fitur-fitur yang tidak memerlupan proses login terlebih dahulu, seperti proses melihat barang yang tersedia di aplikasi dan melihat grafik kategori yang tersedia di dalam aplikasi Cartrity.

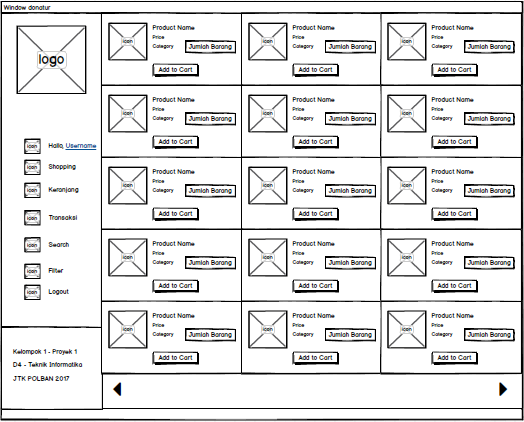
### *Frame* *Admininstrator*



Gambar 2.6 Tampilan frame administrator

*Frame admin* adalah *frame* yang menangani fitur yang berhubungan dengan proses pengelolaan data yang terdapat di dalam aplikasi Cartrity. Untuk mengakses halaman ini pengguna aplikasi harus melakukan *login* ke dalam aplikasi menggunakan *username* dan *password* yang memiliki hak akses *administrator*.

### *Frame* Donatur

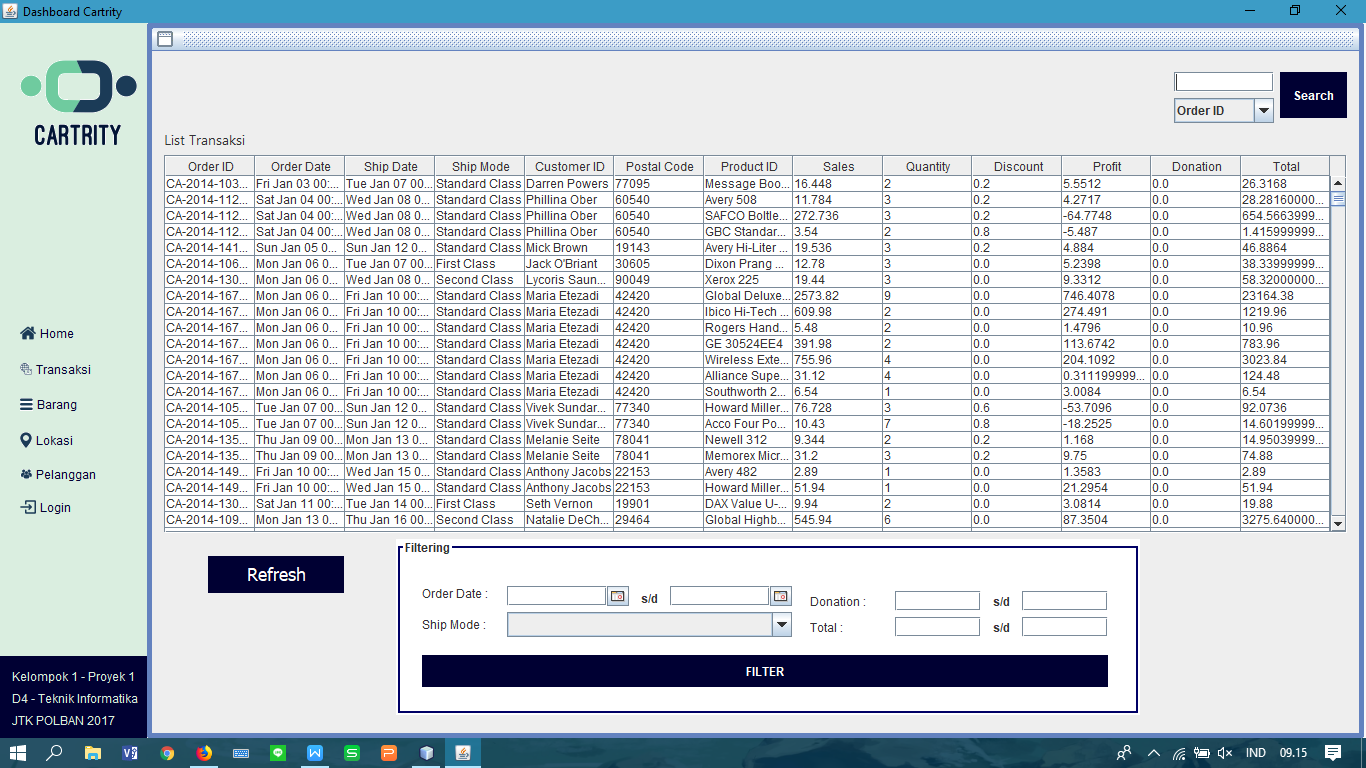


Gambar 2.7 Tampilan frame donatur

*Frame* donatur menangani segala macam proses pembelian yang akan dilakukan oleh donatur melalui aplikasi yang dibuat. *Frame* ini memiliki tampilan yang mirip dengan *frame* *administrator* yang membedakan kedua *frame* tersebut hanyalah pilihan menu yang terdapat di sebelah kiri *frame*.

# BAB III IMPLEMENTASI

## Implementasi Tampilan



panel 2

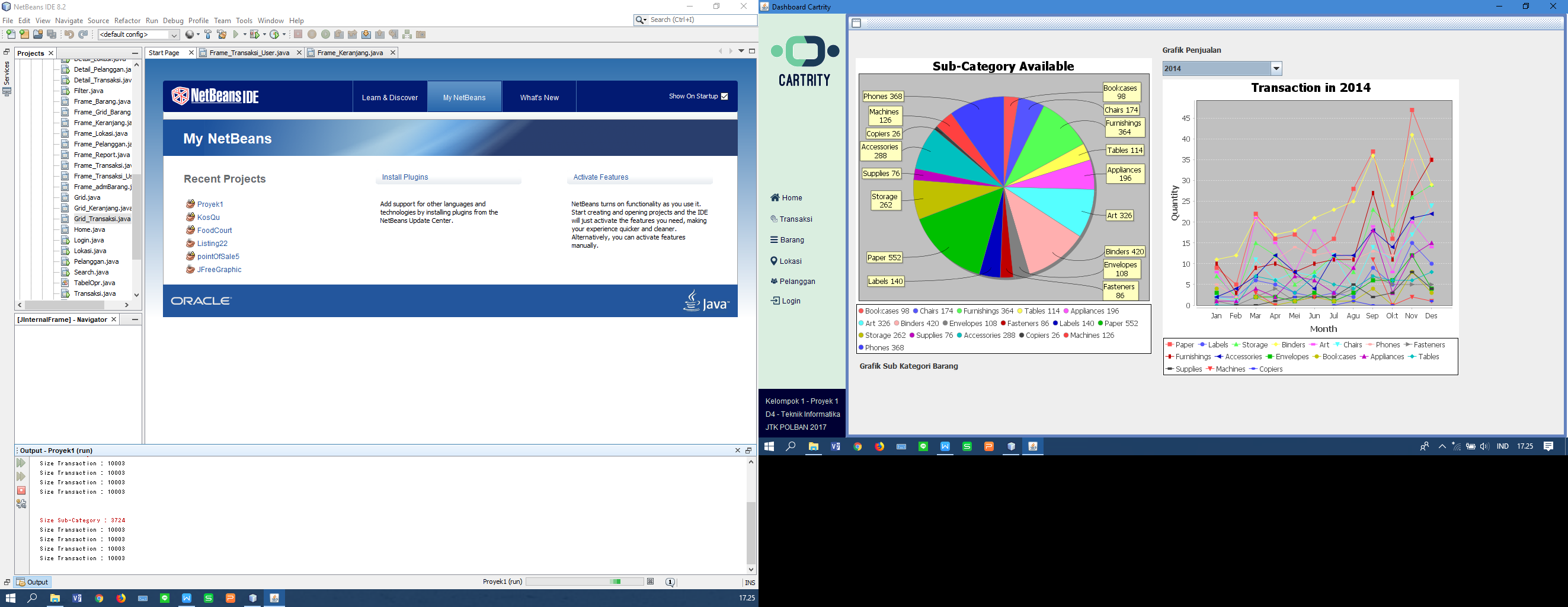
panel 1

Gambar 3.1 JFrame utama yang mewadahi JInternalFrame

Dalam implementasi tampilan program digunakan JFrame sebagai wadah utama tampilan program yang mencangkup 2 (dua) buah JPanel (gambar 3.1), panel 1 yang berisi tombol navigasi halaman program dan panel 2 yang isinya berubah sesuai dengan halaman yang dipilih. Terdapat 3 tipe JFrame yang tersedia untuk : pengunjung tanpa akun, pengunjung yang memiliki akun, dan admin, dimana setiap JFrame tersebut menyediakan tombol navigasi yang berbeda. Untuk pengunjung tanpa akun akan tersedia tombol home, transaksi, barang, lokasi, pelanggan, dan login. Untuk pengunjung dengan akun akan tersedia tombol: shopping, keranjang, transaksi(pengunjung), search, filter, refresh, dan logout. Dan untuk admin tersedia tombol: transaksi, barang, lokasi, pelanggan, logout, dan report.

Berikut adalah beberapa halaman yang perlu diimplementasikan dari desain aplikasi yang terdapat pada BAB II:

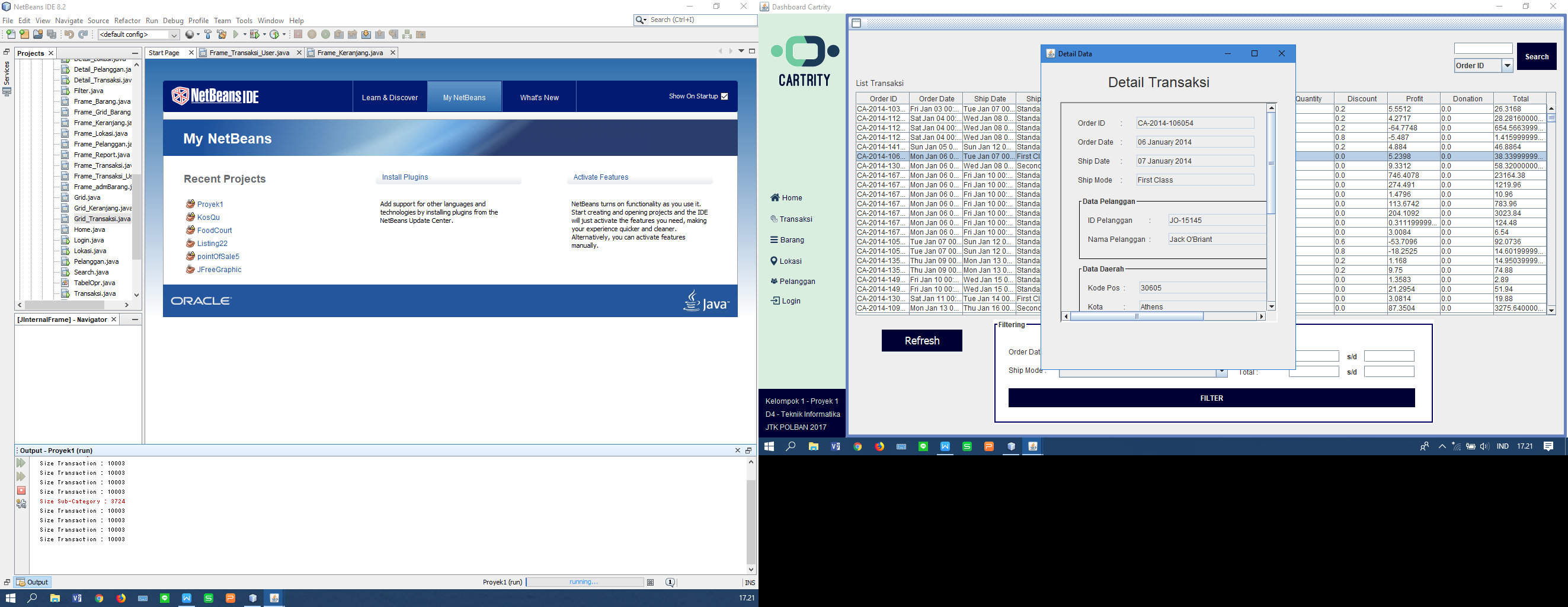
### Implementasi *Frame* Home



Gambar 3.2 Visualisasi Data pada Home

Merupakan *frame* *default* yang akan dibuka saat program pertama kali dijalankan. *Frame* ini berisi diagram pie (*pie chart*) barang yang terjual berdasarkan kategori serta diagram garis (*line chart*) penjualan selama setahun. Pada *frame* ini juga disediakan sebuah JComboBox yang digunakan untuk menentukan tahun yang akan ditampilkan laporan transaksinya dalam bentuk diagram garis (Gambar 3.2).

### Implementasi *Frame* Transaksi

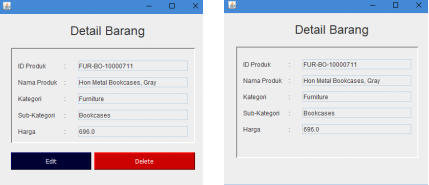


Gambar 3.3 JFrame Detail isi Table

Ditampilkan JTabel yang berisi data proses transaksi yang terdiri dari informasi pelaku transaksi, informasi produk dan pengirimannya, harga dan banyak pembelian, keuntungan, dan besar donasi yang diberikan. Pada panel ini juga tersedia fitur *filtering* untuk menyaring data berdasarkan beberapa kategori yang telah ditentukan, serta fitur pencarian (*searching*) yang digunakan untuk melakukan pencarian data transaksi berdasarkan kata kunci (*keywords*) tertentu.

Data pada tabel juga dapat di-klik untuk mengakses *frame* yang berisikan informasi detail dari transaksi tersebut (Gambar 3.3). Pada *frame* detail juga disediakan fitur untuk menghapus transaksi yang di-klik, dimana fitur ini hanya tersedia untuk hak akses *administrator* (Gambar 3.4).

### Implementasi *Frame* Barang



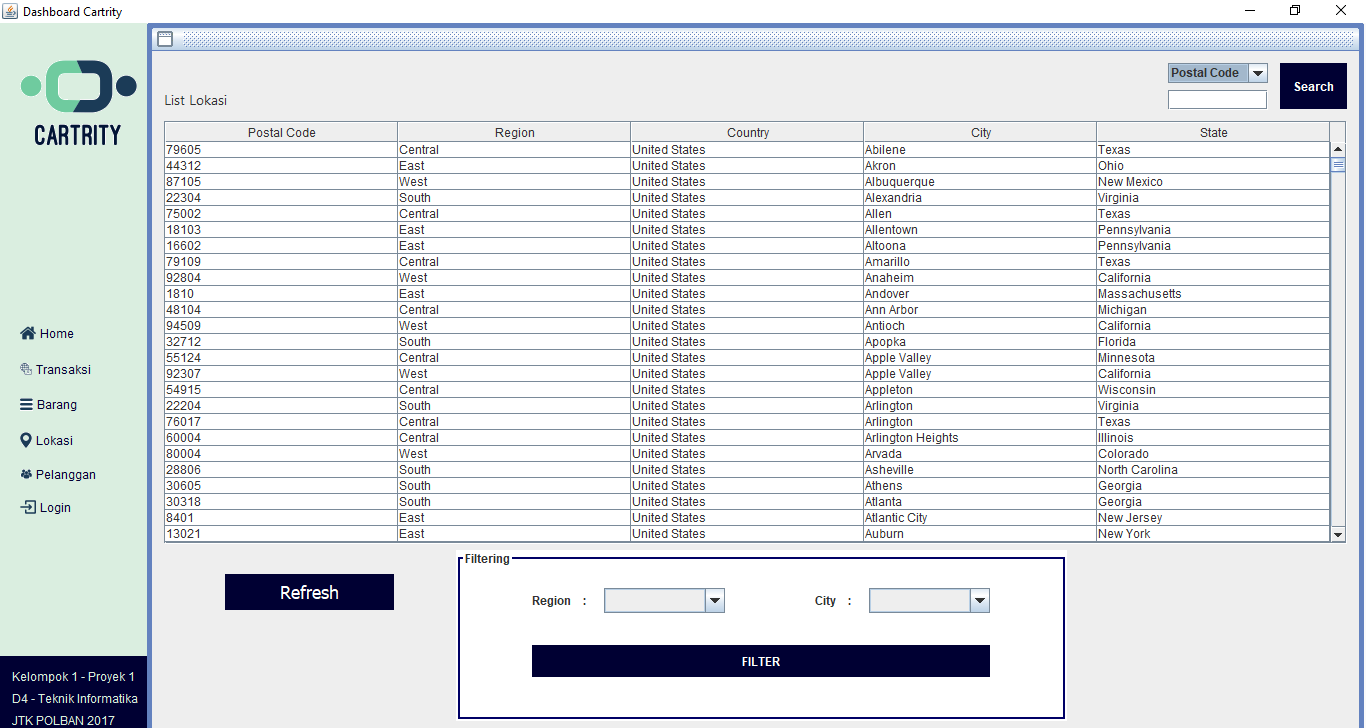
Gambar 3.4 Tombol Edit dan Delete hanya ditampilkan untuk admin

Ditampilkan JTabel berisi data barang yang terdiri dari id, kategori, sub-kategori, nama, dan harga. Sama seperti panel transaksi, *frame* barang juga menyediakan fitur *filtering* dan *searching* yang digunakan untuk melakukan penyaringan data yang akan ditampilkan.

*Frame* barang juga memberikan fitur melihat detail data yang ditampilkan dengan melakukan klik pada data tertentu sama seperti yang diberikan oleh *frame* transaksi. Perbedaan antara *frame* barang dan transaksi terletak pada fitur edit dan tambah barang yang terdapat pada *frame* detail barang, dimana fitur ini hanya dapat diakses oleh pengguna yang memiliki hak akses *administrator* (Gambar 3.4).

### Implementasi *Frame* Lokasi

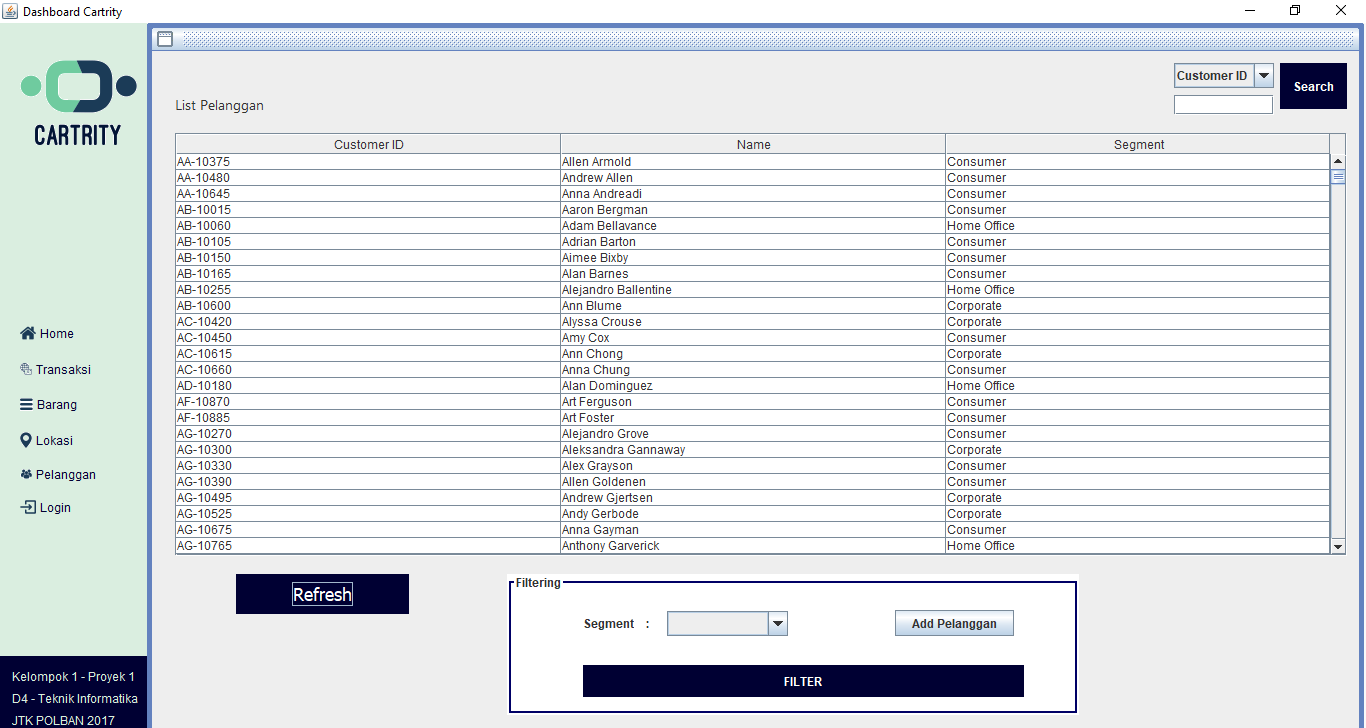
*Frame* ini akan menampilkan JTabel berisi data lokasi terdiri dari kode pos, provinsi, kota, dan negara. Pada panel ini juga tersedia *filtering* dan *searching*, dan akan ditampilkan isi tabel dalam sebuah JFrame apabila data di klik (Gambar 3.3) dan jika admin tombol untuk mengedit dan menghapus data akan ditampilkan (Gambar 3.4).



Gambar 3.5 Tampilan frame lokasi

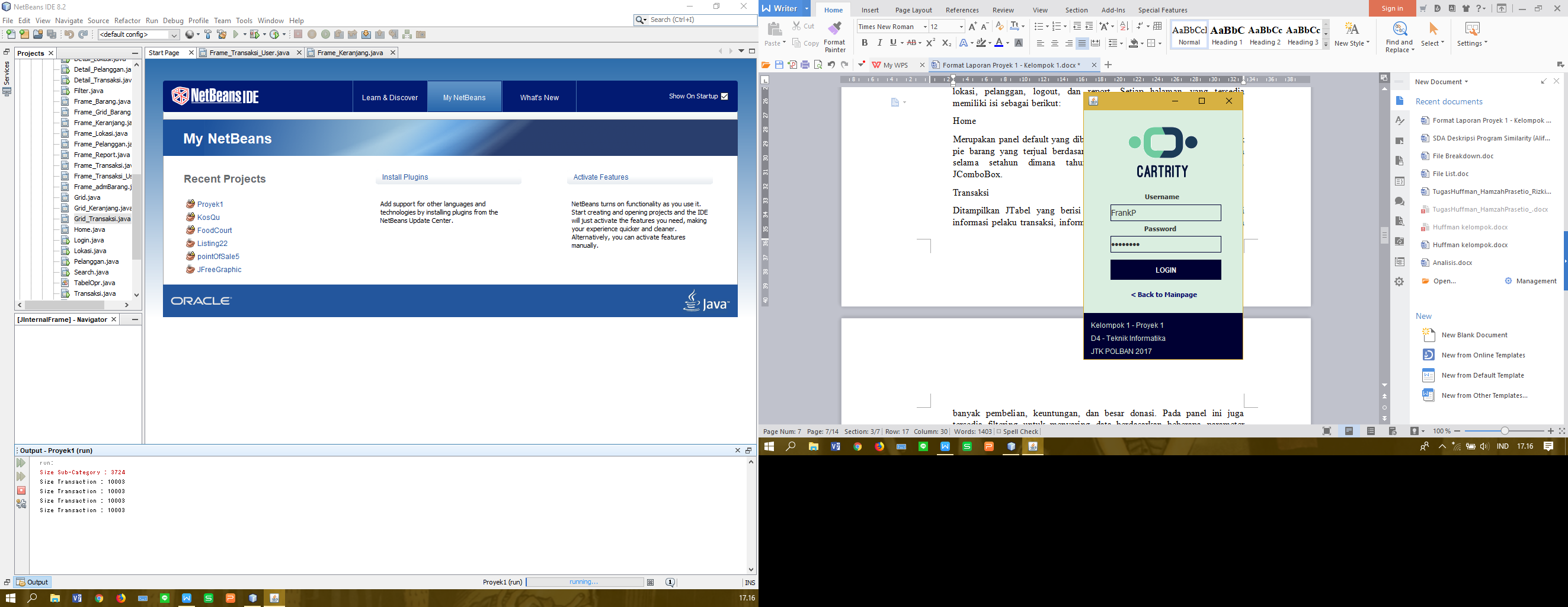
### Implementasi *Frame* Pelanggan

Ditampilkan JTabel berisi data barang terdiri dari id, nama, dan segmen. *Filtering* ,*searching*, dan tombol untuk menampilkan window penambah data pelanggan tersedia. Data pada tabel dapat di klik untuk menampilkan data barang pada JFrame (Gambar 3.3) dimana tersedia pilihan untuk meng-*edit* data dan men-*delete* bagi admin (Gambar 3.6).



Gambar 3.6 Tampilan Frame Pelanggan

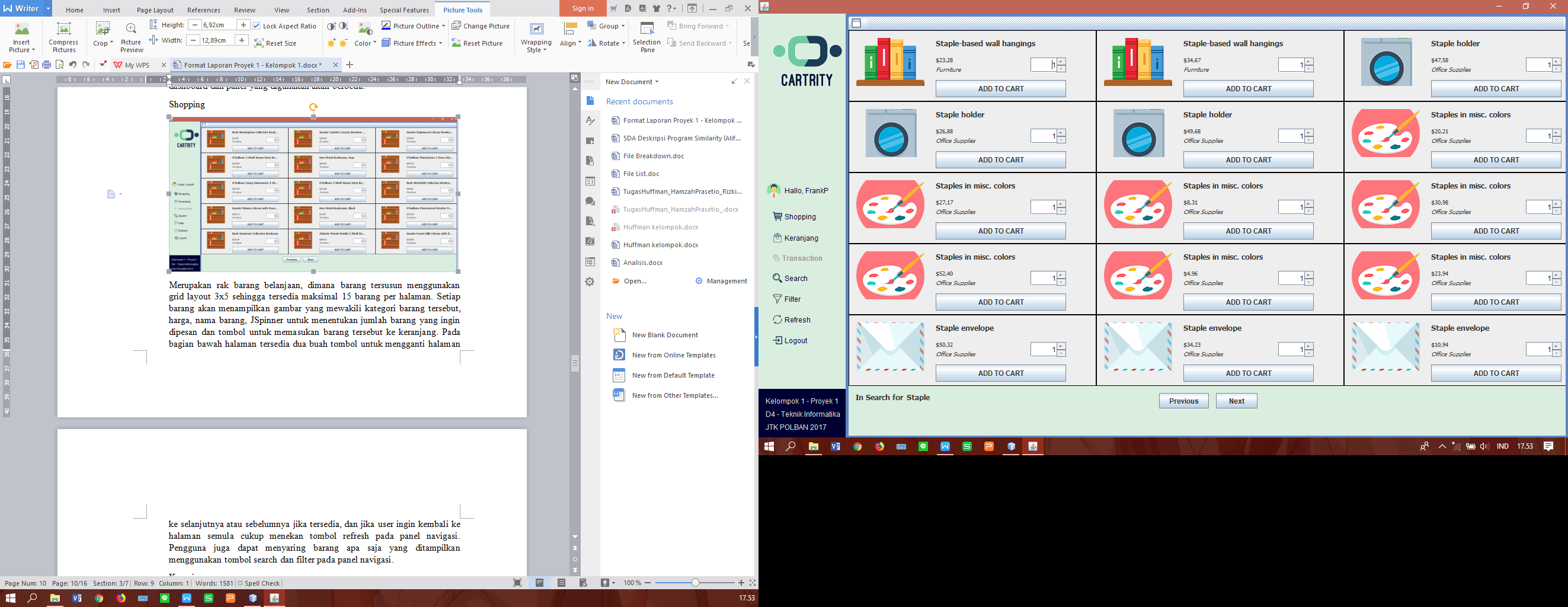
### Implementasi *Frame* *Login*



Gambar 3.7 JFrame Login

Seperti yang telah tergambar pada BAB II, *frame* login akan digunakan untuk menangani fitur *login* di dalam aplikasi yang dibuat. Pada *frame* ini terdapat dua buah *field* yang digunakan untuk memasukan *username* dan *password* dari user yang akan melakukan *login* ke dalam sistem. Seperti yang sudah disampaikan sebelumnya *frame* ini akan mengarahkan pengguna dengan hak akses berbeda menuju *frame* yang sesuai dengan hak akses yang dimiliki, sehingga dapat meningkatkan keamanan dari sistem yang dibuat.

### Implementasi *Frame* *Shopping*

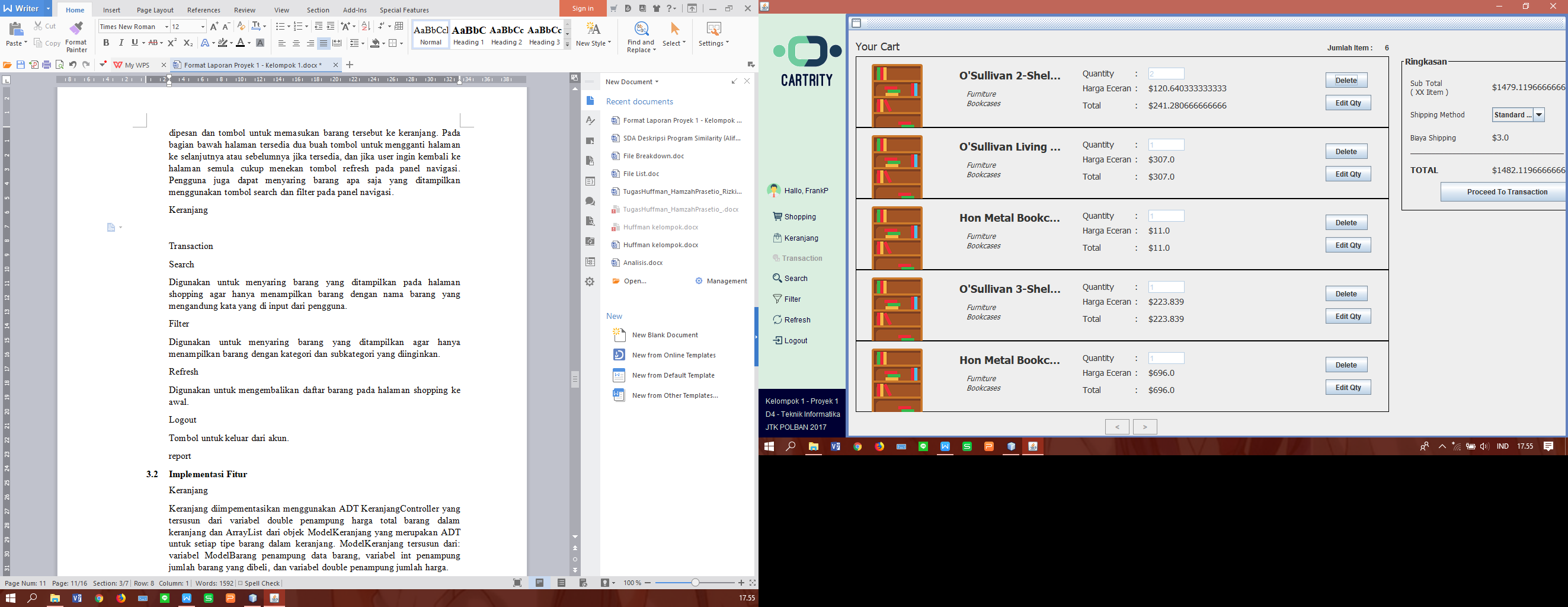


Gambar 3.8 Tampilan frame shopping yang telah di filtering

*Frame* ini merupakan *frame* yang menampilkan seluruh barang yang dapat dibeli oleh pengguna, dimana barang tersusun menggunakan grid layout 3x5 sehingga tersedia maksimal 15 barang per halaman *frame*. Setiap barang akan menyajikan gambar yang mewakili kategori barang tersebut, harga, nama barang, JSpinner untuk menentukan jumlah barang yang ingin dipesan dan tombol ‘add to cart’ untuk memasukan barang tersebut ke dalam keranjang belanja.

Pada bagian bawah *frame* tersedia dua buah tombol untuk mengganti halaman yaitu tombol *next* dan *previous* yang akan mengarahkan pengguna menuju halaman *frame* selanjutnya ataupun sebelumnya jika halaman yang hendak diakses memang tersedia. Pengguna juga dapat menyaring barang apa saja yang ingin ditampilkan menggunakan tombol *search* dan *filter* yang tersedia pada panel navigasi serta tombol *refresh* yang dapat digunakan untuk mengembalikan *frame* pada kondisi semula sebelum dilakukan *filtering* maupun *searching*.

### Implementasi *Frame* Keranjang

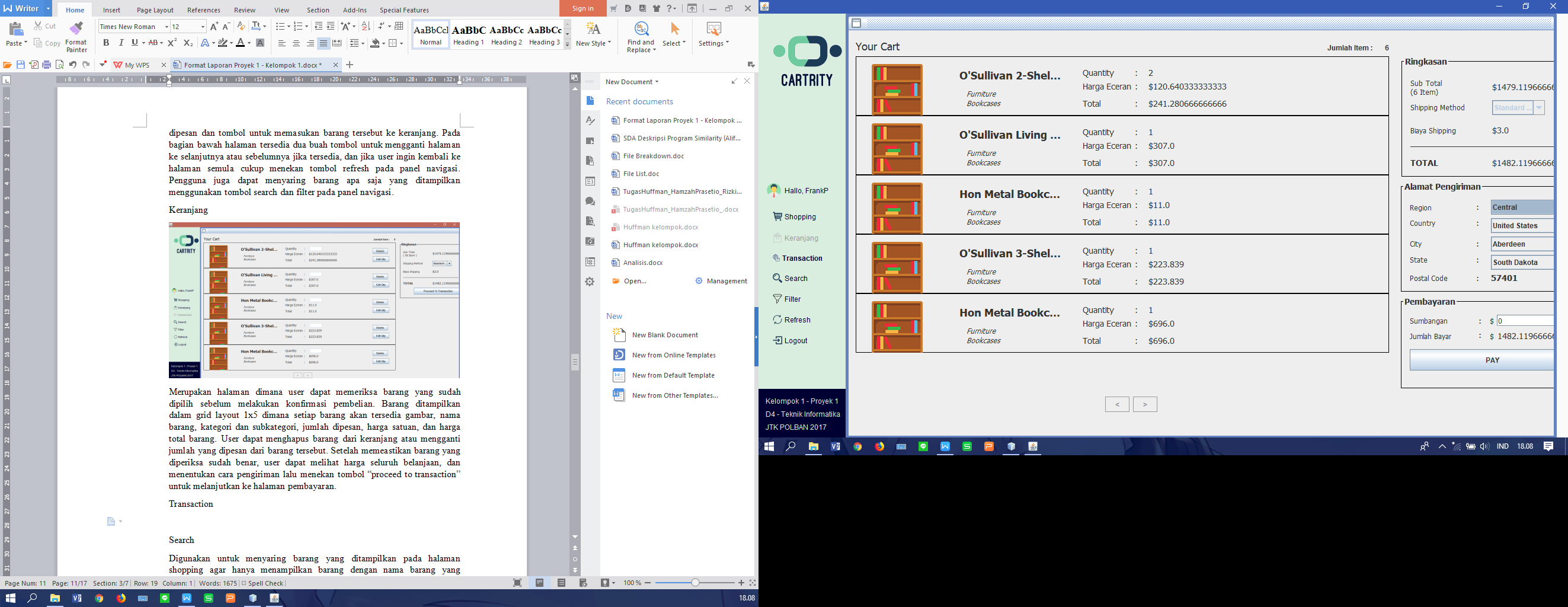
**

Gambar 3.9 Tampilan frame keranjang

Merupakan *frame* dimana *user* dapat memeriksa barang apa saja yang sudah dipilih sebelum melakukan konfirmasi pembelian. Barang ditampilkan dalam bentuk grid layout 1x5 dimana setiap barang akan tersedia gambar, nama barang, kategori, subkategori, jumlah barang yang dipesan, harga satuan barang , dan harga total barang. User dapat menghapus barang dari keranjang atau mengganti jumlah yang dipesan dari barang tersebut memanfaatkan tombol edit dan hapus yang tersedia pada *frame* tersebut.

Setelah memastikan barang yang diperiksa sudah benar, *user* dapat melihat harga seluruh belanjaan, dan menentukan cara pengiriman barang lalu menekan tombol “*proceed to transaction*” untuk melanjutkan ke halaman pembayaran.

### Implementasi *Frame* Transaksi

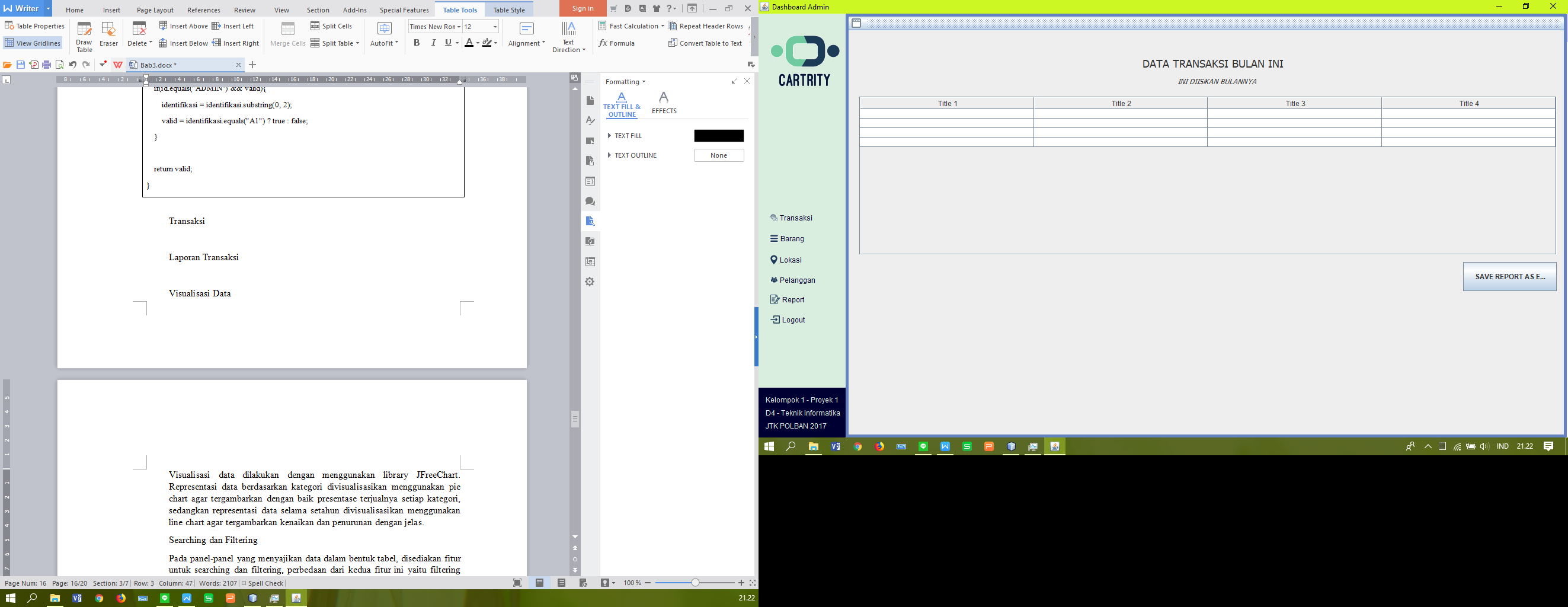


Gambar 3.10 Halaman pembayaran transaksi (logged in)

*Frame* ini adalah *frame* yang dapat diakses oleh pengguna setelah menekan tombol “*proceed to transaction*” pada *frame* keranjang seperti yang dapat dilihat pada Gambar 3.7. *Frame* ini merupakan *frame* terakhir dari rangkaian *frame* yang menangani proses pembelian barang melalui aplikasi Cartrity.

Pada halaman ini pengguna pertama perlu mengisi alamat pengiriman berdasarkan lokasi yang telah disediakan pada JComboBox, dilanjutkan dengan menentukan besarnya donasi yang ingin diberikan oleh pengguna sebelum menekan tombol “*PAY*” untuk menyelesaikan proses transaksi.

### Implementasi *Frame* *Report*



Gambar 3.11 Halaman report proses transaksi selama sebulan

Pada halaman *report,* tabel akan berisi data transaksi yang terjadi dalam jangka waktu sebulan, pada kanan bawah terdapat tombol untuk menuliskan isi tabel kedalam file excel dengan nama file “Report\_<tahun>\_<bulan>”.

## Implementasi Fitur

Fitur-fitur yang sudah dirancang pada BAB II tentunya akan dieksekusi ke dalam bentuk *source* *code* dalam bahasa Java. Fitur yang telah direncanakan atau dirancang tentunya bisa saja mengalami perubahan yang disesuaikan dengan kemampuan dari bahasa pemrograman yang digunakan. Berikut adalah fitur-fitur yang terdapat pada aplikasi Cartrity.

### Implementasi Fitur Keranjang

Keranjang diimpementasikan menggunakan ADT (*Abstract Data Type*) Keranjang, yang diimplementasikan lebih lanjut menggunakan konsep MVC (*Model View Controller*). *Controller* dari keranjang diberi nama KeranjangController yang tersusun dari variabel *double* penampung harga total barang dalam keranjang dan *ArrayList* dari objek *ModelKeranjang* yang merupakan *model* dari ADT keranjang. ModelKeranjang tersusun dari variabel ModelBarang penampung data barang, variabel *integer* penampung jumlah barang yang dibeli, dan variabel *double* penampung jumlah harga.

Pengisian objek ke dalam keranjang dilakukan dalam *frame* “*shopping*” yang dapat diakses oleh pengguna yang memiliki akun atau hak akses donatur. Objek dimasukkan ke dalam keranjang dengan menekan tombol “*add to cart*” pada barang yang diinginkan. Saat dilakukan proses penambahan barang ke keranjang akan dilakukan pemeriksaan terlebih dahulu apa barang tersebut sudah ada di keranjang atau belum. Jika sudah terdapat barang tersebut dalam keranjang, maka akan dilakukan penambahan jumlah barang yang dipesan dan jumlah harga barang tersebut. Jika barang belum ada dalam keranjang maka akan ditambahkan objek baru kedalam list keranjang.

private static List<ModelKeranjang> shoppingCart = new ArrayList<ModelKeranjang>();

boolean ketemu = false;

int i = 0;

BarangController bc = new BarangController();

ModelBarang b = bc.searchObject(ID\_Barang);

if (shoppingCart.isEmpty()) {

//Barang langsung dimasukkan ke keranjang apabila keranjang kosong

} else {

while (!ketemu && i < shoppingCart.size()) {

if (shoppingCart.get(i).getBarang().getProductID().equals(b.getProductID())) {

//dilakukan pencarian barang yang dimasukkan dalam keranjang

ketemu = true;

}

i += 1;

}

if (!ketemu) {

//Jika jenis barang belum ada dalam keranjang, barang ditambahkan ke list

}

}

//dilakukan pencarian barang yang dimasukkan dalam keranjang

ketemu = true;

}

i += 1;

}

if (!ketemu) {

//Jika jenis barang belum ada dalam keranjang, barang ditambahkan ke list

}

}

Untuk memeriksa isi keranjang, pengguna dapat membuka halaman “keranjang”. Pada halaman tersebut pengguna dapat memeriksa isi keranjang, menghapus barang yang tidak diinginkan, atau mengganti jumlah barang yang dipesan sebelum mengkonfirmasi isi keranjang dan melakukan pembayaran. Saat dilakukan penghapusan barang dari keranjang, maka harga total barang dalam keranjang dikurangi dan barang tersebut akan dihapus dari list. Saat dilakukan perubahan jumlah barang yang dipesan, akan dilakukan perhitungan ulang harga total barang.

### Implementasi Fitur *Login*

Dalam penggunaannya program akan memberikan pilihan-pilihan yang berdasarkan status pengguna, apakah menggunakan akun atau tidak, dan jika menggunakan akun apakah admin atau bukan. Untuk pengguna yang tidak memiliki akun, program tidak akan dapat digunakan secara maksimal. Data akun disimpan dalam file excel yang terdiri dari username, password yang dienkripsi, dan id user. Username berfungsi sebagai identitas akun tersebut selama menggunakan program. Id user berfungsi untuk membedakan antara akun admin dan akun biasa; untuk akun admin, id user akan dawali dengan “A1”.

//Algoritma untuk mengenkripsi *password*

public static String encrypt(String data) throws Exception {

Key key = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.ENCRYPT\_MODE, key);

byte[] encVal = c.doFinal(data.getBytes());

return new BASE64Encoder().encode(encVal);

}

//Algoritma pemeriksaan *username* dan *password*

public static boolean masuk(String un, String pass, String id) {

boolean valid = false;

String identifikasi = "";

FileInputStream excelFile = new FileInputStream(new File(FILE\_NAME));

Workbook workbook = new XSSFWorkbook(excelFile);

Sheet sheet = workbook.getSheetAt(7);

Iterator<Row> iterator = sheet.iterator();

while (iterator.hasNext() && !valid) {

//Pemeriksaan *username* dan *password* untuk setiap data pada excel

}

if(id.equals("ADMIN") && valid){

identifikasi = identifikasi.substring(0, 2);

valid = identifikasi.equals("A1") ? true : false;

}

return valid;

}

return valid;

}

### Implementasi Fitur Transaksi dan *Report* Transaksi

Dalam implementasinya, perhitungan dalam proses transaksi dilakukan dalam ADT KeranjangController yang mengurusi keranjang penambung barang-barang yang dibeli. Setelah pengguna mengkonfirmasi barang dalam keranjang dan memilih metode pengiriman, pengguna akan dikirim ke halaman pembayaran transaksi dimana user diminta menentukan alamat pengiriman dari lokasi yang telah disediakan menggunakan JComboBox dan pengguna juga dapat mendonasikan sedikit uangnya.

JComboBox penentu alamat pengiriman telah di persiapkan sedemikian rupa sehingga JComboBox berhubungan dengan lainnya, hal ini agar tidak terjadi salah alamat dimana penguna memilih kota pada negara yang salah. Setelah pengguna mengkonfirmasi pembayaran, informasi transaksi akan disimpan dalam excel dan admin dapat membuka halaman *report* untuk memeriksa data transaksi selama bulan tersebut. Pada halaman tersebut admin juga dapat membuat file excel yang berisi hanya transaksi pada bulan tersebut dengan format nama “Report\_<tahun>\_<bulan>”.

### Implementasi Fitur Visualisasi Data

Visualisasi data dilakukan dengan menggunakan *library* JFreeChart. Fitur visualisasi menjadi penting untuk pengguna karena mampu mempermudah pengguna dalam memahami data yang disajikan di dalam aplikasi, khususnya data-data yang berhubungan dengan angka dan perhitungan. Visualisasi data yang dilakukan adalah visualisasi data yang berhubungan dengan data transaksi yang terdapat di dalam aplikasi, dimana data yang dimiliki akan lebih banyak disajikan menggunakan diagram atau grafik.

Data yang akan yang dimiliki di representasikan ke dalam dua buah diagram atau grafik yang memiliki fungsinya masing-masing yaitu diagram pie (*pie chart*) dan diagram garis (*line chart*). Diagram pie digunakan untuk mereperesntasikan data kategori barang yang tersedia di dalam aplikasi untuk memudahkan pengguna melihat komposisi barang yang tersedia di dalam aplikasi Cartrity, sedangkan diagram garis digunakan untuk merepresentasikan data penjualan barang untuk mempermudah pengguna melihat tren penjualan yang terjadi.

### Implementasi Fitur *Searching* dan *Filtering*

Pada panel-panel yang menyajikan data dalam bentuk tabel, disediakan fitur untuk *searching* dan *filtering*, perbedaan dari kedua fitur ini yaitu *filtering* menyaring data dengan lebih dari satu kategori pencarian dimana kategori tersebut telah ditentukan sebelumnya, sedangkan *searching* melakukan penyaring data hanya berdasarkan satu kategori pencarian dimana kategori tersebut harus diketikan oleh pengguna.

Pada fitur *searching* digunakan JTextField sebagai kotak input parameter *searching*, JComboBox untuk menentukan kolom mana pada tabel yang disaring, dan JButton untuk memulai proses *searching*. Pada fitur *filtering* berdasarkan jenis data yang ada berbagai swing control dapat digunakan. Data dari excel telah tersimpan dalam sebuah *list*, isi setiap *node* pada list tersebut akan diperiksa apakah memenuhi kriteria menggunakan seleksi (*if*) dimana jika bernilai benar maka isi list akan ditampilkan ke tabel.

Berikut adalah potongan sintaks yang menangani *searching* dan *filtering*.

//Algoritma proses *searching*

Object[] isi = { //Header tabel };

DefaultTableModel A = new DefaultTableModel(null,isi);

table1.setModel(A);

List<ModelBarang> lis = new BarangController().getList();

Iterator<ModelBarang> iterator = lis.iterator();

while (iterator.hasNext())

{

ModelBarang hsl = iterator.next();

//pengisian array object “isi” oleh data dari excel

if(isi[cell].toString().contains(searched) ){

A.addRow(isi);

}

}

# BAB IV PENGUJIAN

Aplikasi yang terlah dibuat tentu perlu diuji terlebih dahulu, untuk memastikan Aplikasi tersebut berjalan semestinya dan tidak akan memberikan dampak negatif tehadap seseorang atau suatu pihak.

## Metode Pengujian

Metode pengujian yang digunakan untuk melakukan pengujian terhadap aplikasi Cartrity adalah *Black* *Box* *Testing*. *Black* *Box* *Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengujian pada spesifikasi fungsional *program*.

## Media Pengujian

Pengujian terhadap aplikasi Cartrity dilakukan dengan sebuah *personal* *computer* (*PC*) dengan spesifikasi sebagai berikut :

* Sistem operasi : Windows 10 Home Single Language 64-bit
* *Processor* : Intel Core i5-7200U (2.50 *GHz*)
* *RAM* : 4 *GigaBytes*
* *HardDisk* : 1 *TerraBytes*

*Compiler* atau *IDE* yang digunakan untuk melakukan pengujian terhadap aplikasi Cartrity adalah Netbeans IDE 8.2 dengan JDK 8.

## Rencana Pengujian

Rencana untuk melakukan pengujian terhadap aplikasi Cartrity dengan metode *Black* *Box* *Testing* ditampilkan pada tabel 4.1.

**Tabel 4.1 Rencana Pengujian Aplikasi Cartrity**

|  |  |  |
| --- | --- | --- |
| Kelas Uji | Butir Uji | Detail Uji |
| Display data dengan chart dan tabel | Chart pada tab Home | Run aplikasi dan klik tab Home |
| Tabel pada tab Transaksi | Run aplikasi dan klik tab Transaksi |
| Tabel pada tab Barang | Run aplikasi dan klik tab Barang |
| Tabel pada tab Lokasi | Run aplikasi dan klik tab Lokasi |
| Tabel pada tab Pelanggan | Run aplikasi dan klik tab Pelanggan |
| Fitur searching | Search pada tab Transaksi | Input dengan data yang valid |
| Input dengan data yang tidak valid |
| Search pada tab Barang | Input dengan data yang valid |
| Input dengan data yang tidak valid |
| Search pada tab Lokasi | Input dengan data yang valid |
| Input dengan data yang tidak valid |
| Search pada tab Pelanggan | Input dengan data yang valid |
| Input dengan data yang tidak valid |
| Fitur filtering | Filter pada tab Home | Memilih item dari jComboBox |
| Filter pada tab Transaksi | Memilih item dari jComboBox |
| Input data yang tidak valid |
| Filter pada tab Barang | Memilih item dari jComboBox |
| Filter pada tab Lokasi | Memilih item dari jComboBox |
| Filter pada tab Pelanggan | Memilih item dari jComboBox |
| Fitur login | Login dengan akun | Login dengan akun pelanggan |
| Login dengan akun admin |
| Login dengan akun yang tidak terdaftar |
| Fitur shopping pada dashboard pelanggan | Menambahkan item ke keranjang | Menambahkan item ke keranjang dengan jumlah yang valid |
| Menambahkan item ke keranjang dengan jumlah yang tidak valid |
| Menambahkan item yang sama ke keranjang lebih dari satu kali |
| Search item | Input dengan data yang valid |
| Input dengan data yang tidak valid |
| Filter item | Memilih item dari jComboBox |
| Fitur Keranjang pada dashboard pelanggan | Edit jumlah barang | Mengubah jumlah item dengan jumlah yang valid |
| Mengubah jumlah item dengan jumlah yang tidak valid |
| Delete barang | Menghapus sebuah barang dari keranjang |
| Proses ke transaksi | Melanjutkan ke proses transaksi dengan keranjang yang berisi |
| Melanjutkan ke proses transaksi dengan keranjang yang tidak berisi |
| Fitur Transaction pada dashboard pelanggan | Pengisian data alamat pelanggan | Memilih item dari jComboBox |
| Donasi | Melakukan input dengan jumlah yang valid |
| Melakukan input dengan jumlah yang tidak valid |
| Pembayaran | Melakukan pembayaran |
| Fitur Edit data pada dashboard admin | Edit data pada tab Barang | Mengubah data dengan data yang valid |
| Mengubah data dengan data yang tidak valid |
| Edit data pada tab Lokasi | Mengubah data dengan data yang valid |
| Edit data pada tab Pelanggan | Mengubah data dengan data yang valid |
| Fitur Delete data pada dashboard admin | Delete data pada tab Transaksi | Menghapus sebuah data pada tabel |
| Delete data pada tab Barang | Menghapus sebuah data pada tabel |
| Delete data pada tab Lokasi | Menghapus sebuah data pada tabel |
| Delete data pada tab Pelanggan | Menghapus sebuah data pada tabel |
| Fitur Add data pada dashboard admin | Add produk pada tab Barang | Menambahkan sebuah produk |

## Hasil Pengujian

Hasil pengujian terhadap aplikasi Cartrity dengan metode *Black* *Box* *Testing* ditampilkan pada tabel 4.2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kelas Uji | Butir Uji | Detail Uji | Prosedur | Hasil yang diharapkan  **Tabel 4.2 Hasil Pengujian Aplikasi Cartrity** | Hasil yang didapat | Keterangan |
| Display data dengan chart dan tabel | Chart pada tab Home | Run aplikasi dan klik tab Home | Run aplikasi dan klik tab Home pada bagian kiri dashboard | Data ditampilkan dengan baik | Data ditampilkan dengan baik | Diterima |
| Tabel pada tab Transaksi | Run aplikasi dan klik tab Transaksi | Run aplikasi dan klik tab Transaksi pada bagian kiri dashboard | Data ditampilkan dengan baik | Data ditampilkan dengan baik | Diterima |
| Tabel pada tab Barang | Run aplikasi dan klik tab Barang | Run aplikasi dan klik tab Barang pada bagian kiri dashboard | Data ditampilkan dengan baik | Data ditampilkan dengan baik | Diterima |
| Tabel pada tab Lokasi | Run aplikasi dan klik tab Lokasi | Run aplikasi dan klik tab Lokasi pada bagian kiri dashboard | Data ditampilkan dengan baik | Data ditampilkan dengan baik | Diterima |
| Tabel pada tab Pelanggan | Run aplikasi dan klik tab Pelanggan | Run aplikasi dan klik tab Pelanggan pada bagian kiri dashboard | Data ditampilkan dengan baik | Data ditampilkan dengan baik | Diterima |
| Fitur searching | Search pada tab Transaksi | Input dengan data yang valid | Melakukan input pada jTextField dengan data yang ada pada database | Tabel hanya menampilkan data yang sesuai dengan input | Tabel hanya menampilkan data yang sesuai dengan input | Diterima |
| Input dengan data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Tabel tidak menampilkan apapun | Tabel tidak menampilkan apapun | Diterima |
| Search pada tab Barang | Input dengan data yang valid | Melakukan input pada jTextField dengan data yang ada pada database | Tabel hanya menampilkan data yang sesuai dengan input | Tabel hanya menampilkan data yang sesuai dengan input | Diterima |
| Input dengan data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Tabel tidak menampilkan apapun | Tabel tidak menampilkan apapun | Diterima |
| Search pada tab Lokasi | Input dengan data yang valid | Melakukan input pada jTextField dengan data yang ada pada database | Tabel hanya menampilkan data yang sesuai dengan input | Tabel hanya menampilkan data yang sesuai dengan input | Diterima |
| Input dengan data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Tabel tidak menampilkan apapun | Tabel tidak menampilkan apapun | Diterima |
| Search pada tab Pelanggan | Input dengan data yang valid | Melakukan input pada jTextField dengan data yang ada pada database | Tabel hanya menampilkan data yang sesuai dengan input | Tabel hanya menampilkan data yang sesuai dengan input | Diterima |
| Input dengan data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Tabel tidak menampilkan apapun | Tabel tidak menampilkan apapun | Diterima |
| Fitur filtering | Filter pada tab Home | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Filter pada tab Transaksi | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Input data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Tabel tidak menampilkan apapun | Tabel tidak menampilkan apapun | Diterima |
| Filter pada tab Barang | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Filter pada tab Lokasi | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Filter pada tab Pelanggan | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Tabel hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Fitur login | Login dengan akun | Login dengan akun pelanggan | Melakukan input dengan username dan password yang terdaftar sebagai pelanggan | Masuk ke dashboard untuk pelanggan | Masuk ke dashboard untuk pelanggan | Diterima |
| Login dengan akun admin | Melakukan input dengan username dan password yang terdaftar sebagai admin | Masuk ke dashboard untuk admin | Masuk ke dashboard untuk admin | Diterima |
| Login dengan akun yang tidak terdaftar | Melakukan input dengan username dan / atau password yang tidak terdaftar | Tampil pesan gagal login | Tampil pesan gagal login | Diterima |
| Fitur shopping pada dashboard pelanggan | Menambahkan item ke keranjang | Menambahkan item ke keranjang dengan jumlah yang valid | Input jumlah pada spinner lalu klik tombol ADD TO CART | Item pada keranjang bertambah sesuai jumlah yang di-input | Item pada keranjang bertambah sesuai jumlah yang di-input | Diterima |
| Menambahkan item ke keranjang dengan jumlah yang tidak valid | Input jumlah pada spinner lalu klik tombol ADD TO CART | Permintaan ditolak | Item pada keranjang bertambah sebanyak satu buah item | Ditolak |
| Menambahkan item yang sama ke keranjang lebih dari satu kali | Input jumlah pada barang yang sudah pernah ditambahkan lalu klik tombol ADD TO CART | Item yang sudah ditambahkan sebelumnya bertambah sesuai jumlah yang di-input | Item yang sudah ditambahkan sebelumnya bertambah sesuai jumlah yang di-input | Diterima |
| Search item | Input dengan data yang valid | Melakukan input pada jTextField dengan data yang ada pada database | Aplikasi hanya menampilkan data yang sesuai dengan input | Aplikasi hanya menampilkan data yang sesuai dengan input | Diterima |
| Input dengan data yang tidak valid | Melakukan input pada jTextField dengan data yang tidak ada pada database | Aplikasi tidak menampilkan apapun | Aplikasi tidak menampilkan apapun | Diterima |
| Filter item | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Aplikasi hanya menampilkan data yang sesuai dengan item yang dipilih | Aplikasi hanya menampilkan data yang sesuai dengan item yang dipilih | Diterima |
| Fitur Keranjang pada dashboard pelanggan | Edit jumlah barang | Mengubah jumlah item dengan jumlah yang valid | Klik tombol Edit Qty dan melakukan input pada jTextField, lalu klik tombol Selesai Edit | Jumlah item berubah sesuai dengan input yang diberikan | Jumlah item berubah sesuai dengan input yang diberikan | Diterima |
| Mengubah jumlah item dengan jumlah yang tidak valid | Klik tombol Edit Qty dan melakukan input pada jTextField, lalu klik tombol Selesai Edit | Permintaan ditolak | Jumlah item berubah sesuai dengan input yang diberikan | Ditolak |
| Delete barang | Menghapus sebuah barang dari keranjang | Klik tombol Delete lalu klik Yes pada confirmation dialog | Barang terhapus dari keranjang | Barang terhapus dari keranjang | Diterima |
| Proses ke transaksi | Melanjutkan ke proses transaksi dengan keranjang yang berisi | Pilih salah satu item pada jComboBox lalu klik tombol Proceed to Transaction | Berpindah ke proses transaksi | Berpindah ke proses transaksi | Diterima |
| Melanjutkan ke proses transaksi dengan keranjang yang tidak berisi | Pilih salah satu item pada jComboBox lalu klik tombol Proceed to Transaction | Permintaan ditolak | Berpindah ke proses transaksi | Ditolak |
| Fitur Transaction pada dashboard pelanggan | Pengisian data alamat pelanggan | Memilih item dari jComboBox | Memilih salah satu pilihan dari jComboBox yang telah disediakan | Postal Code berubah sesuai pilihan | Postal Code berubah sesuai pilihan | Diterima |
| Donasi | Melakukan input dengan jumlah yang valid | Melakukan input pada jTextField | Jumlah bayar berubah sesuai dengan input | Jumlah bayar berubah sesuai dengan input | Diterima |
| Melakukan input dengan jumlah yang tidak valid | Melakukan input pada jTextField | Permintaan ditolak | Text berubah menjadi nol | Diterima |
| Pembayaran | Melakukan pembayaran | Klik tombol Pay | Kembali ke shopping mode dan transaksi tercatat pada log | Kembali ke shopping mode dan transaksi tercatat pada log | Diterima |
| Fitur Edit data pada dashboard admin | Edit data pada tab Barang | Mengubah data dengan data yang valid | Klik pada data yang ingin dirubah, klik tombol Edit, ubah sesuai keinginan, klik tombol Save | Data pada database dan pada aplikasi berubah sesuai input | Data pada database dan pada aplikasi berubah sesuai input, namun pada aplikasi harus di-refresh terlebih dahulu agar data yang baru ditampilkan | Diterima |
| Mengubah data dengan data yang tidak valid | Klik pada data yang ingin dirubah, klik tombol Edit, ubah sesuai keinginan, klik tombol Save | Permintaan ditolak | Compiler menampilkan pesan error dan data tidak berubah | Diterima |
| Edit data pada tab Lokasi | Mengubah data dengan data yang valid | Klik pada data yang ingin dirubah, klik tombol Edit, ubah sesuai keinginan, klik tombol Save | Data pada database dan pada aplikasi berubah sesuai input | Data tidak mengalami perubahan | Ditolak |
| Edit data pada tab Pelanggan | Mengubah data dengan data yang valid | Klik pada data yang ingin dirubah, klik tombol Edit, ubah sesuai keinginan, klik tombol Save | Data pada database dan pada aplikasi berubah sesuai input | Data pada database dan pada aplikasi berubah sesuai input, namun pada aplikasi harus di-refresh terlebih dahulu agar data yang baru ditampilkan | Diterima |
| Fitur Delete data pada dashboard admin | Delete data pada tab Transaksi | Menghapus sebuah data pada tabel | Klik pada data yang ingin dihapus, klik tombol Delete | Data yang dihapus hilang dari database dan aplikasi | Data yang dihapus hilang dari database dan aplikasi, namun pada aplikasi harus di-refresh terlebih dahulu agar data yang baru ditampilkan | Diterima |
| Delete data pada tab Barang | Menghapus sebuah data pada tabel | Klik pada data yang ingin dihapus, klik tombol Delete | Data yang dihapus hilang dari database dan aplikasi | Data yang dihapus hilang dari database dan aplikasi, namun pada aplikasi harus di-refresh terlebih dahulu agar data yang baru ditampilkan | Diterima |
| Delete data pada tab Lokasi | Menghapus sebuah data pada tabel | Klik pada data yang ingin dihapus, klik tombol Delete | Data yang dihapus hilang dari database dan aplikasi | Compiler menampilkan pesan error dan data tidak terhapus | Ditolak |
| Delete data pada tab Pelanggan | Menghapus sebuah data pada tabel | Klik pada data yang ingin dihapus, klik tombol Delete | Data yang dihapus hilang dari database dan aplikasi | Data yang dihapus hilang dari database dan aplikasi, namun pada aplikasi harus di-refresh terlebih dahulu agar data yang baru ditampilkan | Diterima |
| Fitur Add data pada dashboard admin | Add produk pada tab Barang | Menambahkan sebuah produk | Klik tombol Add Product, isi form sesuai keinginan, klik tombol Create | Data pada database dan pada aplikasi bertambah sesuai input | Compiler menampilkan pesan error dan data tidak bertambah | Ditolak |

# BAB V PENUTUP

## Kesimpulan

Perkembangan *e-commerce* saat ini sangat berkembang, telah banyak dibuktikan dengan makin maraknya *e-commerce* baru yang menyajikan berbagai layanan jual beli yang mempermudah manusia dalam melakukan transaksi jual beli berbagai jenis barang. Meningkatnya transaksi yang terjadi di ­­*e-commerce* tentunya dapat dimanfaatkan dengan menyediakan suatu *e-commerce* yang menyediakan fitur donasi yang dapat meningkatkan perekonomian masyarakat (penjual) sekaligus membantu pihak-pihak yang memang membutuhkan untuk meningkatkan taraf hidupnya menjadi lebih baik.

Dengan adanya aplikasi Cartrity, diharapkan dapat membuat donasi menjadi sebuah kebiasaan di dalam masyarakat sekaligus membantu orang-orang untuk melakukan jual-beli secara mudah, cepat, dan tepat.

## Saran Pengembangan

Aplikasi yang dibuat tentunya tidak akan lepas dari kekurangan yang dapat diperbaiki pada pengembangan selanjutnya. Proses pengembangan memang sudah menjadi alur dari *software development life cycle* (SDLC) yang harus dilewati setelah suatu program atau aplikasi dibuat, sehingga penulis merasa perlu menambahkan sub bab saran pengembangan untuk mempermudah pihak manapun yang hendak melakukan pengembangan pada aplikasi Cartrity yang telah dibuat sebelumnya. Berikut adalah beberapa saran pengembangan yang diberikan oleh penulis beserta tim pengembang aplikasi Cartrity.

1. Melakukan pembuatan versi *platform* website aplikasi Cartrity untuk mempermudah pengguna dalam mengakses aplikasi Cartrity.
2. Melakukan penambahan pengelolaan akun pengguna dengan hak akses donatur, sehingga donatur dapat melakukan kostumisasi terhadap akun yang dimiliki.
3. Menambahkan fitur lupa *password* untuk mengantisipasi pengguna yang hendak melakukan *login* namun lupa *password*.
4. Menambahkan fitur daftar untuk donatur baru.
5. Menyeragamkan bahasa yang digunakan pada aplikasi.
6. Ubah mata uang yang digunakan ke dalam bentuk Rupiah.

# LAMPIRAN

Lampiran 1 : *Class Model*

1. ModelBarang.java

public class ModelBarang {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String productID;

private ModelSubKategori subcategory;

private String productName;

private double price;

private String imageSource;

}

1. ModelDaerah.java

public class ModelDaerah {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String postalCode;

private String region;

private String country;

private String city;

private String state;

}

1. ModelDonation.java

public class ModelDonation {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String donationID;

private ModelPelanggan pelanggan;

private double jmlDonasi;

}

1. ModelKategori.java

public class ModelKategori {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String id\_kategori;

private String kategori;

}

1. ModelKeranjang.java

public class ModelKeranjang {

private ModelBarang barang;

private int qty;

private double total;

}

1. ModelPelanggan.java

public class ModelPelanggan {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String customerID;

private String customerName;

private String segment;

}

1. ModelShipMode.java

public class ModelShipMode {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String shipID;

private String shipMode;

private double price;

}

1. ModelSubKategori.java

public class ModelSubKategori {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String id\_sub;

private ModelKategori kategori;

private String subKategori;

private String imageSource;

}

1. ModelTransaksi.java

public class ModelTransaksi {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String orderID;

private Date orderDate;

private Date shipDate;

private ModelShipMode shipMode;

private ModelPelanggan pelanggan;

private ModelDaerah postal;

private ModelBarang product;

private double sales;

private int quantity;

private double discount;

private double profit;

private Double donation;

private double total;

}

1. ModelUser.java

public class ModelSubKategori {

public static final String FILE\_NAME = "\*/../src/Excel/DataTransaksi.xlsx";

private String username;

private String password;

private String userID;

}

# DAFTAR PUSTAKA

Badang Pengembangan dan Pembinaan Bahasa Kementrian Pendidikan dan Kebudayaan Indonesia. KBBI Daring. <https://kbbi.kemdikbud.go.id/entri/Jual%20Beli>. [2 Juli 2018]

Heyder Affan. Ketimpangan sosial di Indonesia meningkat: Apakah maknanya bagi rakyat biasa?. <http://www.bbc.com/indonesia/indonesia-42989531>. [30 Juni 2018]

M. Sidi Mustaqbal, dkk. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. <http://jitter.widyatama.ac.id/index.php/jitter/article/view/70/50>. [2 Juli 2018]

Muhammad Sufyan Abdurrahman. Pertumbuhan e-Commerce Indonesia Tertinggi di Dunia. <https://www.liputan6.com/tekno/read/2957050/pertumbuhan-e-commerce-indonesia-tertinggi-di-dunia>. [30 Juni 2018]

Risky Maulana. KitaBisa Alami Pertumbuhan Donasi Hingga 230 Persen di Tahun 2017. <https://id.techinasia.com/pencapaian-kitabisa-2017>. [3 Juli 2018]