BAB I

PENDAHULUAN

A. Latar Belakang

Dengan perkembangan teknologi dan kebutuhan penyebaran informasi yang lebih cepat, penggunaan komputer dan internet juga telah diterapkan pada bidang perdagangan. Dengan menggunakan komputer ini, Anda dapat menyelesaikan perdagangan komoditas tatap muka yang biasa dilakukan di dunia maya (melalui *Internet*). Ini akan menghemat waktu dan tenaga Anda dalam mencari barang yang Anda butuhkan. Hal ini juga berlaku untuk perusahaan yang bergerak di bidang penjualan pakaian. Sistem penjualan yang ada saat ini adalah setiap pembeli harus datang langsung ke toko untuk melakukan transaksi pembelian sandang. Selain tidak tertata dengan baik dalam proses pembuatan laporan dan penyimpanan data, hal ini mengakibatkan biaya yang relatif tinggi dan ruang terbatas yang berujung pada penumpukan atau hilangnya data transaksi penjualan.

Salah satu media alternatif untuk meningkatkan *volume* penjualan dengan biaya yang relatif lebih efektif adalah dengan menggunakan fasilitas *Internet*, yaitu salah satu bentuk layanan yang dapat digunakan melalui *internet*, yaitu melalui promosi dan penjualan produk terbaru secara cepat dan mudah. Mengingat pentingnya hal diatas maka diperlukan juga dukungan sistem informasi yang cepat dan akurat. Oleh karena itu untuk mengatasi masalah tersebut penulis menggunakan bahasa pemograman berbasis *android*. Karena dengan pemograman berbasis

android maka dapat menghubungkan banyak jaringan dalam suatu wilayah, negara atau wilayah. Informasi yang ditampilkan lebih mudah diperbarui antar wilayah, dan bahkan antar negara, sehingga beban biaya dan efisiensi waktu dapat dikurangi.

Berdasarkan uraian diatas bahwa sistem informasi sangat diperlukan dalam suatu perusahaan sehingga penulis melakukan penelitian dan menuangkan dalam bentuk skripsi dengan judul "**SISTEM INFORMASI PENYEDIA JASA PENJUALAN PAKAIAN BERBASIS *ANDROID***".

B. Rumusan Masalah

Berdasarkan latar belakang tersebut, maka dapat dirumuskan bahwa masalah yang ada :

1. Bagaimana cara merancang sebuah aplikasi sistem informasi penyedia jasa penjualan pakaian berbasis *android.*
2. Bagaimana membuat aplikasi yang dapat mempermudah interaksi jual beli antara calon pembeli dengan penjual?

C. Tujuan Penelitian

Tujuan pembuatan aplikasi ini adalah bagaimana kita bisa membantu seseorang yang ingin mencari dan menjual barang, dimana mereka masih sering kesulitan menemukan barang-barang atau jasa yang hendak mereka cari, maka dari itu kiranya aplikasi ini dapat membantu orang-orang tersebut untuk menpermudah menemukan apa yang mereka ingin cari.

D. Batasan Masalah

Adapun batasan masalah dalam judul skripsi ini sebagai berikut :

1. Aplikasi ini hanya di khususkan untuk pakaian.
2. Aplikasi ini hanya dapat digunakan oleh *user* yang telah mendaftar.
3. Aplikasi ini hanya menampilkan iklan dari penjual.
4. Aplikasi ini tidak melayani masalah transaksi penjualan

E. Manfaat Penelitian

Adapun manfaat penelitian dari perancangan Sistem Informasi Penyedia Jasa Penjualan Pakaian Berbasis *Android* sebagai berikut :

1. Manfaat yang diperoleh dari penelitian ini dapat membantu masyarakat dalam melakukan transaksi jual beli pakaian secara *online*.
2. Bagi penulis bermanfaat sebagai wadah untuk menerapkan dan mengimplementasikan ilmu yang diperoleh selama perkuliahan agar menjadi bekal didunia kerja.
3. Bagi akademik bermanfaat sebagai materi evaluasi bagi pengembangan peningkatan pendidikan dimasa yang akan datang .

F. Sistematika Penulisan

Sistem penulisan yang digunakan dalam penelitian ini adalah:

Bab I : Pendahuluan

Bab ini merupakan pengantar yang menguraikan secara umum tentang latar belakang masalah, rumusan masalah, tujuan penulisan, lingkup dan batasan penelitian, dan sistematika penulisan.

Bab II : Landasan Teori

Dalam bab ini disajikan teori-teori berdasarkan penelitian kepustakaan yang relevan dengan pokok persoalan dalam penelitian ini.

Bab III : Metodologi Penelitian

Bab ini mengemukakan tentang pendekatan dari teori-teori, kemudian diuraikan menjadi suatu usulan pemecahan masalah.

Bab IV : Hasil dan Pembahasan

Bab ini mencakup proses pengolahan dan hasil analisis dari data yang telah dikumpulkan, yang selanjutnya dibahas dan dilakukan interpretasi.

Bab V : Kesimpulan dan Saran

Bab ini membahas kesimpulan berdasarkan hasil evaluasi dan pembahasan yang telah dilakukan, serta memberikan saran bagi penelitian selanjutnya untuk memperhatikan beberapa keterbatasan dalam penelitian ini.

BAB II

LANDASAN TEORI

A. Aplikasi

Aplikasi adalah *sub* kategori perangkat lunak komputer, yang dapat langsung menggunakan fungsi komputer untuk melakukan tugas-tugas yang diperlukan oleh pengguna. Bisa juga dikatakan bahwa program aplikasi menerjemahkan perintah-perintah yang dijalankan oleh pengguna komputer untuk diteruskan atau diolah oleh perangkat keras.

Program aplikasi (biasanya disebut perangkat lunak) adalah program komputer yang konten pengajarannya dapat dengan mudah diubah. Program aplikasi biasanya digunakan untuk mengontrol perangkat keras (biasanya disebut *driver* perangkat), melakukan kalkulasi, dan berinteraksi dengan aplikasi lain yang lebih mendasar (seperti sistem operasi dan bahasa pemrograman). Secara umum, aplikasi dapat dibagi menjadi 3 level, yaitu *level* aplikasi (misalnya aplikasi seperti *Microsoft Office*), *level* sistem operasi (misalnya, sistem operasi seperti *Microsoft Windows*), dan *level* bahasa pemrograman (misalnya , *PHP*).

B. *Android*

Android adalah sistem operasi perangkat seluler berbasis *Linux*, yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* memberi pengembang *platform* terbuka untuk membuat aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* Ini adalah pemula dalam membuat perangkat lunak *ponsel* / *ponsel* pintar. Kemudian mengembangkan *Android* dan mendirikan *Open Mobile Phone*

Alliance, yang terdiri dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. *Android* memiliki 4 ciri khas, yaitu sebagai berikut:

1. Terbuka, *Android* dibangun agar benar-benar terbuka, sehingga aplikasi dapat memanggil fungsi inti ponsel, seperti melakukan panggilan, mengirim pesan teks, menggunakan kamera, dll. *Android* adalah "Jurnal Informatika", *Volume* 8, Edisi 1, Juni 2019 ISSNOnline 2528-0090 Program Penelitian Teknik Informatika Universitas Dananu 34 http://ejournal.unidayan.ac.id/index.php Mesin *virtual* / JIU dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras di perangkat. *Android* adalah *open source* dan dapat diperluas secara bebas untuk menyertakan teknologi baru dan yang lebih canggih. *Platform* tersebut akan terus berkembang untuk membangun aplikasi seluler yang inovatif.
2. Semua aplikasi adalah sama, *Android* dan *ponsel* serta aplikasi pihak ketiga *(third-party application)* tidak berpengaruh pada aplikasi utama. Semua aplikasi dapat dibangun untuk menyediakan berbagai layanan dan kemampuan aplikasi kepada pengguna menggunakan *ponsel* dengan pijakan yang setara.
3. Mendobrak batasan aplikasi, *Android* mendobrak batasan dalam membangun aplikasi baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari *Web* dengan data di ponsel

seseorang (seperti kontak, kalender, atau lokasi geografis pengguna) untuk mengembangkan aplikasi dengan cepat dan mudah.

4. *Android* memberi pengguna rentang izin akses yang lebih luas untuk menggunakan aplikasi yang terus meningkat. *Android* memiliki serangkaian alat yang dapat digunakan untuk membantu pengembang meningkatkan efisiensi kerja saat membangun aplikasi.

C. *Javascript*

Javascript adalah bahasa *scripting*, pertama kali dikembangkan oleh *Netscape* pada tahun 1995. *Javascript* ditulis dalam dokumen *HTML*, dan pemanggilan program bergantung pada *browser* (navigator) yang digunakan untuk memanggil halaman yang terdapat dalam skrip. *Javascript* juga membutuhkan kompiler atau penerjemah untuk menjalankannya

Mac Bride juga menjelaskan hal ini dalam bukunya "*Javascript*". Menurut *Bride* , *Javascript* adalah bahasa pemrograman berbasis *browser*. Kode-kode ini langsung ditulis ke dalam *HTML* halaman *web*, dan diterjemahkan serta dijalankan sesuai dengan aktivitas di halaman *web*.

Karakteristik *Javascript*.Menurut Sidik , *Javascipt* memuat konten spesifik sebagai berikut:

1. Gunakan blok awal "{" dan blok akhir "}".
2. Konversi otomatis dalam pengoperasian tipe data yang berbeda.
3. Peka huruf besar / kecil, jadi *programmer* harus berhati-hati saat menggunakan *variabel*, fungsi, dll.
4. Ekstensi *file* yang umum digunakan adalah "* .js".

5. Setiap kalimat bisa diakhiri dengan ";" tapi bisa atau mungkin tidak.
6. Jika *browser* lama tidak mendukung skrip, Anda dapat menyembunyikan *skrip* di antara tag "<! -" dan "->".
7. Jika program dalam satu baris terlalu panjang, Anda dapat melanjutkan ke baris berikutnya dengan karakter "\" di awal baris.

Menurut definisi *Javascript* yang diberikan, *Bride* lebih lanjut mendeskripsikan *Javascript* sebagai berikut:

1. *Javascript* berinteraksi dengan *browser*. *Javascript* dapat membaca informasi dari halaman *web*, seperti data yang dimasukkan melalui bidang dalam formulir.
2. *Javascript* adalah bahasa *event-driven* berbasis *objek*. *Javascript* memberikan respons untuk aktivitas yang dilakukan pada objek di halaman *web*.

Javascript bukanlah bahasa yang rumit dan memiliki keterbatasan dalam penggunaannya. Misalnya, *Javascript* dapat digunakan tidak hanya untuk menampilkan kotak pesan atau pemberitahuan saat tombol diklik. *Javascript* juga digunakan untuk aplikasi permainan interaktif yang kompleks atau memeriksa pesanan di situs belanja *online*. Satu-satunya batasan adalah bahwa setiap input atau output dari aplikasi dengan *Javascript* hanya dapat dilakukan / disediakan melalui *browser web*.

D. *Framework7*

Framework7 adalah *framework HTML* gratis dan *open-source* (sumber terbuka) untuk membangun aplikasi *mobile hybrid* atau aplikasi web dengan *iOS* dan *Android* agar terlihat dan terasa lebih asli (*native*).

Framework ini juga merupakan alat aplikasi *prototipe* yang sangat diperlukan untuk menunjukkan *prototipe* aplikasi yang berfungsi dengan sangat cepat jika diperlukan.

Pendekatan utama dari *Framework7* adalah memberikan kesempatan kepada para *developer* untuk menciptakan aplikasi *iOS* dan *Android* dengan *HTML*, *CSS*, dan *JavaScript* dengan lebih mudah dan jelas. *Framework7* memberi Anda kebebasan dalam menciptakan aplikasi.

Framework ini tidak membatasi Anda dalam menciptakan aplikasi dan menawarkan berbagai cara atau solusi dari berbagai permasalahan pengembangan.

Dengan rangkaian komponen *UI* yang menakjubkan yang disediakan *Framework7* langsung dari kotaknya, ini memungkinkan untuk membuat aplikasi *web*, aplikasi *web progresif* atau *Progresive Web App* (*PWA*) dan aplikasi *iOS* dan *Android* dengan tampilan dan nuansa asli. *Framework7* yang dipasangkan dengan alat tambahan seperti *Electron* dan *NW.js* memungkinkan untuk membangun aplikasi *desktop* asli.

Framework7 merupakan ekosistem yang kaya, karena selain *library JavaScript*, *Framework7* hadir dengan komponen *Vue.js*, *React* dan *Svelte* untuk menghadirkan komponen-sintaksis, data terstruktur, dan pengikatan data dengan kekuatan dan kesederhanaan *Vue.js*, *React* atau *Svelte*.

Fitur-fitur yang dimiliki *Framework7* adalah sebagai berikut :

1. *Framework7* adalah *framework* dengan sumber terbuka *(open source)* dan dapat digunakan secara gratis.
2. *Framework7* memiliki *syntax* yang mudah dan mirip dengan *jQuery* untuk memulai tanpa penundaan.
3. *Framework7* memiliki pustaka built-in *FastClick* yang berguna untuk melakukan kontrol terhadap penundaan klik ketika menyentuh UI.
4. *Framework7* memiliki sistem *grid layout built-in* untuk mengatur elemen Anda secara *responsive.*
5. *Framework* dapat secara dinamis memuat halaman dari *template* melalui *flexible router api* (*router* api yang *fleksibel*).

Kelebihan yang dimiliki oleh *Framework7* :

1. Lebih mudah mengembangkan aplikasi untuk *iOS* dan *Android*.
2. Kurva pembelajaran untuk *Framework7* sangat mudah.
3. *Framework7* memiliki banyak *widget* / komponen *pre-style*.
4. *Framework7* memiliki perpustakaan pembantu *built-in*.
5. *Framework7* tidak bergantung pada pustaka pihak ketiga mana pun bahkan untuk manipulasi *DOM*. Sebaliknya, ia memiliki *DOM7* khusus sendiri.
6. *Framework7* juga dapat digunakan dengan *framework Vue, React* dan *Svelte*.
7. Anda dapat mulai membuat aplikasi setelah mengetahui *HTML, CSS*, dan beberapa *JavaScript* dasar.
8. *Framework7* mendukung pengembangan yang lebih cepat melalui *Browser*.

9. Mudah untuk mengembangkan aplikasi untuk *iOS* dan *Android* tanpa mempelajarinya.

Kekurangan *Framework7*

Namun, *Framework7* App tidak kompatibel dengan berbagai *platform.* Aplikasi ini fokus pada *IOS* dan desain *Google Material* untuk membawa pengalaman terbaik dan simple. Aplikasi ini sangat direkomendasi untuk para *developer* yang familiar dengan dasar-dasar *HTML, CSS,* dan *JavaScript* untuk memulai pembuatan aplikasi dengan *Framework7*.

Dokumentasi yang disediakan oleh *Framework*7 tidak mencakup proses kompilasi aplikasi *Framework7* ke aplikasi *Cordova*. Dokumentasi yang ada saat ini lebih tentang *Framework7* dan bagaimana menggunakan semua komponennya.

Untuk mempelajari lebih lanjut tentang cara menyusun aplikasi *Framework7* ke aplikasi *Cordova hybrid* yang dapat dirilis di *IOS* dan *Google Play Store*, lihat Dokumentasi Resmi *Cordova*. *Framework7 Kitchen Sink* juga merupakan tempat yang tepat untuk memulai, karena memiliki banyak contoh untuk semua komponen *Framework7* dan mencakup sebagian besar aspek.

Kekurangan lainnya adalah, dukungan komunitas *online* untuk kerangka *Framework7* kurang dibandingkan dengan *IOS* dan *Android.*

E. *PHP (Hypertext Preprocessor)*

(*Hypertext Preprocessor*) PHP adalah bahasa pemrograman yang banyak digunakan untuk menangani pembuatan dan pengembangan situs *web*, dan dapat digunakan bersama dengan *CSS* dan *HTML*. *PHP* pertama kali dibuat pada tahun 1994 oleh *Rasmus Lerdoff*. Awalnya, *PHP* adalah singkatan dari "alat beranda

pribadi". Kemudian diubah menjadi FI ("*Form Interpreter*"). Mulai dari versi 3.0, nama bahasa telah diubah menjadi "*PHP* (*Hypertext Preprocessor*)".

Pada tahun 1997, sebuah perusahaan bernama *Zend* menulis ulang penerjemah *PHP* hingga empat kali direvisi. Pada bulan Juni 2004, *Zend* merilis *PHP* 5.0. Dalam versi ini, inti dari *interpreter PHP* telah mengalami perubahan besar. Versi ini juga mengintegrasikan model pemrograman berorientasi objek ke dalam *PHP* sebagai respons terhadap perkembangan bahasa pemrograman hingga paradigma berorientasi objek.

Sistem database yang dapat didukung oleh *PHP* adalah *Oracle*, *MySQL*, *Ms.Access, Sybase* dan *PostgreSQL*. PHP dapat berjalan di berbagai system operasi seperti *windows* 98/NT, UNIX/LINUX, solaris maupun *macintosh.* Keunggulan lainnya dari *PHP* adalah *PHP* juga mendukung komunikasi dengan layanan seperti *protocol* IMAP, SNMP, NNTP, POP3 bahkan HTTP.

Setiap bahasa pemrograman memiliki aturan *coding* sendiri. Begitu pula dengan PHP. *Sintaks* dasarnya dibuka dengan **<?php** dan ditutup dengan **?>** sebagai terlihat di contoh berikut:

```
<?php
echo “Selamat datang”;
?>
```

Gambar 2.1 *Script PHP*

F. *MySQL*



Gambar 2.2 *MySQL*

Sumber (www.wikimedia.org)

MySQL adalah perangkat lunak sistem manajemen basis data *SQL* (*database management system*) atau *DBMS*, yang bersifat *multi-threaded* dan *multi-user*. *MySQL* adalah sistem manajemen basis data *relasional* (*RDBMS*), didistribusikan secara gratis di bawah *GPL* (Lisensi Publik Umum). Setiap orang bebas menggunakan *MySQL*, tetapi tidak dapat digunakan sebagai sumber tertutup atau produk turunan komersial. *MySQL* sebenarnya merupakan turunan dari *SQL* (*Structured Query Language*), salah satu konsep utama dalam *database* lama.

SQL merupakan konsep operasi basis data, khususnya konsep operasi basis data yang digunakan untuk memilih atau memilih dan memasukkan data, sehingga operasi data dapat diselesaikan dengan mudah dan otomatis. Keunggulan sistem database (*DBMS*) dapat dilihat dari cara pengoptimal memproses perintah SQL yang dibuat oleh pengguna dan aplikasinya. Sebagai *server database*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *MySQL* bisa

sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase.*

G. Php*Myadmin*



Gambar 2.3 phpMyAdmin

Sumber (www.wikipedia.org)

PhpMyAdmin adalah aplikasi *open source* yang digunakan untuk memfasilitasi administrasi *MySQL*. Dengan menggunakan *phpmyadmin*, kita bisa menggunakan *GUI* untuk membuat *database*, membuat tabel, menyisipkan, menghapus dan meng*update* data, dan rasanya lebih mudah tanpa harus mengetik perintah *SQL* secara manual.

H. *Domain*

Domain adalah sebuah nama atau alamat yang dipakai untuk mengunjungi suatu *website*. *Level domain* ada 3, yaitu :

1. *Top Level Domain (TLD)* adalah suatu ekstensi atau akhiran suatu nama unik *domain* tersebut. Contohnya, *.com* (*dotcommercial*), *.net* (*dotnetwork*), *.org* (*dotorganization*), dan lain-lain. *Domain* TLD ada dua macam, yaitu *Global Top Level Domain (TLD)* dan *Country Code Top Level Domain* (ccTLD). Untuk gTLD, adalah *domain-domain* yang sangat

umum digunakan secara *global* (banyak orang) seperti contoh *domain* di atas. Sedangkan untuk ccTLD, adalah TLD yang diperuntukkan untuk masing-masing negara. Contohnya seperti *domain* Indonesia dengan kode ID, misalnya, *.co.id, .net.id,* dan *.or.id*. Setiap negara memiliki kode TLD masing-masing.

2. *Second Level Domain* adalah nama *domain* yang kita daftarkan. Misalnya nama *domain* yang akan didaftarkan adalah KaosSmk5.net, maka KaosSmk5 adalah SLDnya dan *.net* adalah TLDnya.
3. *Third Level Domain* adalah nama sebelum *Second Level Domain*, contohnya *support.* kaos.net. dari contoh tersebut, *Third* Level *Domain*nya adalah *support,* SLDnya adalah kaos dan TLDnya adalah .net. *Third Level Domain* biasa dikenal atau disebut dengan *"subdomain".* (Wahana Komputer, 2015)

I. *Hosting*

Secara teknis *hosting* merupakan tempat penyimpanan *file-file* atau data dari sebuah *website*. Untuk itulah, jika kita membuat *website*, *domain* saja tidak cukup. Dengan adanya *hosting*, *file* apa saja yang kita *upload*, dapat diakses oleh pengguna *internet* di seluruh dunia melalui situs *web* kita di *internet. Hosting* diibaratkan sebuah rumah yang dipenuhi dengan barang-barang penting yang saling berguna. *Hosting* ada 2 jenis, yakni *hosting* berbayar dan *hosting* gratis.

Pengguna lebih banyak menggunakan *hosting* berbayar, karena fiturnya lebih lengkap dari pada yang gratis. Mengenai biaya, *hosting* menggunakan sistem sewa sama seperti *htm*. Biaya *hosting* tergantung dari fitur dan spesifikasi yang kita

pakai. Semakin besar fitur dan spesifikasi *hosting*, semakin besar pula tarif sewanya. Biasanya fitur dan spesifikasi *hosting* yang besar ini, digunakan oleh perusahaan-perusahaan besar. Harga sewa *hosting* juga tergantung dari penyedia layanan yang kita pakai. Kadang ada yang murah, adapula yang mahal, bahkan sangat mahal. Penyedia *hosting* yang cukup terkenal di *internet* seperti *AsianBrainHosting.com*, dan *Hostagor.com.*

J. HTML

Hypertext Markup Language (HTML) adalah bahasa *markup* yang digunakan untuk membuat halaman *web*, menampilkan berbagai informasi di *browser Web Internet* dan format *hypertext* sederhana yang ditulis dalam *file* format *ASCII* untuk menghasilkan tampilan terintegrasi. Dengan kata lain, ini adalah *file* yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCII* normal, sehingga *file* tersebut menjadi halaman *web* dengan perintah *HTML*.

Mulai dari *SGML* (*Standard General Markup Language*), bahasa yang banyak digunakan dalam penerbitan dan percetakan, *HTML* merupakan standar yang banyak digunakan untuk menampilkan halaman *web*. *HTML* saat ini merupakan standar *Internet* yang ditentukan dan dikendalikan oleh *Konsorsium Internet* (W3C). *HTML* dibuat oleh *Caillau TIM* dan *Berners-lee Robert* pada tahun 1989 bekerja sama dengan Organisasi Eropa untuk Riset Nuklir (*CERN*).

Berikut adalah contoh dokumen sederhana HTML:

```
<html>
<head>
<title>HyperText Markup Language </title>
</head>
<body>
```

```
<p> <i>HyperText Markup Language</i> (HTML) merupakan suatu bahasa markah yang
digunakan untuk membuat halaman web, menampilkan berbagai informasi di dalam sebuah  <i>
browser web</i></p>
</body>
</html>
```

Gambar 2.4 Contoh *HTML*

K. CSS

Ascading style sheets atau *CSS* merupakan aturan untuk menyusun beberapa komponen dalam suatu *web* agar dapat lebih terstruktur dan teratur. *CSS* bukanlah bahasa pemrograman, sama seperti gaya dalam aplikasi pengolah kata seperti *Microsoft Word*, *CSS* menetapkan berbagai gaya, seperti *heading*, *section*, *body*, *footer*, gambar, dan gaya lainnya. Agar dapat digunakan bersama dalam banyak *file*, *CSS* sering digunakan untuk memformat tampilan halaman *web* yang dibuat dalam *HTML*.

CSS dapat mengontrol ukuran gambar, warna tubuh dalam teks, warna tabel, ukuran batas, warna batas, warna *hyperlink*, warna *hover mouse*, jarak antar paragraf, spasi antar teks, margin kiri, kanan, atas dan bawah dan parameter lainnya. *CSS* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan menggunakan *CSS*, kita dapat menampilkan halaman yang sama dalam format yang berbeda.

L. *Apache Cordova*

Apache Cordova adalah *platform* yang menggunakan *HTML*, *CSS*, dan *JavaScript* untuk membangun *native mobile applications*. Aplikasi seluler asli yang didukung termasuk *Android*, *iOS*, *Windows Phone*, dan *Blackberry*.

Apache Cordova menyertakan satu *set API* (antarmuka pemrograman aplikasi) untuk mengakses perangkat dari perangkat seluler. Perangkat ini termasuk kamera, *GPS* (*Global Positioning System*), perangkat penyimpanan, dan perangkat lain. Dengan menggunakan *framework* UI (*User Interface*) (seperti *jQuery Mobile*, *Dojo Mobile* atau *Sencha Touch*), kita dapat mengakses *API* ini. Dengan kata lain kita hanya bisa membangun aplikasi dengan menggunakan *HTML*, *CSS* dan *Javascript*.

Dengan menggunakan *API Cordova*, kita tidak perlu menggunakan kode *native* (seperti *Java*, *Objective-C*, dll.) Untuk membangun aplikasi. Kami hanya menggunakan teknologi *Web* dan bahasa pemrograman *Web* bersama-sama dan menginstalnya di perangkat seluler terkait. Karena menggunakan *Javascript*, aplikasi yang kita bangun pada suatu *platform mobile* dapat digunakan di *platform* lain dengan sedikit atau tanpa perubahan.

Aplikasi yang dihasilkan dari *Cordova* adalah dikemas dalam aplikasi menggunakan *SDK* masing-masing *platform* dan dapat diterapkan ke dalam platform lain menggunakan *SDK platform* tersebut.

Cordova cocok untuk:

1. Seorang *programmer* untuk perangkat *mobile* yang *muti platform*, cukup membuat sekali *coding* tanpa harus melakukan implementasi ulang pada setiap *platform*.
2. Seorang *web developer* di mana ada aplikasi yang perlu dijalankan pada perangkat *mobile* untuk mengakses aplikasi yang dibuatnya.

M. *Internet*

Internet atau jaringan interkoneksi adalah sistem komunikasi *global* yang menghubungkan komputer *global* dan jaringan komputer. Setiap komputer dan jaringan terhubung secara langsung atau tidak langsung ke beberapa jalur utama yang disebut tulang punggung *Internet*. Tujuan utama kerja *Internet* adalah untuk memaksimalkan interoperabilitas, yaitu memaksimalkan kemampuan berbagai sistem dan program komputer di jaringan untuk berkomunikasi secara handal dan efisien.

N. UML (*Unified Modelling Language*)

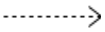
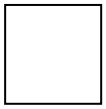
Unified Modeling Language (*UML*) adalah bahasa pemodelan *universal* dan pengembangan dalam rekayasa perangkat lunak, yang bertujuan untuk menyediakan metode standar desain sistem *visual*.

UML pada awalnya didorong oleh keinginan untuk membakukan berbagai simbologi dan metode perangkat lunak. Metode dan metode dikembangkan oleh *Grady Booch*, *Ivar Jacobson*, dan *James Rumbaugh* dari *Rational Software* dari tahun 1994 hingga 1995, dan dipimpin oleh mereka hingga tahun 1996. Pengembangan lebih lanjut .

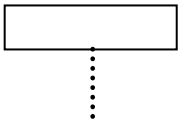
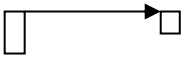
Pada tahun 1997, *UML* diadopsi sebagai standar oleh *Object Management Group* (*OMG*) dan telah dikelola oleh organisasi sejak saat itu. Pada tahun 2005, *UML* juga diterbitkan sebagai standar *ISO* yang disetujui oleh Organisasi Internasional untuk Standardisasi (*ISO*), dan secara teratur direvisi untuk mencakup versi terbaru *UML*.

Adapun daftar simbol UML yaitu :

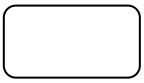
Tabel 0.1 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri*(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 0.2 Simbol *Sequnce Diagram*

No	Gambar	Nama	Keterangan
1.		*LifeLine*	Objek *entity*, antarmuka yang berinteraksi.
2.		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Tabel 0.3 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali.
3		*Final State*	Bagaimana objek di bentuk dan dihancurkan.
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 0.4 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek di bentuk atau diawali
4		*Activiti final Node*	Bagaimana objek di bentuk dan di hancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

O. *Black Box*

Black Box adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian *Black Box*, mengevaluasi hanya dari tampilan luarnya (*interface*-nya), fungsionalitasnya tanpa mengetahu apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output).*

P. *White Box*

Untuk menguji sistem informasi yang dirancang, maka digunakan pendekatan *White Box* yang merupakan salah satu metode pengujian yang menggunakan *struktur control design proscedure* untuk memperoleh *test case*. Jadi dengan menggunakan metode ini, penulis dapat mengetahui cara kerja

sistem yang dirancang secara terperinci sesuai spesifikasi dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah sesuai dengan baik dan benar.

Dengan pengujian ini dapat dipastikan bahwa:

1. Semua jalur *independent patch*telah digunakan minimal satu kali
2. Telah menggunakan semua keputusan logika pada kedua sisinya baik itu *true* atau *false*
3. Telah mengeksekusi semua *loop* pada batasan maksimal yang mungkin terjadi.
4. Menggunakan struktur data internal untuk menjamin validasi.

Sebagai alternatif pengembangan *White Box Testing*, digunakan metode Basis *Patch Testing.* Metode *Basis Patch* digunakan untuk menentukan ukuran kompleksitas logika (*Logika Complecitu Measure*) dari suatu desain. Ukuran kompleksitas logika berguna untuk menentukan banyaknya pengujian yang dilakukan untuk menjaring semua jalur diuji setidaknya sekali. Teknik ini mempunyai 3 langkah dalam pelaksanaannya yaitu:

1. Teknik prosedural dengan menggambarkan *flowgraph* perancangan prosedural diberi nomor untuk memudahkan pembuatan *flowgraph.*
2. Menentukan *Cyclomatic Complexity* untuk *flowgraph* yang dibuat. dengan rumusan V(G) = E – N + 2, dimana V(G) adalah jumlah *region* pada *flowgraph.*
3. Menentukan *independent patch* pada *flowgraph*. *Independent patch* adalah jalu yang melintas dalam sistem dimana sekurang-kurangnya tedapat ada *flowgraph,* E adalah jumlah *edge,* dan N adalah jumlah *node*

pada proses perintah yang baru. Dalam istilah *flowgraph, independent patch* harus bergerak sekurang-kurangnya pada *edge*, dimana pada *edge* yang akan dilewatkan tidak dapat dilewati *edge* sebelum jalur tersebut didefenisikan.

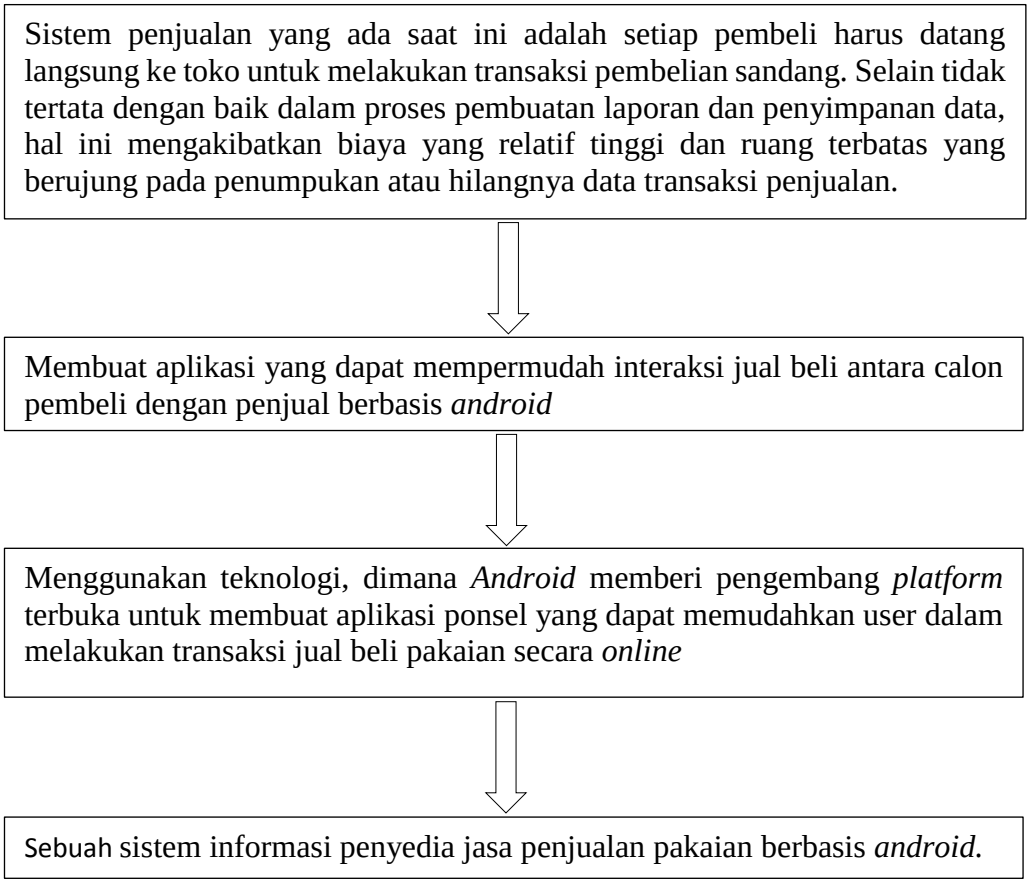
Berdasarkan penjelasan diatas, terlihat bahwa *Basis Patch Texting* ini berguna untuk:

1. Mengukur kompleksitas *logic* dari *design procedure* dan menggunakannya sekaligus sebagai pedoman untuk mendapatkan konsisten dari jalur aplikasi.
2. Pengujian yang dilakukan dijamin menggunakan *statement* dalam program minimal satu kali selama pengujian.
3. Menghitung *cyclometris complecity* sebagai ukuran kuantitif untuk menentukan jumlah *independent patch* sebagai jalur yang perlu diuji.

Dalam pengujian menggunakan *basis patch*, ada beberapa elemen penting yang harus diketahui antara lain:

1. *Node* (N), yaitu *symbol* yang mewakili suatu proses yang ada pada setiap *flowchart*.
2. *Edge* (E), yaitu garis-garis yang menghubungkun *node* satu dengan *node* yang lainnya pada *flowchart*.
3. Prediksi (P), yaitu *node* yang mempunyai minimal dua buah *edge*.
4. *Region* (R), yaitu suatu wilayah tertutup yang terbentuk dari *Node* dan *Edge*.

Q. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Adapun Jenis yang digunakan dalam penelitian ini secara tidak langsung dan secara langsung yaitu :

1. Secara Tidak Langsung (Studi Literatur)

   Metode tidak langsung ini maksudnyadam ialah mengumpulkan data-data maupun informasi yang terkait seperti mempelajari buku-buku pustaka atau artikel yang berasal dari media *internet*.

2. Secara Langsung (Observasi)

   Metode secara langsung yaitu mengumpulkan data-data atau informasi yang terkait dengan perancangan program aplikasi berupa pengamatan langsung, wawancara dan pengumpulan dokumen.

B. Lokasi dan Waktu

1. Lokasi

   Lokasi penelitian dilakukan pada jl. Poros Pare-pare – Makassar, Palanro.

2. Waktu

   Adapun waktu yang dipergunakan untuk pelaksanaan penelitian ini adalah sebagai berikut :

Tabel 0.1 Waktu Penelitian

No	Kegiatan	Bulan					
		Nov 2020	Des 2020	Jan 2021	Feb 2021	Mar 2021	Apr 2021
1	Persiapan Data						
2	Desain dan Pembuatan Program						
3	Implementasi dan Testing						
4	Penyusunan Laporan						

C. Teknik Pengumpulan Data

Metode Pengumpulan data dilakukan melalui studi pustaka, terutama yang berhubungan data-data sekunder. Sementara data primer dilakukan melalui studi lapangan yaitu berupa:

1. Observasi, yaitu mengumpulkan data dengan cara mengetahui informasi dan data awal tentang keadaan objek penelitian.
2. Dokumentasi, yaitu mengumpulkan data dalam bentuk dokumen atau catatan tertulis.
3. Wawancara, yaitu dengan melakukan wawancara terhadap beberapa penjual.

D. Metode pengujian

Metode pengujian yang dilakukan pada peneltian ini menggunakan metode pengujian sebagai berikut:

1. Metode *Black Box*. memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.
2. Metode *White Box,* yaitu pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya (*basis path*) atau proses *looping* (pengulangan).

E. Jenis data

Jenis data yang dikumpulkan dalam penelitian ini adalah data *primer* dan data *sekunder* yang relevan dengan masalah yang akan dibahas.

1. **Data *Primer***

   Data *Primer* adalah data yang berasal atau data yang diperoleh langsung dari sumber-sumber yang berada di lapangan yang berasal dari wawancara dengan pihak-pihak terkait dengan penelitian ini.

2. **Data *Sekunder***

   Data *Sekunder* adalah data yang diperoleh tidak secara langsung dari objek penelitian. Penelitian medapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Perangkat keras komputer tidak berarti tanpa perangkat lunak begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain.Perangkat keras hanya berfungsi jika diberikan instruksi-intruksi kepadanya. Instruksi-instruksi inilah disebut dengan perangkat lunak. Dalam penelitian ini penyusun menggunakan beberapa perangkat lunak dan perangkat keras ,antara lain:

1. **Kebutuhan perangkat keras.**

   Spesifikasi perangkat keras sebagai berikut :

   Tabel 0.1 Kebutuhan Perangkat Keras

Jenis	Spesifikasi
Laptop	ACER Aspire ES 14
Processor	*Intel® Celeron®* CPU N3350 @1.10Ghz (2 CPUs), (1.1 GHz)
Memory	4 GB *RAM*
Harddisk	500 MB
Handphone	*Realme 5 Pro*

2. **Kebutuhan perangkat lunak**

   Spesifikasi perangkat lunak sebagai berikut :

   Tabel 0.2 Kebutuhan Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	*Windows 10 Home Single Language X64-bit.*
Software Lainnya	*Xampp versi 7.3.6* *Visual Studio Code.* *Cordova.* *PHP versi 7.3* *Google Chrome*

B. Perancangan

1. *Use case diagram*

Use case diagram berfungsi untuk menjelaskan alur sistem jika dilihat menurut pandangan orang yang berada diluar sistem.

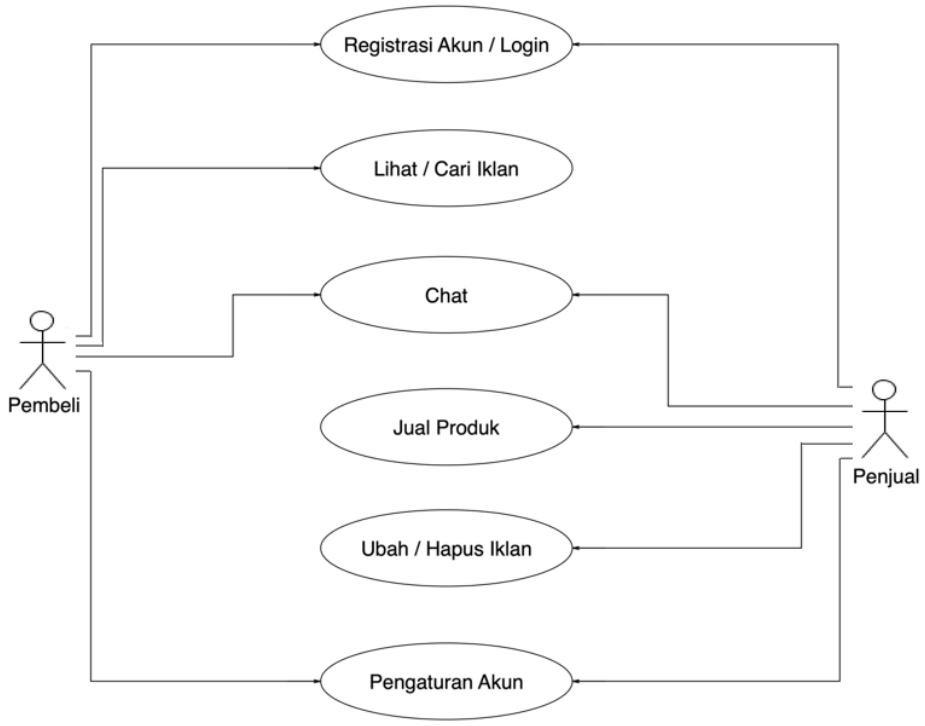
Aktor menggambarkan segala pengguna aplikasi. *Actor* dalam Aplikasi Penyediaan Jasa Penjualan Pakaian antara lain sebagai berikut:

1. **Penjual *(User)***

   Actor yang memiliki hak akses sebagai penjual.

2. **Pembeli *(User)***

   Actor yang memiliki hak akses sebagai pembeli.



Gambar 4.1 *Use Case Diagram*

Penjelasan *Use Case Diagram*:

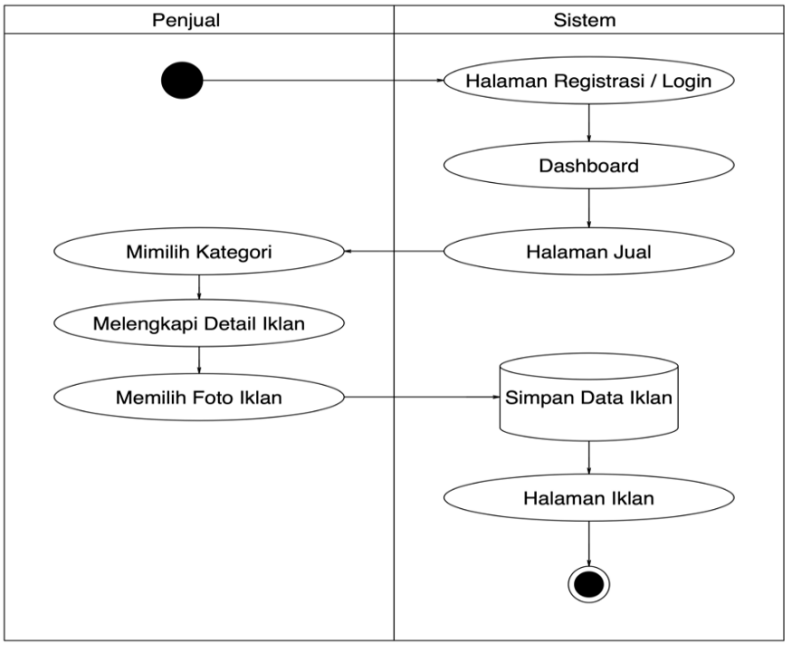
Tabel 0.3 Penjelasan *Use Case Diagram*

Nama *Use Case*	**Deskripsi *Use Case***
Registrasi Akun / Login	*Use Case* ini menjelaskan proses dimana *actor registrasi* atau *login*
Lihat / Cari Iklan	*Use Case* ini menjelaskan proses dimana *actor* dapat melihat dan mencari iklan produk
Jual Produk	*Use Case* ini menjelaskan proses dimana *actor* dapat menambahkan iklan atau menjual produk
Ubah / Hapus Iklan	*Use Case* ini menjelaskan proses dimana *actor* penjual dapat melakukan ubah atau hapus iklannya
Pengaturan Akun	*Use Case* ini menjelaskan proses *actor* dapat mengubah data akun

2. *Activity Diagram*

Activity diagram memodelkan alur kerja (*workflow)* sebuah proses bisnis dan urutan aktifitas dalam suatu proses. *Activity diagram* yang ada di *system* yang penulis rancang sebagai berikut:

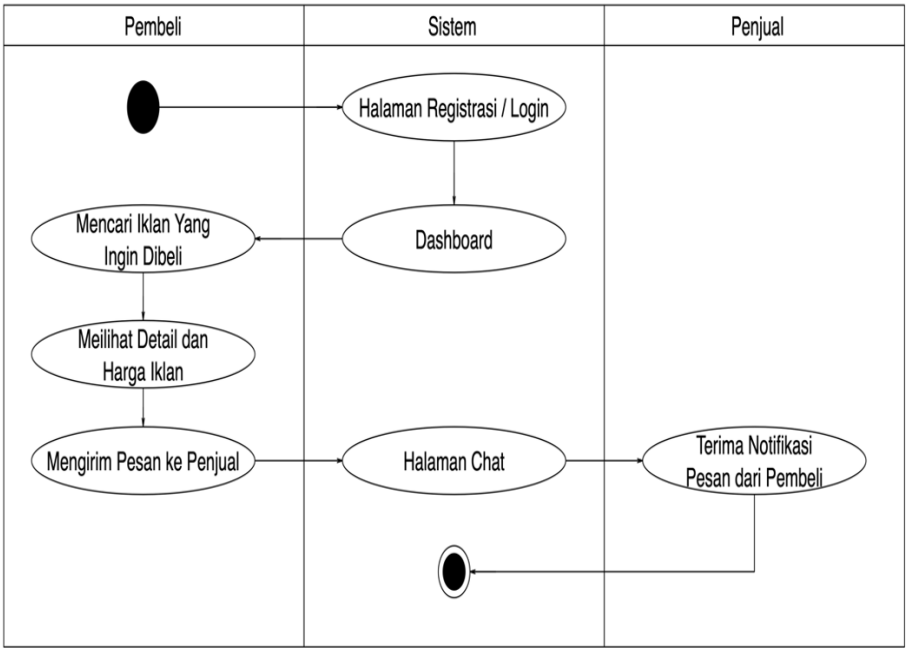
a. *Activity diagram* penjual



Gambar 4.2 *Activity Diagram* Penjual

Activity ini menjelaskan mengenai proses urutan yang dilalui oleh penjual dalam melakukan menambahkan iklan, dan melengkapi data *detail* iklan besarta foto iklan yang ingin jual.

b. *Activity diagram* pembeli



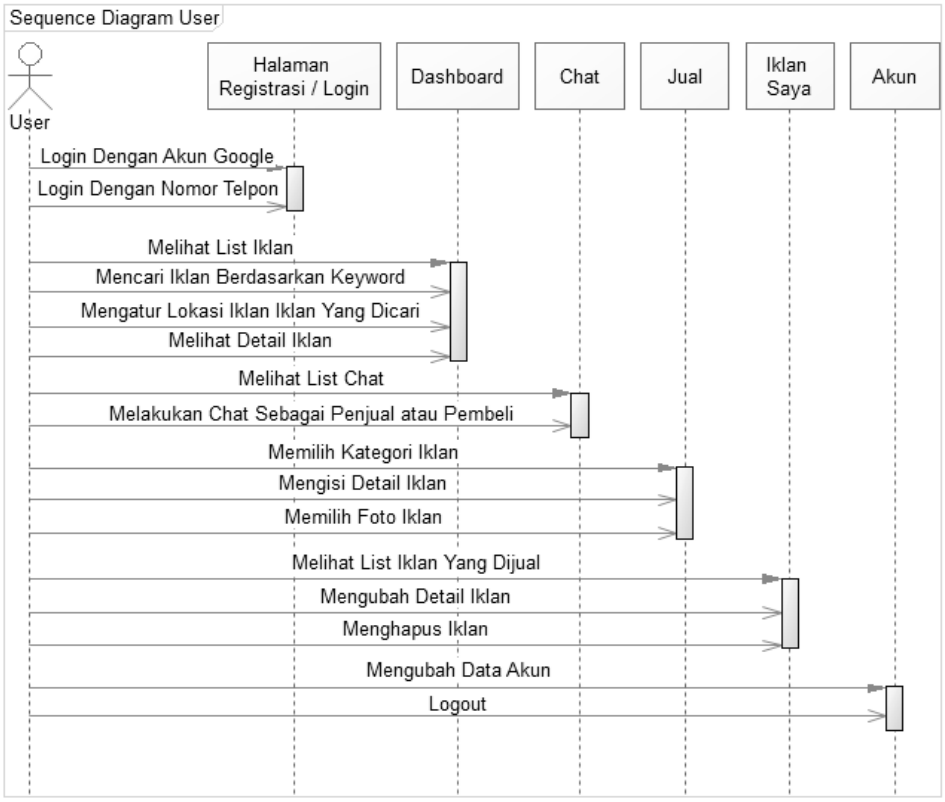
Gambar 4.3 *Activity Diagram* Pembeli

Activity ini menjelaskan mengenai proses dimana calon pembeli melakukan pembelian produk didalam aplikasi, mulai dari *login* lalu mencari iklan yang ingin dibeli kemudian mengirim pesan ke penjual.

3. *Sequence Diagram*

Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah *diagram sequence*nya.

a. *Diagram sequence user* (penjual dan pembeli)



Gambar 4.4 *Sequence Diagram User*

Berdasarkan gambar 4.4 diatas atau *Sequence Diagram User*, dapat dilihat bahwa terdapat 6 *page* atau halaman yang terdiri Halaman Registrasi / *Login, Dashboard, Chat,* Jual, Iklan Saya dan Akun.

4. Tabel Implementasi

Tabel 0.4 Tabel *User*

Tbl_*user*			
No	***Name***	***Type***	***Size***
1	*UID*	*Varchar*	50
2	*Email*	*Varchar*	50
3	Nama	*Varchar*	50
4	Telp	*Varchar*	15
5	Foto	*Text*	-
6	Token	*Text*	-
7	*Latitude*	*Float*	-
8	*Longitude*	*Float*	-

Tabel 0.5 Tabel Produk

Tbl_produk			
No	***Name***	***Type***	***Size***
1	*ID*	*Varchar*	18
2	*UID*	*Varchar*	50
3	Judul	*Text*	-
4	*Deskripsi*	*Text*	-
5	Kondisi	*Varchar*	10
6	Harga	*Double*	-
7	*Images*	*JSON*	-
8	Id_kecamatan	*Bigint*	20
9	Dilihat	*Int*	11
10	Ditambahkan_pada	*Datetime*	-
11	Diubah_pada	*Timestamp*	-

Tabel 0.6 Tabel Provinsi

Tbl_provinsi			
No	***Name***	***Type***	***Size***
1	*ID*	*Bigint*	20
2	*Name*	*Varchar*	50
3	*Latitude*	*Float*	-
4	*Longitude*	*Float*	-

Tabel 0.7 Tabel Kabupaten

Tbl_kabupaten			
No	***Name***	***Type***	***Size***
1	*ID*	*Bigint*	20
2	*Province_id*	*Bigint*	20
3	*Name*	*Varchar*	50
4	Latitude	*Float*	-
5	Longitude	*Float*	-

Tabel 0.8 Tabel Kecamatan

Tbl_kecamatan			
No	***Name***	***Type***	***Size***
1	*ID*	*Bigint*	20
2	*Regency_id*	*Bigint*	20
3	*Name*	*Varchar*	50
4	Alt_name	*Text*	-
5	*Latitude*	*Float*	-
6	*Longitude*	*Float*	-

C. Implementasi

Program aplikasi ini adalah aplikasi *android.* pada aplikasi ini *user* yang telah mendaftar akan bisa menjual dan membeli barang.

1. Hasil implementasi aplikasi

Implementasi aplikasi ini dilakukan setelah proses analisa dan perancangan *system* selesai dilakukan. Untuk menjalankan aplikasi ini maka dibutuhkan sebuah perangkat laptop dengan *XAMPP* 7.3.6.

2. Implementas desain *interface*

Implementasi desain *interface* hanya satu bagian, yaitu *interface* penjual dan pembeli yang memiliki *interface* yang sama.

a. Menu *registrasi / login*

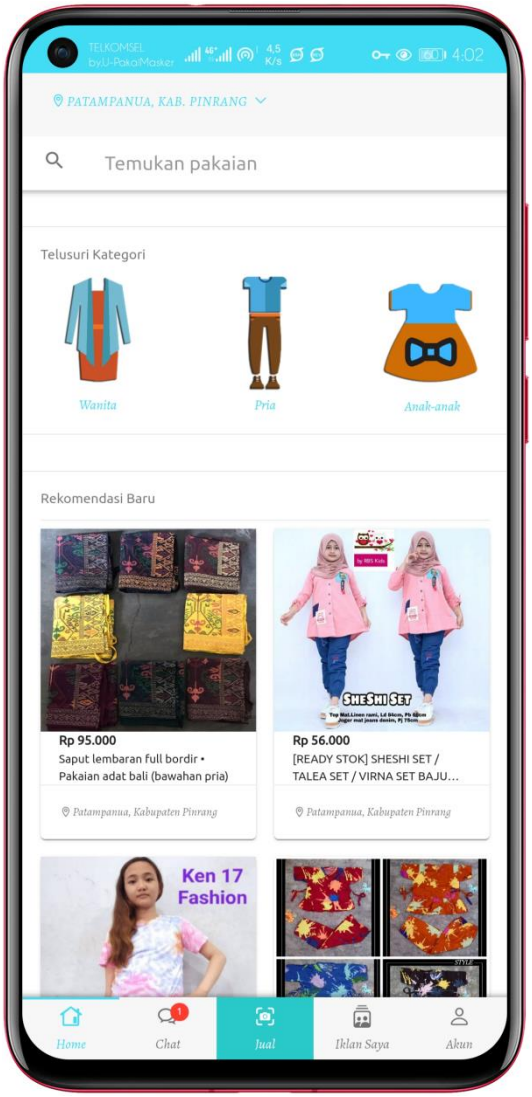
Merupakan tampilan menu yang digunakan *user* untuk *registrasi* maupun *login*



Gambar 4.5 Tampilan *Login*

Pada tampilan halaman *login* terdapat dua tombol yang terdiri dari masuk dengan akun *google* dan masuk dengan nomor telpon. Ketika ingin melakukan *login* dengan akun *google* aplikasi akan menampilkan *list* akun yang telah terdaftar di *device* pengguna. Dan ketika ingin melakukan *login* dengan nomor telpon, *user* akan menerima sms kode verifikasi.

b. Menu *Home*

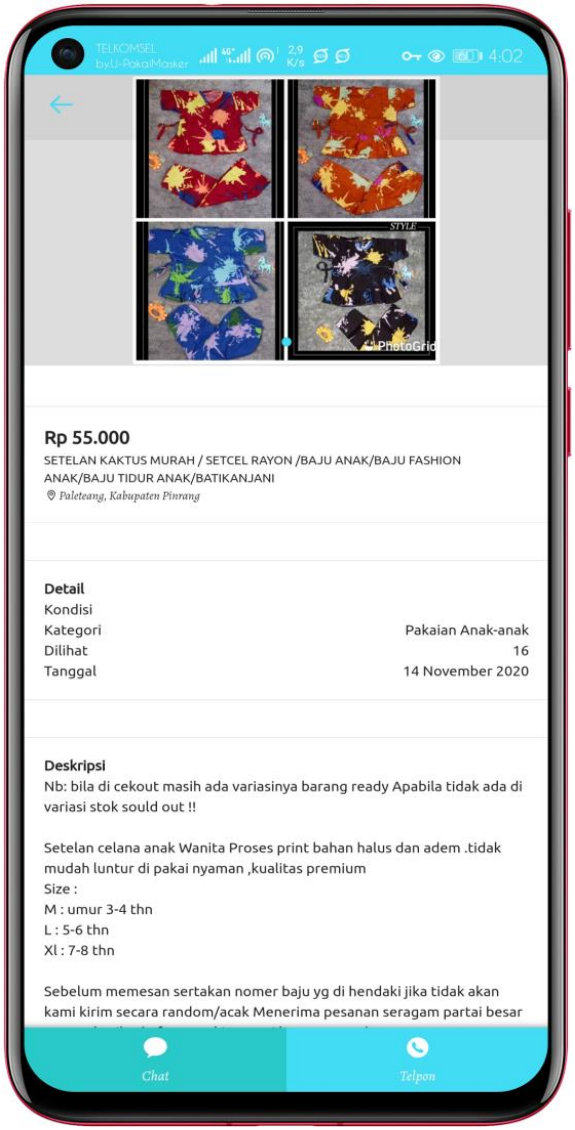


Gambar 4.6 Tampilan Menu *Home*

Halaman di atas merupakan tampilan menu utama setelah *user* melakukan *login.* Pada halaman ini *user* dapat melihat *list* iklan berdasarkan lokasi terdekat. *User* dapat melakukan pencarian iklan berdasarkan *keyword* ataupun kategori. *User* juga dapat mengubah lokasi untuk melihat iklan disekitar.

c. Halaman *detail* iklan

Merupakan tampilan *detail* iklan yang dipilih

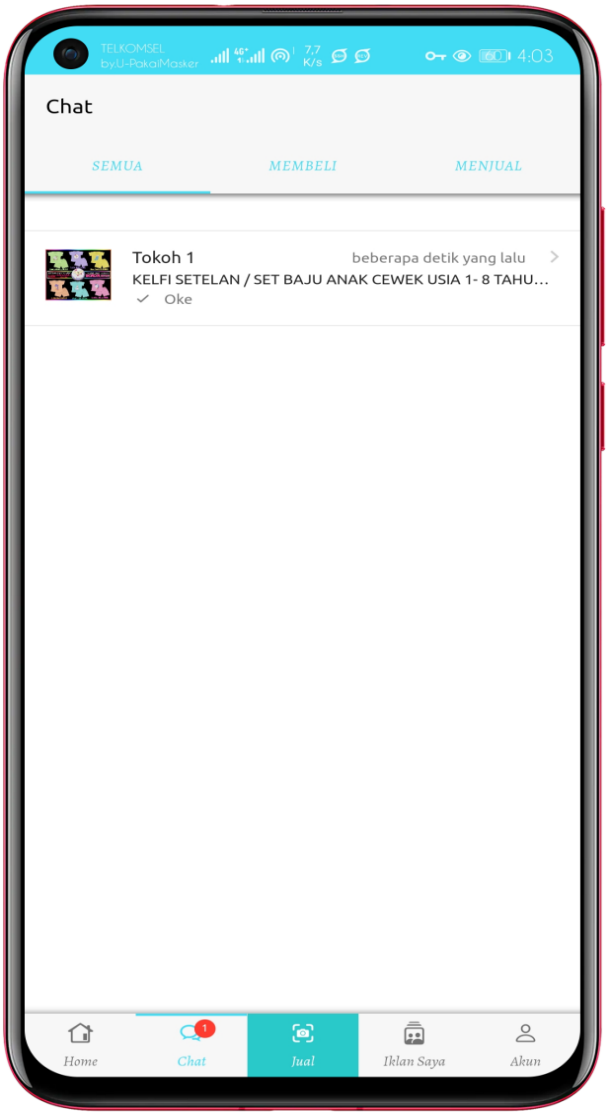


Gambar 4.7 Tampilan Halaman *Detail* Iklan

Pada halaman *detail* iklan, *user* (pembeli) dapat melakukan *chat* dengan penjual atau bisa langsung menelpon nomor telpon penjual.

d. Halaman *Chat*

Merupakan tampilan dimana daftar *chat user* sebagai pembeli dan penjual.

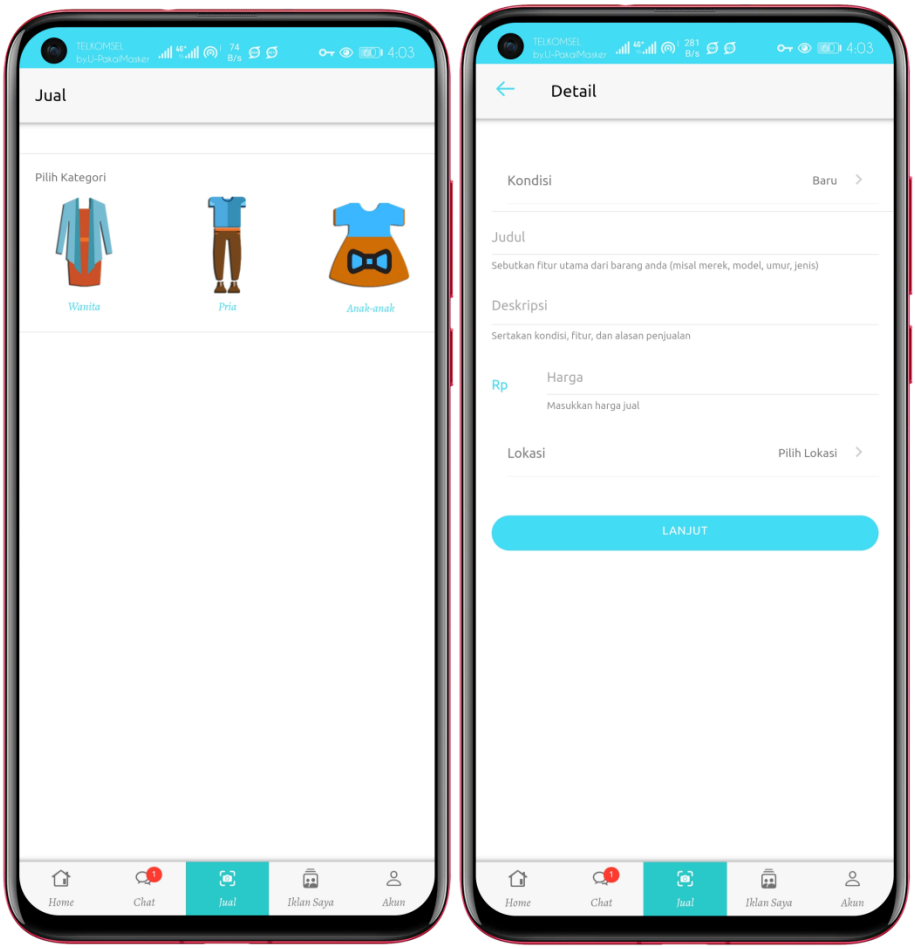


Gambar 4.8 Tampilan Halaman *Chat*

Pada halaman ini, *user* dapat melihat semua daftar *chat* yang masuk sebagai penjual dan juga pembeli.

e. Halaman Jual

Merupakan tampilan dimana *user* dapat melakukan jual barang atau memasang iklan.

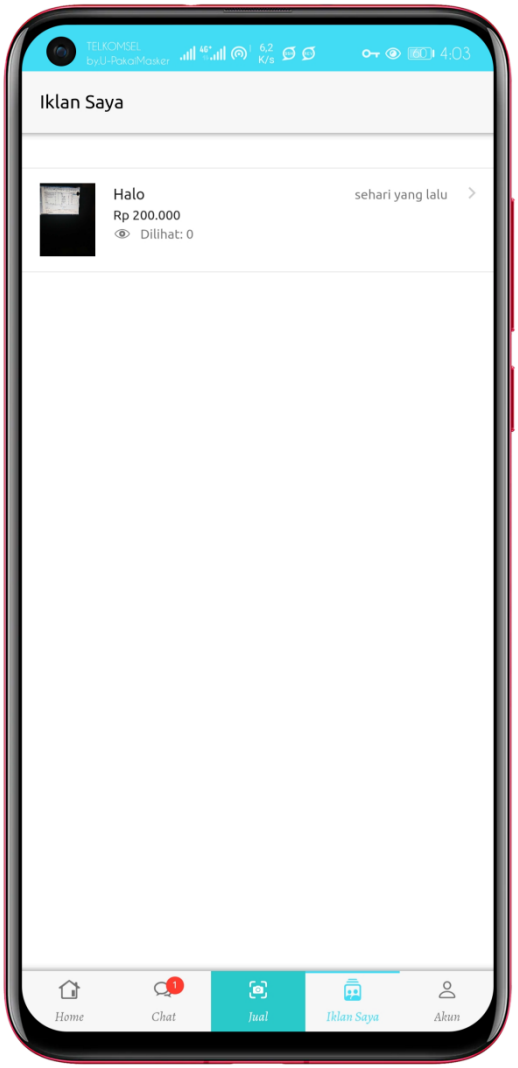


Gambar 4.9 Tampilan Halaman Jual

Pada halaman ini setelah *user* selesai mengisi *detail* iklan, aplikasi akan mengarahkan *user* untuk memilih foto dari kamera atau galeri untuk di*upload* sebagai foto iklan yang akan dijual.

f. Halaman iklan saya

Merupakan tampilan daftar iklan yang dijual oleh *user*

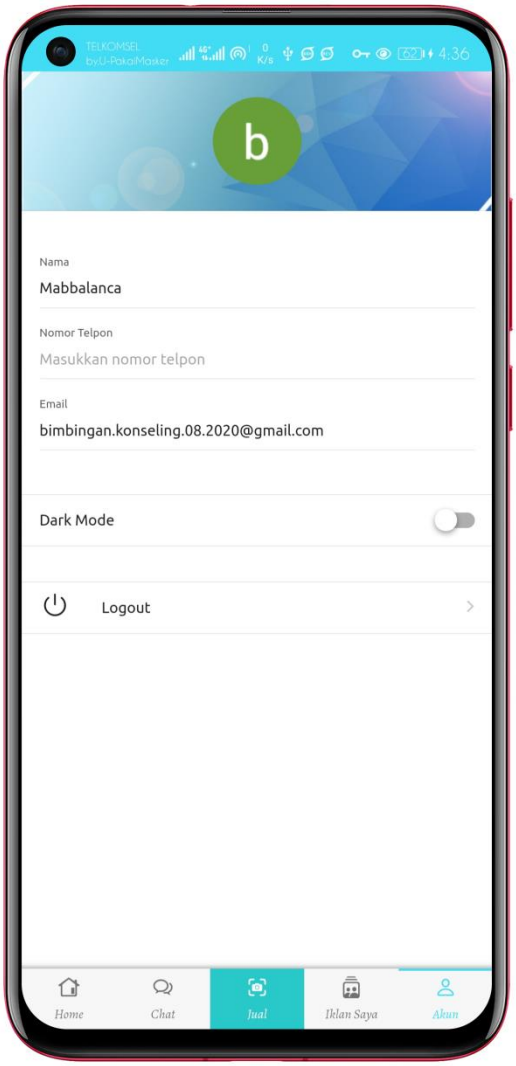


Gambar 4.10 Tampilan Halaman Iklan Saya

Pada halaman ini terdapat dua tombol untuk *edit* dan hapus iklan.

g. Halaman Akun

Merupakan tampilan halaman Akun



Gambar 4.16 Tampilan halaman *check out*

Pada halaman ini, *user* dapat mengubah data akun. *User* juga dapat mengubah ke mode gelap (*Dark Mode*), dan *user* juga dapat melakukan *logout*.

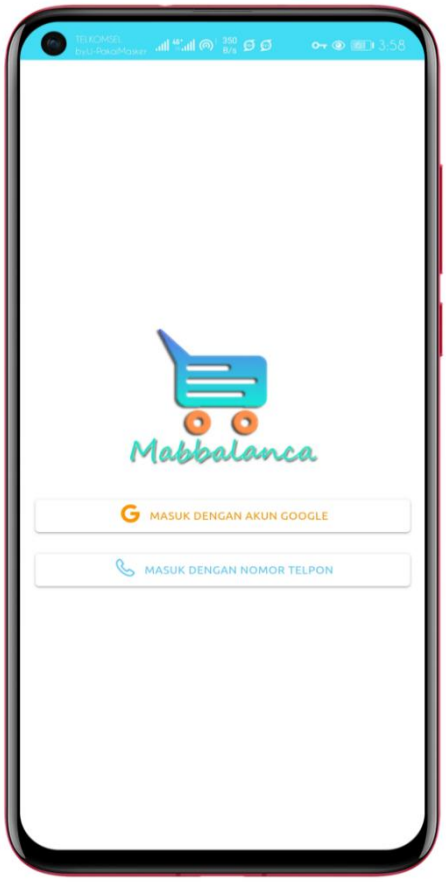
D. Pengujian

Pengujian Sistem dengan Pengujian *Blackbox*. Pengujian ini bersifat menguji waktu perintah sesuai dengan fungsinya yang tanpa ada kesalahan atau error.

1. Pengujian *Black-Box*

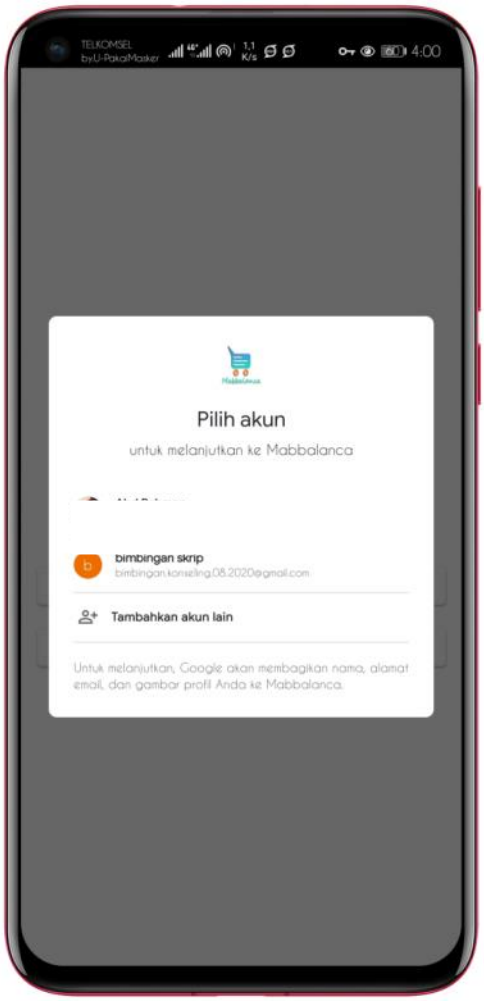
Tabel 0.9 Pengujian membuka aplikasi

Test Factor	**Hasil**	**Keterangan**
Membuka aplikasi *android*	✓	Berhasil, karena Halaman *Login* dapat ditampilkan
Screen Shoot		



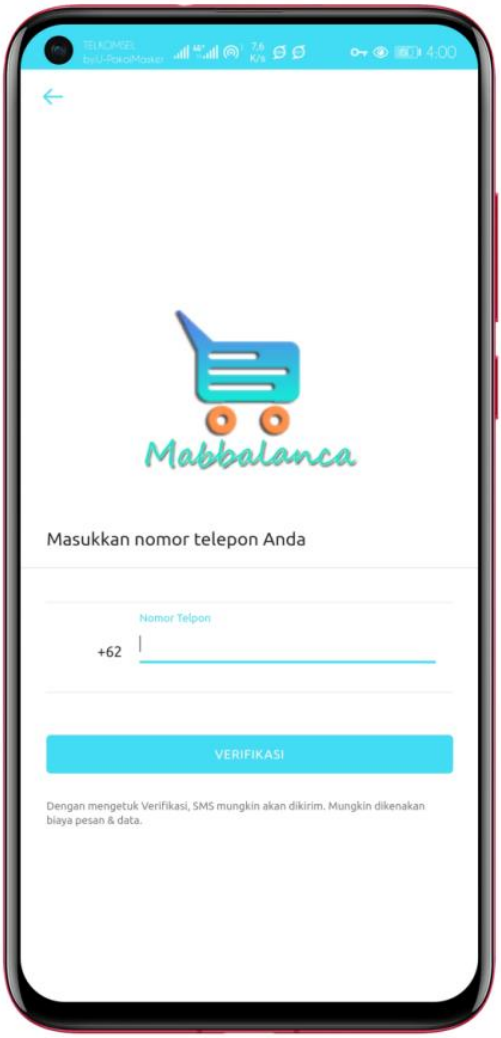
Tabel 0.10 Pengujian *login* akun *google*

Test Factor	**Hasil**	**Keterangan**
Login dengan akun *google*	✓	Berhasil, karena daftar akun *google* dapat dipilih untuk melakukan *login*
Screen Shoot		



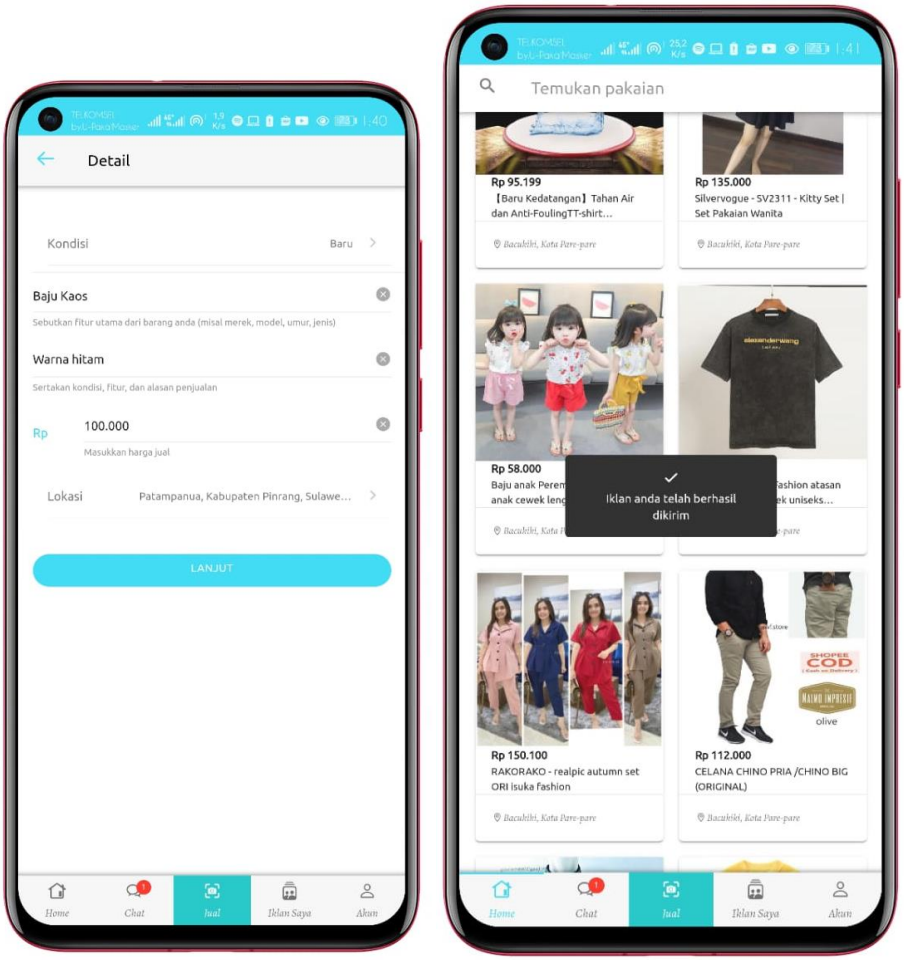
Tabel 0.11 Pengujian *login* dengan nomor *telephone*

Test Factor	**Hasil**	**Keterangan**
Login dengan nomor *telephone*	✓	Berhasil, karena dapat melakukan *login* dengan nomor *telephone*
Screen Shoot		



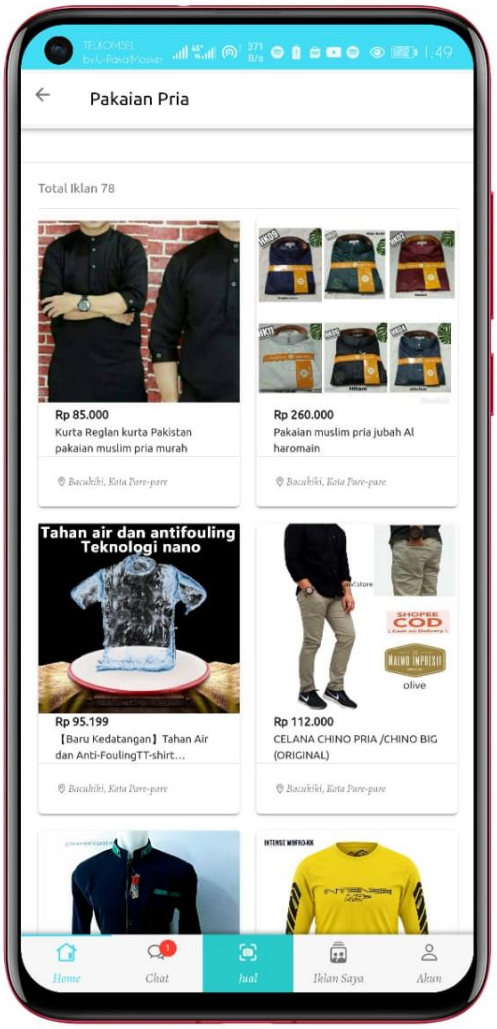
Tabel 0.12 Pengujian tambah iklan

Test Factor	**Hasil**	**Keterangan**
Tambah Iklan	✓	Berhasil, karena iklan dapat ditambahkan
Screen Shoot		



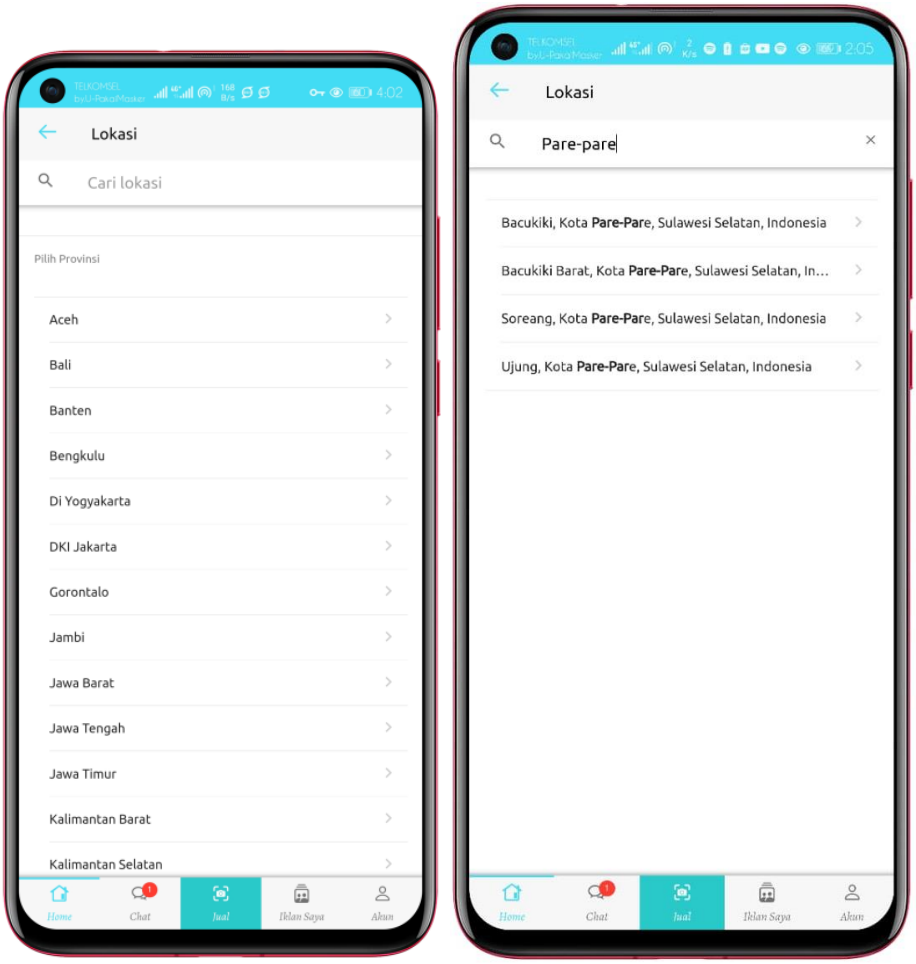
Tabel 0.13 Pengujian pencarian iklan dengan *keyword*

Test Factor	**Hasil**	**Keterangan**
Pencarian iklan dengan *keyword*	✓	Berhasil, karena iklan dapat ditampilkan berdasarkan *keyword*
Screen Shoot		



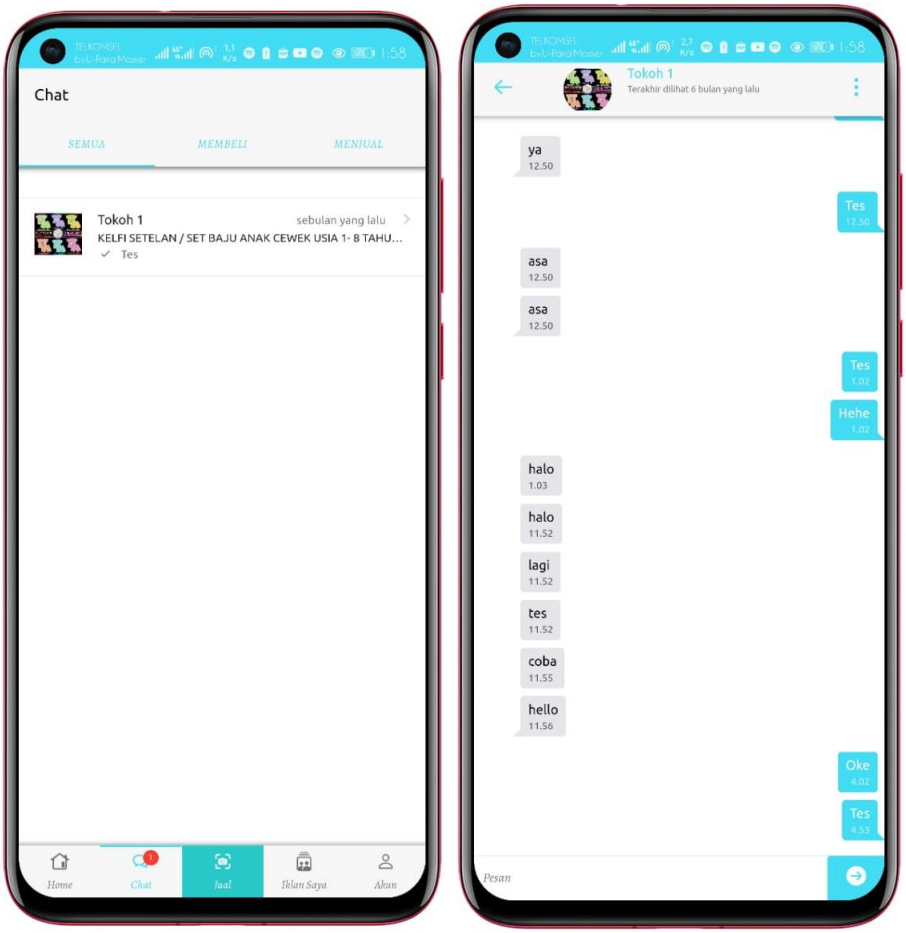
Tabel 0.14 Pengujian percarian iklan berdasarkan lokasi

Test Factor	**Hasil**	**Keterangan**
Pencarian iklan dengan lokasi	✓	Berhasil, karena iklan dapat ditampilkan berdasarkan lokasi yang dipilih
Screen Shoot		



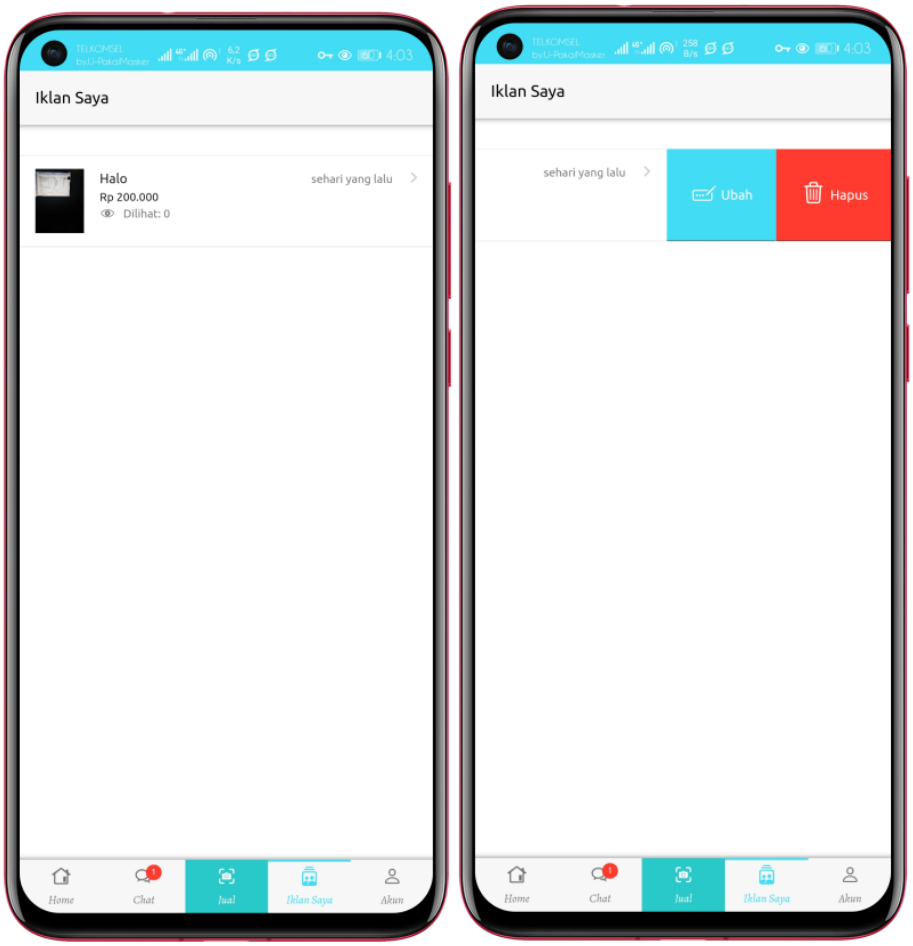
Tabel 0.15 Pengujian kirim *chat*

Test Factor	**Hasil**	**Keterangan**
Kirim *chat*	✓	Berhasil, karena *chat* dapat dikirim dan diterima
Screen Shoot		



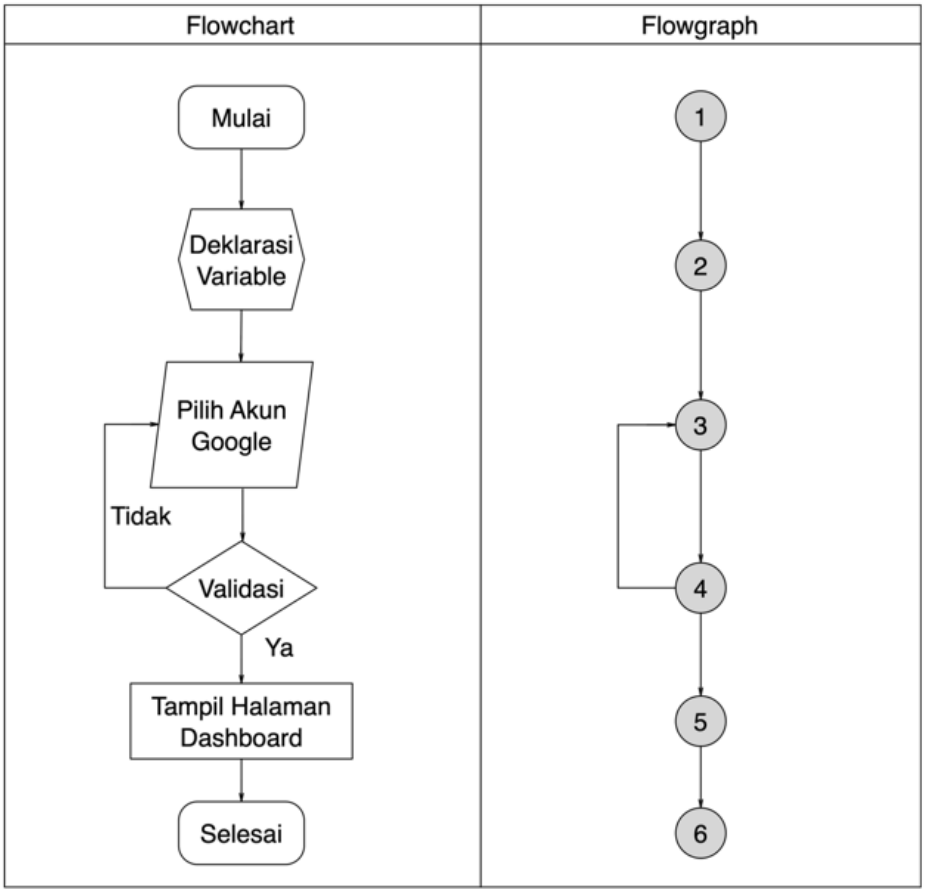
Tabel 0.16 Pengujian ubah iklan

Test Factor	**Hasil**	**Keterangan**
Ubah iklan	✓	Berhasil, karena iklan dapat diubah
Screen Shoot		



2. Pengujian *White-Box*

a) *Login*



Gambar 4.11 *Flowchart Login*

Dari gambar *flowgraph Menu Login* di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G) dari Egde* dan *Node*:

   Dengan Rumus : V (G) = E – N + 2

   Dengan E (*edge*) = 6

   N (*node*) = 6

   Predikat *Node* (P) = 1

   Penyelesaian :V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent Path* pada *flowgraph* diatas adalah :

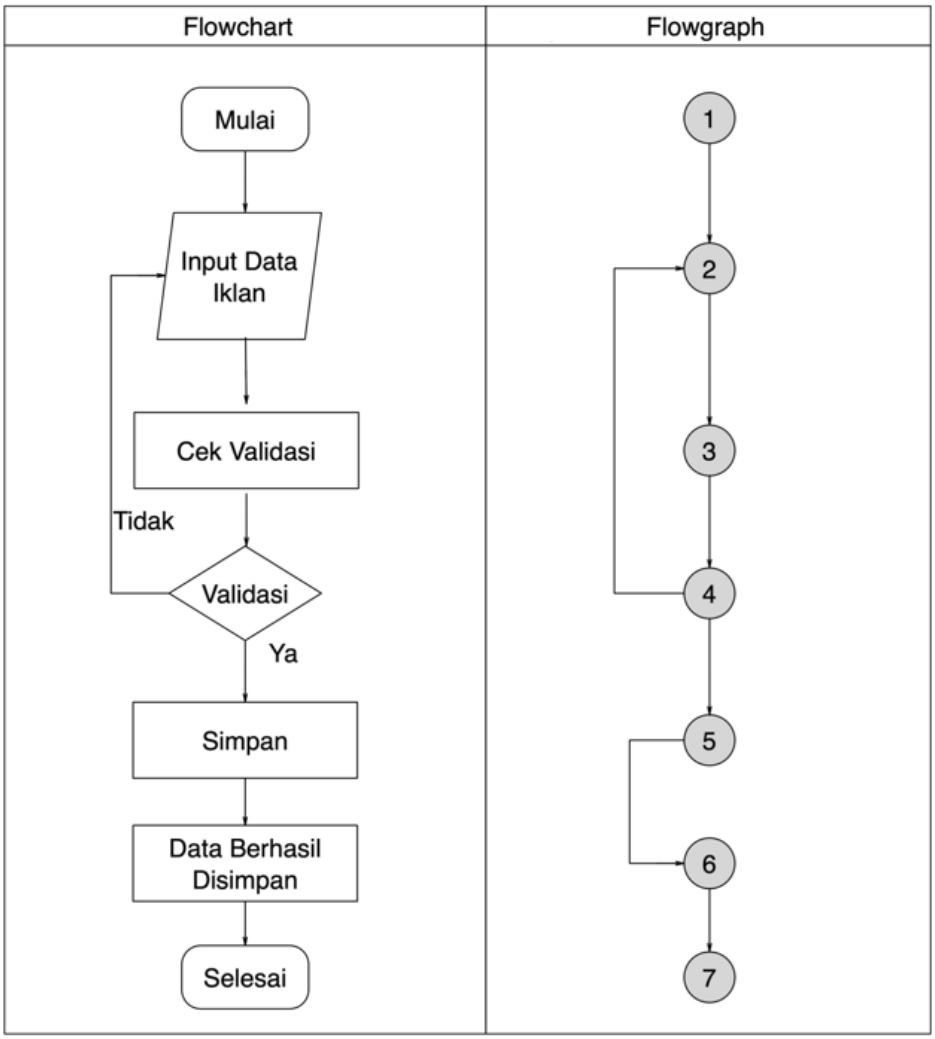
   Path 1 = 1 – 2 – 3 – 4 – 3

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6
4. *Grafis Matriks*

   Tabel 0.17 Grafik *Matriks* dari *Flowgraph Login*

	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3				1			1-1=0
4			1		1		2-1=1
5						1	1-1=0
6							0
Sum (E) + 1							1+1=2

b) *Input* Data



Gambar 4.12 *Flowchart Input* Data

Dari gambar *flowgraph* Input Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G)* dari *Egde* dan *Node*:

   Dengan Rumus : V (G) = E – N + 2

   Dengan E (*edge*) = 7

   N (*node*) = 7

   Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 7 – 7 + 2 = 2

Predikat (P) = P + 1

= 1 + 1 = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent Path*pada *flowgraph* diatas adalah :

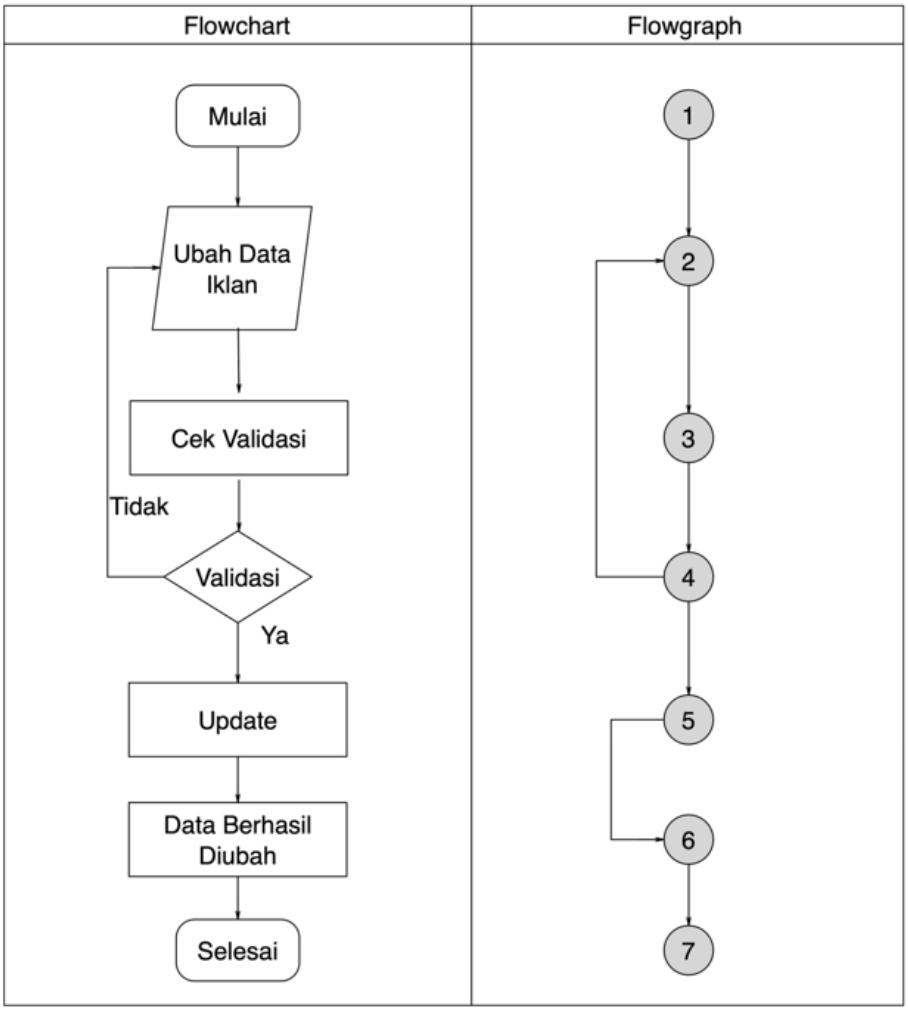
   Path 1 = 1 – 2 – 3 – 4 – 2

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7
4. *Grafik Matriks*

Tabel 0.18 *Grafik Matriks* pada *Flowgraph Input* Data

	1	2	3	4	5	6	7	E – 1 = 0
1		1						1 – 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0
4		1			1			2 – 1 = 1
5						1		1 – 1 = 0
6							1	1 – 1 = 0
7								0
Sum (E) + 1								1 + 1 = 2

c) *Edit* Data



Gambar 4.13 *Flowchart Edit* Data

Dari gambar *flowgraph Edit* Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G) dari Egde* dan *Node*:

   Dengan Rumus : V (G) = E – N + 2

   Dengan E (*edge*) = 8

   N (*node*) = 8

   Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 8 – 8 + 2 = 2

Predikat (P) = P + 1

= 1 + 1 = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent Path*pada *flowgraph* diatas adalah :

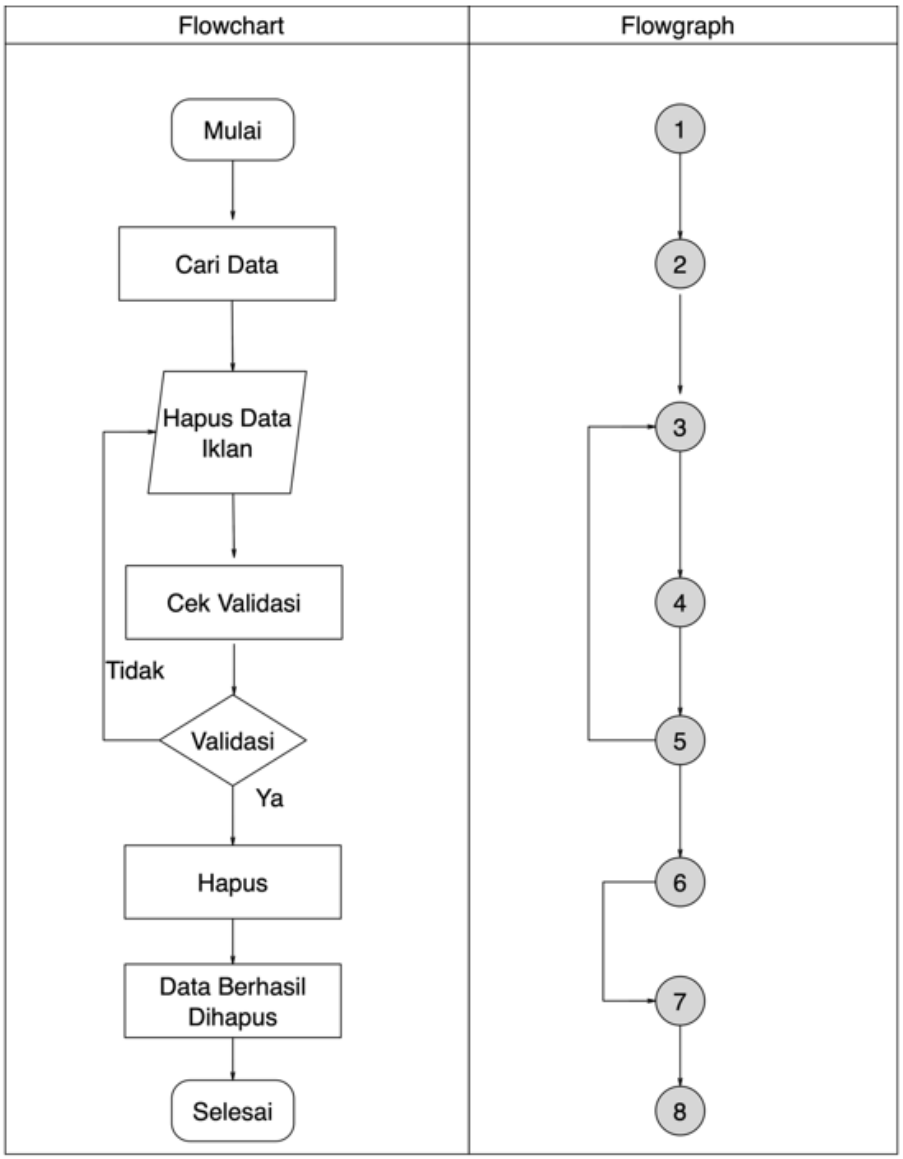
   Path 1 = 1 – 2 – 3 – 4 – 5 – 3

   Path 2 = *1 – 2 – 3 – 4 – 5 – 6 – 7* – 8
4. *Gra*fik *Matriks*

Tabel 0.19 *Grafik Matriks* pada *Flowgraph Edit* Data

	1	2	3	4	5	6	7	8	E – 1 = 0
1		1							1 – 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5			1			1			2 – 1 = 1
6							1		1 – 1 = 0
7								1	1 – 1 = 0
8									0
Sum (E) + 1									1 + 1 = 2

d) Hapus Data



Gambar 4.14 *Flowchart* Hapus Data

Dari gambar *flowgraph* Hapus Data di atas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic ComplexcityV(G) dari Egde* dan *Node*:

   Dengan Rumus : V (G) = E – N + 2

   Dengan E (*edge*) = 8

N (*node*) = 8

Predikat *Node* (P) = 1

Penyelesaian :V(G) = E – N + 2

= 8 – 8 + 2 = 2

Predikat (P) = P + 1

= 1 + 1= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent Path*pada *flowgraph* diatas adalah :

   Path 1 = 1 – 2 – 3 – 4 – 5 – 3

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

4. *Grafik Matriks*

Tabel 0.20 *Grafik Matriks* pada *Flowgraph* Hapus Data

	1	2	3	4	5	6	7	8	E – 1 = 0
1		1							1 – 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5			1			1			2 – 1 = 1
6							1		1 – 1 = 0
7								1	1 – 1 = 0
8									0
Sum (E) + 1									1 + 1 = 2

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem aplikasi untuk sarana jual beli pakaian secara *online* melalui sebuah aplikasi *android* yang efisien guna mempermudah *customer* untuk melakukan pemesanan dan juga mempermudah penjual untuk mengiklankan barang jualannya.
2. Aplikasi ini menggunakan *database firebase* dan dilengkapi dengan menu *Home, Chat,* Jual, Iklan saya, Dan pengaturan *akun*.
3. Aplikasi dilengkapi dengan layanan *realtime chat* dapat mempermudah interaksi jual beli antara calon pembeli dengan penjual.
4. Dengan aplikasi ini, pelanggan dapat melihat kondisi barang ataupun stok barang, lalu kemudian menyelesaikan transaksi dengan sistem COD (*Cash On Delivery)* atau mengirim barang dengan jasa kurir sesuai dengan kesepakatan antara penjual dengan pembeli.
5. Untuk mengakses aplikasi, pengguna harus *login* terlebih dahulu menggunakan akun *google* atau nomor telpon.
6. Pelanggan dapat menghubungi penjual secara langsung menggunakan kontak yang tersedia.

B. Saran

Setelah melakukan penelitian, maka penulis memberikan saran sebagai berikut :

1. Penulis berharap agar aplikasi ini dapat diterapkan, dimanfaatkan dan dikembangkan di kemudian hari
2. Penulis menyarankan agar kedepannya akan ada pengembangan terkait aplikasi sistem penyedia jasa penjualan berbasis *android.*
3. Penulis menyarankan agar kedepannya skripsi ini dapat menjadi referensi dan dapat dipergunakan sebaik mungkin.

DAFTAR PUSTAKA

Andy, 2019, "Pengertian Xampp lengkap dengan fungsi dan cara instalasi", https://qwords.com/blog/pengertian-xampp/, 13 Maret 2019.

Arief, Ramadhan. 2018. Pemrograman Web dengan Html, Css, dan Javascript.Jakarta :PT Elex Media Komputindo.

Edy, Winarno, & Smitdev, community. 2016. Pemrograman Web Berbasis Html, Php, dan Javascript. Jakarta :PT Elex Media Komputindo.

Harianto, Kristanto. 2017. Konsep Dan Perancangan Database. Yogjakarta : Andi

Jubilee, Enterprise. 2015. Msql untuk pemula. Jakarta :PT Elex Media Komputindo.

Jubilee, Enterprise.2016. Pemrograman Bootstrap untuk Pemula.Jakarta :PT Elex Media Komputindo.

Mudyahardjo, Redja. 2019. *Pengantar Pendidikan Komputer*. Jakarta : Rajawali Pers Tirtarahadja, U. dan La Sula. 2000.

Taryana, Suryana, & Koesheryatin.2019. Aplikasi *Internet* Menggunakan HTML, CSS, & Javascript.Jakarta : PT Elex Media Komputindo.

Pahmi, Ritonga,(2015), "Pengertian Unified Modeling Language (UML) dan Modelnya Menurut Pakar dan Ahli", https://bangpahmi.com/pengertian-unified-modeling-language-uml-dan-modelnya-menurut-pakar-dan-ahli/, 13 Maret 2019.

Yuniar, Supriadi, dan Defri, Faizal, Maulana, S. 2018. Buku Mahir Web Programming.Jakarta :PT Elex Media Komputindo.

Wikipedia, (2019), "Kuis", https://id.wikipedia.org/wiki/Kuis, 13 Maret 2019