

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi pada zaman ini semakin maju dan telah didukung suatu teknologi sistem informasi, yaitu teknologi informasi. Dimana kemajuan teknologi informasi telah berdampak pada sejumlah bidang, salah satunya dibidang bisnis. Dengan menggunakan teknologi dan sistem pemesanan yang cepat, teliti dan tepat akan membantu aktivitas pembeli dalam melakukan pemesanan dan juga membantu pemilik toko dalam melihat penjualan, operasional yang dapat diakses dimana-mana. Es kristal adalah jenis es batu yang memiliki struktur kristal yang lebih teratur dan padat dibandingkan es batu biasa. Es kristal umumnya dibuat dengan proses pembekuan air yang terkontrol dan lambat, sehingga menghasilkan es yang lebih jernih, keras, dan tahan lama. Es kristal banyak digunakan untuk berbagai keperluan, seperti mendinginkan minuman, menghias makanan, dan terapi dingin. Toko Es Kristal Atika merupakan salah satu toko yang menjual es Kristal yang beralamat Jl. Mangga Tengah No.47, Labukkang, Kec.Ujung, Kota Parepare, Sulawesi Selatan 91111. Pemesanan es Kristal di toko Atika hanya mengandalkan referensi dari mulut ke mulut atau pelanggan secara langsung datang ke toko untuk memesan es Kristal. Untuk itu diperlukan rancangan sistem yang dapat digunakan sebagai media penjualan dan pemesanan bagi *customer* dan pemilik toko, berdasarkan uraian diatas adapun judul skripsi yang diajukan penulis yaitu “**APLIKASI SISTEM**

PENJUALAN DAN PEMESANAN ES KRISTAL PADA TOKO BERBASIS ANDROID”.

A. Rumusan Masalah

Berdasarkan latar belakang penelitian yang ditulis di atas, maka yang menjadi rumusan masalah pada penelitian adalah bagaimana membuat aplikasi sistem penjualan dan pemesanan Es Kristal pada Toko Atika berbasis *android*?

B. Batasan Masalah

Agar penelitian ini terhindar dari penyimpangan atau perluasan pokok masalah agar tetap pada jalurnya, maka penulis memberikan batasan masalah antara lain:

1. Cakupan penelitian melingkup seputar penjualan dan pemesanan Es Kristal pada Toko Atika.
2. Pembuatan sistem penjualan dan pemesanan es Kristal pada toko, peneliti menggunakan MsqL sebagai database, bahasa pemrograman java dan *android studio* sebagai perancangan sistem.

C. Tujuan Penelitian

Tujuan dari pembuatan rancangan aplikasi ini adalah sebagai berikut:

Merancang sebuah aplikasi sistem penjualan dan pemesana Es Kristal pada Toko Atika berbasis *android*.

D. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Bagi Peneliti

Manfaat yang didapat bagi peneliti adalah dapat memberi pengetahuan bagaimana merancang sebuah aplikasi android.

2. Bagi Pembaca

Penelitian ini diharapkan dapat memberikan informasi secara tertulis maupun sebagai referensi mengenai teknologi.

3. Bagi Pengguna

Untuk meringankan dalam pengolahan data, penjualan dan pemesanan es kristal pada toko tersebut.

E. Sistematika Penulisan

Berdasarkan latar belakang yang dikemukakan diatas, sistematika penulisan dari penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Dalam bab ini dijelaskan mengenai hal yang terdiri dari Latar Belakang, Rumusan Masalah, Tujuan dan Manfaat Penelitian , Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Dalam tinjauan pustaka ini berisi tentang teori-teori yang berhubungan dengan Penelitian Proposal Skripsi dari berbagai sumber mulai dari buku dan jurnal.

BAB III METODE PENELITIAN

Dalam bab ini dijelaskan tentang rencana penelitian, objek penelitian, teknik pengumpulan data dengan observasi dan kondisi tempat penelitian, bab ini juga berisi metode pengembang *Prototype* yang digunakan untuk menerapkan penjualan dan pemesanan es kristal berbasis *android*, dibab ini juga berisi tentang

perbandingan penelitian terdahulu dan analisis masalah yang ada pada proses bisnis yang sedang berjalan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi usulan proses bisnis baru dalam penerapan penjualan dan pemesanan es kristal di toko mulai dari usulan kebutuhan fungsional dan non - fungsional usulan proses setelah diterapkan dalam bentuk rancangan diagram *use case diagram*, *class diagram* , *activity diagram*, dan *sequence diagram*.

BAB V KESIMPULAN

Pada bab ini berisi kesimpulan yang telah didapat dari penelitian terkait laporan tugas akhir.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Es Kristal



Gambar 2 .1 Es Kristal Tube (Bambu)

Es kristal adalah es yang diproduksi dengan menggunakan mesin yang menghasilkan bentuk seperti kristal dengan lubang di tengahnya. Pelaku usaha seperti rumah makan, pedagang kecil, bahkan pedagang ikan maupun daging dipasar kerap menggunakan es kristal sebagai pelengkap produknya. Tak hanya pelaku usaha kini masyarakat umum juga banyak menggunakan es kristal secara pribadi sebagai pelengkap hidangan khususnya minuman karena bentuknya yang unik, (Anjani et al., 2023)

2. Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan

mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpaku pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*.(Muhammad Saed Novendri, Ade Saputra, 2019).

3. Penjualan

Definisi Penjualan menurut Mulyadi (2008) adalah: “kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli”. Penjualan dapat terjadi bila adanya penyerahan barang atau jasa, dan dilakukan pembayaran terhadap pembelian. Sedangkan untuk penjualan kredit, yakni adanya tenggang waktu di antara penyerahan barang atau jasa. Untuk penjualan kredit, adanya tanda bukti penerimaan barang yang terjadi saat penyerahan barang atau jasa. Keuntungan atas transaksi penjualan tunai merupakan hasil penjualan yang diterima bentuk kas (Fitriyana & Sucipto, 2020).

4. Pemesanan

Pemesanan dalam arti umum adalah perjanjian antara 2 pihak atau lebih, perjanjian tersebut dapat berupa produk atau jasa (Mamay Syani & Nindi Werstantia, 2018).

5. *Android*



Gambar 2 .2 Logo *Android*

Android adalah sebuah sistem operasi untuk perangkat lunak *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel / *smartphone*. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *Htc*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *android* dibawah *lisensi Apache*, sebuah *lisensi* perangkat lunak dan *open platform* perangkat seluler. Didunia ini terdapat dua jenis distributor sistem operasi *android*. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar - benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal

sebagai *Open Handset Distribution (OHD)*.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini *android* menjadi pesaing utama dari *Apple* pada sistem operasi Table PC. Pesatnya pertumbuhan *Android* selain faktor yang disebutkan diatas adalah karena *android* itu sendiri adalah *platform* sangat lengkap baik itu sistem operasinya, aplikasi dan *Tool Developmen, Market* aplikasi *android* serta dukungan yang sangat tinggi dari komunitas *Open source* didunia, sheingga *android* terus berkembang pesat dari segi teknologi maupun dari segi jumlah *device* yang ada didunia (Yunus et al., 2015).

6. *Android studio*



Gambar 2 .3 Logo Android Studio

Android Studio merupakan sebuah IDE (*Integrated Development Environment*) untuk mengembangkan aplikasi *Android*. Aplikasi ini diterbitkan oleh Google pada tanggal 16 Mei 2013 dan tersedia secara gratis dengan lisensi *Apache 2.0*. *Android Studio* menggantikan perangkat lunak pengembangan *Android* sebelumnya yaitu *Eclipse*. IDE (*Integrated Development Environment*) adalah aplikasi untuk pengembang perangkat lunak yang berisi fungsi-fungsi terintegrasi yang diperlukan untuk membangun perangkat lunak, seperti editor kode, debugger, kompiler, dan sebagainya.

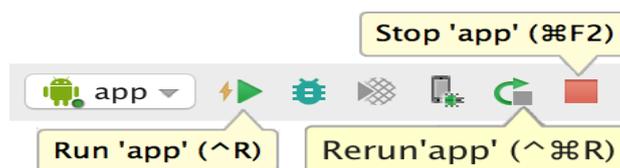
Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA*, mirip dengan *Eclipse*, disertai dengan plugin ADT (*Android Development Tools*).

Android studio memiliki fitur:

- a. Proyek berdasarkan *Gradle Build*.
- b. *Refactoring* cepat dan perbaikan *bug*.
- c. Tools baru bernama "*Lint*" mengklaim dapat dengan cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.
- d. Mendukung *Proguard* dan penandatanganan aplikasi untuk keamanan
- e. Memiliki GUI aplikasi *Android* lebih mudah
- f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para *programmer*, khususnya programmer tingkat dasar yang ingin mempelajari lebih lanjut tentang *Android*. Meski menggunakan *Android studio* memakan cukup banyak RAM pada perangkat PC, namun *Android Studio* memiliki sejumlah keunggulan lain, yaitu:

1) *Instant RUN*

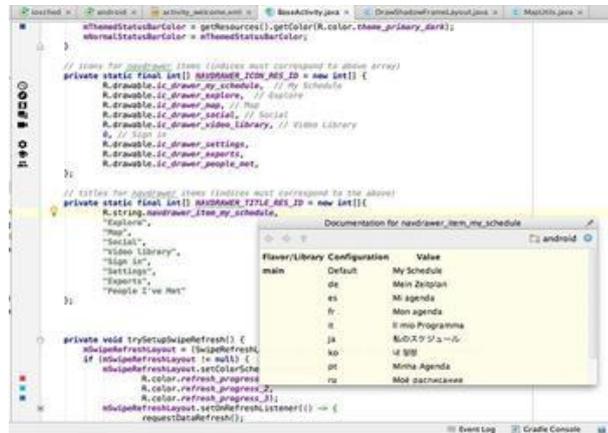


Gambar 2.4 *Instant Run*

Fitur *Instant Run* dapat memastikan program berjalan dengan cepat tanpa perlu mengkompilasi ulang aplikasi atau membuat ulang APK saat melakukan

perubahan kode, sehingga proses yang dihasilkan lebih cepat.

2) *Intelligent Code Editor*



Gambar 2.5 *Intelligent Code Editor*

Android Studio memiliki *Intelligent Code Editor* yang memudahkan analisis kode dan memberikan saran kode untuk digunakan dengan sistem *auto complete*. Saat kita mengetik kode, *Android Studio* akan secara otomatis menyarankan kelas jika kita memiliki kelas yang diinstal dan kita dapat menekan tombol TAB untuk memasukkan kode jika sesuai dengan kebutuhan. Fitur-fitur tersebut tentunya mempercepat pembuatan program sehingga membuat kinerja pembuat program menjadi lebih produktif.

3) *Sistem Versi yang Fleksibel*

Android Studio menawarkan otomatisasi versi, manajemen dependensi, dan konfigurasi versi yang dapat disesuaikan. Anda dapat mengonfigurasi proyek Anda untuk menyertakan pustaka lokal dan yang *dihosting*, serta menentukan varian versi yang berisi kode berbeda. Teman-teman bisa mengkonfigurasi dan menginstall *library* yang memudahkan teman-teman dalam membuat aplikasi *Android*. Fitur ini merupakan bagian dari *fleksibilitas Android Studio*.

4) Dioptimalkan untuk semua perangkat *Android*

Android Studio memberi Anda tempat untuk membuat aplikasi untuk berbagai perangkat *Android*, seperti tablet *Android*, *Android Wear*, *Android TV*, dan *Android Auto*. Fungsi terstruktur ini memungkinkan Anda membagi proyek menjadi unit-unit fungsional yang bisa Anda buat, uji, dan men-debug sesuai keinginan Anda.

5) Di desain untuk Tim

Android Studio memiliki integrasi dengan beberapa kontrol versi populer seperti *Git* dan *Subversion*. Bahkan untuk memudahkan kolaborasi, kita juga bisa menggunakan layanan *Github* langsung dari *Android Studio*. Dengan cara ini, pembaca dan tim terus bekerja secara efektif dengan proyek-proyek yang mudah diakses satu sama lain (Erni Sri Wahyuni, 2019).

7. Java



Gambar 2 .6 Logo Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh *James Gosling* saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih

sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi berbasis *java* umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin *Virtual Java (JVM)*. *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa platform sistem operasi yang berbeda, *java* dikenal pula dengan slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini *java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*.

Java adalah bahasa pemrograman yang *multiplatform* dan *multidevice*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang *support Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam code dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan nonspesifik. Kelebihan *Java* yang pertama tentu saja *multiplatform*. *Java* dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang sudah dibahas sebelumnya. Yang kedua adalah OOP atau *Object Oriented Programming*.

Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam *Java*. Hal ini akan memudahkan

pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas *Java*. Setiap hal pasti memiliki kelebihan dan kekurangan (Irsan, 2015).

8. *MySql*



Gambar 2 .7 *Logo MySql*

MySQL adalah perangkat lunak sistem manajemen *basic* 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 komersional. *MySQL* sebenarnya merupakan turunan dari *SQL* (*Structured Query Language*), salah satu konsep utama dalam database lama. Menurut Abdul Kadir *MySQL* adalah nama database server. Database server yang berfungsi untuk menangani database. Database adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan *MySQL*, kita bias menyimpan data dan kemudian data bias diakses dengan yang mudah dan cepat (Muhammad et al., 2021).

SQL merupakan konsep operasi basis data, khususnya konsep operasi basis data yang digunakan untuk 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* (Ramadhan & Mukhaiyar, 2020).

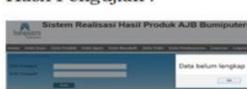
9. Black Box Testing

Blackbox testing terfokus pada fungsional dari program yang ada. Pada *Blackbox testing* diuji dengan cara menjalankan program kemudian diamati apakah program tersebut apakah berhasil atau tidak. *Blackbox testing* menggunakan teknik *equivalence partitions* yang merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada program, setiap menu masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasar fungsinya.

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode *BlackboxTesting* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batasatas dari data yang di harapkan,Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan

diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid. Solusi praktis peningkatan akurasi perlu dilakukan segera guna memperbaiki celah *error* yang telah ditemukan, selanjutnya dilakukan pengujian keamanan secara intensif melalui jaringan internal (*whitebox penetration testing*) secara berkala oleh *System Administrator* atau Pengelola Sistem Informasi, khususnya bagi yang mengelola perangkat lunak tersebut dan Untuk mencapai tingkat akurasi, dimana semua parameter akurasi yang terkait aspek kerahasiaan, integritas data, dan avabilitas data dapat terpenuhi, maka harus dipertimbangkan metode lain yang dapat dijadikan tolak ukur standar keamanan informasi (Cholifah et al., 2018).

Berikut ini adalah pengujian *black box* pada sistem *Monitoring Inventory Control* pada Asuransi Jiwa Bersama (AJB) Bumiputera untuk fungsi menu Laporan (2016), yaitu sebagai berikut:

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<p>Tambah Laporan (Data yang di <i>input</i> tidak lengkap), lalu klik "Simpan"</p> <p><i>Test Case :</i></p> 	<p>Sistem tidak akan menyimpan ketika kolom tidak terisi semua, maka akan menampilkan pesan "Data belum lengkap"</p> <p><i>Hasil Pengujian :</i></p> 	Valid
2	<p>Tambah Data laporan <i>input</i> data Dari Tanggal dan S/D Tanggal lalu klik <i>Print</i></p> <p><i>Test Case :</i></p> 	<p>Sistem sukses, berhasil mencetak Laporan maka akan menampilkan Laporan Perperiode yang diinginkan</p> <p><i>Hasil Pengujian :</i></p> 	Valid

Gambar 2 .8 Contoh *Blackbox Testing*

10. *White Box Testing*

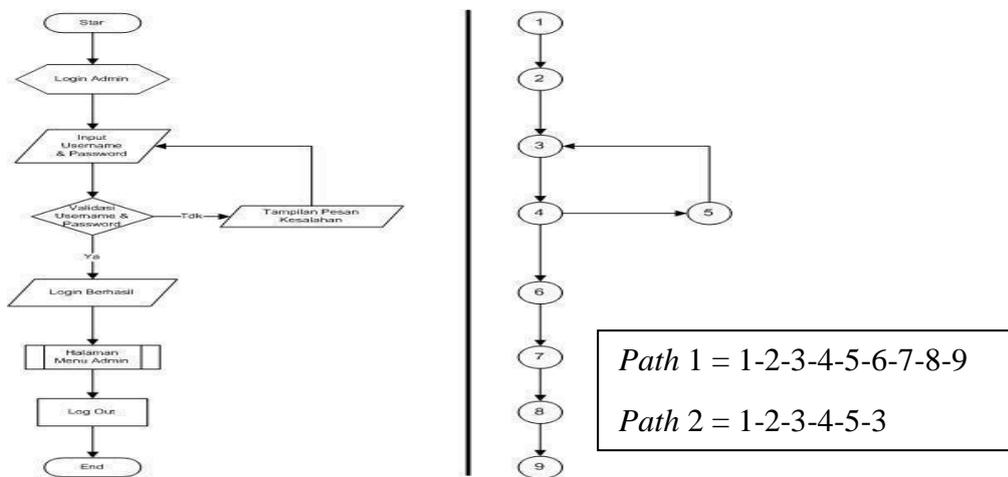
Whitebox testing bertujuan untuk mengetahui apakah struktur pada aplikasi yang dibuat sudah sesuai dengan ketentuan. *Whitebox testing* menitik beratkan pada pengujian dengan mengecek detail perancangan perangkat lunak. *Whitebox testing* dinilai dengan mendefinisikan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian, kemudian menguji kasus tersebut untuk memperoleh hasilnya.

White box testing disebut sebagai pengujian structural. Yang mana perangkat lunak yang diuji merupakan hal transparan kepada penguji. Dalam pengujian, uji dirancang dari perspektif pengembang dikarenakan struktur internal dikenal dengan menguji segala bagian kode yang mampu untuk diuji dengan tujuan untuk menentukan kesalahan logis dari kode sumber perangkat lunak. Tipologi acak kesalahan bisa ditemukan dan mampu memunculkan asumsi yang salah dalam sebuah program. Pengujian *white box* digunakan di segala tingkatan perancangan perangkat lunak. Penguji perlu melihat sumber kode dan mampu menentukan bagian mana kode yang tidak berfungsi dengan baik dan benar. *White box testing* membutuhkan akses kepada kode sumber dan dilakukan

pengembangan perangkat lunak. Pengujian *white box* dikenal dengan nama pengujian *clear box*, *glass box*, atau *open box*. *White box testing* memiliki kelebihan, seperti mampu menghapus bagian asing dari kode-kode yang tersembunyi, melakukan pengujian secara menyeluruh karena seluruh bagian dari struktur atau logika dieksplorasi, membantu dalam mengoptimalkan kode, dan pengujian *white box testing* dapat dimulai walaupun GUI masih dalam masa

pengembangan (Praniffa et al., 2023).

Berikut ini adalah contoh pengujian *white box* pada sistem *login* yang terdapat pada Sistem Informasi laporan Data Statistik Desa Pananrang berbasis *Web*, yaitu :



Gambar 2 .9 Contoh *Whitebox Testing*

Perhitungan
cyclomatic

complexity dari *flowgraph* diatas memiliki $region = 2$. menghitung *cyclomatic complexity* dari *edge* dna *node*, dengan rumus :

$$V(G) = E - N + 2$$

$$E(edge) = 9, N(node) = 9$$

$$\text{Penyelesaian : } V(G) = 9 - 9 + 2$$

$$V(G) = 2$$

Jadi jumlah path pada *flowgraph* di atas adalah 2. *Independent path* pada

flowgraph diatas adalah :

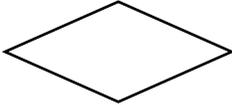
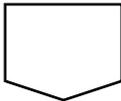
Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9

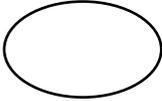
Path 2 = 1 - 2 - 3 - 4 - 5 - 3

11. *Flowchart*

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso, 2017).

Tabel 2. 1 *Symbol dan Fungsi Flowchart*

Simbol	Fungsi
	Permulaan sub program
	Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	Penghubung bagian- bagian <i>flowchart</i> yang berada pada halaman berbeda
	Permulaan/akhir program

	Arah aliran program
	Proses inisialisasi/pemberian harga awal
	Proses penghitung/ proses pengolahan data
	Proses <i>input/output</i> data
	Penghubung bagian- bagian <i>flowchart</i> yang berada pada satu halaman.

12. UML (*Unified Modelling Language*)

(Wardhani, 2014) UML (*Unified Modelling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek. Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

1. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML):

- a. *Structural things*, bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
- b. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu

- c. *Grouping things*, bagian yang pengorganisasian dalam *UML*. Dalam penggambaran model *UML* yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.
- d. *An notational things*, merupakan bagian yang memperjelas model *UML*. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model *UML*.

2. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

- a. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen *independent* akan mempengaruhi elemen yang bergantung padanya.
 - b. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
 - c. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi.
 - d. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.
- ## 3. Diagram, *Unified Modelling Language* (UML) menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan

dinamis).

Diagram *use case* menyajikan intraksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun.

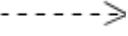
Adapun simbol-simbol *use case* diagram antara lain :

Tabel 2. 2 *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
4		<i>Generalization</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Extend</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Association</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboratio</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

NO.	GAMBAR	NAMA	KETERANGAN
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2. 3 *Symbol Class Diagram*

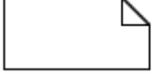
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 4 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
---	---	----------------	---

Tabel 2. 5 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai <i>Link</i> pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 6 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

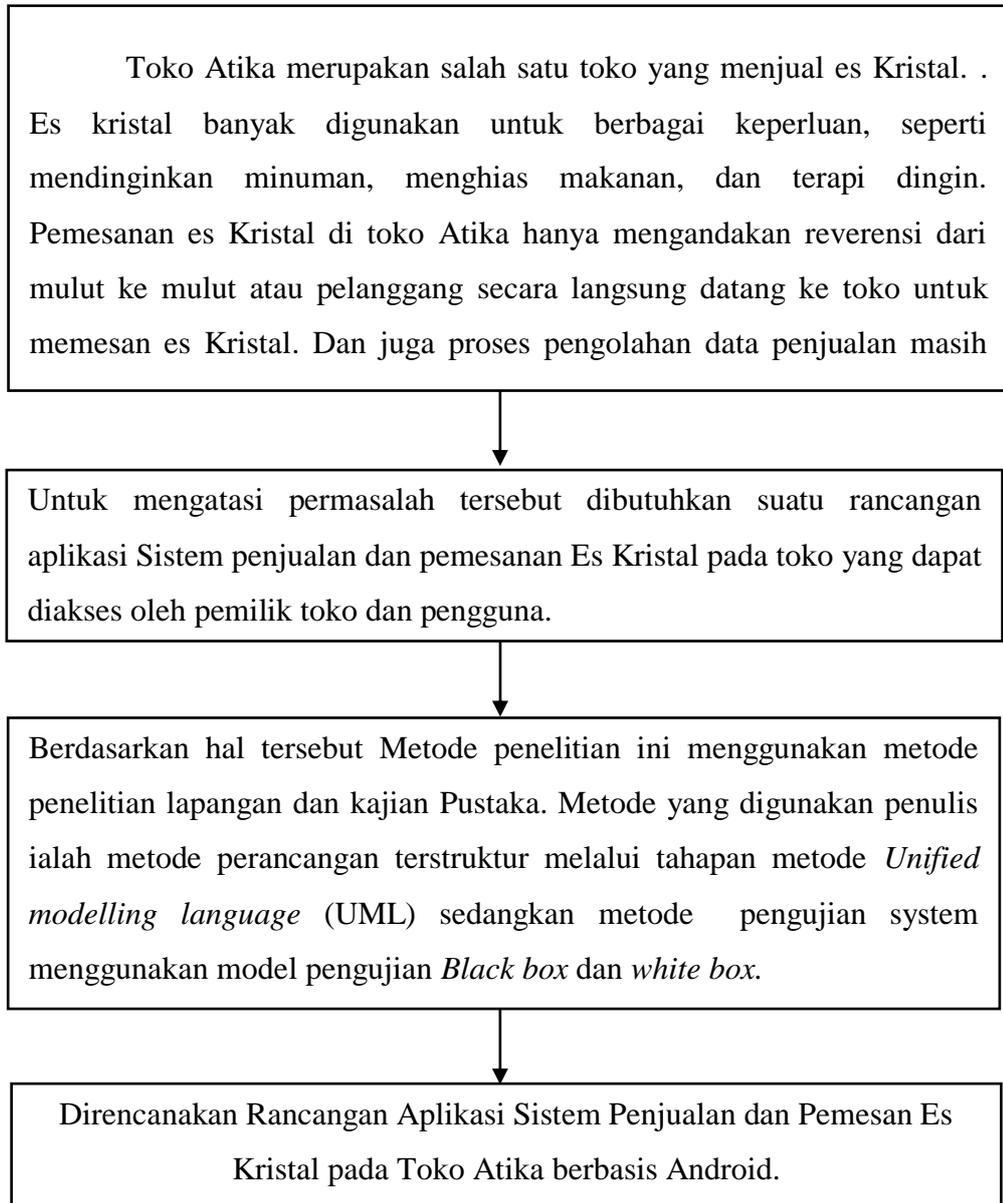
B. Kajian Penelitian Terdahulu

Penelitian mengacu pada beberapa penelitian sebelumnya yang menjadi acuan dalam penelitian ini, penelitian yang relevan tersebut antara lain:

1. (Kharisma Rizki A. Wijaksono, Muh.Fadli Hasa, 2022) dengan judul penelitian ” **RANCANG BANGUN APLIKASI PENJUALAN IKAN LAUT BERBASIS ANDROID** “ hasil riset yang dilakukan oleh aplikasi dapat membantu nantinya penjual dan pembeli bisa langsung melakukan transaksi jual beli tanpa harus pembeli datang secara langsung ke pasar ikan, aplikasi ini akan mempermudah para ibu rumah tangga yang tidak bisa datang secara langsung dikarenakan memiliki kesibukan dan mungkin dengan cara ini dapat membantu sebuah UMKM hasil laut dalam meningkatkan penjualan mereka.
2. (P. Eni & F. Siti, 2021) dengan judul penelitian “**PERANCANGAN APLIKASI PENJUALANAN BERBASIS ANDROID SEBAGAI MEDIA PEMESANAN PADA DISTRO ONLINE**” Analisis dari aplikasi startup yang dilakukan, diperoleh kesimpulan sebagai berikut. Pertama, Aplikasi ini diharapkan dapat menjadi solusi bagi penjual untuk fokus pada penjualan produk distro secara online dimasa pandemi.

C. Kerangka Pikir

Untuk memahami kerangka pikir penelitian ini dapat dilihat dibawah ini.



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Proses pengumpulan data kualitatif dapat dilakukan dengan berbagai cara yang didapatkan dengan terjun langsung ke lapangan. Caranya bisa melalui pengamatan atau observasi, kuisioner, wawancara mendalam dengan objek penelitian, pengkajian dokumen, hingga fokus *discussion group*.

B. Tempat Dan Waktu Penelitian

1. Tempat

Toko Es Kristal Atika di Jl. Mangga Tengah No.47, Labukkang, Kec.Ujung, Kota Parepare, Sulawesi Selatan 91111.

2. Waktu penelitian

Penelitian ini berlangsung kurang lebih 2 bulan.

C. Metode Pengumpulan Data

1. Observasi

Observasi adalah metode mengumpulkan data dengan melakukan pengamatan secara langsung ketempat.

2. Wawancara

Wawancara adalah teknik pengumpulan data yang dilakukan melalui hubungan personal dan tanya jawab langsung kepada narasumber.

3. Studi literatur

Studi literatur yaitu pengumpulan data dengan dengan cara mengambil data, membaca, mempelajari literatur dari sumber-sumber seperti buku, skripsi, jurnal yang berhubungan dengan penelitian.

D. Tahapan Penelitia

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh *user*.

E. Alat dan bahan penelitian

1. Perangkat Keras

- a. Asus vivobook
- b. *Processor Intel® Core™ i3-4005U (3MB, 1.7GHz)*
- c. *Installed RAM* : 8.00 GB
- d. *SSD* : 475 GB

2. Perangkat Lunak

- a. Windows 11
- b. *Android Studio*
- c. *Java*

F. Metode pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *blackbox testing* dan *whitebox testing*:

1. Blackbox testing

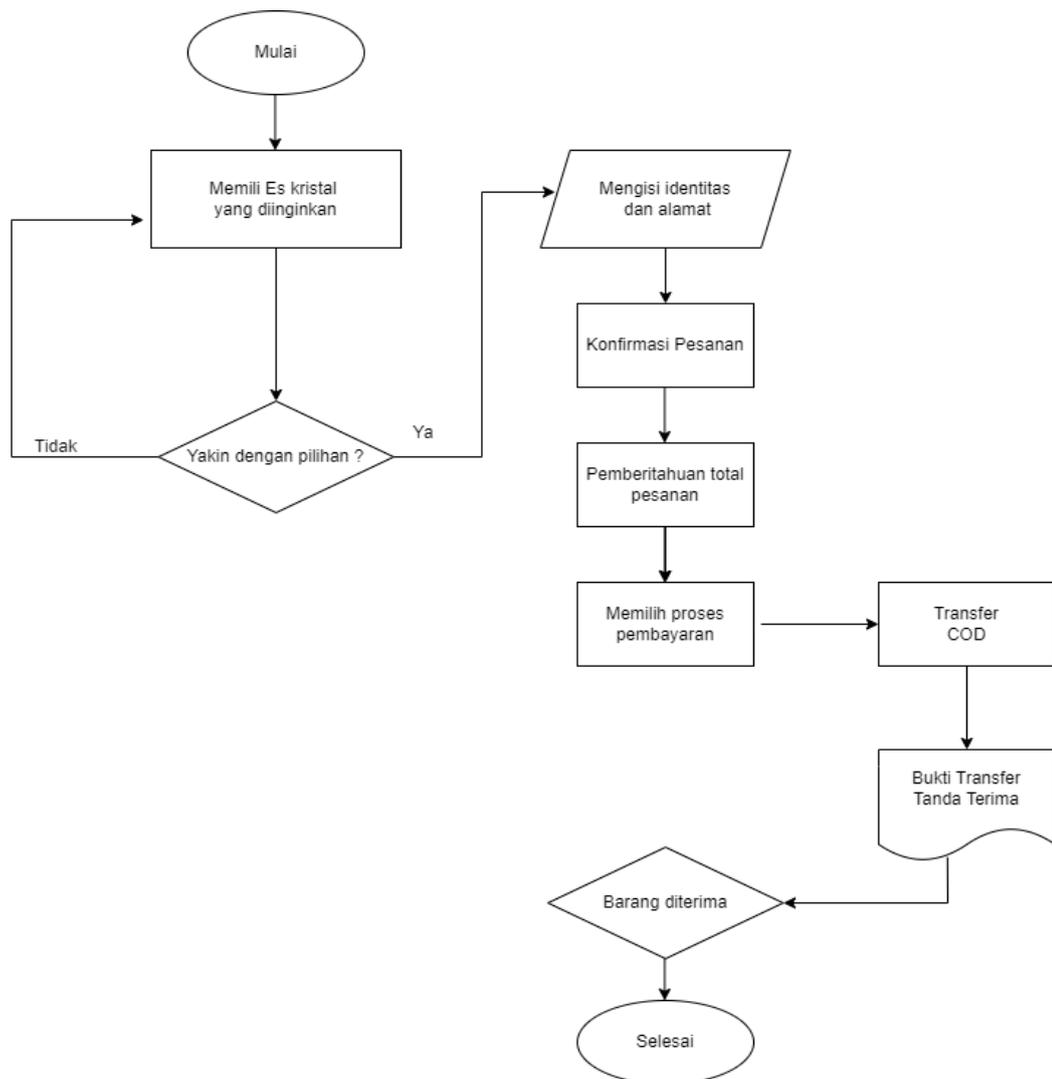
Blackbox testing yaitu untuk menguji program yang sedang berjalan

apakah aplikasi *wedding organizer* yang dibuat sudah berhasil atau tidak.

2. Whitebox testing

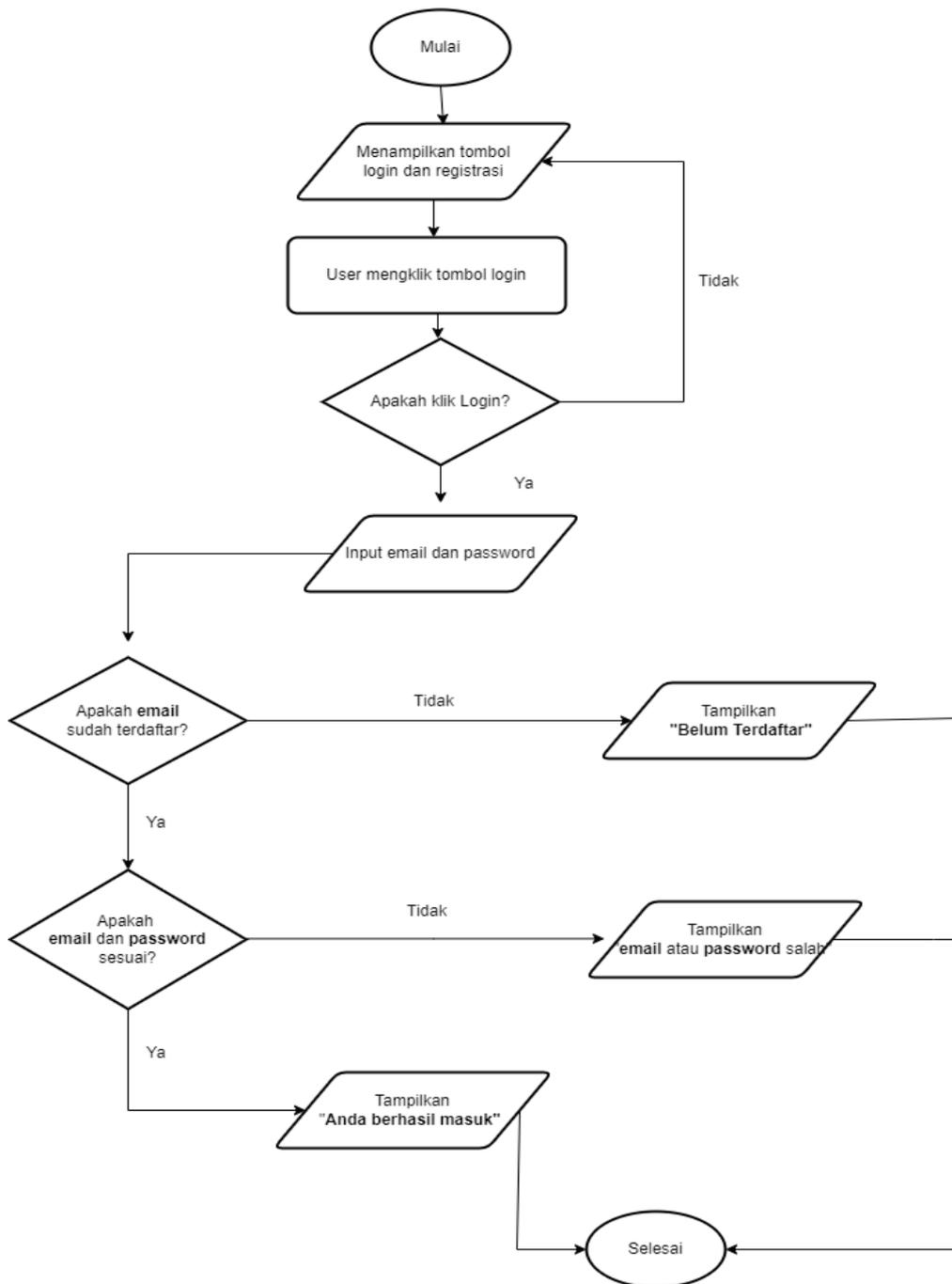
Whitebox testing yaitu untuk mengetahui struktur dan alur pada aplikasi *wedding organizer* apakah sudah sesuai dengan ketentuan kemudian menguji aplikasi tersebut untuk memperoleh hasil.

G. Diagram Alir



Gambar 3.1 Flowchart Pemesanan

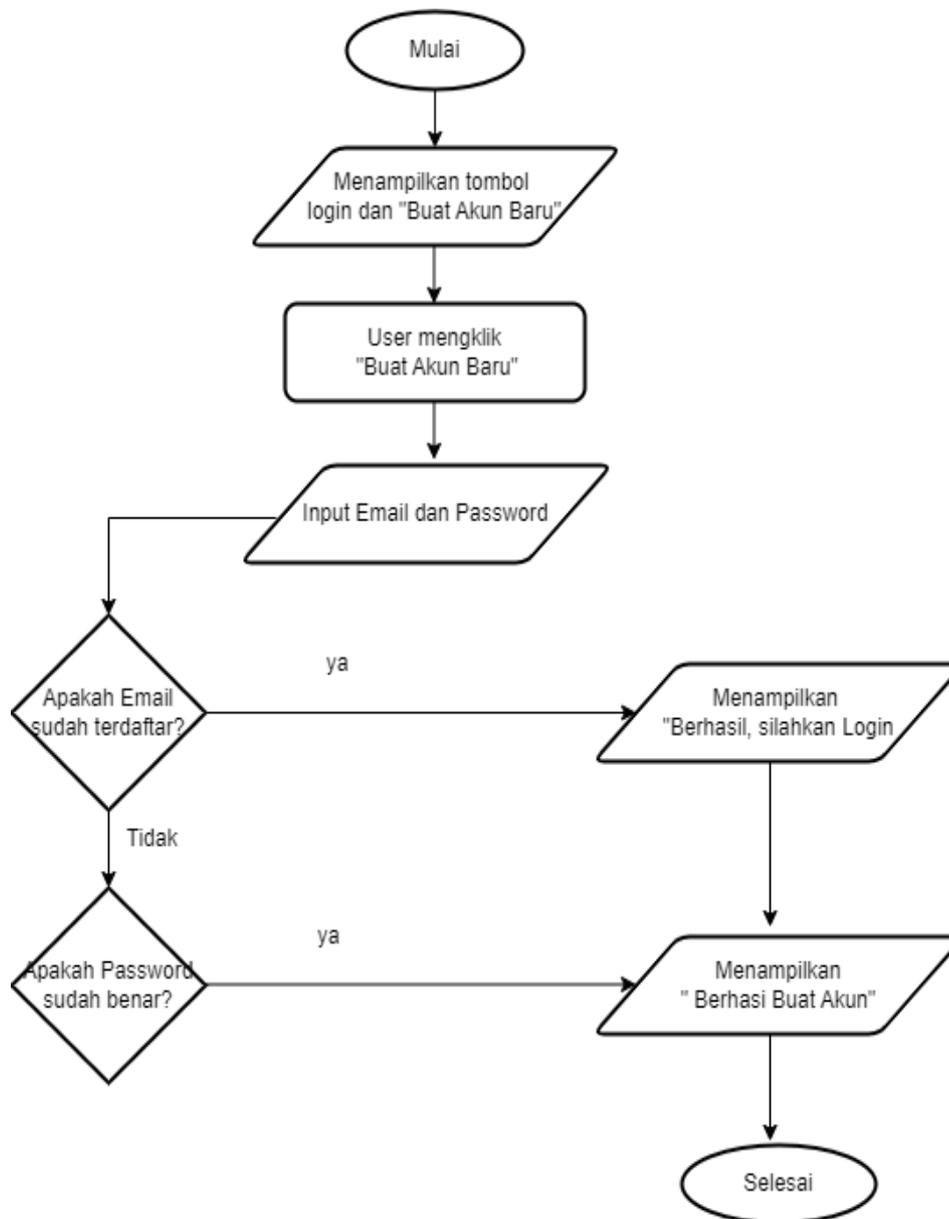
Pada gambar di atas menjelaskan mengenai alur dari proses pemesanan yaitu *costomer* akan mengakses aplikasi kemudian memilih produk yang diinginkan dan apabila yakin dengan produk maka akan mengisi identitas dan alamat lengkap apabila tidak maka kembali ke halaman memilih produk kemudian apabila sudah mengisi identitas dan alamat lengkap maka akan menampilkan konfirmasi pesanan kemudian menampilkan pemberitahuan total pembayaran kemudian melakukan proses pembayaran dan terbagi menjadi dua yaitu melalui *Cash On Delivery(COD)* dan *transfer* kemudian menampilkan bukti transfer dan tanda terima kemudian barang di terima dan selesai.



Gambar 3. 2 *Flowchart Login User*

Pada gambar di atas menjelaskan mengenai alur dari *login user* yaitu *user* akan di tampilkan dengan halaman login kemudian *user* mengklik tombol *login* dan apabila *login* berhasil akan masuk ke *input email* dan *password* dan apabila tidak maka kembali ke halaman *login* dan setelah menginput email dan *password*

maka *user* akan di tampilkan halaman apakah *email* sudah terdaftar apabila tidak maka akan menampilkan belum terdaftar dan apabila *email* dan *password* belum sesuai maka akan menampilkan *email* dan *password* salah dan apabila berhasil maka akan muncul anda berhasil masuk dan selesai.

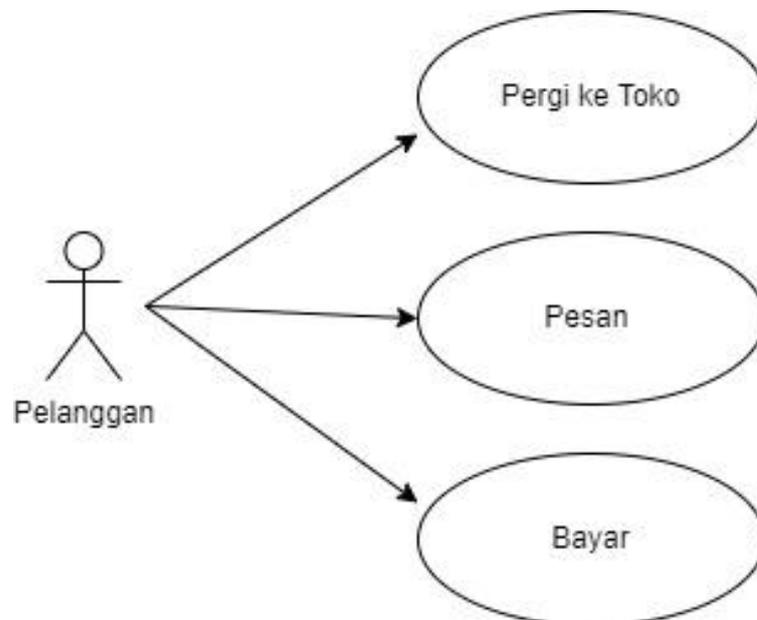


Gambar 3.3 *Flowchart Registrasi User*

Pada gambar di atas menjelaskan mengenai alur dari register *user* yaitu *user* akan ditampilkan dengan menu *register* dan mengklik tombol register apabila tidak maka akan kembali ke halaman register dan apabila berhasil maka *user* akan menginput *email*, *password* satu dan dua kemudian akan di tampilkan apakah *email* sudah terdaftar dan apabila ya maka akan menampilkan sudah terdaftar silahkan *login* dan apabila tidak maka akan menampilkan *password* satu dan dua harus sama dan apabila ya maka akan menampilkan anda berhasil *register* dan apabila tidak maka akan menampilkan *password* satu dan *password* dua harus sama kemudian akan menampilkan apakah user mau mengulangi lagi apabila maka kembali ke halaman tombol *register* dan apabila tidak maka selesai.

H. Desain Sistem

1. Sistem yang berjalan



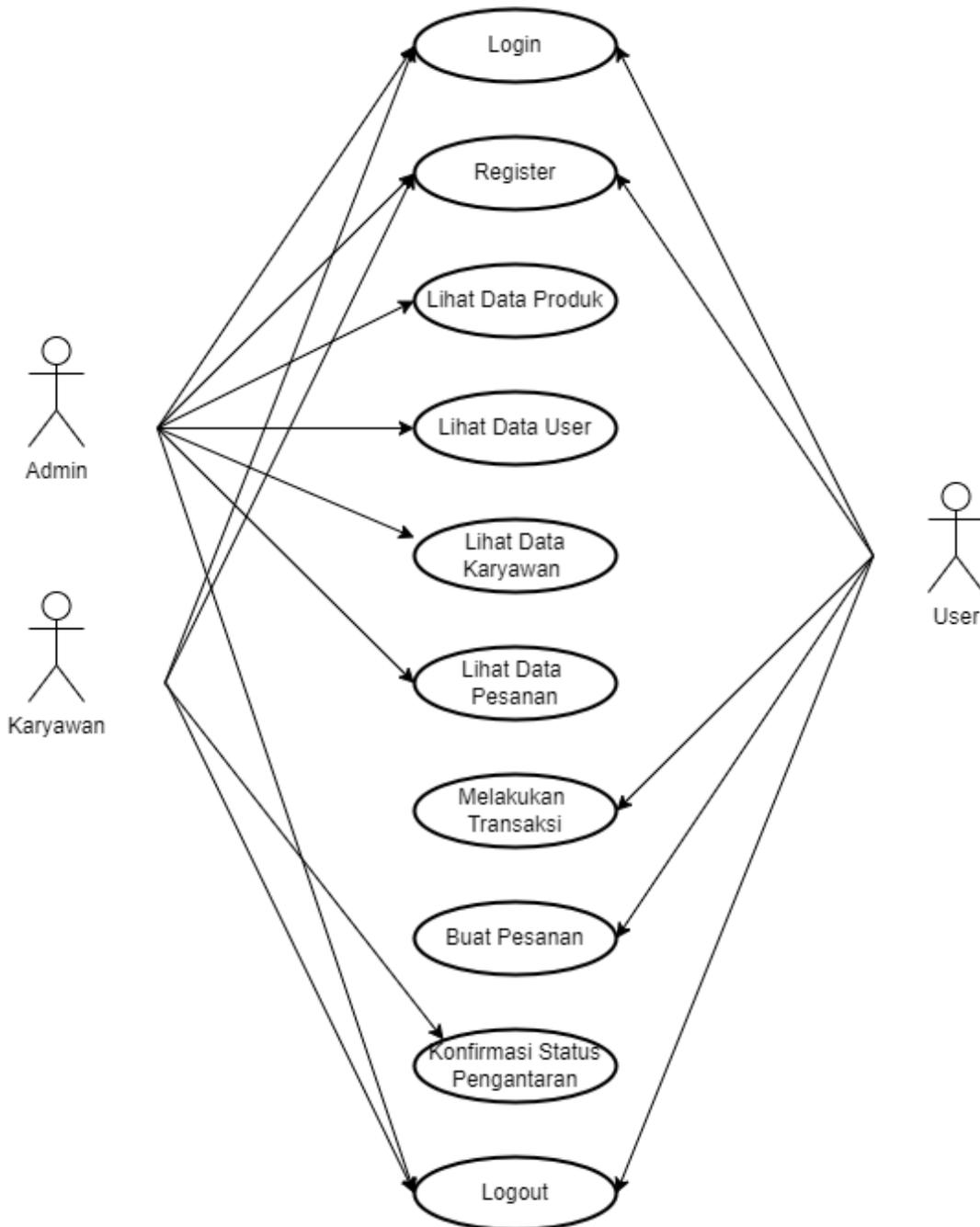
Gambar 3. 4 Sistem yang berjalan

Pada gambar di atas sistem yang berjalan pada saat ini, *user* langsung ke toko untuk membeli langsung dan pemilik atau karyawan di toko tersebut mencatat pemesanan yang dilakukan *costumer* di buku.

Tabel 3. 1 Penjelasan Sisten Berjalan

Nama use case	Deskripsi
<i>User ke toko</i>	<i>Use case</i> ini menjelaskan tentang sistem yang berjalan bahwa <i>user</i> langsung ke toko
Membeli langsung	<i>Use case</i> ini menjelaskan bahwa <i>user</i> membeli langsung
Bayar	<i>Use case</i> ini menjelaskan bahwa <i>user</i> melakukan pembayaran langsung.

2. Sistem yang diusulkan



Gambar 3. 5 Sistem yang diusulkan

Pada gambar di atas terdapat tiga peran, pertama yang bertindak sebagai admin, di mana admin memiliki tanggung jawab untuk mengelola aplikasi baik

itu berupa data-data materi ataupun informasi terkait *user/costumer*. Kedua yaitu *user/costumer*, *user* atau *costumer* dapat mengakses aplikasi dimana nantinya terdapat beberapa *fitur* dalam melakukan pemesanan. Ketiga, yaitu karyawan pengantar dapat mengakses aplikasi untuk mengecek pesanan yang perlu diantarkan di tiap harinya.

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Berdasarkan hasil penelitian yang telah Peneliti laksanakan selama kurang lebih 2 bulan di Toko Atika menunjukkan beberapa data yang telah dihasilkan dapat dilihat pada tabel berikut:

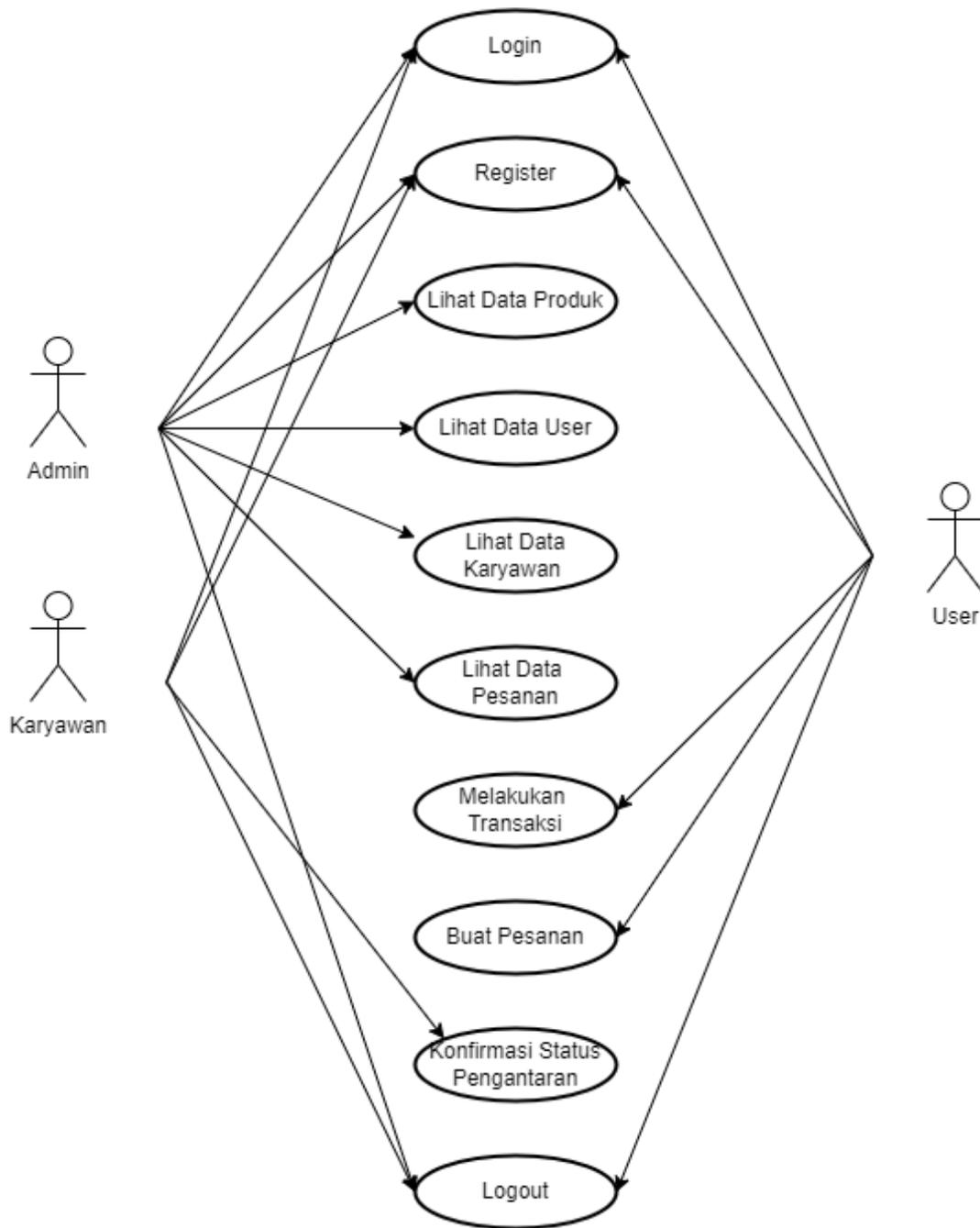
Tabel 4. 1 Jenis Es Kristal

Jenis	Harga
3,7kg	Rp. 5000
7,5kg	Rp. 10.000
11,5kg	Rp. 15.000

B. Analisis Kebutuhan dan Rancangan

1. *Use Case Diagram*

Use Case Diagram berfungsi untuk melakuka kelebihan sistem jika dilihat menurut sudut pandang individu yang berada di luar sistem (*user* dan *admin*)



Gambar 4. 1 Use Case Diagram

a. Aktor (Admin)

Tabel 4. 2 Penjelasan Sistem Yang Diusulkan

Nama use case	Deskripsi
<i>login</i>	<i>Use case ini menjelaskan proses dimana admin login terlebih dahulu.</i>
Registrasi	<i>Use case ini menjelaskan proses dimana admin melakukan registrasi jika belum memiliki akun.</i>
Lihat data produk	<i>Use case ini menjelaskan bahwa admin dapat melihat produk yang ada di dalam aplikasi.</i>
Lihat data <i>user</i>	<i>Use case ini menjelaskan admin dapat melihat data user.</i>
Lihat data Karyawan	<i>Use case ini menjelaskan admin dapat melihat data karyawan</i>
Lihat data pesanan	<i>Use case ini menjelaskan bahwa admin dapat melihat data pesanan user.</i>
<i>Logout</i>	<i>Use case ini menjelaskan bahwa admin dapat logout.</i>

b. Aktor (User)

Tabel 4. 3 Penjelasan Sistem Yang Diusulkan

Nama use case	Deskripsi
<i>login</i>	<i>Use case ini menjelaskan proses dimana user login terlebih dahulu.</i>

Registrasi	<i>Use case ini menjelaskan proses dimana user melakukan registrasi jika belum memiliki akun</i>
Buat pesanan	<i>Use case ini menjelaskan bahwa user membuat pesanan.</i>
Melakukan transaksi	<i>Use case ini menjelaskan bahwa user dapat melakukan melakukan transaksi.</i>
<i>Logout</i>	<i>Use case ini menjelaskan bahwa user dapat logout</i>

c. Aktor (Karyawan)

Tabel 4. 4 Penjelasan Sistem Yang Diusulkan

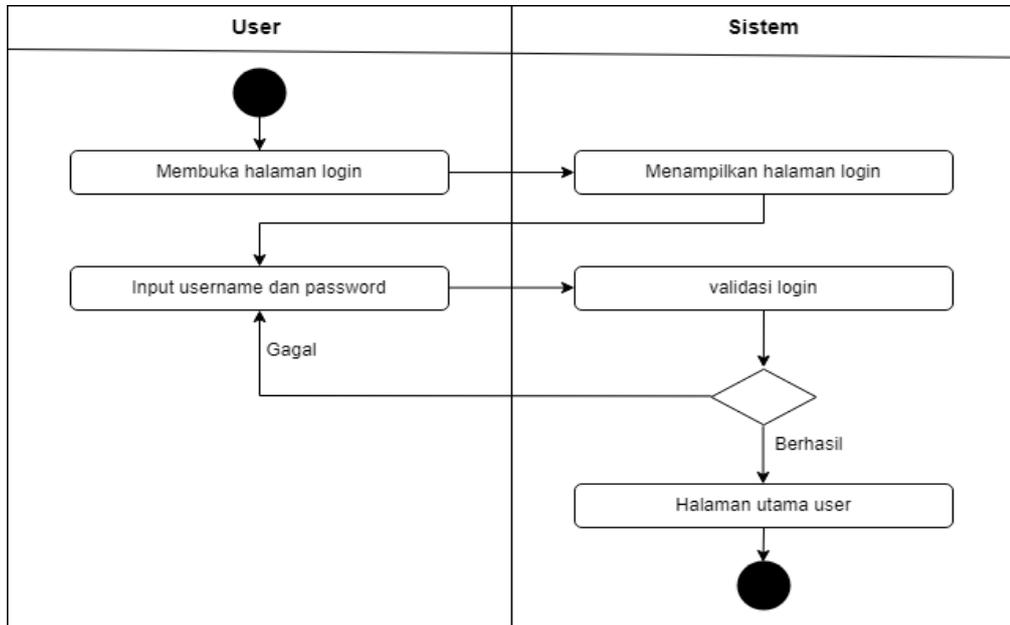
Nama use case	Deskripsi
<i>login</i>	<i>Use case ini menjelaskan proses dimana user login terlebih dahulu</i>
Registrasi	<i>Use case ini menjelaskan proses dimana user melakukan registrasi jika belum memiliki akun</i>
Konfirmasi status pengantaran	<i>Use case ini menjelaskan bahwa karyawan dapat mengubah atau mengkonfirmasi status pengantaran.</i>
<i>Logout</i>	<i>Use case ini menjelaskan bahwa user dapat logout.</i>

2. Activity Diagram

Activity diagram merupakan bahasa spesifikasi standar yang dipakai untuk mendokumentasikan, menspesifikasikan, dan mengembangkan perangkat lunak. Diagram ini menggambarkan berbagai aktivitas dalam aliran proses sebuah sistem.

a. *Activity Diagram User*

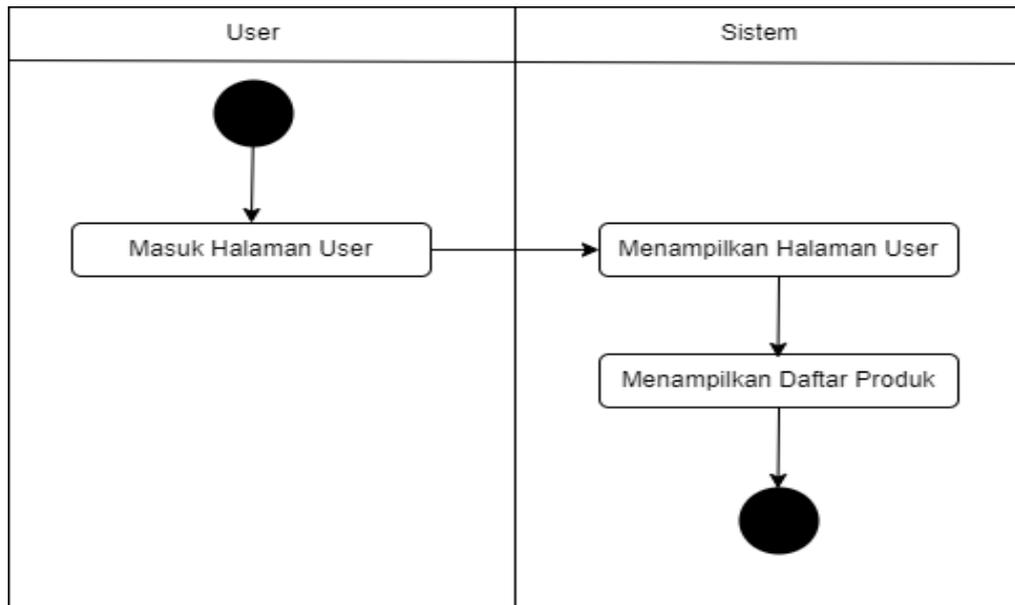
1) *Activity diagram login*



Gambar 4. 2 *Activity Diagram Login*

Gambar di atas menjelaskan proses *login* dari *user*. Sistem akan menampilkan *form login*, kemudian *user* memasukkan *username* dan *password*. Selanjutnya sistem akan melakukan validasi, jika benar maka akan dilanjutkan ke halaman namun jika salah maka akan kembali ke halaman *login* dengan menampilkan pesan kesalahan.

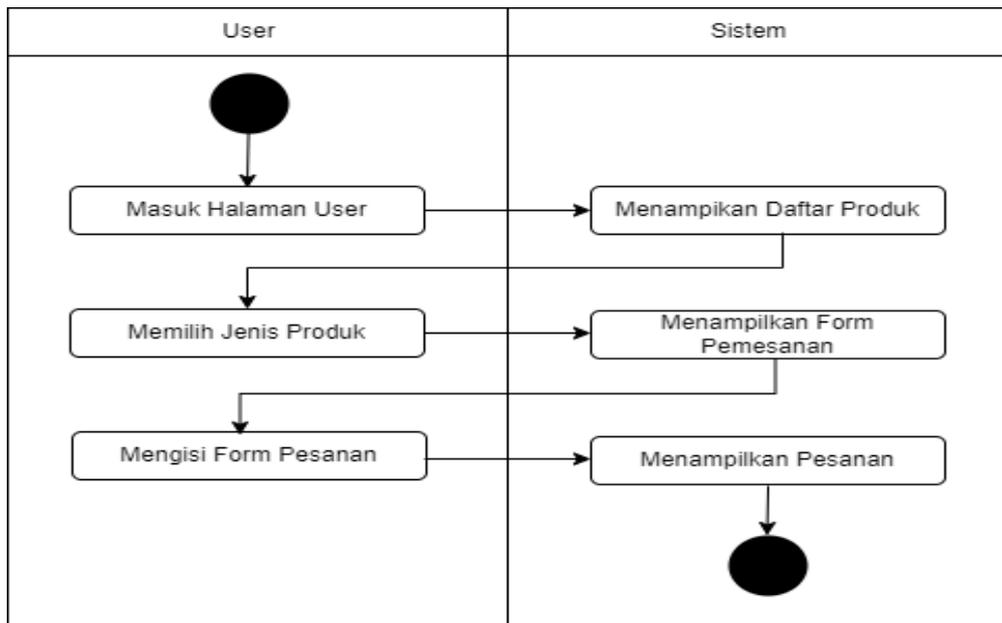
2) *Activity diagram lihat daftar produk*



Gambar 4. 3 *Activity Diagram* Lihat Daftar Produk

Pada gambar di atas menjelaskan bagaimana proses yang dilakukan *user* jika ingin menampilkan halaman daftar produk atau jenis es kristal.

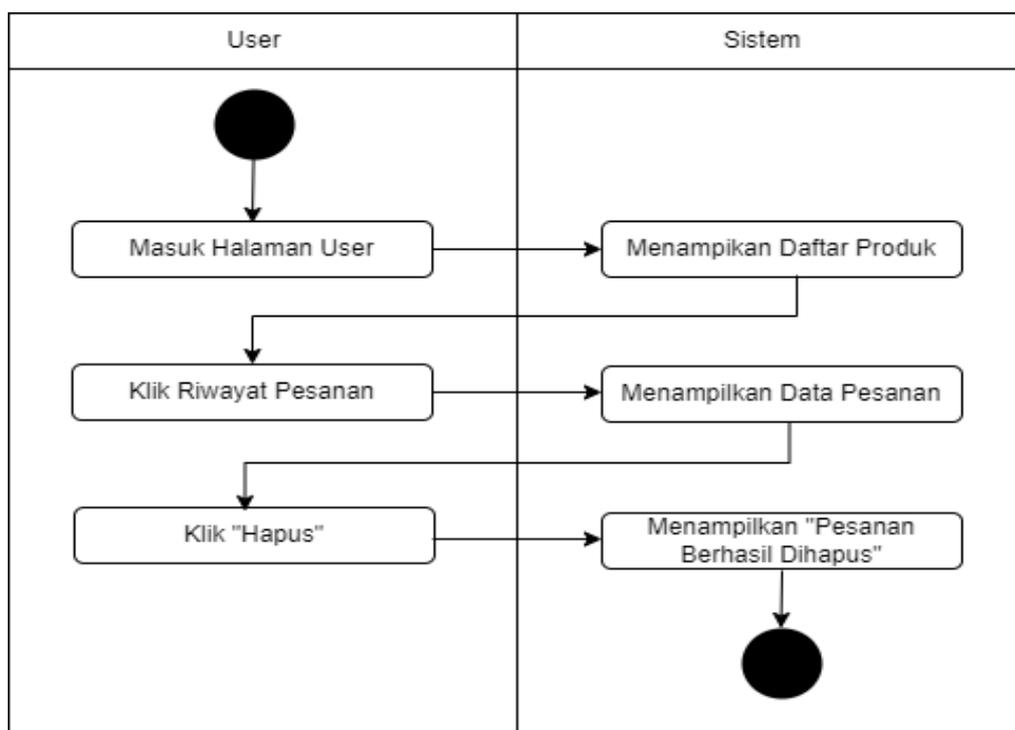
3) *Activiy diagram* pemesanan



Gambar 4. 4 *Activity Diagram* Pemesanan

Pada gambar di atas menjelaskan bagaimana proses pemesanan yang dilakukan *user*. *User* masuk pada halaman utama kemudian sistem menampilkan produk yang akan dipilih oleh *user*. Setelah itu *user* memilih produk kemudian mengisi *form* pemesanan setelah itu *user* dapat mengklik ikon pesan maka sistem akan list pesanan.

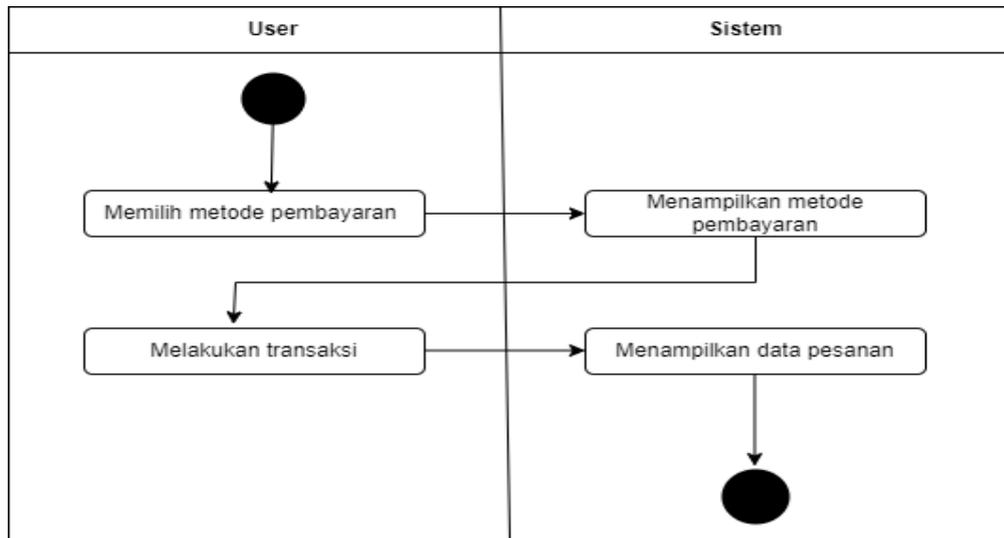
4) Activity diagram hapus data pesanan



Gambar 4. 5 Activity Diagram Hapus Data Pesanan

Pada gambar di atas menjelaskan bagaimana proses yang dilakukan *user* jika ingin menghapus data pesanan. Pertama yang dilakukan adalah membuka halaman *user* kemudian sistem akan menampilkan halaman *user*. Selanjutnya *user* diharapkan mengklik list pesanan, kemudian sistem akan menampilkan pesanan dipilih *user*, selanjutnya *user* memilih *icon* hapus data. Maka sistem akan menampilkan konfirmasi hapus.

5) *Activity diagram transaksi*

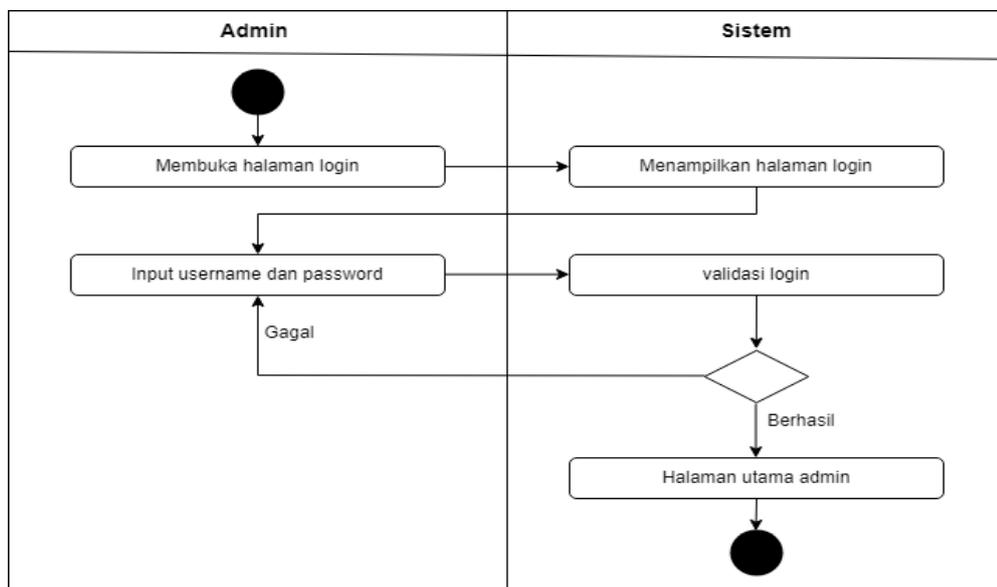


Gambar 4.6 *Activity Diagram Transaksi*

Pada gambar di atas menjelaskan bagaimana proses yang dilakukan *user* jika ingin melakukan transaksi.

b. *Activity diagram admin*

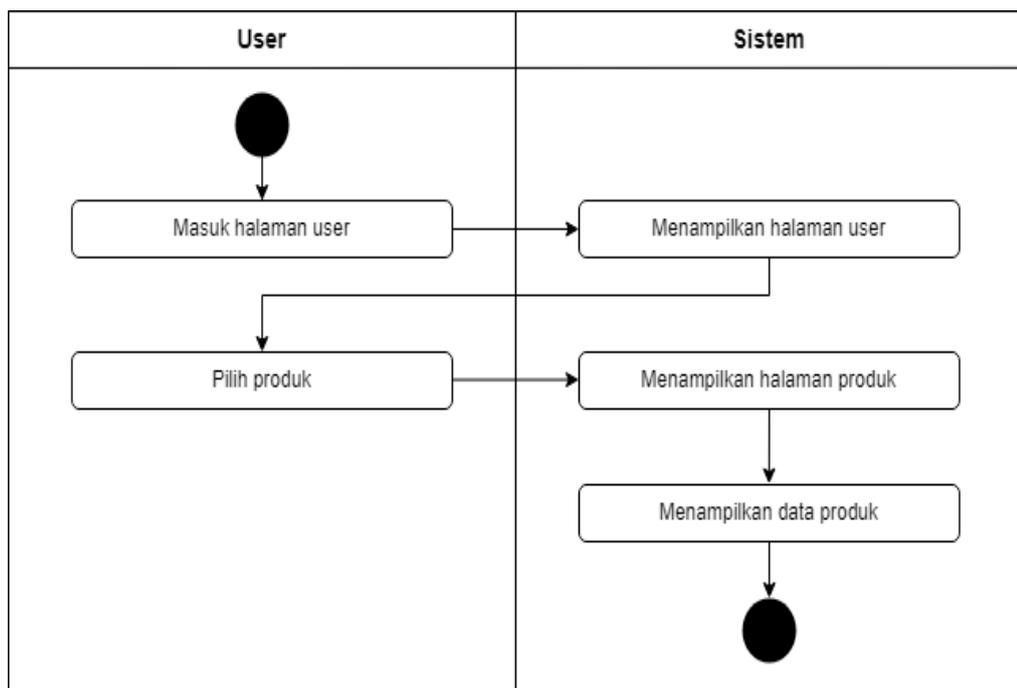
1) *Activity diagram login*



Gambar 4.7 *Activity Diagram Login Admin*

Gambar di atas menjelaskan proses *login* dari *admin*. Sistem akan menampilkan *form login*, kemudian user memasukkan *username* dan *password*. Selanjutnya sistem akan melakukan validasi, jika benar maka akan dilanjutkan ke halaman namun jika salah maka akan kembali ke halaman login dengan menampilkan pesan kesalahan.

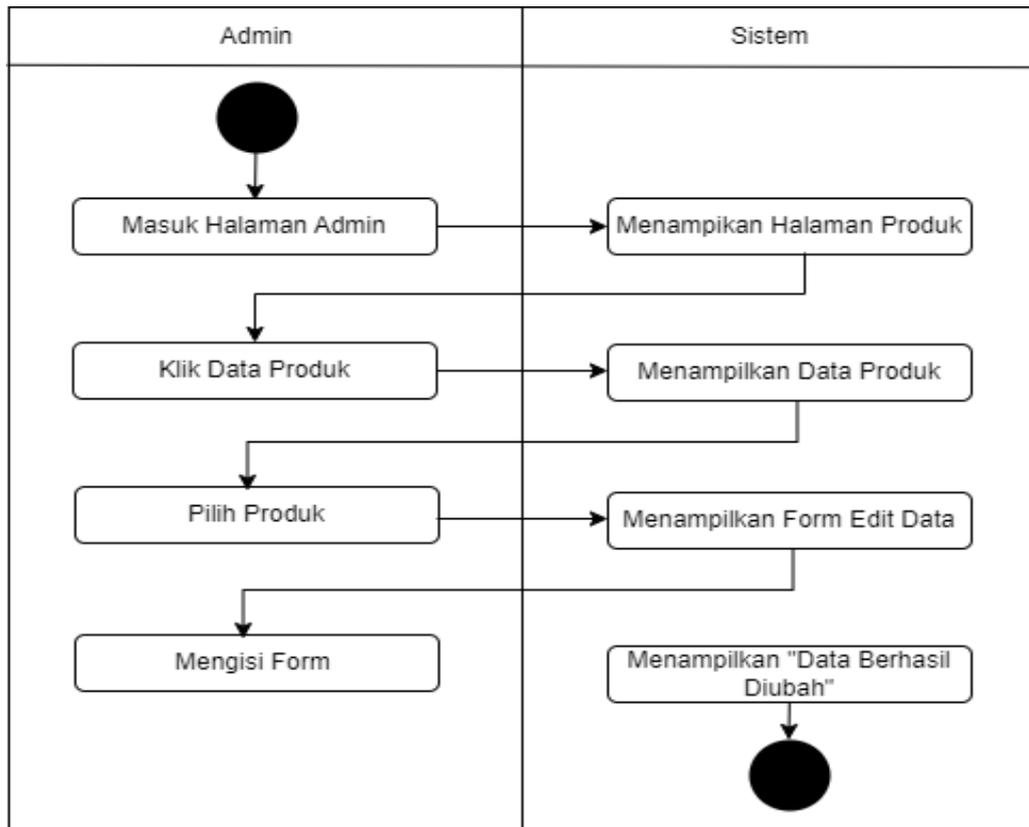
2) Activity diagram lihat data produk



Gambar 4. 8 Activity Diagram Lihat Data Pesanan

Gambar di atas menjelaskan proses admin jika ingin melihat data produk, kemudia sistem akan menampilkan halaman produk dan data produk.

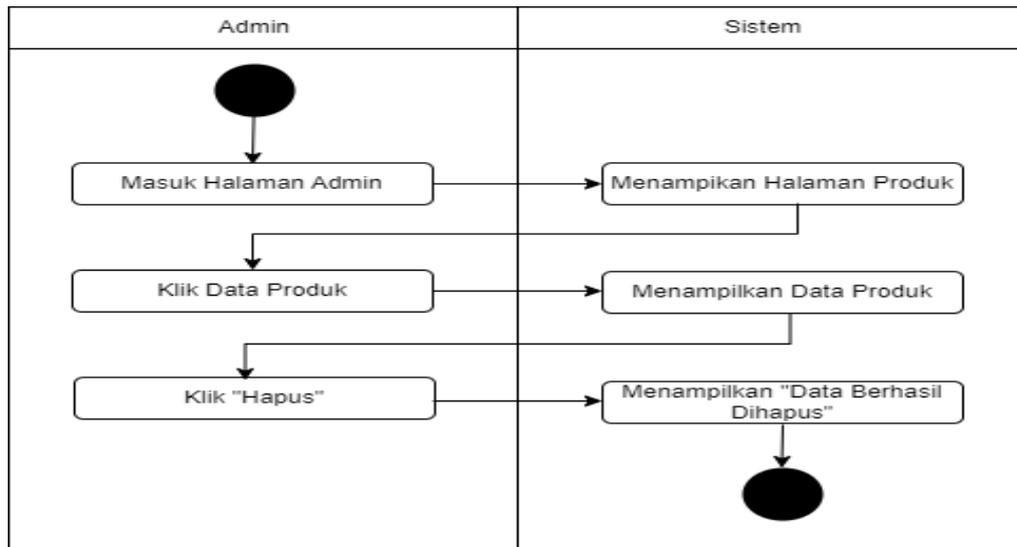
3) Activity diagram edit data produk



Gambar 4. 9 *Activity Diagram* Edit Data Produk

Gambar di atas menjelaskan proses *admin* jika ingin mengedit data produk, sistem akan menampilkan halaman admin dan admin kemudian memilih data produk maka sistem menampilkan produk. Kemudian admin mengklik ikon edit data dan mengisi form data produk dan mensubmit hasil data yang sudah diedit.

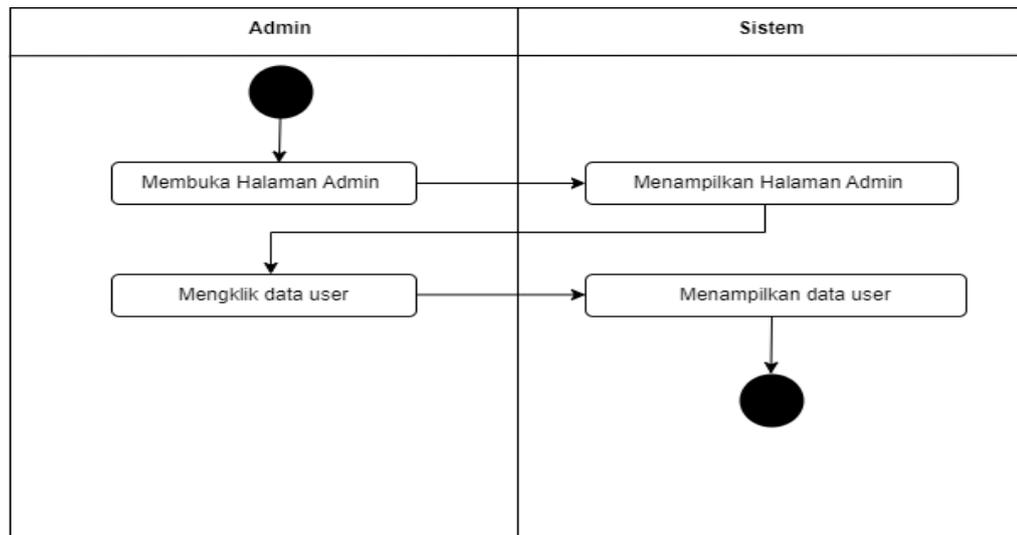
4) Activity diagram hapus data produk



Gambar 4. 10 *Activity Diagram* Hapus Data Produk

Gambar di atas menjelaskan proses *admin* jika ingin menghapus data produk, sistem akan menampilkan halaman admin dan admin kemudian memilih produk dan memilih hapus produk.

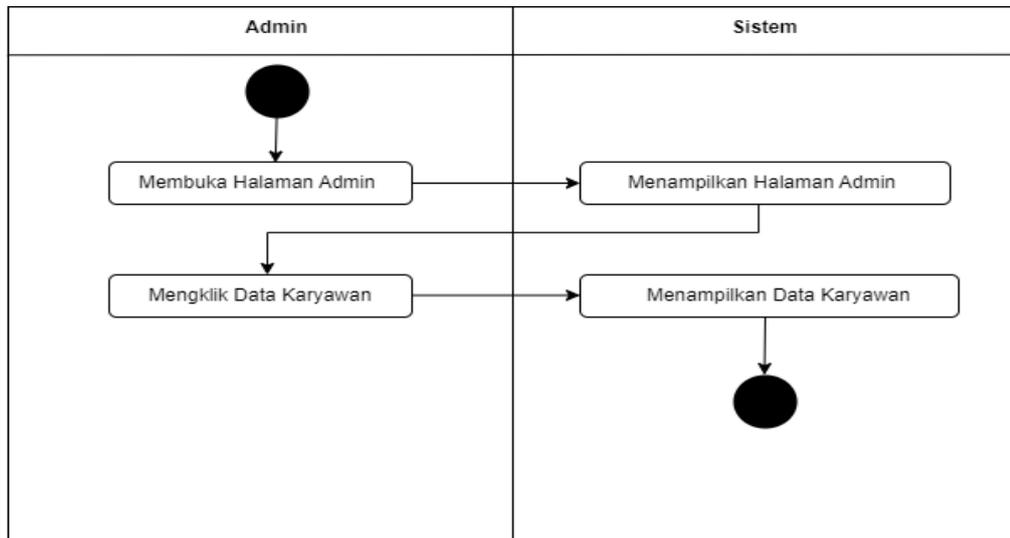
5) *Activity diagram* lihat data user



Gambar 4. 11 *Activity Diagram* Lihat Data User

Gambar di atas menjelaskan proses *admin* jika ingin melihat data *user*, *admin* mengklik data *user* maka sistem akan menampilkan halaman data *user*.

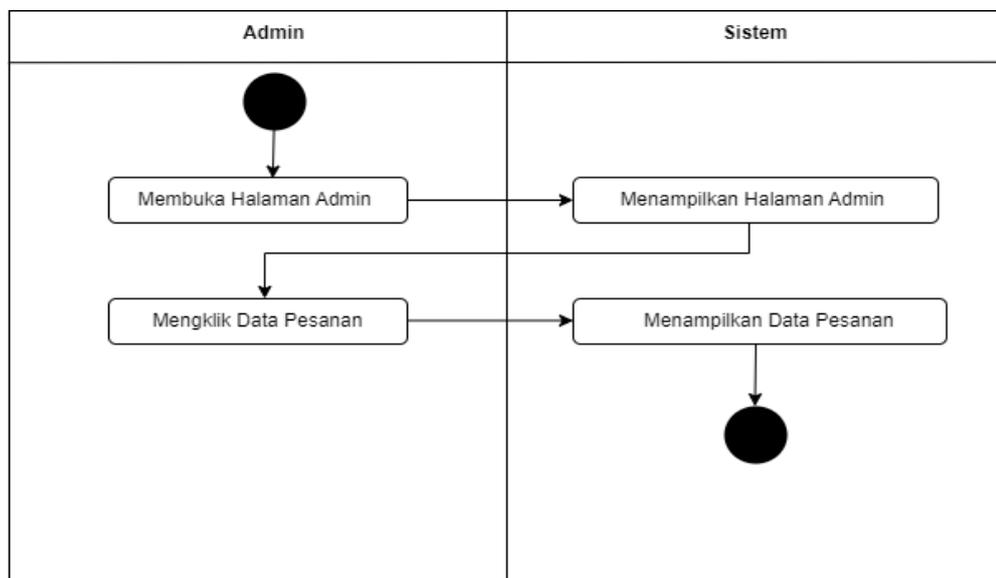
6) *Activity diagram* lihat data karyawan



Gambar 4. 12 *Activity Diagram* Lihat Data Karyawan

Gambar di atas menjelaskan proses *admin* jika ingin melihat data karyawan, *admin* mengklik data karyawan maka sistem akan menampilkan data karyawan.

7) *Activity diagram* lihat data pesanan

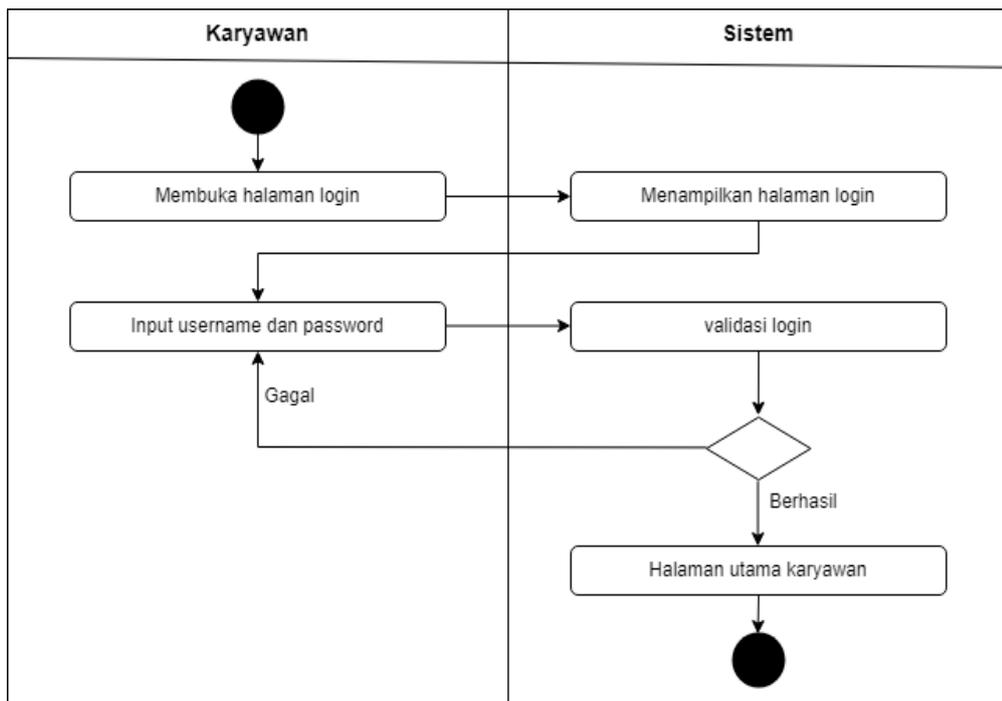


Gambar 4. 13 *Activity Diagram* Lihat Data Pesanan

Gambar di atas menjelaskan proses *admin* jika ingin melihat data pesanan, *admin* mengklik data pesanan maka sistem akan menampilkan data pesanan.

c. Activity diagram karyawan

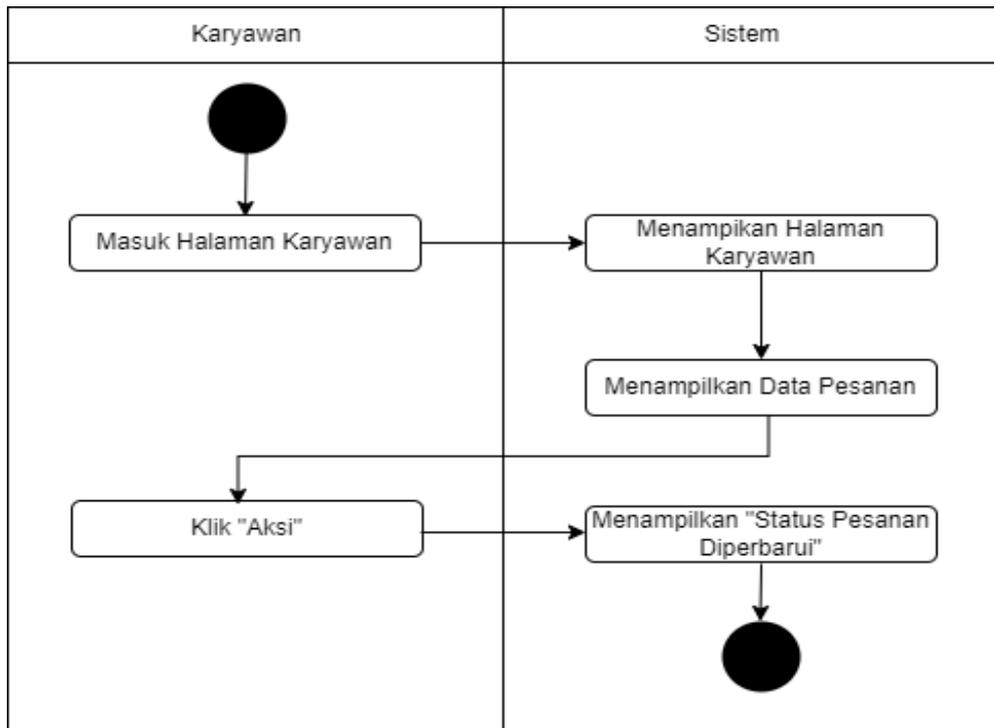
1) Activity diagram login



Gambar 4. 14 Activity Diagram Login Karyawan

Gambar di atas menjelaskan proses *login* dari karyawan. Sistem akan menampilkan *form login*, kemudian karyawan memasukkan *username* dan *password*. Selanjutnya sistem akan melakukan validasi, jika benar maka akan dilanjutkan ke halaman namun jika salah maka akan kembali ke halaman login dengan menampilkan pesan kesalahan.

2) Activity diagram konfirmasi status pengantaran



Gambar 4. 15 *Activity Diagram* Konfirmasi Status Pengantaran

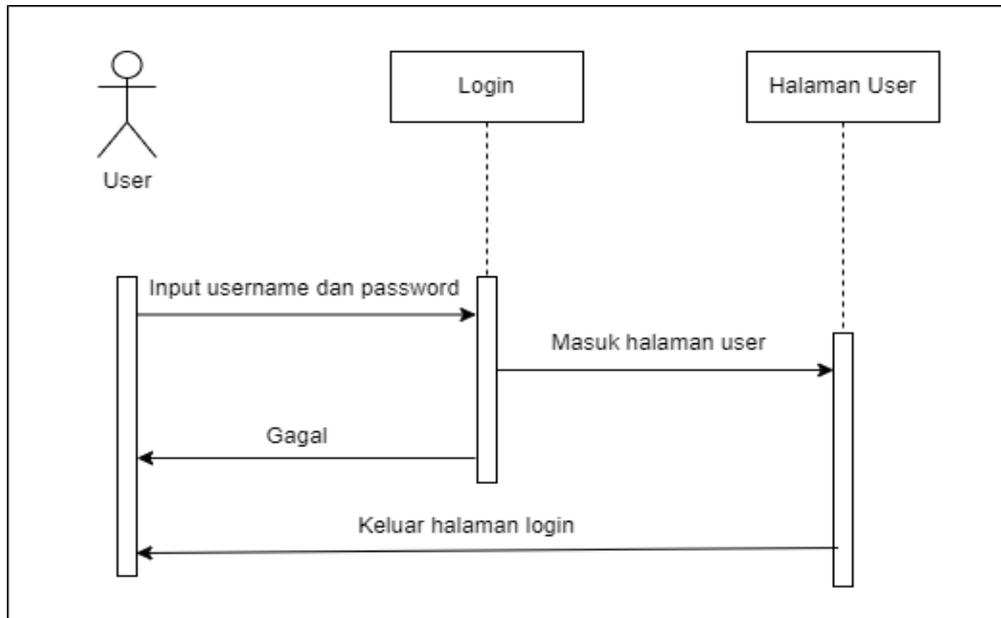
Gambar di atas menjelaskan proses konfirmasi status pengantaran oleh karyawan, dimana karyawan membuka halaman karyawan dan sistem akan menampilkan data pesanan yang dilakukan oleh user (pembeli) kemudian karyawan mengklik aksi(diproses atau dikirim. Sistem akan menampilkan validasi dikirim atau diantar.

3. *Sequence Diagram*

Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

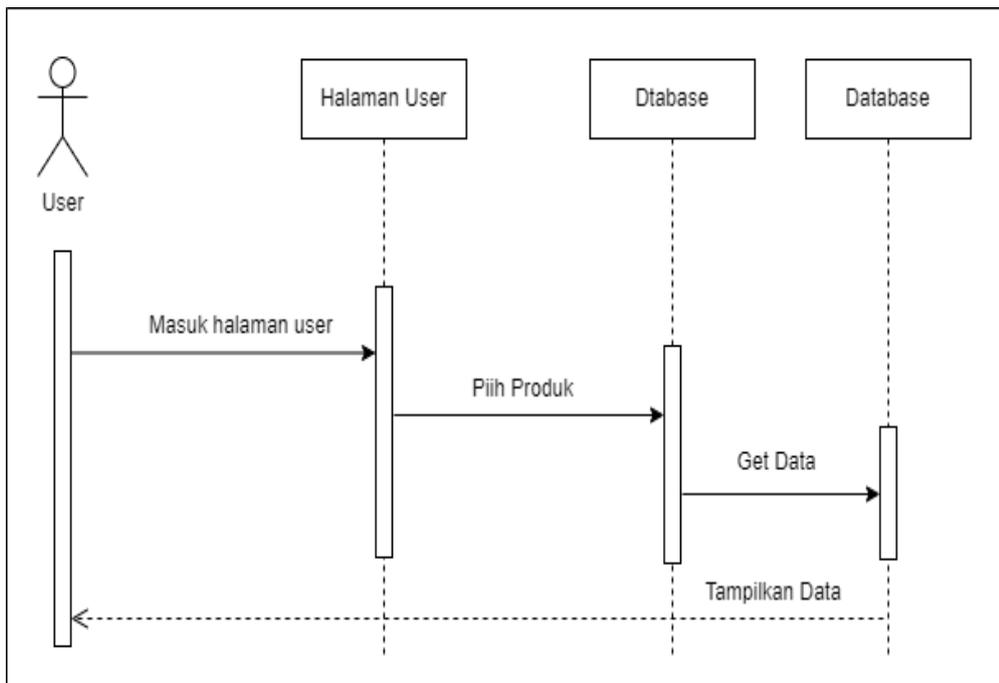
a. *Sequence diagram user*

1) *Sequence diagram login*



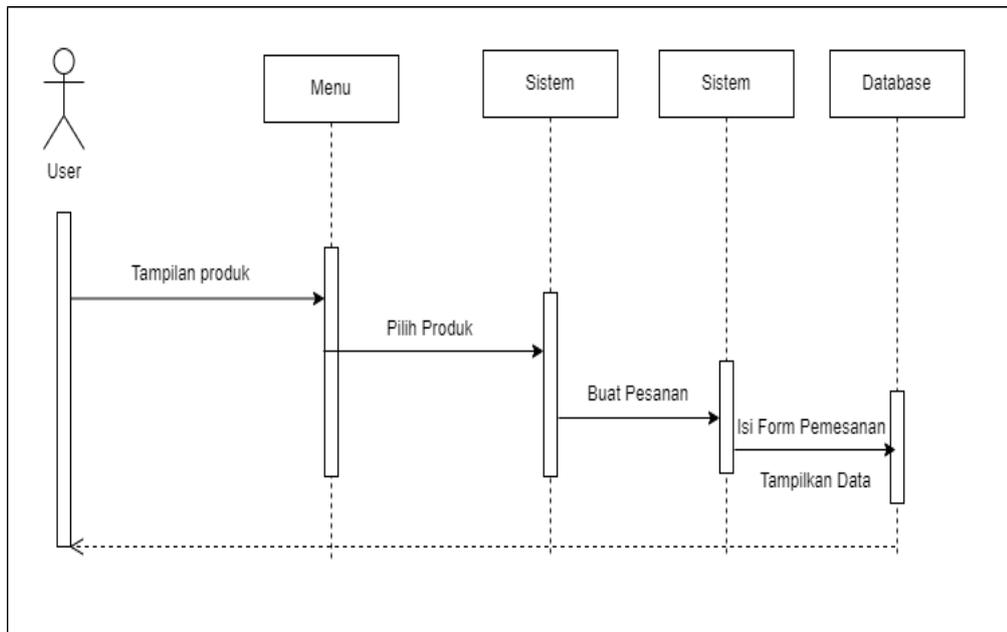
Gambar 4.16 *Sequence Diagram Login User*

2) *Sequence diagram lihat daftar produk*



Gambar 4. 17 *Sequence Diagram Lihat Daftar Produk*

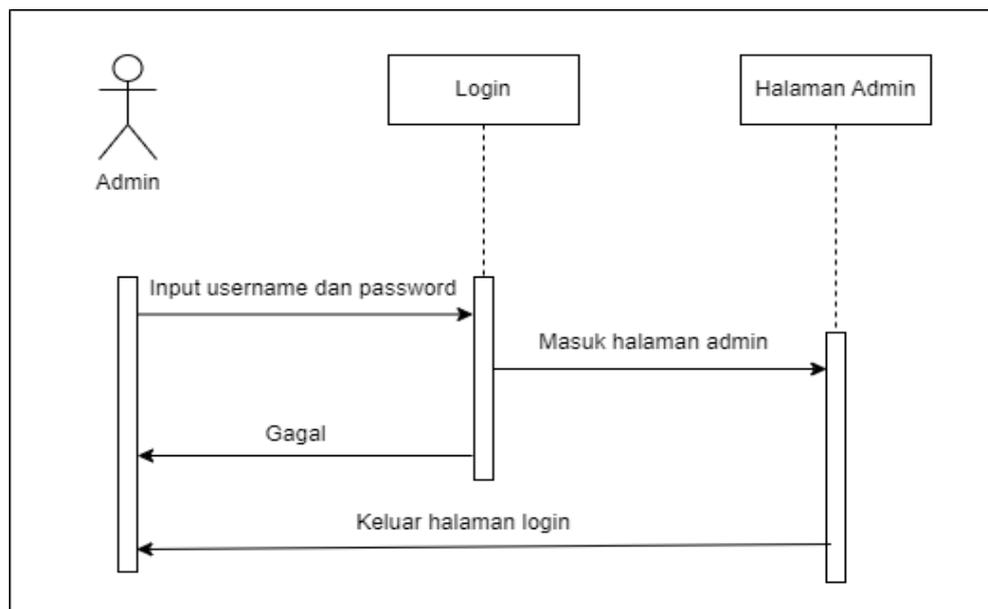
3) *Sequence diagram pemesanan*



Gambar 4. 18 *Sequence Diagram Pemesanan*

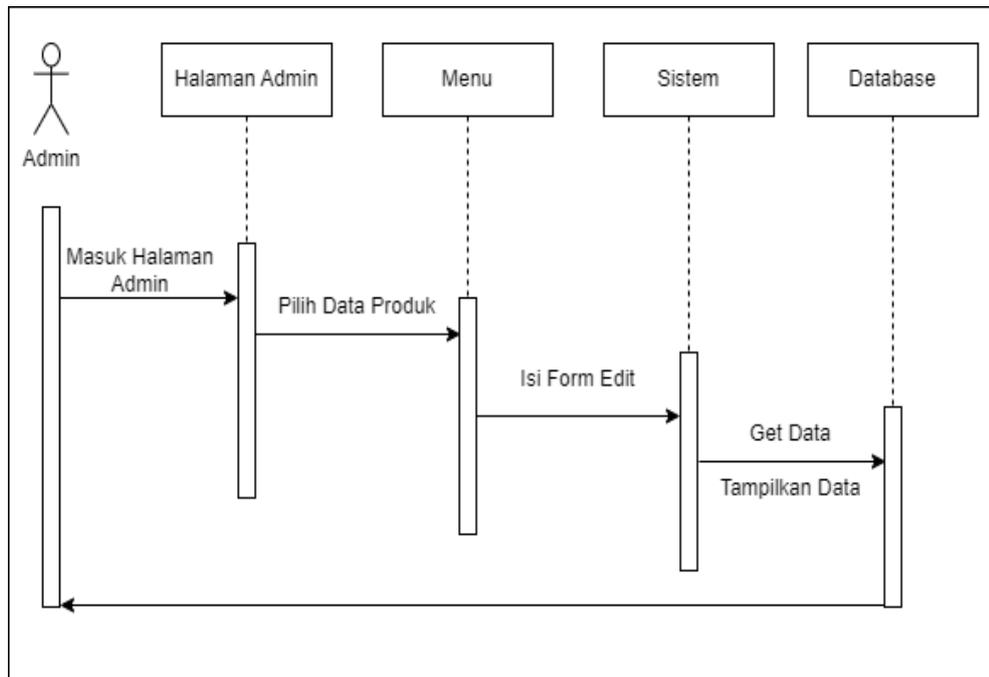
b. *Sequence diagram admin*

1) *Sequence diagram login admin*



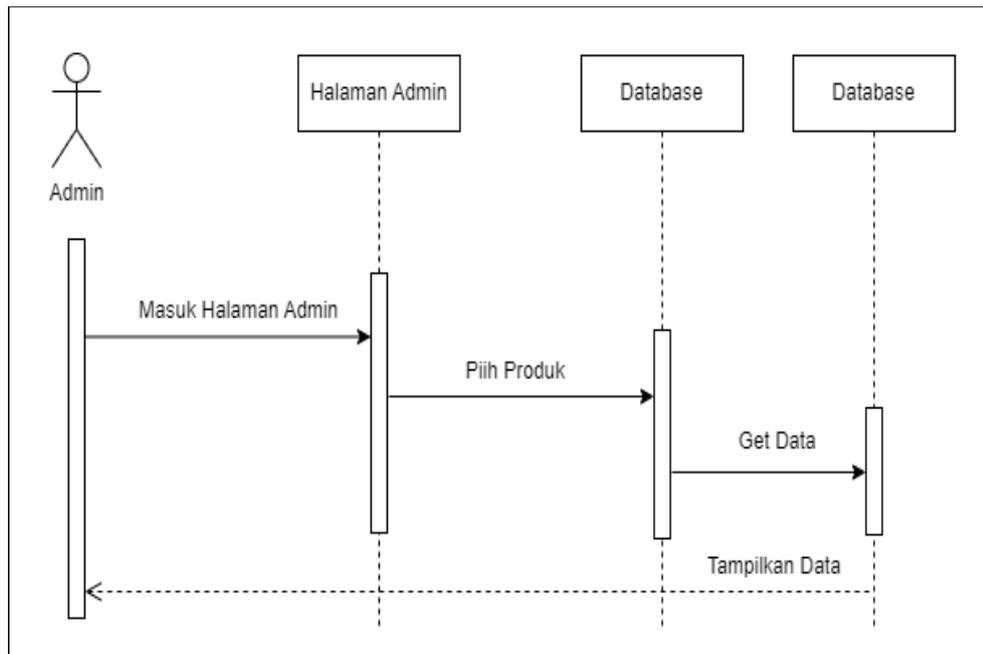
Gambar 4.19 *Sequence Diagram Login Admin*

2) *Sequence diagram edit data produk*



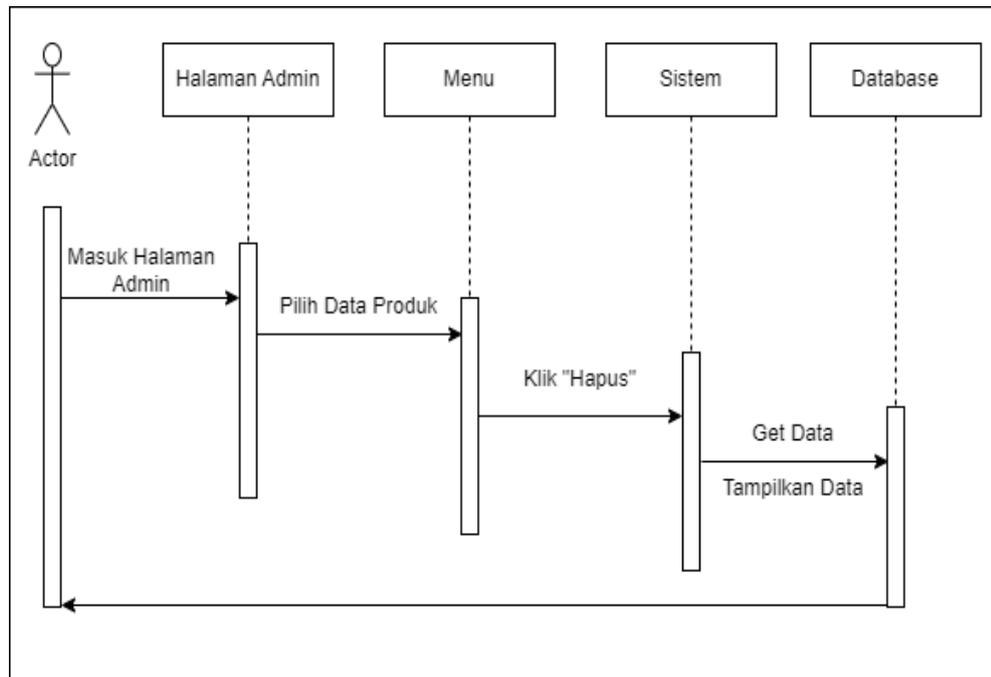
Gambar 4. 20 *Sequence Diagram* Edit Data Produk

3) *Sequence diagram lihat data produk*



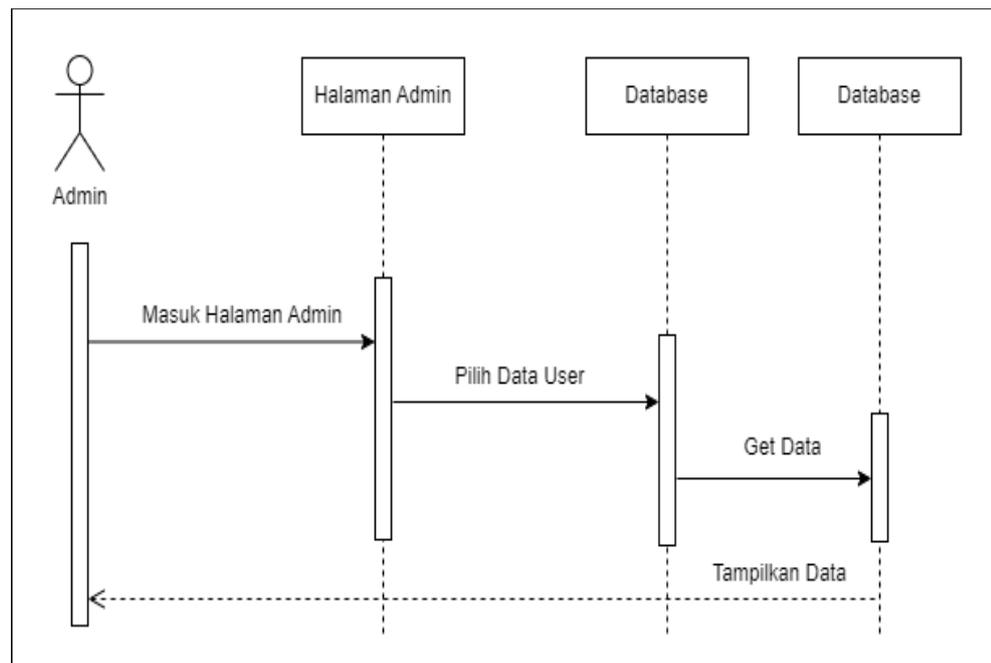
Gambar 4. 21 *Sequence Diagram* Lihat Data Produk

4) *Sequence diagram hapus data produk*



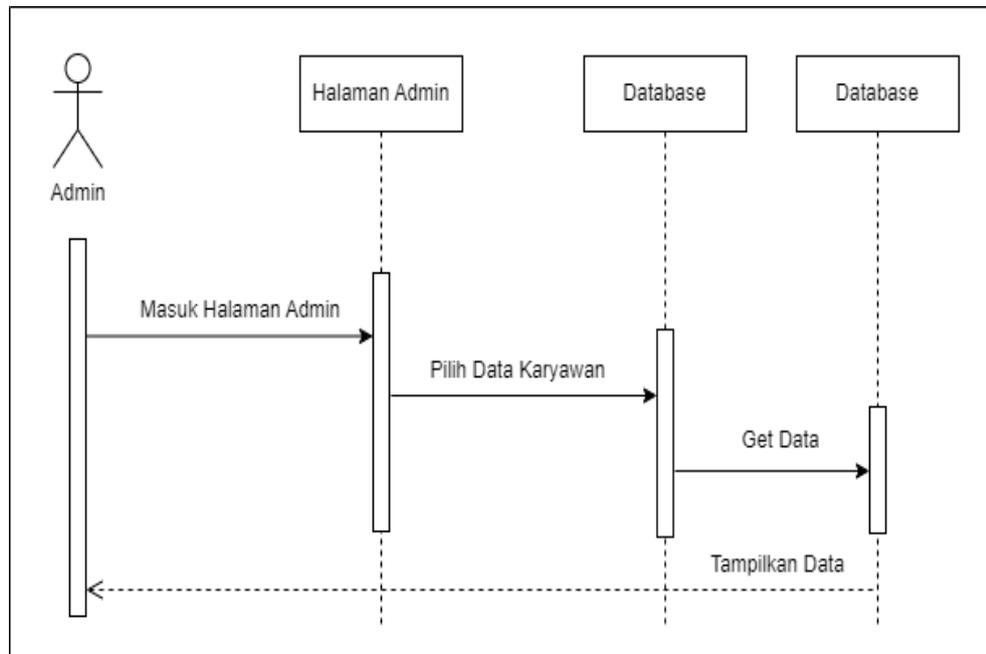
Gambar 4. 22 *Sequence Diagram Hapus Data*

5) *Sequence diagram lihat data user*



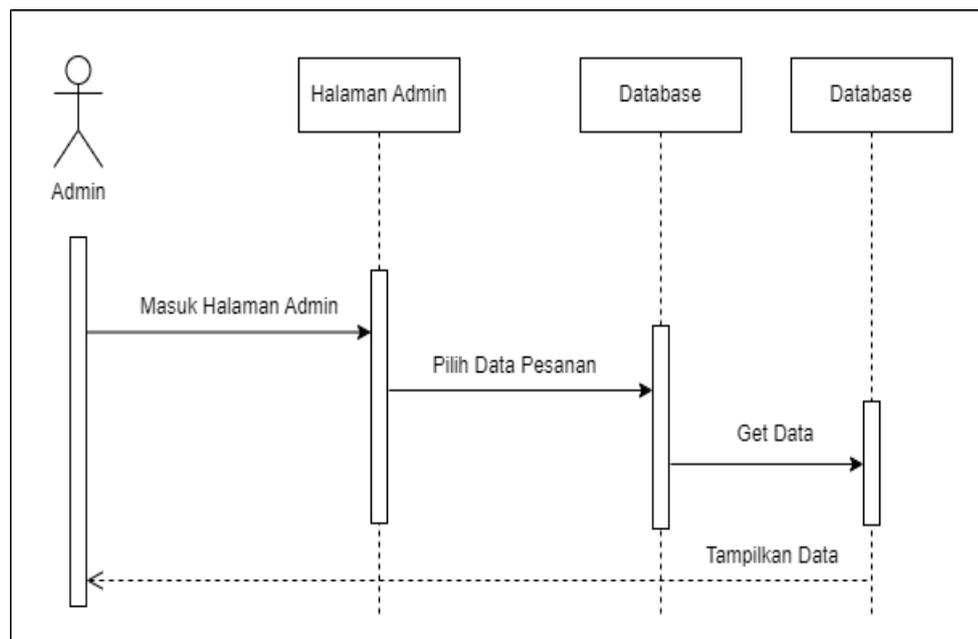
Gambar 4. 23 *Sequence Diagram Lihat Data User*

6) *Sequence diagram* lihat data karyawan



Gambar 4. 24 *Sequence Diagram* Lihat Data Karyawan

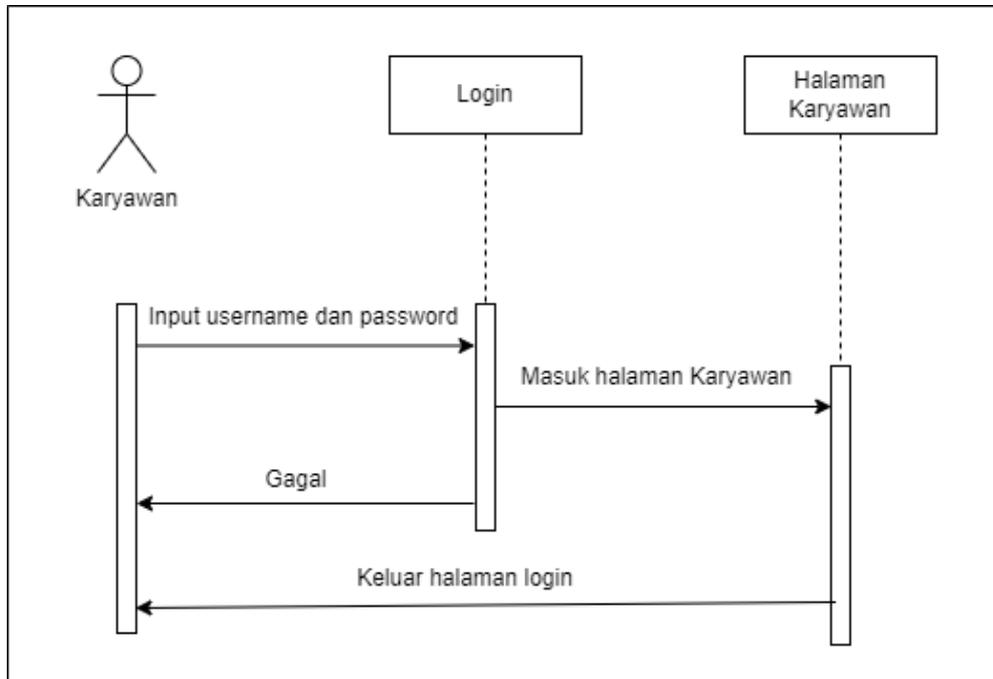
7) *Sequence diagram* lihat data pesanan



Gambar 4. 25 *Sequence Diagram* Lihat Data Pesanan

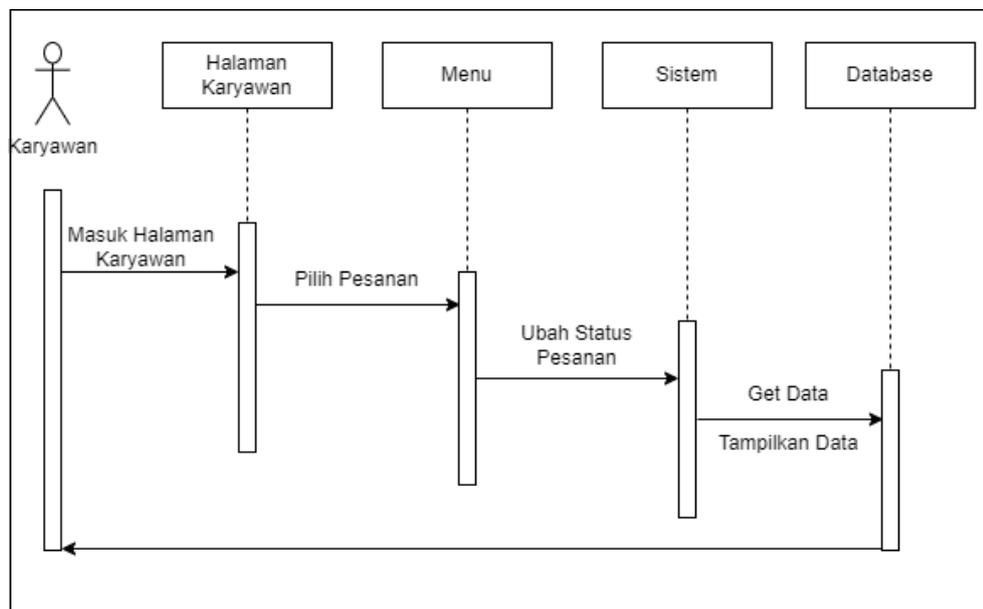
c. *Sequence diagram karyawan*

1) *Sequence diagram login karyawan*



Gambar 4. 26 *Sequence Diagram Login Karyawan*

2) *Sequence iagram konfirmasi status pengantaran*



Gambar 4. 27 *Sequence Diagram Konfirmasi Status Pengantaran*

C. Implementasi *Interface*

1. Admin

a. Tampilan *Splash Screen*

Merupakan layar pembukaan yang muncul ketika aplikasi pertama kali dijalankan yang terdiri dari logo aplikasi, nama dan sebagainya.

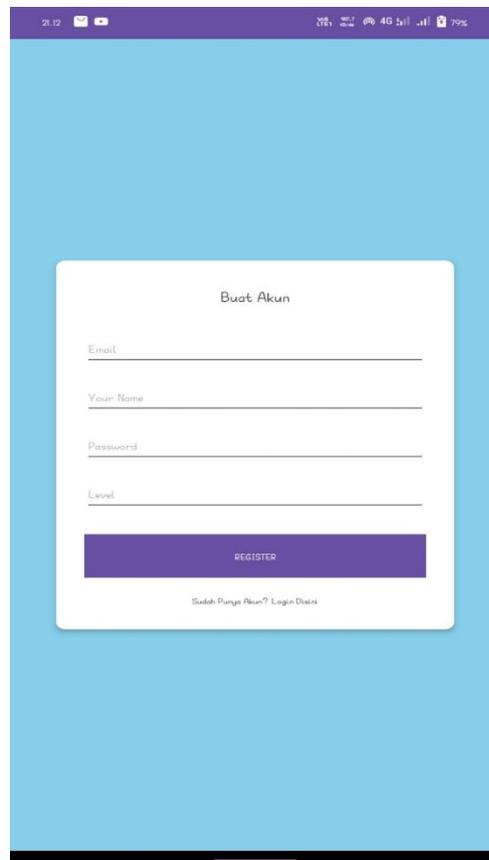


Gambar 4. 28 Halaman *Splash Screen*

Untuk menyusun tampilan pada gambar di atas, menggunakan *RelativeLayout* yang di dalamnya terdapat elemen *source code* android: layout_width="match_parent" untuk memanjangkan lebar penuh dari parent, android: layout_height="match_parent" untuk memanjangkan tinggi penuh dari *parent*. Atribut android: background="@drawable/splash" digunakan untuk mengatur gambar atau *drawable* sebagai latar belakang (*background*).

b. Halaman *Register*

Merupakan halaman awal yang digunakan admin untuk membuat akun agar bisa login ke halaman utama.



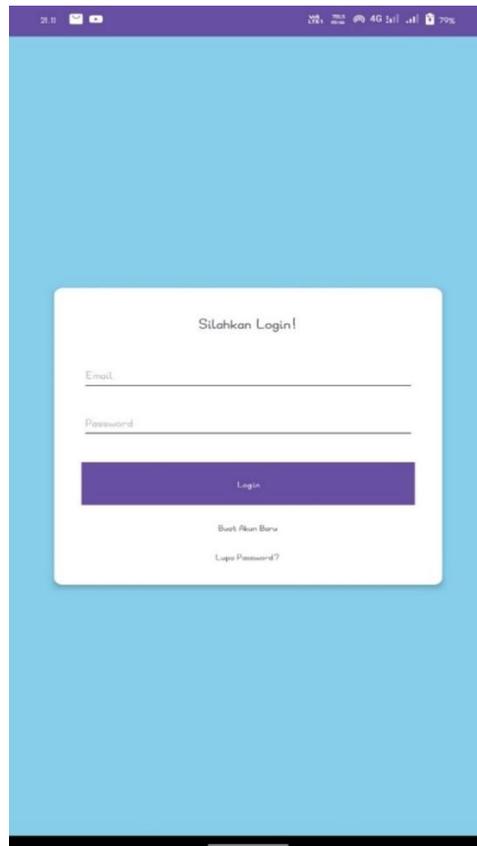
Gambar 4. 29 Halaman *Registrasi Admin*

Pada gambar diatas, dalam membuat tampilan UInya terdiri dari atribut *LinearLayout* yang digunakan untuk menata elemen-elemen seperti *TextView*, *EditText*, dan *Button*. Pada *TextView* “Sudah Punya Akun?Login Disini” dengan *source code* `android:textAlignment="center"` untuk mengatur posisi teks, `android:layout_gravity="center_horizontal"` `android:layout_marginTop="16dp"` untuk mengatur jarak antara elemen di dalam *LinearLayout*. Pada *TextView* terdapat elemen *EditText* untuk memungkinkan mengetik teks seperti mengisi

email, nama, *password*, dan level. Kemudian untuk “*Registrasi*” menggunakan elemem button yang didalamnya terdapat source code android:id="@+id/btnRegister" sebagai ID, android:layout_width="match_parent" android:layout_height="wrap_content" untuk mengatur lebar dan tinggi elemen button, android:backgroundTint="@color/buttonColor" untuk mengatur warna *background* button, android:textColor="@android:color/white" untuk mengatur warna teks”*Register*”.

c. Halaman *Login*

Merupakan halaman kedua *admin* untuk mendapatkan akses masuk ke halaman utama.

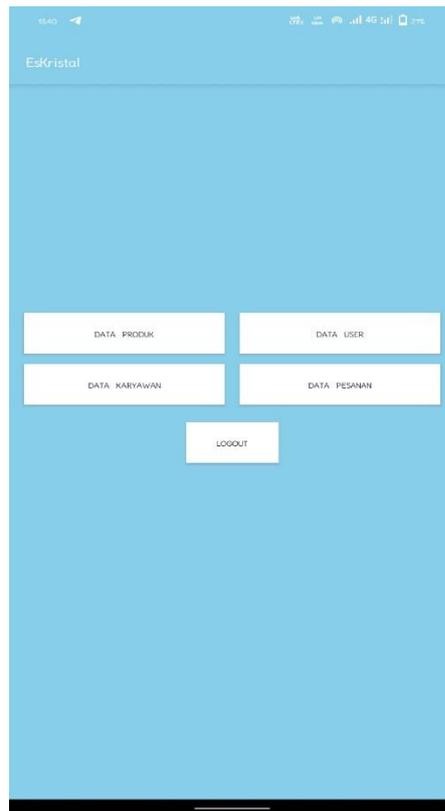


Gambar 4. 30 Halaman *Login Admin*

Pada gambar di atas, dalam membuat tampilan UInya terdiri dari atribut *LinearLayout* yang didalamnya terdapat atribut, *TextView*, *EditText*, dan *Button*. Untuk mengatur *TextView* “SilahkanLogin, BuatAkunBaru, LupaPassword” dengan *source code* `<TextView android:id="@+id/tv" untuklogin, lupa password, dan buat akun. Android: layout_width="wrap_content" android:layout_height="wrap_content" untuk menentukan bagaimana tinggi elemen UI harus diatur, android:text="" untuk mengisi teks silahkan login, lupa password dan buat akun baru. Android: textSize="24sp" untuk mengatur ukuran teks, android:textAlignment="center", untuk mengatur posisi teks. Untuk mengatur EditText” Email, Password dengan source code android:id="@+id/et" sebagai id username dan password. Android:layout_width="match_parent" dan android:layout_height="wrap_content" untuk menentukan tinggi elemen, android:hint="" untuk menampilkan teks sementara yaitu email dan password, android:drawablePadding="16dp" untuk mengatur jarak antara teks”email dan password”, android:layout_marginBottom=""/mengatur jarak antara bagian bawah elemen dan batas bawah kontainer. Mengatur Button”Login”, android:textColor="@android:color/white" untuk mengatur warna teks, android:background="@color/buttonColor" untuk mengatur warna background.`

d. Halaman Menu utama

Pada halaman menu utama terdapat beberapa data yang dapat dilihat oleh admin yaitu, data produk, data *user*, data karyawan dan data pesanan.



Gambar 4. 31 Halaman Menu

Pada gambar di atas, terdiri dari 5 *button* yaitu data *user*, data produk, data karyawan, data pesanan dan *logout*. `android:id="@+id/btn"` untuk button data *user*, data produk, data karyawan, data pesanan dan *logout*, `android:layout_width="0dp"`, `android:layout_height="wrap_content"` untuk menentukan tinggi elemen, `android:layout_marginTop="268dp"` untuk mengatur bagian atas elemen dan batas atas kontainer, `android:text=""` untuk mengisi teks, `android:background="@color/putih"` untuk mengatur warna background button, `android:textColor="@color/hitam"` untuk mengatur warna teks dalam button, `app:layout_constraintEnd_toEndOf="parent"` untuk mengatur posisi kanan kiri *button*, `app:layout_constraintStart_toEndOf="@+id/btn"` data produk, data *user*, data karyawan, data pesanan dan *logout*, `app:la`

yout_constraintTop_toTopOf="parent" untuk mengatur posisi atas bawah *button*.

e. Halaman Daftar Produk

Merupakan halaman data produk yang telah diinput oleh admin.



Gambar 4. 32 Halaman Daftar Produk

Tampilan pada gambar di atas, terdapat *progresbar* yang terdiri dari 6 kontainer, *floating action button* dan *imageview*. Pada *progresbar* dengan *source code*, `android:id="@+id/progressbar"`, `android:layout_width="48dp"` dan `android:layout_height="48dp"` untuk mengatur lebar dan tinggi elemen, `app:layout_anchorGravity="center"` untuk mengatur posisi setiap kontainer. Untuk *imageview* dengan *source vode* `<ImageView android:id="@+id/img_item_photo",android:layout_width="0dp" dan android:layout_height="0dp"`

untuk mengatur ukuran gambar didalam kontainer, `android:layout_marginBottom="4dp"` untuk mengatur posisi elemen atas dengan kontainer atas.

f. Halaman Data User

Pada halaman ini semua data user yang telah membuat akun akan tampil.



Gambar 4. 33 Halaman Data *User*

Tampilan pada gambar di atas terdiri dari *progressbar* dan *floating action button* dalam *ProgressBar* terdapat teks nama, username, password dan level *user*. `Android:id="@+id/pb_data"`, `android:layout_width="wrap_content"` dan `android:layout_height="wrap_content"` untuk mengatur ukuran kontainer layout. Untuk *floating action button* dengan *source code* `android:id="@+id/fab_tambah"` untuk *floating action button*, `android:layout_width="wrap_content"` dan `android:layout_height="wrap_content"` untuk

mengatur ukuran *floating action button*, `android:layout_margin="32dp"` untuk mengatur ukuran pada *floating action button*.

g. Halaman Data Karyawan

Pada halaman ini menampilkan data admin dan data karyawan.



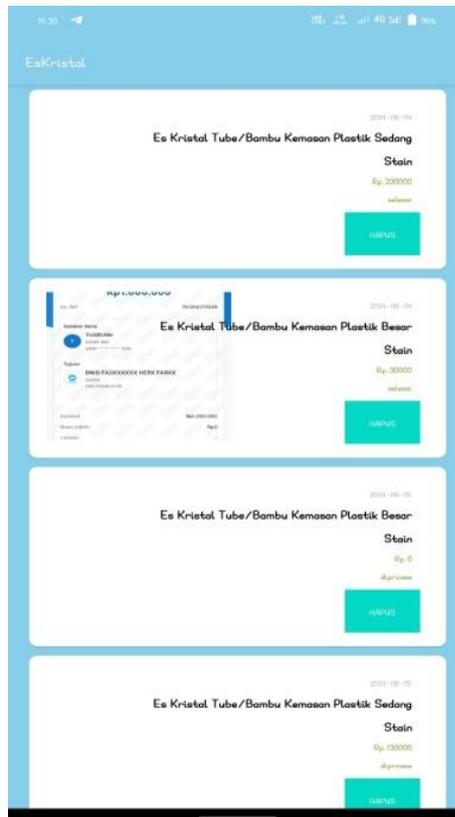
Gambar 4. 34 Halaman Data Karyawan

Tampilan pada gambar di atas terdiri dari *progressbar* dan *floating action button* dalam *ProgressBar* terdapat teks nama, *username*, *password* dan level *user*. `Android:id="@+id/pb_data"`, `android:layout_width="wrap_content"` dan `android:layout_height="wrap_content"` untuk mengatur ukuran kontainer layout. Untuk *floating action button* dengan *source code* `android:id="@+id/fab_tambah"` untuk *floating action button*, `android:layout_width="wrap_content"` dan `android:layout_height="wrap_content"` mengatur

ukuran *floating action button*, `android:layout_margin="32dp"` untuk mengatur ukuran pada *floating action button*.

h. Halaman Data Pesanan

Pada halaman ini akan tampil beberapa pesanan dari beberapa user.



Gambar 4. 35 Halaman Data Pesanan

Tampilan pada gambar di atas terdapat kontainer *layout* yang didalamnya terdapat atribut *TextView* dan *Button*. Untuk *Textview*nya terdapat *Textview* `android:id="@+id/"` item produk, item *user*, item tanggal, dan item proses. `Android:layout_width="0dp"` dan `android:layout_height="wrap_content"` untuk menentukan lebar elemen, `android:layout_marginLeft="16dp"` dan `android:layout_marginRight="16dp"`, `android:layout_marginTop="8dp"` untuk mengatur jarak antara sisi atas dari elemen UI dengan sisi atas elemen

kontainer, `android:textColor="@android:color/darker_gray"` untuk mengatur warna teks `android:textSize="12sp"` untuk mengatur ukuran teks `tools:text=""` untuk mengatur teks *user*, produk, tanggal, dan proses. Untuk `imageView` dengan `source code` `<ImageView android:id="@+id/img_item_photo",android:layout_width="0dp" dan android:layout_height="0dp"` untuk mengatur ukuran gambar didalam kontainer, `android:layout_marginBottom="4dp"` untuk mengatur posisi elemen atas dengan kontainer atas. Untuk `button` "hapus" dengan `source code` `<Button android:id="@+id/btn_delete"` sebagai id, `android:layout_width="70dp, android:layout_height="50dp"` untuk mengatur ukuran `button`, `android:text=""` untuk teks, `android:textColor="@android:color/white"` untuk mengatur `android:background="@color/colorAccent"` untuk mengatur warna *background button*.

2. *User*

a. *Halaman Register*

Merupakan halaman awal yang digunakan *user* untuk membuat akun sebelum *login* untuk masuk ke halaman utama.



Gambar 4. 36 Halaman *Registrasi User*

Pada gambar di atas, dalam membuat tampilan UInya terdiri dari atribut *LinearLayout* yang digunakan untuk menata elemen-elemen seperti *TextView*, *EditText*, dan *Button*. Pada *TextView* “Sudah Punya Akun? Login Disini” dengan *source code* `android:textAlignment="center"` untuk mengatur posisi text, `android:layout_gravity="center_horizontal"` `android:layout_marginTop="16dp"` untuk mengatur jarak antara elemen didalam *linearlayout*. Pada *TextView* terdapat elemen *EditText* untuk memungkinkan mengetik teks seperti mengisi email, nama, *password*, dan level. Kemudian untuk “*Registrasi*” menggunakan elemem button yang didalamnya terdapat *source code* `android:id="@+id/btnRegister"` sebagai ID, `android:layout_width="mat`

ch_parent" android: layout_height="wrap_content" untuk mengatur lebar dan tinggi elemen *button*, android: background="@color/buttonColor" untuk mengatur warna *background button*, android: textColor="@android: color/white" untuk mengatur warna teks”*Register*”.

b. Halaman Login

Merupakan halaman kedua setelah halaman *registrasi user* untuk mendapatkan akses masuk ke halaman utama.



Gambar 4. 37 Halaman *Login User*

Pada gambar di atas, dalam membuat tampilan UInya terdiri dari atribut *LinearLayout* yang didalamnya terdapat atribut, *TextView*, *EditText*, dan *Button*. Untuk mengatur *TextView* “*SilahkanLogin*, *BuatAkunBaru*, *LupaPas*

sword” dengan *source code* `<TextView android:id="@+id/tv" untuk login, lu`
pa *password*, dan buat akun. `Android: layout_width="wrap_content" android:`
`layout_height="wrap_content"` untuk menentukan bagaimana tinggi elemen
UI harus diatur, `android:text=""` untuk mengisi teks silahkan login, lupa
password dan buat akun baru. `Android: textSize="24sp"` untuk mengatur ukur
an teks, `android:textAlignment="center"`, untuk mengatur posisi teks. Untuk
mengatur `EditText`” *Email, Password* dengan *source code* `android:id="@+id"`
sebagai id username dan password. `Android:layout_width="match_parent" da`
n `android:layout_height="wrap_content"` untuk menentukan tinggi elemen,
`android:hint=""` untuk menampilkan teks sementara yaitu email dan
password, `android:drawablePadding="16dp"` untuk mengatur jarak anatar
teks”email dan password”, `android:layout_marginBottom=""/>` untuk
mengatur jarak antara bagian bawah elemen dan batas bawah kontainer.

c. Halaman Menu Utama

Merupakan halaman utama setelah *user* melakukan proses login yang
dimana pada halaman utama terdapat tampilan produk.



Gambar 4. 38 Halaman Menu Utama *User*

Tampilan pada gambar di atas, terdapat *progresbar* yang terdiri dari 3 kontainer, *button logout* dan riwayat pesanan. Pada *progresbar* dengan *source code*, `android:id="@+id/progressbar"`, `android:layout_width="48dp"` dan `android:layout_height="48dp"` untuk mengatur lebar dan tinggi elemen, `app:layout_anchorGravity="center"` untuk mengatur posisi setiap kontainer. Untuk button logout dan riwayat pesanan dengan *source code* `android:id="@+id/btnLeft"` untuk posisi *logout* dan `android:id="@+id/btnRight"` untuk posisi riwayat pesanan, `android:layout_width="150dp"`, `android:layout_height="wrap_content"` `android:text=""` untuk teks *logout* dan riwayat pesanan, `app:layout_constraintStart_toStartOf="parent"`, untuk

mengatur posisi elemen UI terhadap sisi awal (kiri), `app:layout_constraintTop_toTopOf="parent"` dan `app:layout_constraintBottom_toBottomOf="parent"` untuk mengatur posisi elemen kontainer terhadap sisi atas dan sisi bawah, `android:background="@color/buttonColor"` untuk mengatur warna *background button*, dan `android:textColor="@color/putih"/>` untuk mengatur warna teks yang terdapat dalam *button*.

d. Halaman Buat Pesanan

Merupakan halaman Dimana user mengisi beberapa data.



Gambar 4. 39 Halaman Buat Pesanan

Tampilan pada gambar di atas terdapat elemen `ImageView`, `EditText` dan `Button`. Pada elemen `ImageView` dengan source code `<ImageVi`

d:layout_width="match_parent" dan android:layout_height="wrap_content" untuk menentukan lebar dan tinggi pada gambar, android:scaleType="centerInside" untuk menentukan bagaimana gambar yang ditampilkan di *ImageView* akan diubah ukurannya dan ditempatkan dalam batas mengatur, android:id="@+id/imgHolderProduk"/> digunakan untuk memberikan identifikasi unik pada elemen UI, sehingga Anda dapat merujuknya dalam kode *Java*. Untuk elemen *EditText* dengan *source code* android:id="@+id/" untuk id edit produk, harga, jumlah, alamat, *user*, noHp, total, android:layout_width="match_parent" dan android:layout_height="wrap_content" untuk mengatur lebar dan tinggi elemen, android:hint="" untuk menampilkan teks, dan android:maxLines="1" /> untuk memastikan bahwa elemen hanya akan menampilkan satu baris teks. Untuk *button* dengan *source code* android:id="@+id/", android:layout_width="match_parent" dan android:layout_height="wrap_content" untuk mengatur lebar dan tinggi elemen, android:text="" untuk mengisi teks, android:textColor="@android:color/white" untuk mengatur warna teks, android:background="@color/buttonColor"/> untuk mengatur warna *background button*.

e. **Halaman Transaksi**

Merupakan halaman jika user ingin melakukan transaksi



Gambar 4. 40 Halaman Transaksi

Tampilan pada gambar di atas, terdapat elemen *RadioGroup* yang terdiri dari *RadioButton* dengan source code `<RadioGroup, android:id="@+id/radio_group_pembayaran", android:layout_width="match_parent" dan android:layout_height="wrap_content" untuk mengatur lebar dan tinggi elemen, android:layout_marginBottom="16dp">` untuk mengatur batas bawa dengan elemen kontainer bagian bawah. Untuk `<RadioButton` terdapat 2 teks yaitu *COD* dan *transfer*, `android:id="@+id/"` untuk id *COD* dan *transfer*, `android:layout_width="wrap_content" dan android:layout_heigh`

t="wrap_content" untuk mengatur lebar dan tinggi elemen, android:text="" />
untuk mengisi teks.

f. Halaman Riwayat Pesanan

Merupakan halaman yang berisi daftar pesanan *user*



Gambar 4. 41 Halaman Riwayat Pesanan

Tampilan pada gambar di atas terdapat kontainer *layout* yang didalamnya terdapat atribut *TextView* dan *Button*. Untuk *TextView*nya terdapat *TextView* android:id="@+id/" item produk, item *user*, item tanggal, dan item proses. Android:layout_width="0dp" dan android:layout_height="wrap_content" untuk menentukan lebar elemen, android:layout_marginLeft="16dp" dan android:layout_marginRight="16dp", android:layout_marginTop="8dp"

untuk mengatur jarak antara sisi atas dari elemen UI dengan sisi atas elemen kontainer, `android:textColor="@android:color/darker_gray"` untuk mengatur warna teks `android:textSize="12sp"` untuk mengatur ukuran teks `tools:text=""` untuk mengatur teks *user*, produk, tanggal, dan proses. Untuk *imageview* dengan *source code* `<ImageView android:id="@+id/img_item_photo",android:layout_width="0dp" dan android:layout_height="0dp"` untuk mengatur ukuran gambar didalam kontainer, `android:layout_marginBottom="4dp"` untuk mengatur posisi elemen atas dengan kontainer atas. Untuk *button* "hapus" dengan *source code* `<Button android:id="@+id/btn_delete"` sebagai id, `android:layout_width="70dp, android:layout_height="50dp"` untuk mengatur ukuran button, `android:text=""` untuk teks, `android:textColor="@android:color/white"` untuk mengatur warna teks, `android:backgroundTint="@color/color"` untuk mengatur warna *background button*.

3. Karyawan

a. Halaman Daftar Pesanan



Gambar 4. 42 Halaman Daftar Pesanan

Tampilan pada gambar di atas terdapat kontainer *layout* yang didalamnya terdapat atribut *TextView* dan *Button* Aksi. Untuk *TextView*nya terdapat `<TextView android:id="@+id/"` item produk, item *user*, item tanggal, dan item proses. `Android:layout_width="0dp"` dan `android:layout_height="wrap_content"` untuk menentukan lebar elemen, `android:layout_marginLeft="16dp"` dan `android:layout_marginRight="16dp"`, `android:layout_marginTop="8dp"` untuk mengatur jarak antara sisi atas dari elemen UI dengan sisi atas elemen kontainer, `android:textColor="@android:color/darker_gray"` untuk mengatur warna teks `android:textSize="12sp"` untuk mengatur ukuran teks `tools:text=""` untuk mengatur teks *user*, produk, tanggal, dan proses. Untuk *imageview* dan

gan *source code* <ImageView android:id="@+id/img_item_photo",android:layout_width="0dp" dan android:layout_height="0dp" untuk mengatur ukuran gambar didalam kontainer, android:layout_marginBottom="4dp" mengatur posisi elemen atas dengan kontainer atas. Untuk button"aksi" dengan *source code* <Button android:id="@+id/btn_action" sebagai id,android:layout_width="70dp dan android:layout_height="50dp" mengatur ukuran *button*, android:text="" untuk teks, android:textColor="@android:color/white" mengatur warna teks, android:background="@color/color" untuk mengatur warna *background button*.

D. Pengujian

Pengujian harus mencakup unit testing, yang mengecek validasi dari prosedur dan fungsifungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan. Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit *testing* yaitu *black box testing* dan *white box testing*

1. Black Box Testing

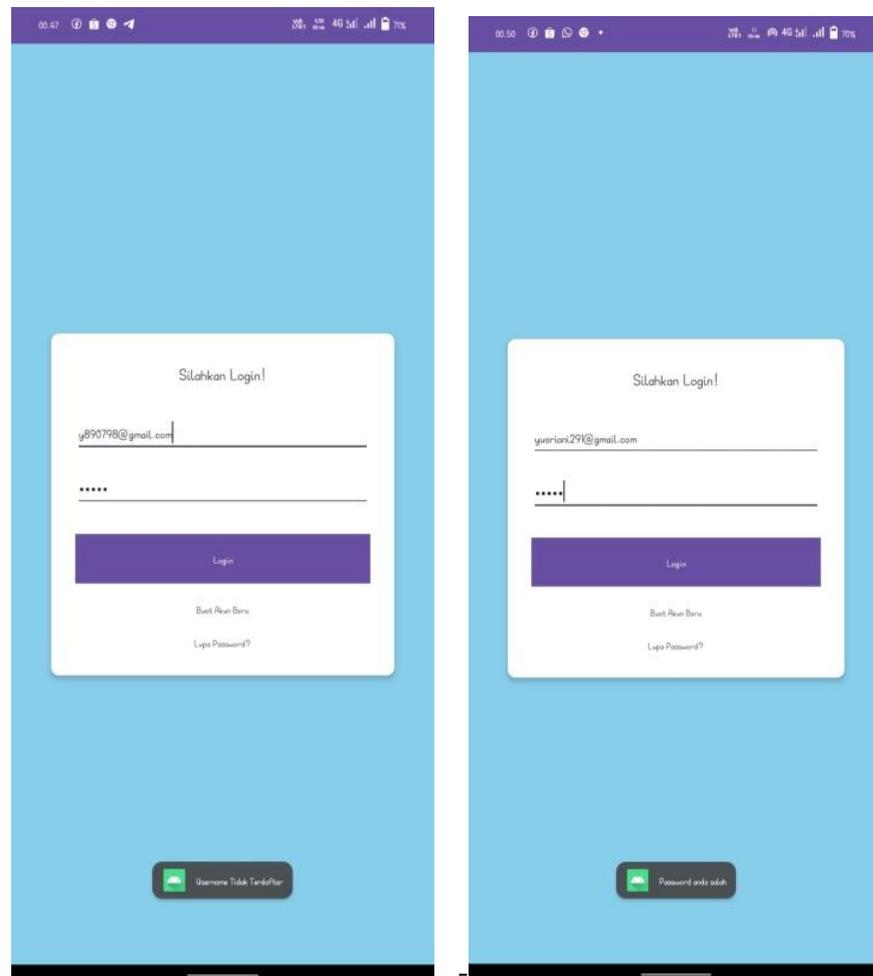
a. Admin

a) Black Box Testing Kesalahan Email dan Password

Tabel 4. 5 Black Box Testing Kesalahan Email dan Password

Tes Faktor	Hasil	Keterangan
Admin memasukkan email dan password yang tidak sesuai	✓	Berhasil, jika email dan password tidak sesuai maka sistem menampilkan pesan “username dan password tidak terdaftar”

Screenshot

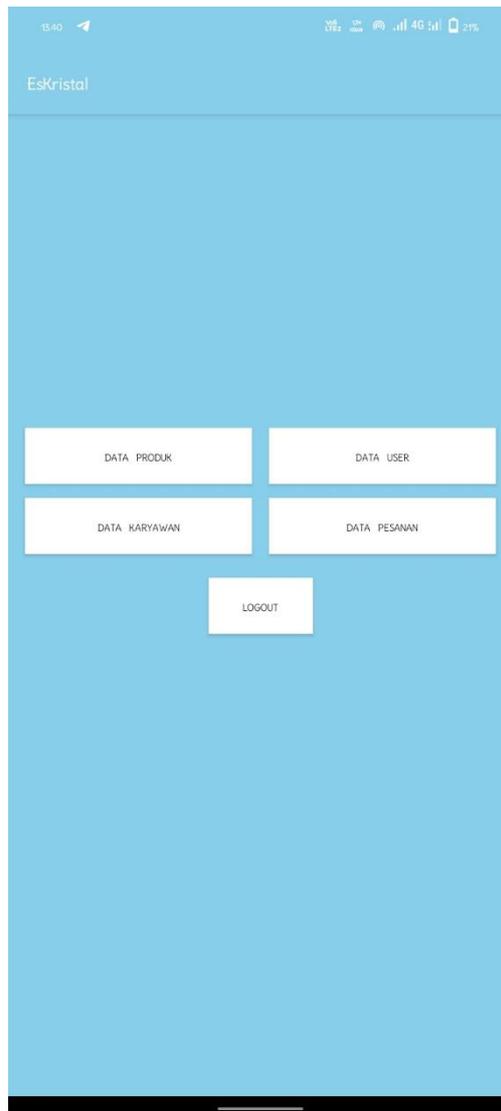


b) Black Box Testing Login Berhasil

Tabel 4. 6 Black Box Testing Login Berhasil

Tes Faktor	Hasil	Keterangan
<i>Admin</i> memasukkan <i>email</i> dan <i>password</i> yang sesuai	✓	Berhasil, sistem akan menampilkan halaman utama <i>admin</i>

Screenshot

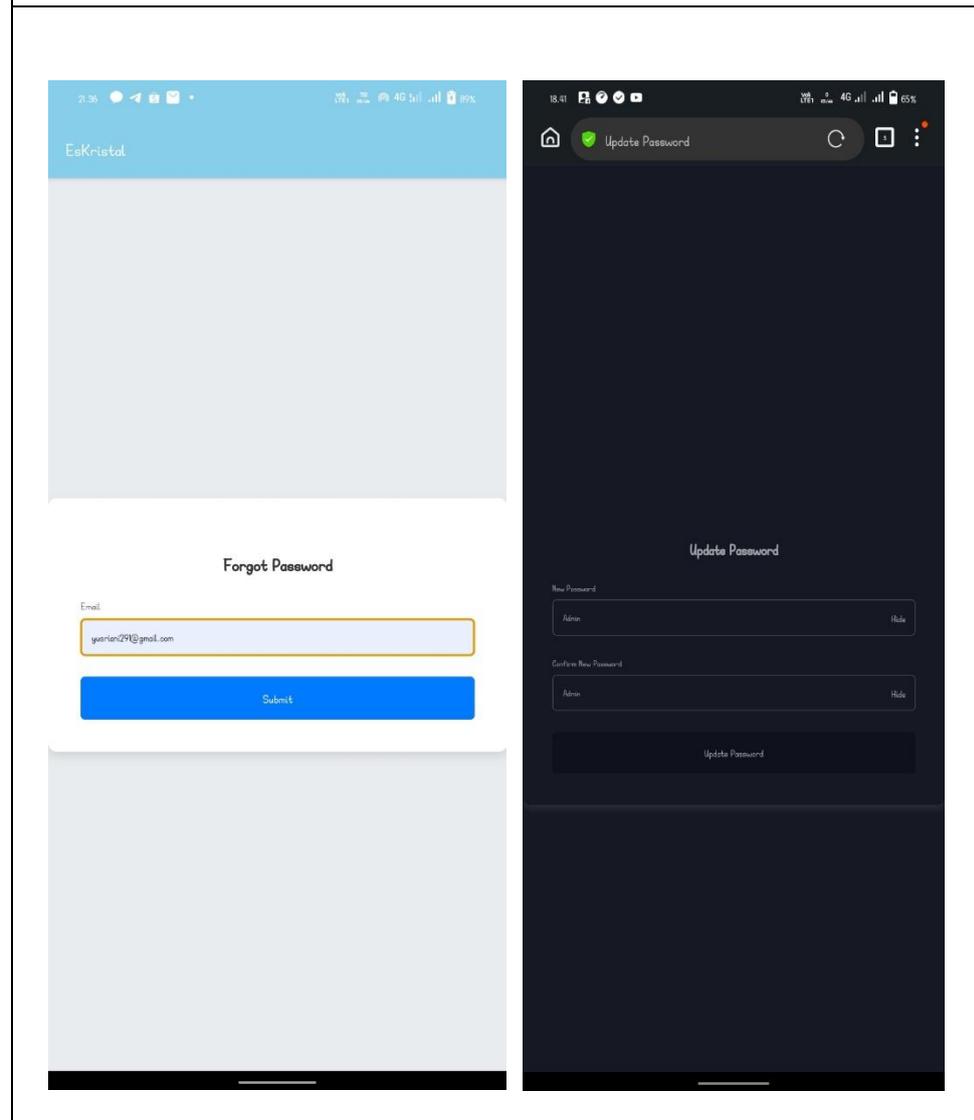


c) Black Box Testing Reset Password

Tabel 4. 7 Black Box Testing Reset Password

Tes Faktor	Hasil	Keterangan
Admin mengklik “lupa password”	✓	Berhasil, sistem akan menampilkan halaman <i>forgot password</i> dan mengupdate <i>password</i> dari pesan <i>email</i> .

Screenshot

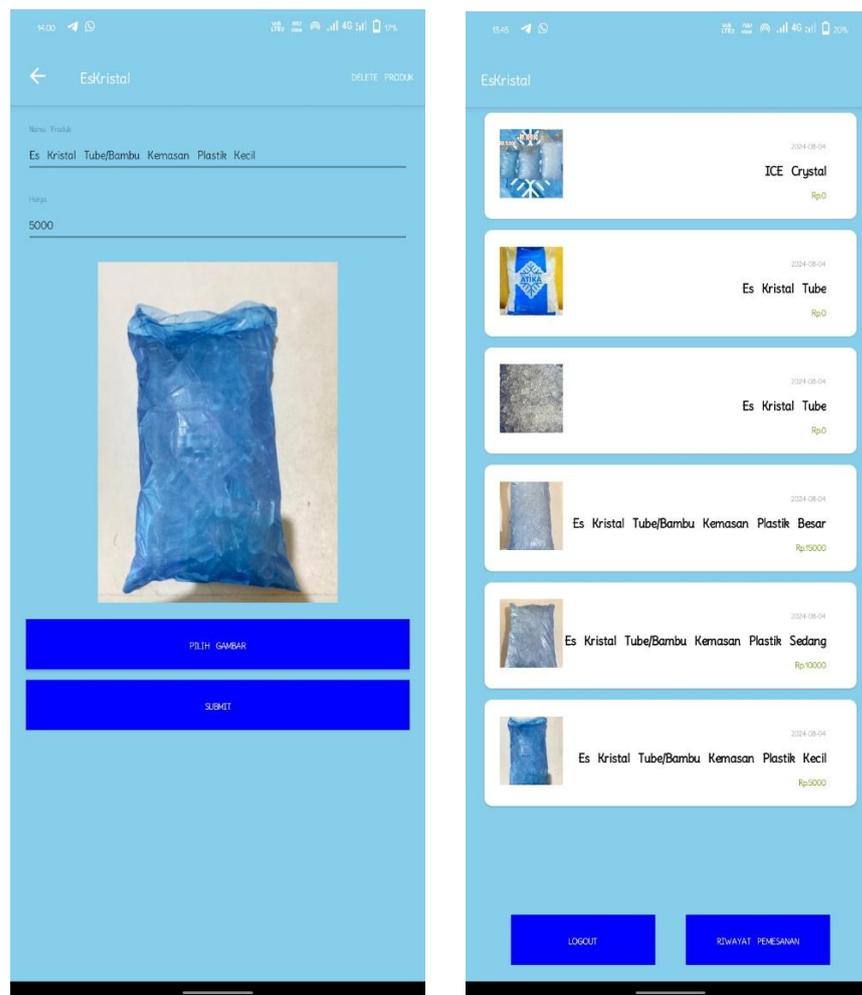


d) Black Box Testing Tambah Data Produk

Tabel 4. 8 Black Box Testing Tambah Data Produk

Tes Faktor	Hasil	Keterangan
Admin mengisi form data untuk menambah data produk dengan mengklik ikon tambah	✓	Berhasil, tampil data yang telah ditambahkan

Screenshot

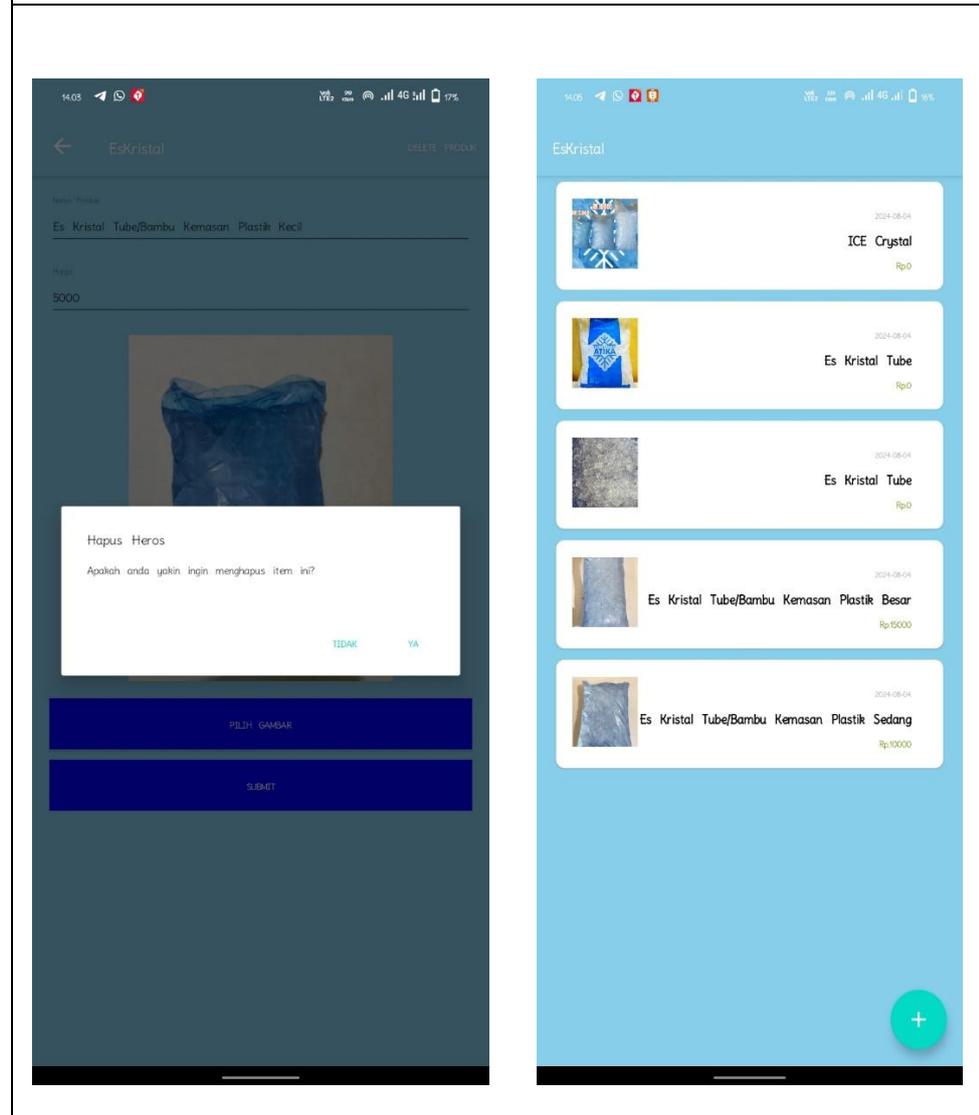


e) **Black Box Testing Hapus Data Produk**

Tabel 4. 9 *Black Box Testing Hapus Data Produk*

Tes Faktor	Hasil	Keterangan
Admin mengklik “DELETE”	✓	Berhasil, sistem tidak lagi menampilkan data produk.

Screenshot

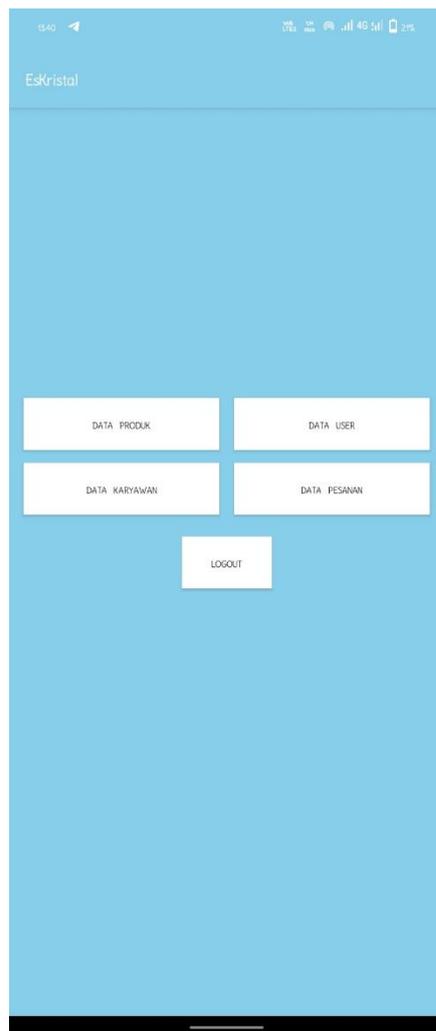


f) **Black Box Testing Melihat Data Produk**

Tabel 4. 10 *Black Box Testing* Melihat Data Produk

Tes Faktor	Hasil	Keterangan
Admin mengklik data produk	✓	Berhasil, Ketika <i>admin</i> mengklik data produk, sistem akan menampilkan data tersebut

Screenshot

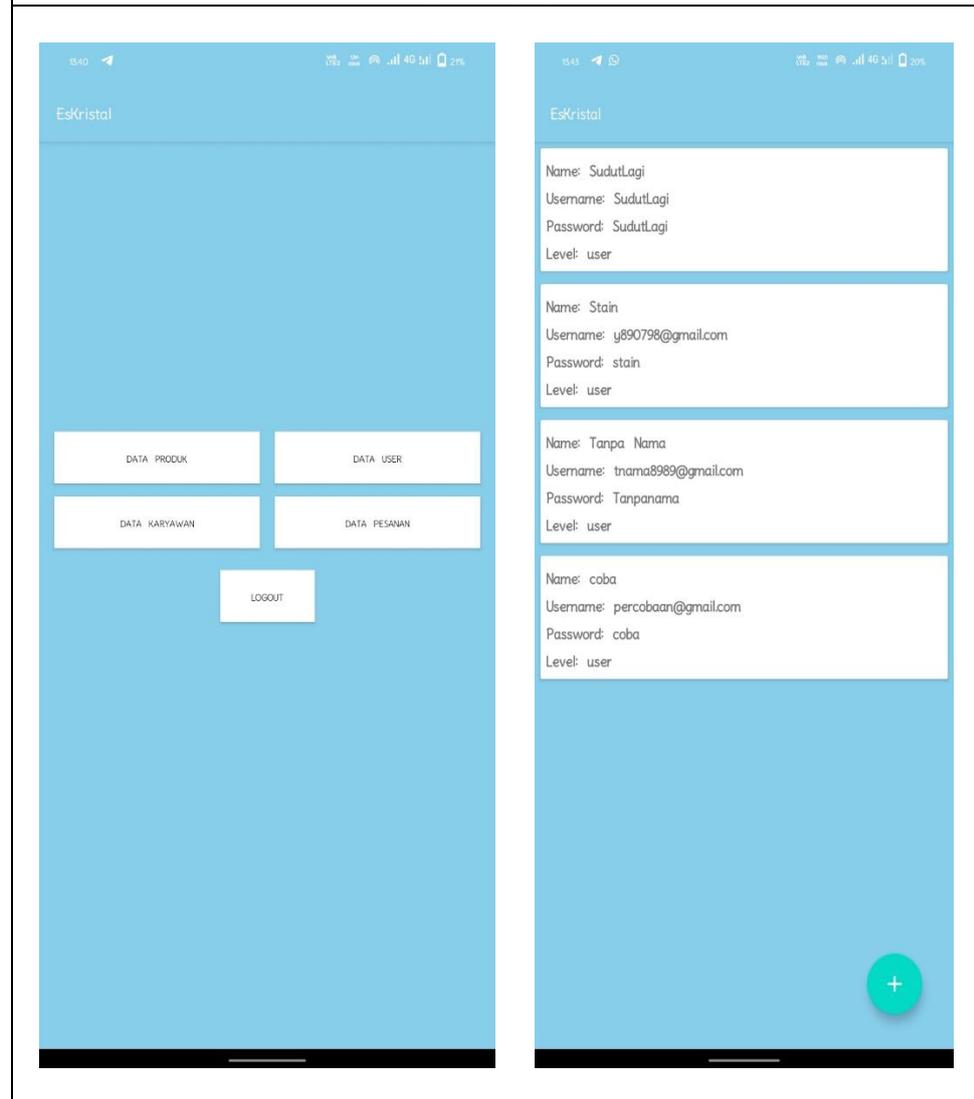


g) Black Box Testing Melihat Data User

Tabel 4. 11 Black Box Testing Melihat Data User

Tes Faktor	Hasil	Keterangan
<i>Admin</i> melihat data <i>user</i>	✓	Berhasil, Ketika <i>admin</i> mengklik data <i>user</i> , sistem akan menampilkan data tersebut

Screenshot

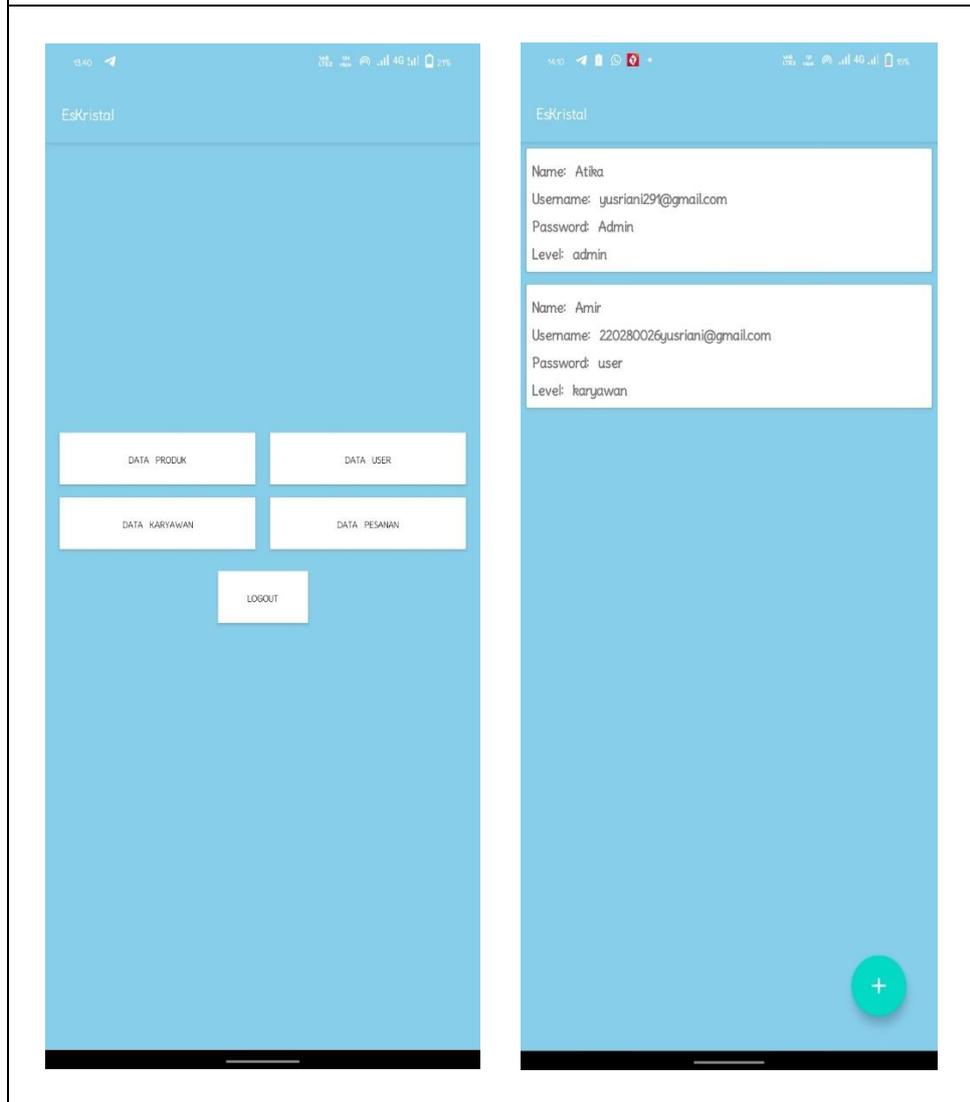


h) Black Box Testing Melihat Data Karyawan

Tabel 4. 12 Black Box Testing Melihat Data Karyawan

Tes Faktor	Hasil	Keterangan
Admin mengklik data karyawan	✓	Berhasil, Ketika admin mengklik data karyawan, sistem akan menampilkan data tersebut

Screenshot



i) Black Box Testing Melihat Data Pesanan

Tabel 4. 13 Black Box Testing Melihat Data Pesanan

Tes Faktor	Hasil	Keterangan
Admin mengklik data pesanan	✓	Berhasil, Ketika <i>admin</i> mengklik data pesanan, sistem akan menampilkan data tersebut

Screenshot

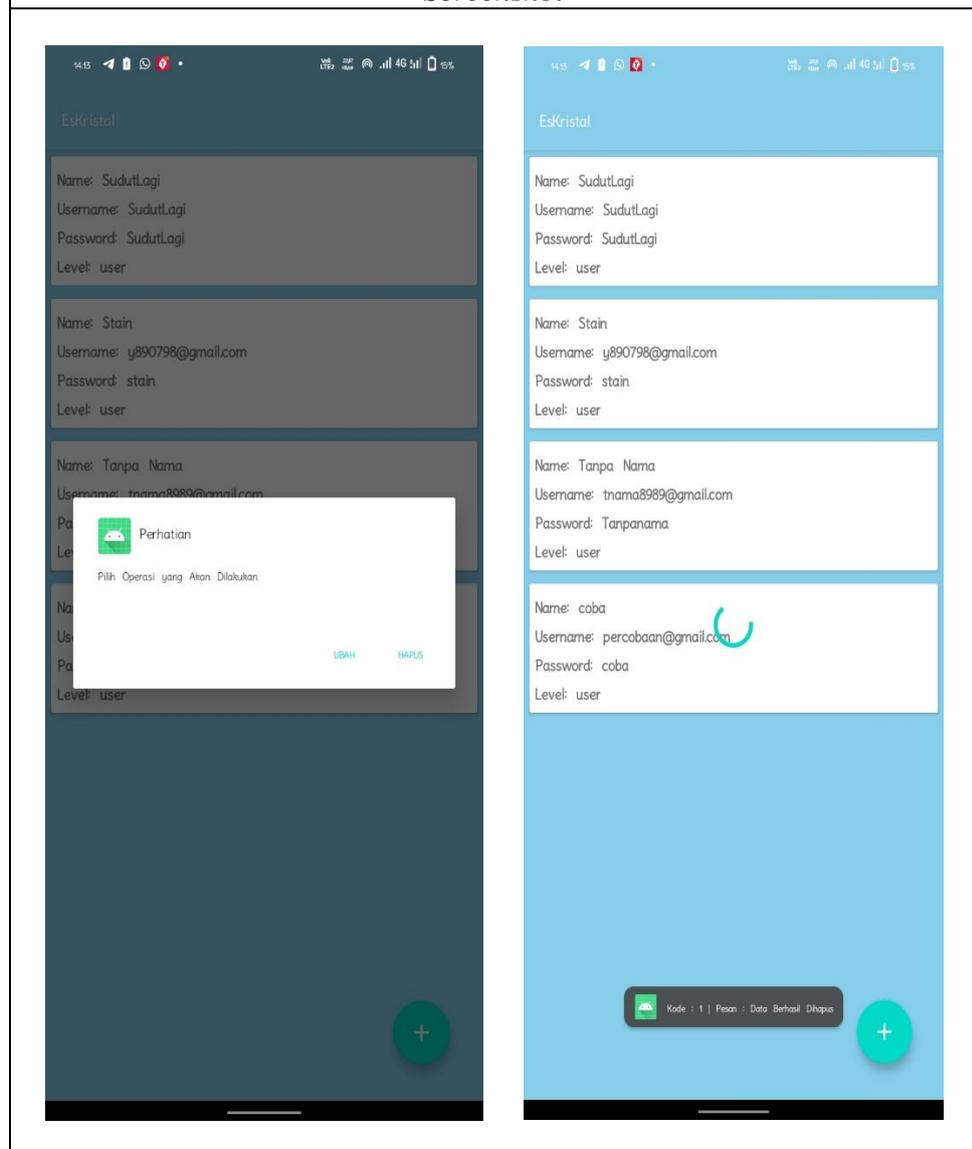


j) Black Box Testing Berhasil Menghapus Data User

Tabel 4. 14 Balck Box Testing Menghapus Data User

Tes Faktor	Hasil	Keterangan
Admin memilih “HAPUS”	✓	Berhasil, Ketika <i>admin</i> memilih “HAPUS” sistem akan menampilkan pesan “Data berhasil dihapus”

Screenshot

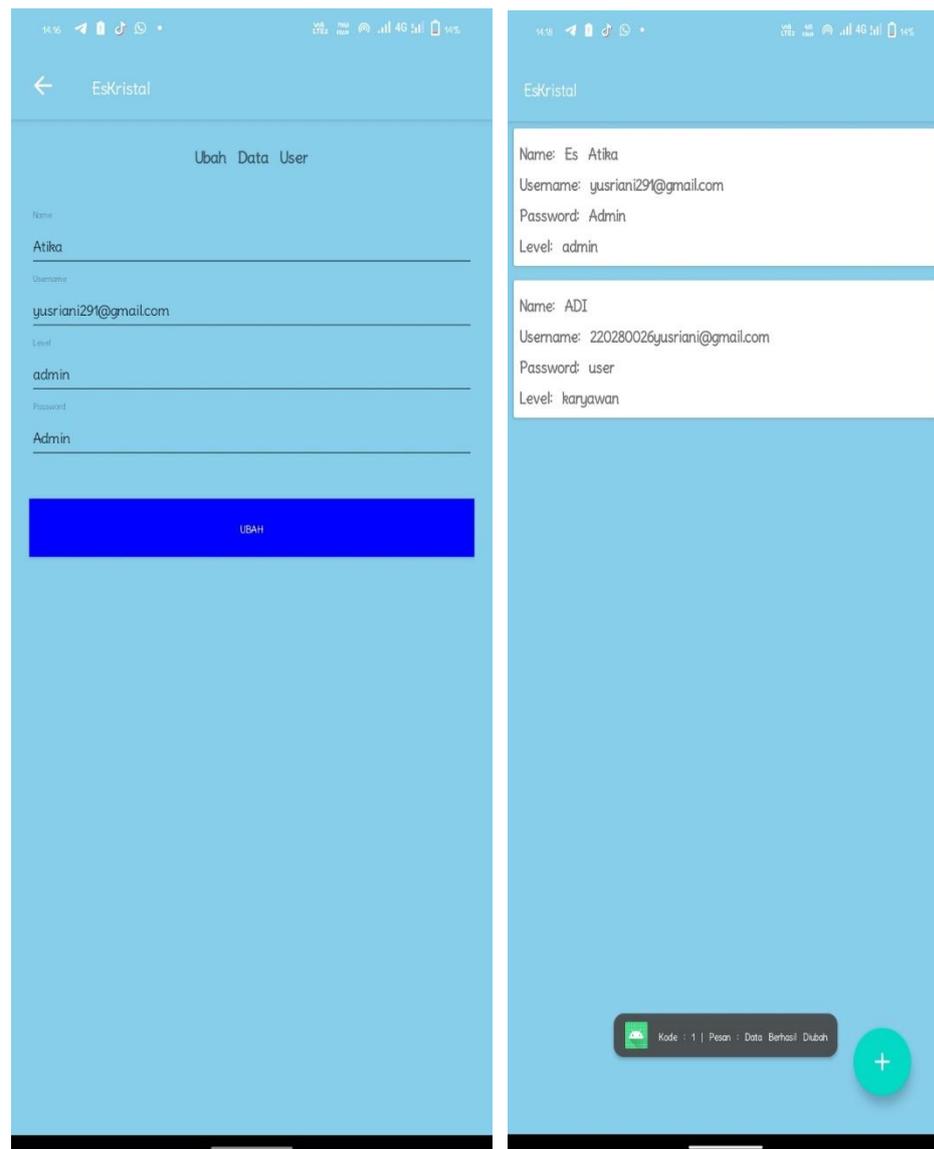


k) Black Box Testing Mengubah Data Admin

Tabel 4. 15 *Black Box Testing Mengubah Data Admin*

Tes Faktor	Hasil	Keterangan
Admin memilih “UBAH”	✓	Berhasil, Ketika <i>admin</i> memilih “UBAH” sistem akan menampilkan pesan “Data berhasil diubah”

Screenshot

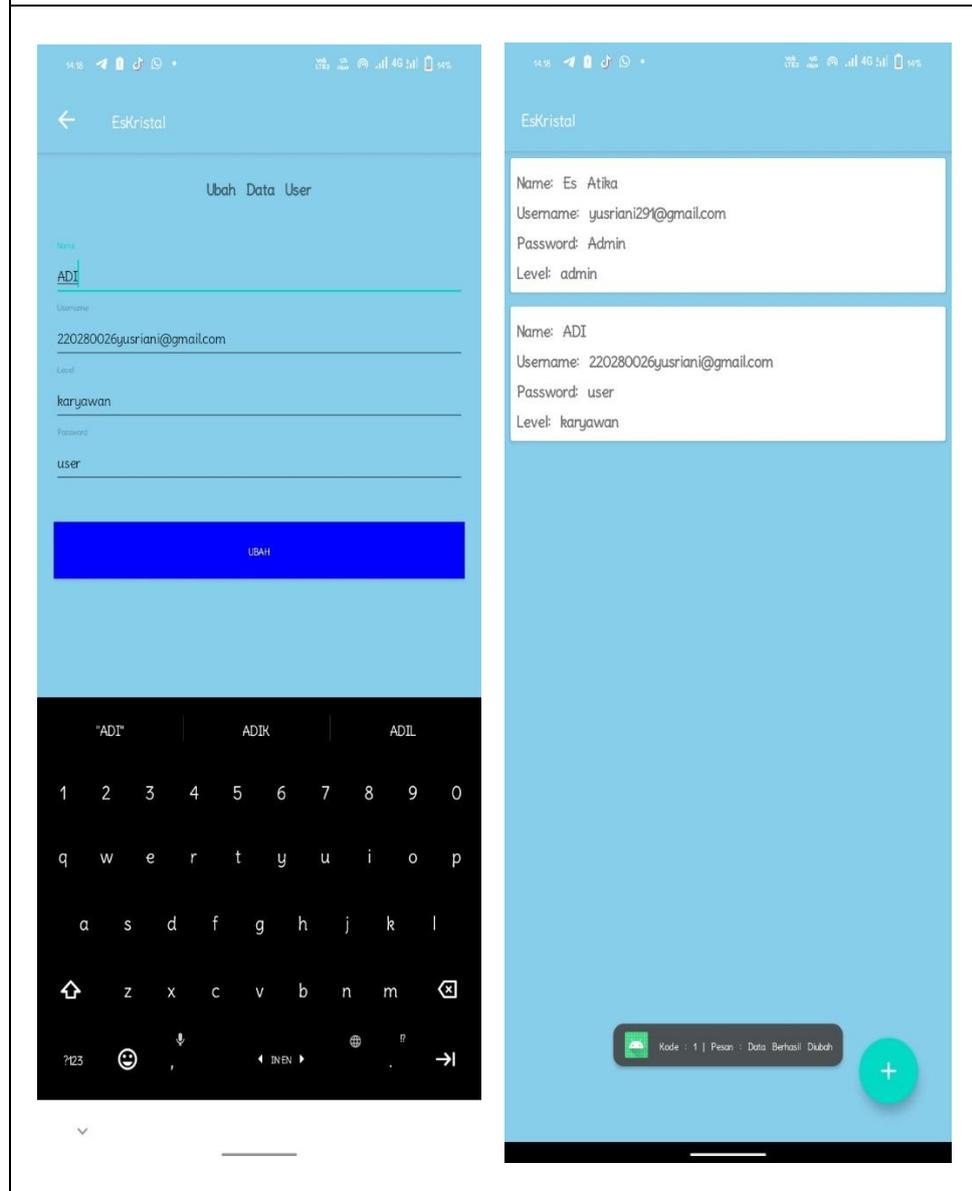


1) **Black Box Testing Menghapus Data Kayawan**

Tabel 4. 16 Black Box Testing Menghapus Data Karyawan

Tes Faktor	Hasil	Keterangan
Admin memilih “HAPUS”	✓	Berhasil, Ketika admin memilih “HAPUS” sistem akan menampilkan pesan “Data berhasil dihapus”

Screenshot

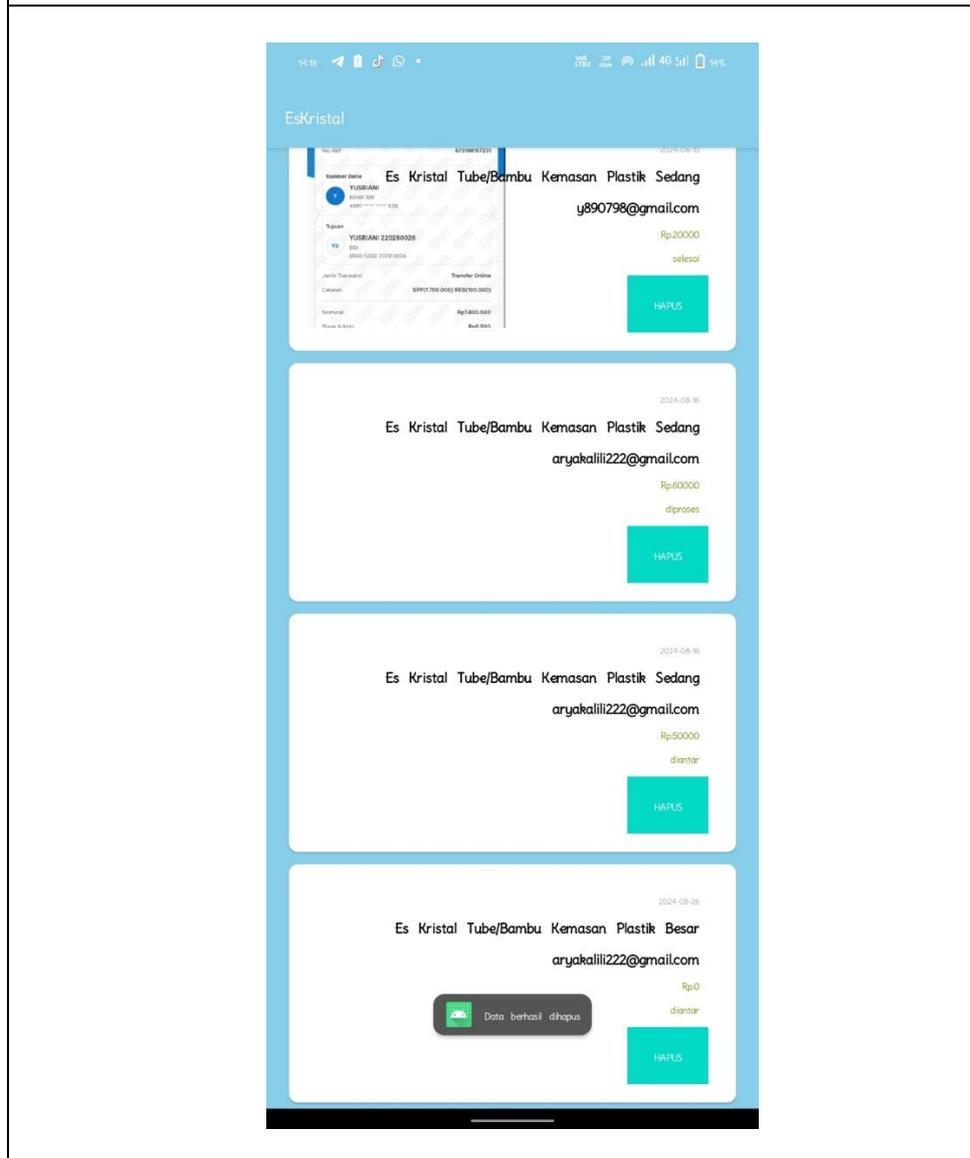


m) **Black Box Testing Menghapus Data Pesanan**

Tabel 4. 17 Black Box Testing Menghapus Data Pesanan

Tes Faktor	Hasil	Keterangan
Admin memilih “HAPUS”	✓	Berhasil, Ketika admin memilih “HAPUS” sistem akan menampilkan pesan “Data berhasil dihapus”

Screenshot



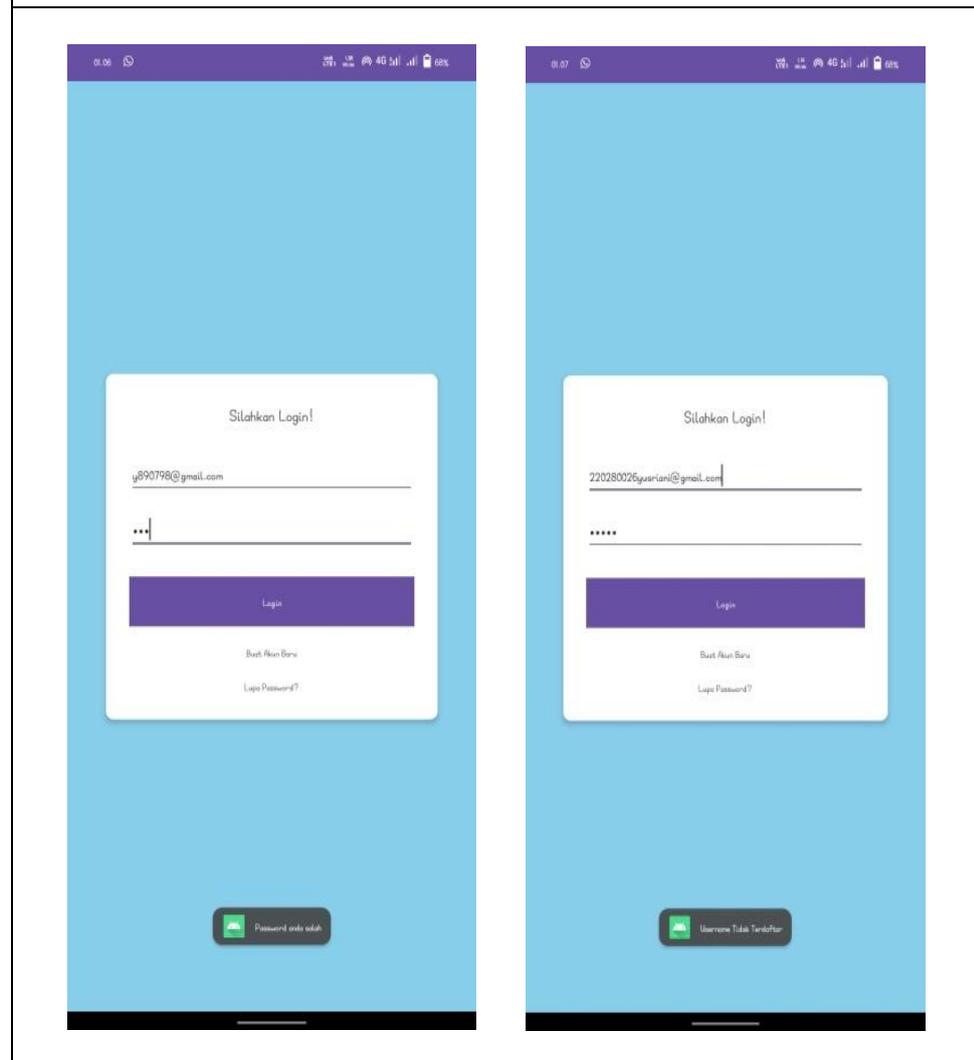
b. User

a) Black Box Testing Kesalahan Email dan Password

Tabel 4. 18 Black Box Testing Kesalahan Email dan Password

Tes Faktor	Hasil	Keterangan
Admin memasukkan <i>email</i> dan <i>password</i> yang tidak sesuai	✓	Berhasil, jika <i>email</i> dan <i>password</i> tidak sesuai maka tampil pesan <i>username</i> dan <i>password</i> tidak terdaftar

Screenshot



b) **Black Box Testing Login Berhasil**

Tabel 4. 19 *Black Box Testing Login Berhasil*

Tes Faktor	Hasil	Keterangan
<i>Admin</i> memasukkan <i>username</i> dan <i>password</i> yang sesuai	✓	Berhasil, sistem dapat menampilkan halaman utama <i>user</i>
<i>Screenshot</i>		
		

c) Black Box Testing Berhasil Melakukan Pesanan

Tabel 4. 20 Black Box Testing Buat Pesanan

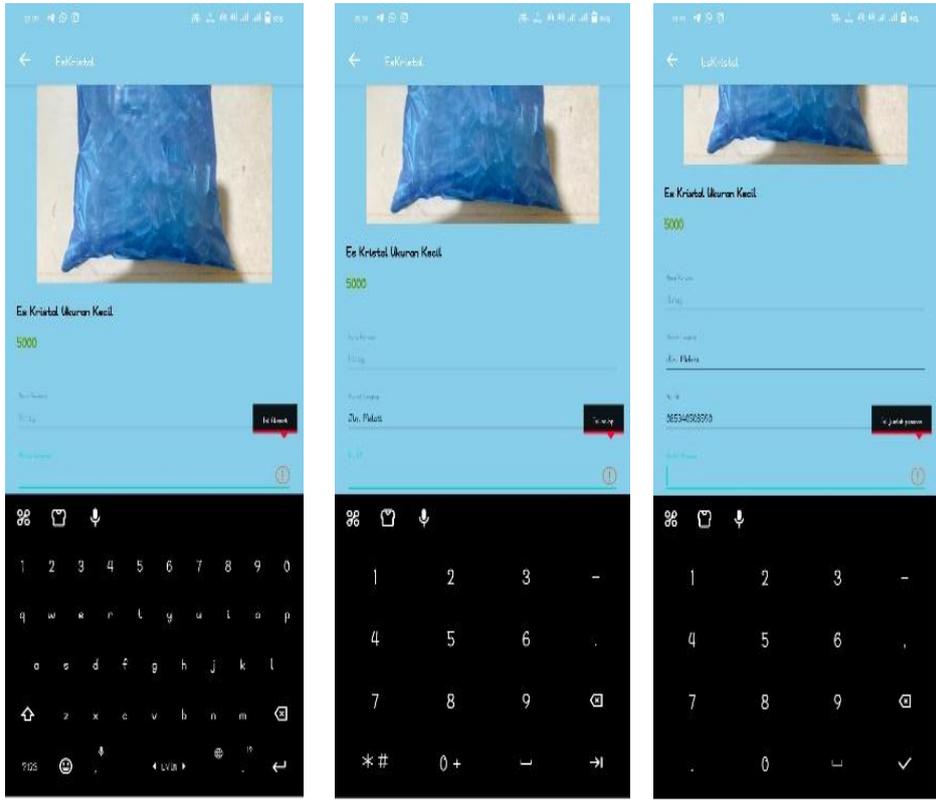
Tes Faktor	Hasil	Keterangan
Admin Mengisi semua Form pesanan	✓	Berhasil, sistem dapat menampilkan pesan “Berhasil Melakukan Pesanan”

Screenshot



d) Black Box Testing Tidak mengisi Alamat, NoHp, dan Jumlah Pesanan

Tabel 4. 21 Black Box Testing Tidak Mengisi Alamat, NoHp dan Jumlah Pesanan

Tes Faktor	Hasil	Keterangan
Admin tidak Mengisi Form alamat, no.hp dan jumlah pesanan	✓	Berhasil, sistem akan menampilkan pesan “Isi alamat”, “Isi No.Hp”, “Isi Jumlah Pesanan”
<i>Screenshot</i>		
		

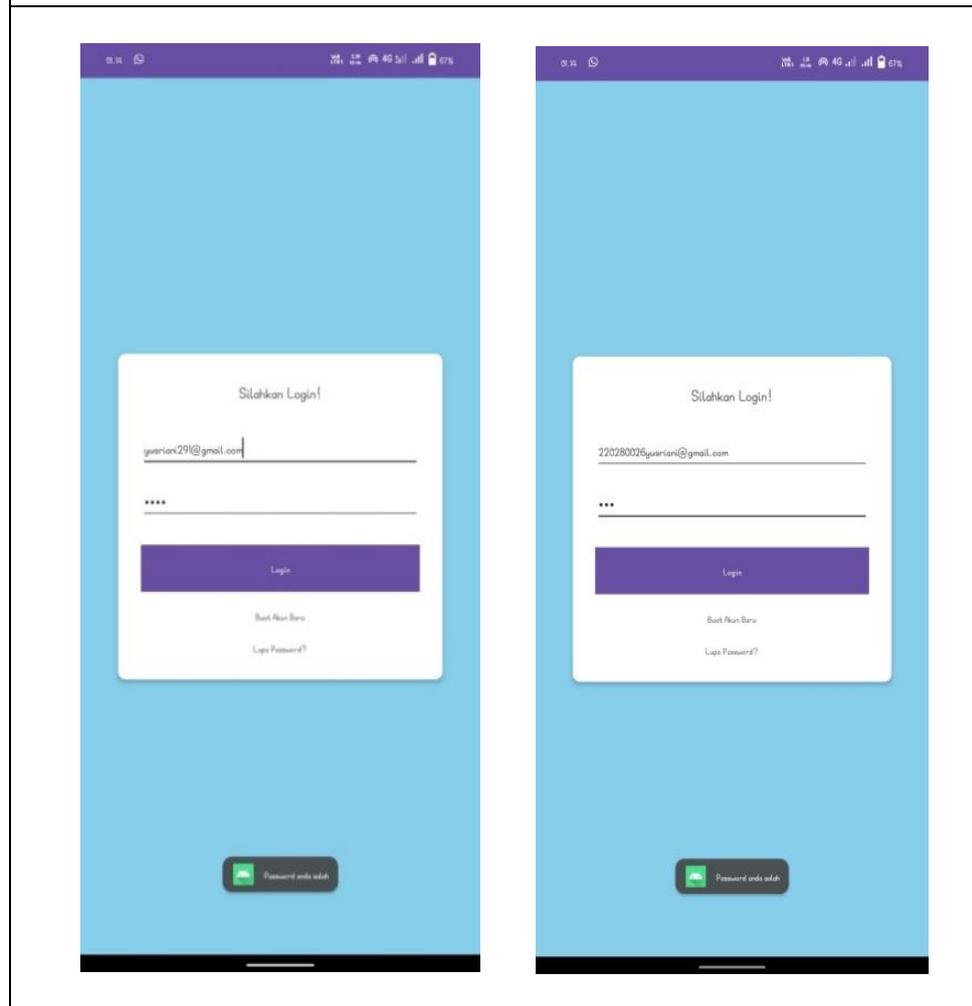
c. Karyawan

a) Black Box Testing Kesalahan Email dan Password

Tabel 4. 22 Black Box Testing Kesalahan Email dan Password Karyawan

Tes Faktor	Hasil	Keterangan
Karyawan memasukkan <i>email</i> dan <i>password</i> yang tidak sesuai	✓	Berhasil, sistem menampilkan pesan “ <i>username</i> tidak terdaftar” dan “ <i>password</i> salah”

Screenshot



b) **Black Box Testing Login Berhasil**

Tabel 4. 23 Black Box Testing Login Berhasil

Tes Faktor	Hasil	Keterangan
Admin memasukkan email dan password yang sesuai	✓	Berhasil, sistem dapat menampilkan halaman utama admin

Screenshot

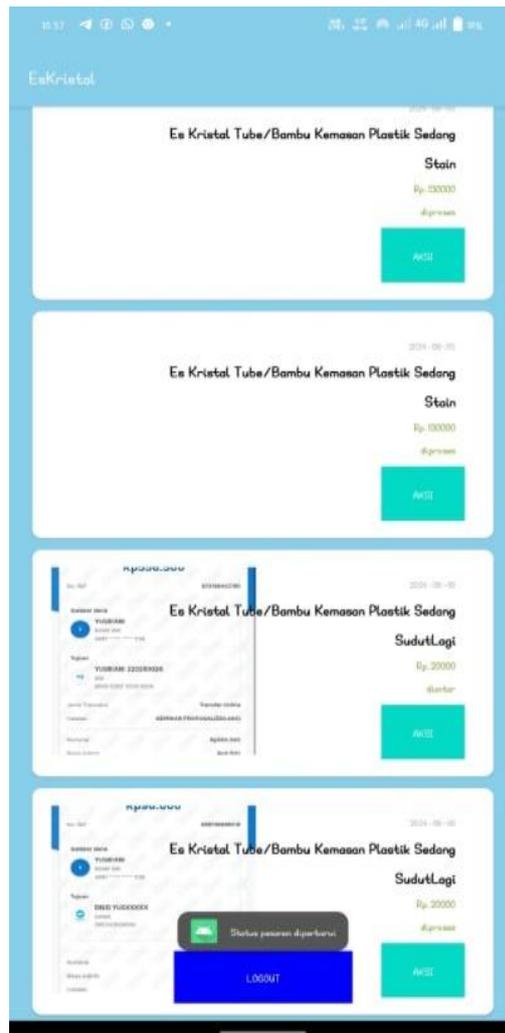


c) **Black Box Testing Berhasil Mengubah Status Pengantaran**

Tabel 4. 24 *Black Box Testing* Mengubah Status Pengantaran

Tes Faktor	Hasil	Keterangan
Admin mengubah status pengantaran	✓	Berhasil, sistem akan menampilkan pesan “Status Pesanan Diperbarui”

Screenshot

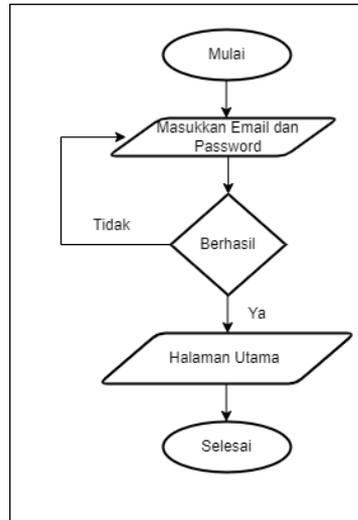


2. *White Box Testing*

a. Admin

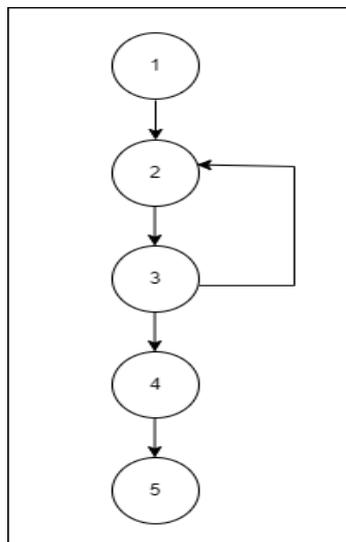
1) *White Box Testing Kesalahan Email dan Password*

a) *Flowchart*



Gambar 4. 43 *Flowchart Kesalahan Email dan Password*

b) *Flowgraph*



Gambar 4. 44 *Flowgraph Kesalahan Email dan Password*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat } \textit{node}) = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\textit{Path 1} = 1-2-3-2$$

$$\textit{Path 2} = 1-2-3-4-5$$

(4) Grafik matriks kesalahan *username* dan *password*

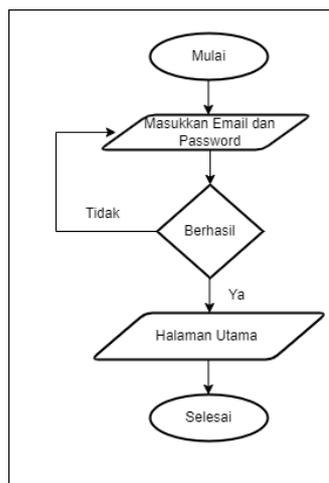
Tabel 4. 25 Grafik Matriks Kesalahan *Username* dan *Password*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$

4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$1 + 1 = 2$

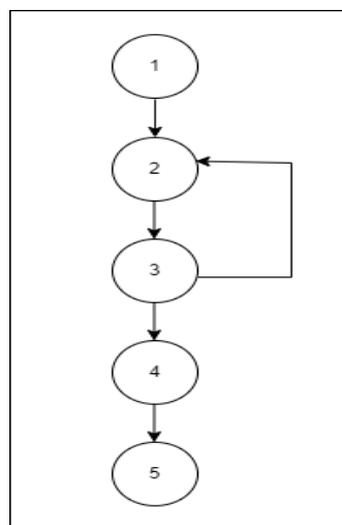
2) *White Box Testing Login Berhasil*

a) *Flowchart*



Gambar 4. 45 *Flowchart Login Berhasil*

b) *Flowgraph*



Gambar 4. 46 *Flowgraph Login Berhasil*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1-2-3-2$$

$$\text{Path 2} = 1-2-3-4-5$$

(4) Grafik matriks *login* berhasil

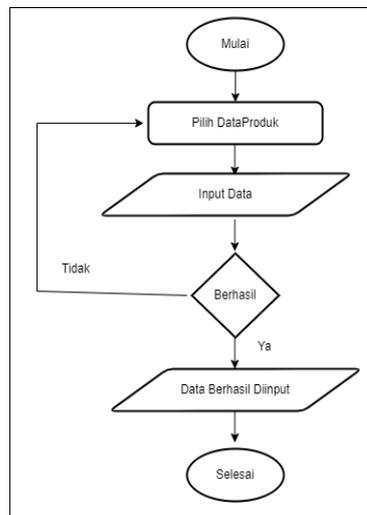
Tabel 4. 26 Grafik Matriks *Login* Berhasil

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$

2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$1 + 1 = 2$

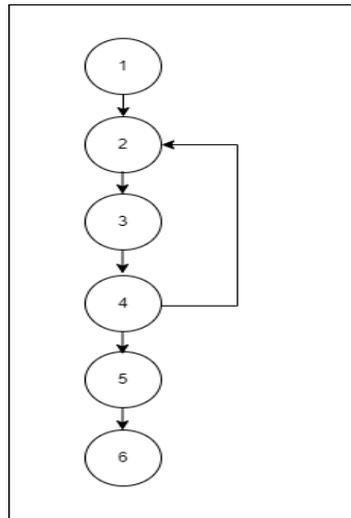
3) *White Box Testing Berhasil Tambah Data*

a. *Flowchart*



Gambar 4. 47 *Flowchart Berhasil Tambah Data*

b. *Flowgraph*



Gambar 4. 48 *Flowgraph* Berhasil Tambah Data

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-2

Path 2 = 1-2-3-4-5-6

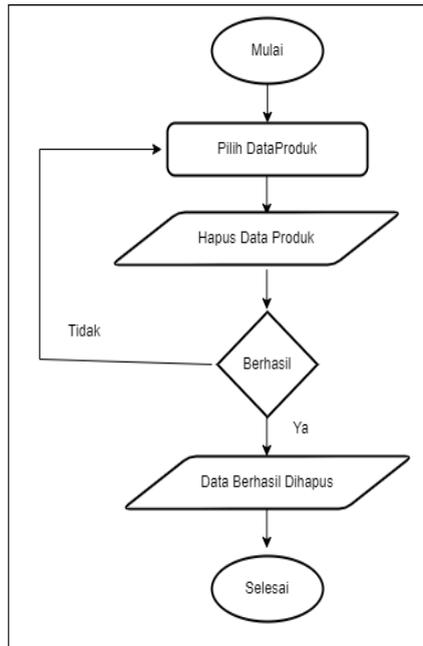
(4) Grafik matriks tambah data produk

Tabel 4. 27 Grafik Matriks Tambah Data

	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

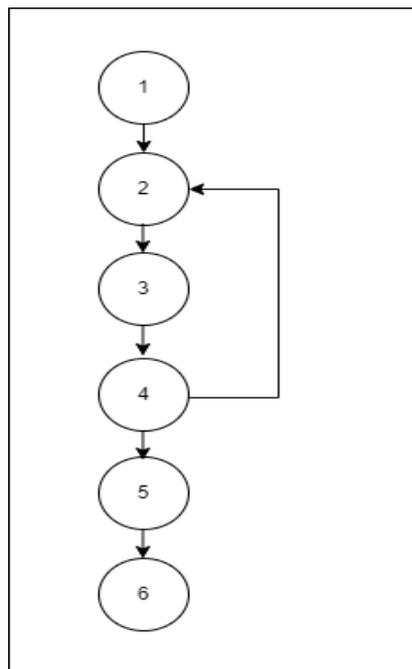
4) *White Box Testing* Hapus Data Produk

a. *Flowchart*



Gambar 4. 49 *Flowchart* Hapus Data

b. Flowgraph



Gambar 4. 50 *Flowgraph* Hapus Data

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 6

N (*node*) = 6

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 6 - 6 + 2$

$= 2$

$\text{Predikat (P)} = P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-3

Path 2 = 1-2-3-4-5-6

(4) Grafik matriks hapus data produk

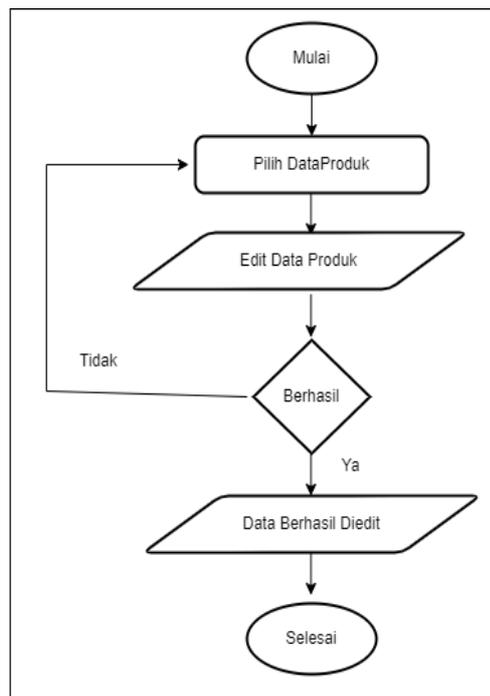
Tabel 4. 28 Grafik Matriks Hapus Data Produk

	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)							$1+1 = 2$

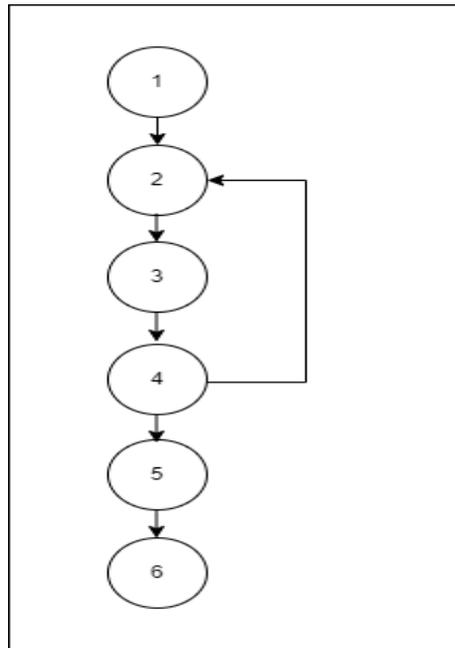
5) **White Box Testing Edit Data Produk**

a. **Flowchart**



Gambar 4. 51 *Flowchart* Edit Data

b. **Flowgraph**



Gambar 4. 52 *Flowgraph* Edit Data

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 6

N (*node*) = 6

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 6 - 6 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-3

Path 2 = 1-2-3-4-5-6

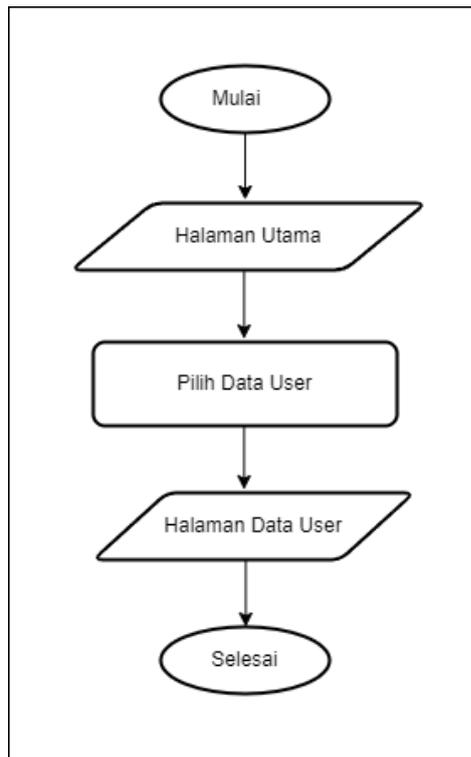
(4) Grafik matriks edit data produk

Tabel 4. 29 Grafik Matriks Edit Data Produk

	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

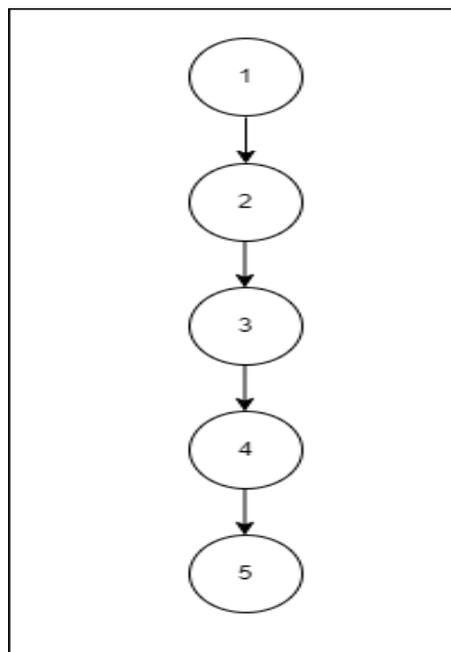
6) *White Box Testing* Lihat Data User

a. *Flowchart*



Gambar 4. 53 *Flowchart Lihat Data User*

b. Flowgraph



Gambar 4. 54 *Flowgraph Lihat Data User*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1-2-3-2$$

$$\text{Path 2} = 1-2-3-4-5$$

(4) Grafik matriks lihat data *user*

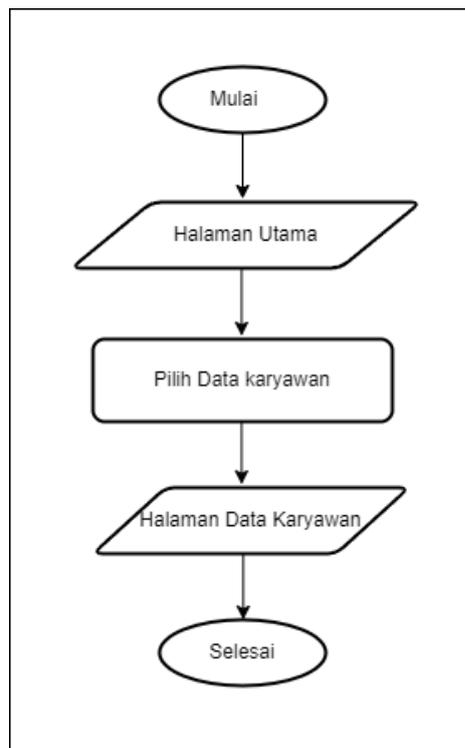
Tabel 4. 30 Grafik Matriks Lihat Data *User*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$

3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$0 + 1 = 1$

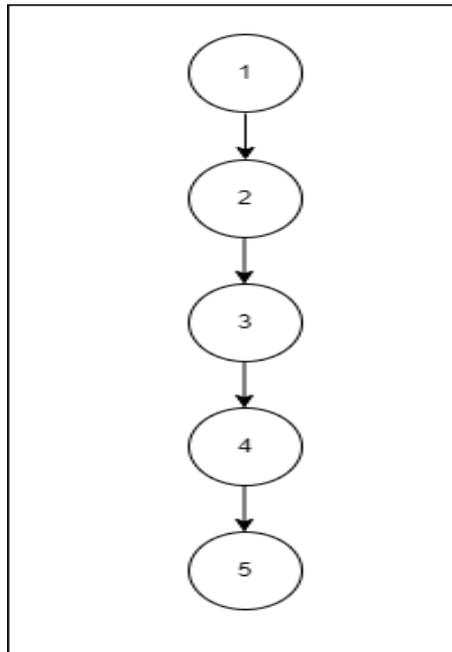
7) **White Box Testing Lihat Data Karyawan**

a. **Flowchart**



Gambar 4. 55 *Flowchart* Lihat Data Karyawan

b. **Flowgraph**



Gambar 4. 56 *Flowgraph* Lihat Data Karyawan

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian:

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 0 + 1$

$= 1$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-2

Path 2 = 1-2-3-4-5

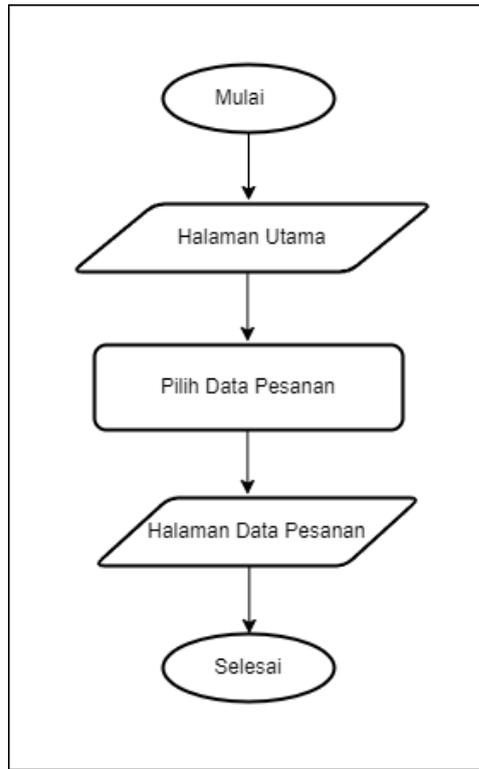
(4) Grafik matriks lihat data karyawan

Tabel 4. 31 Grafik Matriks Lihat Data Karyawan

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
SUM (E +1)						$0 + 1 = 1$

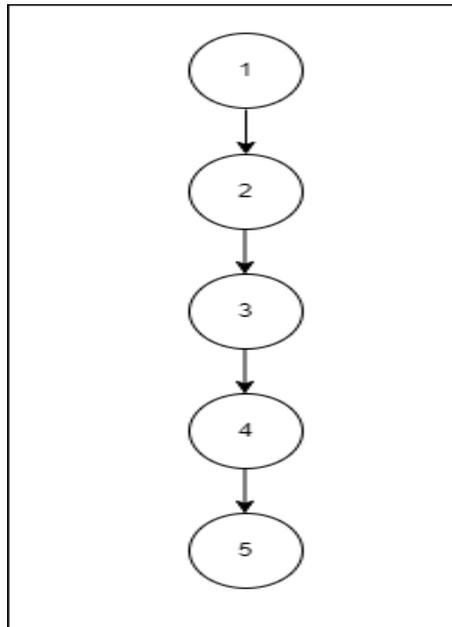
8) White Box Testing Lihat Data Pesanan

a. Flowchart



Gambar 4. 57 *Flowchart* Lihat Data Pesanan

b. Flowgraph



Gambar 4. 58 *Flowgraph* Lihat Data Pesanan

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian:

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

P (*Predikat (P)*) = $P + 1$

$= 0 + 1$

$= 1$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-2

Path 2 = 1-2-3-4-5

(4) Grafik matriks lihat data pesanan

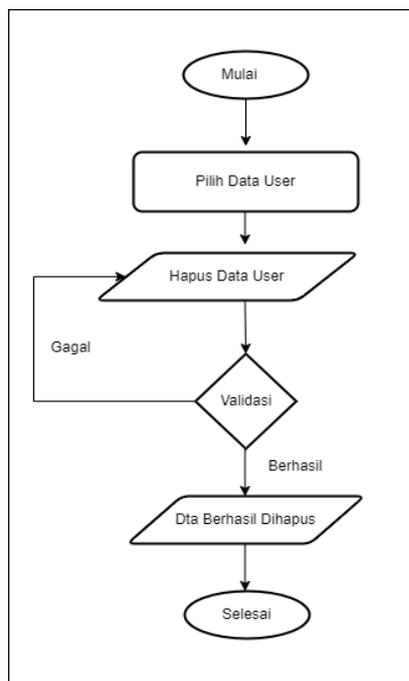
Tabel 4. 32 Grafik Matriks Lihat Data Pesanan

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$

3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$0 + 1 = 1$

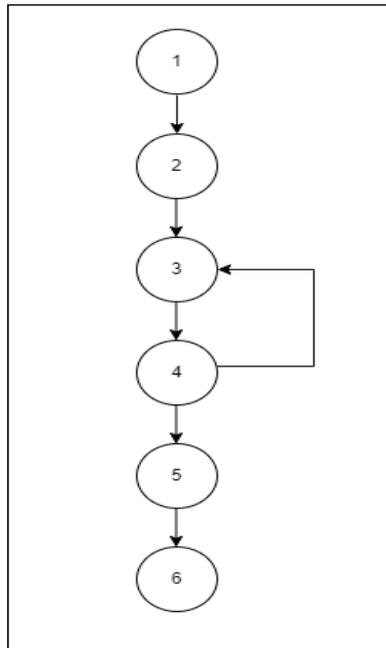
9) *White Box Testing Hapus Data User*

a. *Flowchart*



Gambar 4. 59 *Flowchart Hapus Data User*

b. *Flowgraph*



Gambar 4 60 *Flowgraph* Hapus Data *User*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-3

Path 2 = 1-2-3-4-5-6

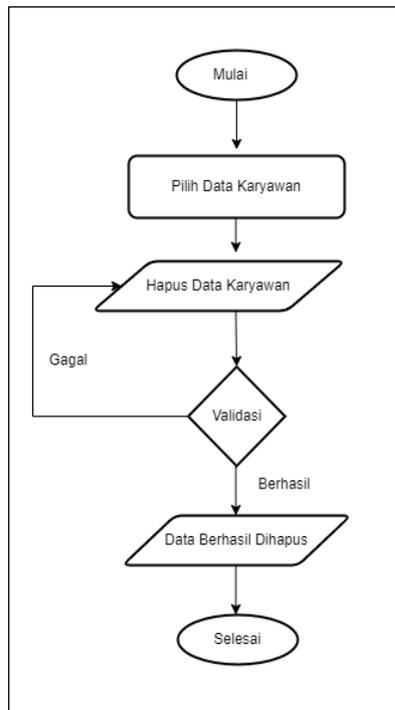
(4) Grafik matriks hapus data *user*

Tabel 4. 33 Grafik Matriks Hapus Data *User*

	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

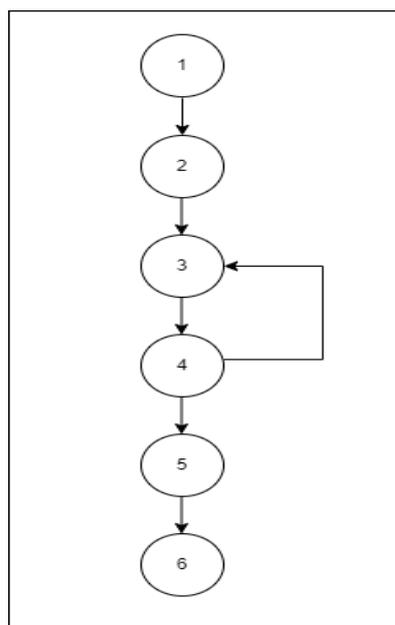
10) White Box Testing Hapus Data Karyawan

a) *Flowchart*



Gambar 4. 61 *Flowchart* Hapus Data Karyawan

b) Flowgraph



Gambar 4. 62 *Flowgraph* Hapus Data Pesanan

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 6

N (*node*) = 6

P (Predikat *node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 6 - 6 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-3

Path 2 = 1-2-3-4-5-6

(4) Grafik matriks hapus data karyawan

Tabel 4. 34 Grafik Matriks Hapus Data Karyawan

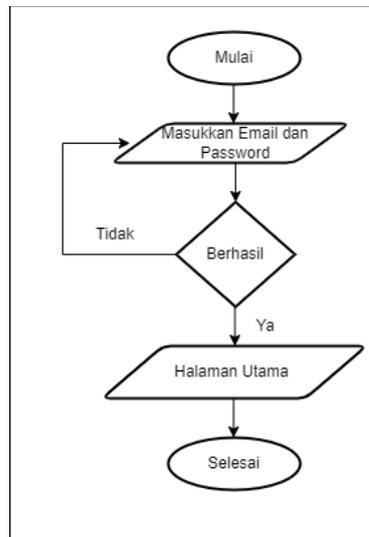
	1	2	3	4	5	6	E-1
1		1					1-1 = 0
2			1				1-1 = 0
3				1			1-1 = 0
4			1		1		2-1 = 1

5						1	1-1 = 0
6							0
SUM (E + 1)							1+1 = 2

c. User

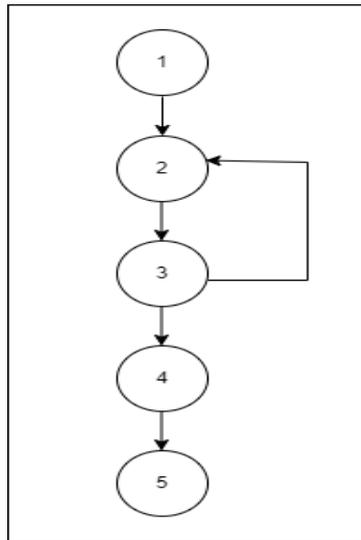
1) **White Box Testing Kesalahan Email dan Password**

a. **Flowchart**



Gambar 4. 63 *Flowchart Kesalahan Email dan Password*

b. **Flowgraph**



Gambar 4. 64 *Flowgraph* Kesalahan *Email* dan *Password*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-2

Path 2 = 1-2-3-4-5

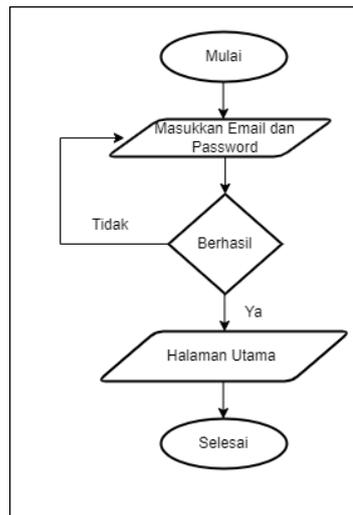
(4) Grafik matriks kesalahan *username* dan *password*

Tabel 4. 35 Grafik Matriks Kesalahan *Username* dan *Password*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
SUM (E +1)						$1 + 1 = 2$

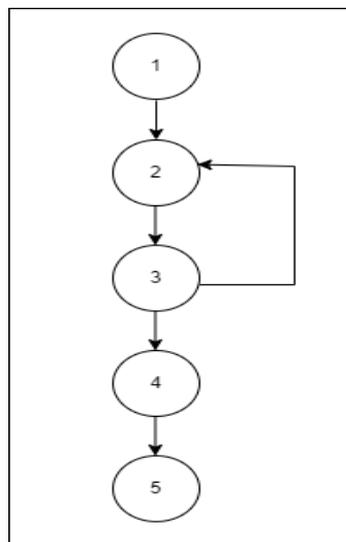
2) ***White Box Testing Login Berhasil***

a. ***Flowchart***



Gambar 4. 65 *Flowchart* Berhasil Login

b. Flowgraph



Gambar 4. 66 *Flowgraph* Berhasil Login

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

$$P \text{ (Predikat node)} = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-2

Path 2 = 1-2-3-4-5

(4) Grafik matriks *login* berhasil

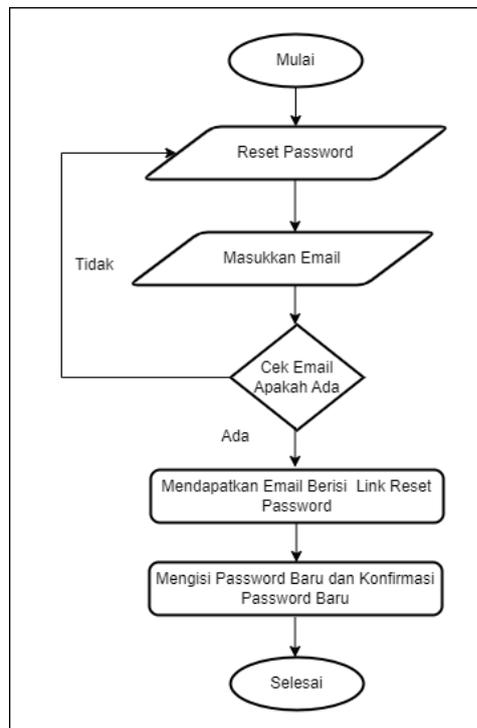
Tabel 4. 36 Grafik Matriks *Login* Berhasil

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$

4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$1 + 1 = 2$

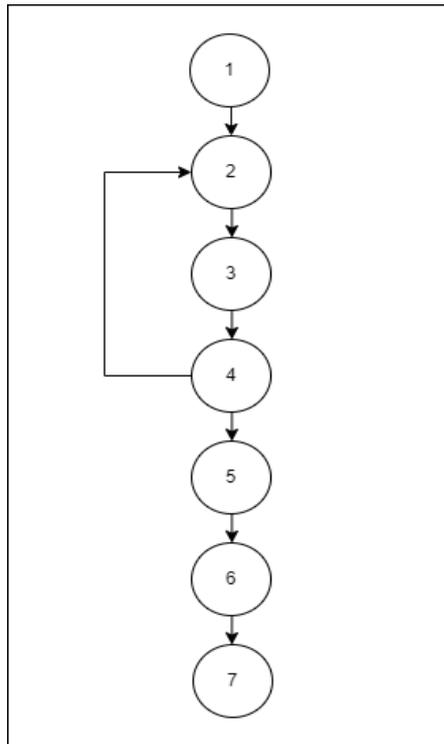
3) *White Box Testing Lupa Reset Password*

a. *Flowchart*



Gambar 4. 67 *Flowchart Reset Password*

b. *Flowgraph*



Gambar 4. 68 *Flowgraph Reset Password*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 7

N (*node*) = 7

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 7 - 7 + 2$

$= 2$

Predikat (P) = $P + 1$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-2

Path 2 = 1-2-3-4-5-6-7

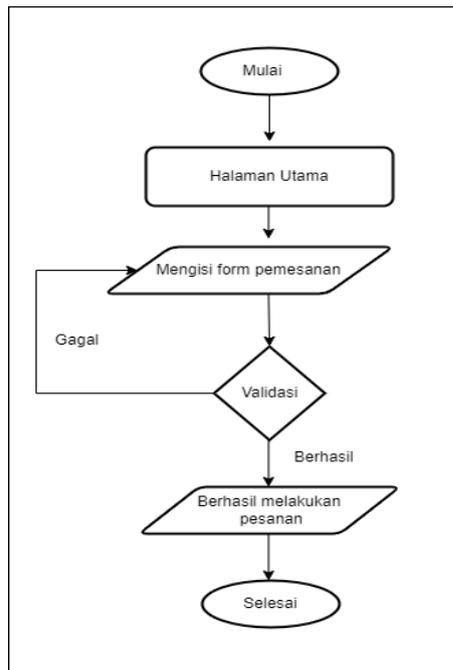
(4) Grafik matriks *reset password*

Tabel 4. 37 Grafik Matriks *Reset Password*

	1	2	3	4	5	6	7	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							1	1-1=0	
7								0	
	SUM (E + 1)								1+1=2

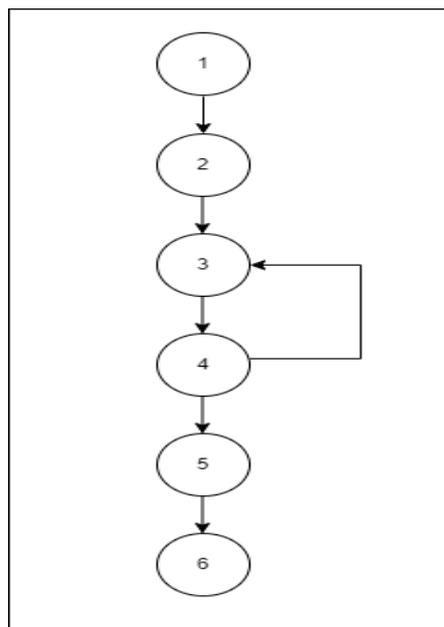
4) *White Box Testing* Berhasil Buat Pesanan

a. *Flowchart*



Gambar 4. 69 *Flowchart* Berhasil Buat Pesanan

b. Flowgraph



Gambar 4. 70 *Flowgraph* Berhasil Buat Pesanan

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

- (1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 6

N (*node*) = 6

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

= 6 - 6 + 2

= 2

$\text{Predikat (P)} = P + 1$

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1-2-3-4-3

Path 2 = 1-2-3-4-5-6

(4) Grafik matriks buat pesanan

Tabel 4. 38 Grafik Matriks Buat Pesanan

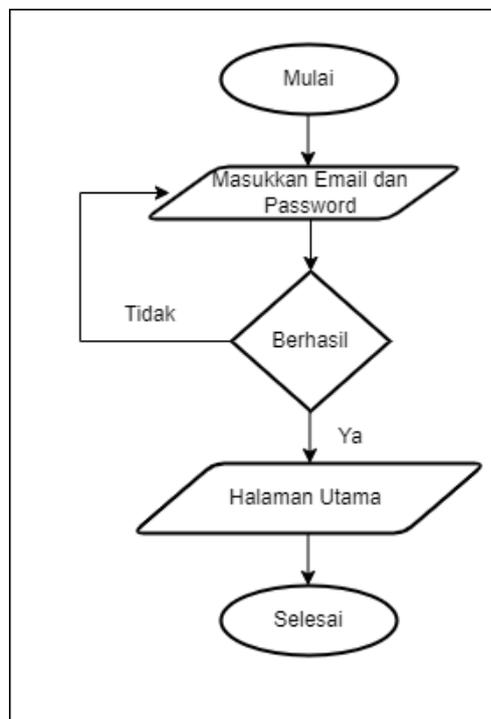
	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0

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

c. Karyawan

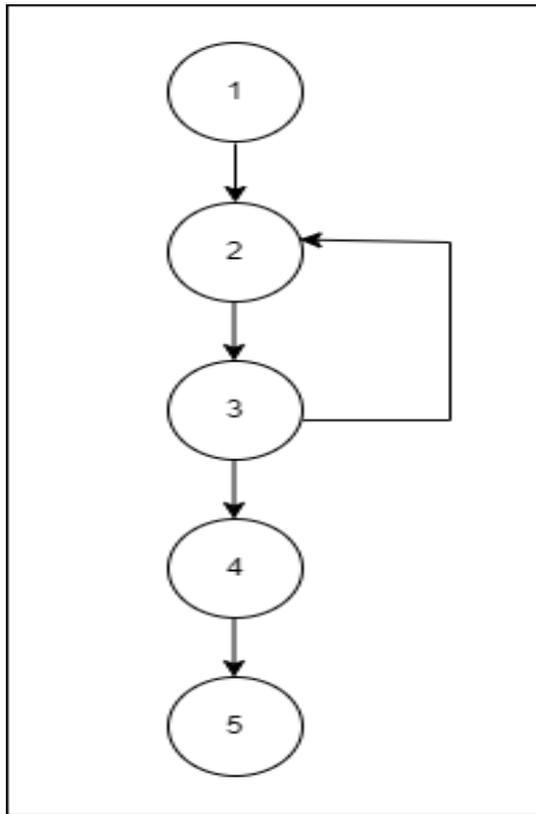
1) White Box Testing Kesalahan Username dan Password

a. Flowchart



Gambar 4. 71 Flowchart Kesalahan Email dan Password

b. Flowgraph



Gambar 4. 72 *Flowgraph* Kesalahan *Email* dan *Password*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus: $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (*Predikat node*) = 1

Penyelesaian:

$V(G) = E - N + 2$

$= 5 - 5 + 2$

$= 2$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1-2-3-2$$

$$\text{Path 2} = 1-2-3-4-5$$

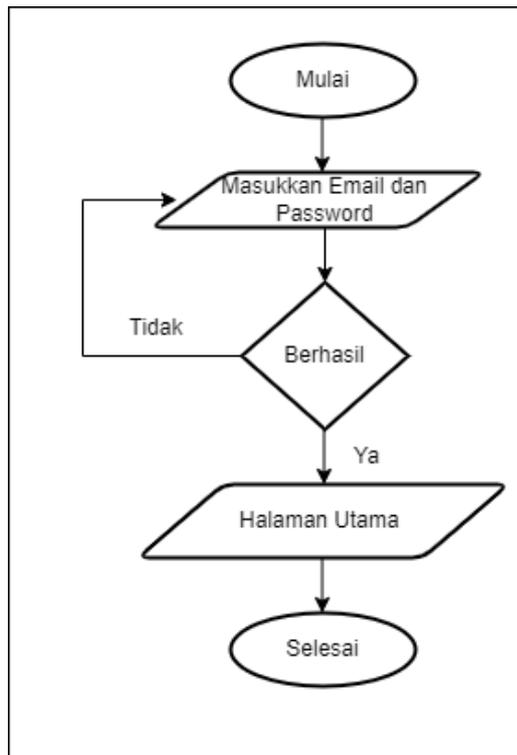
(4) Grafik matriks kesalahan *username* dan *password*

Tabel 4. 39 Grafik Matriks Kesalahan *Username* dan *Password*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$1 + 1 = 2$

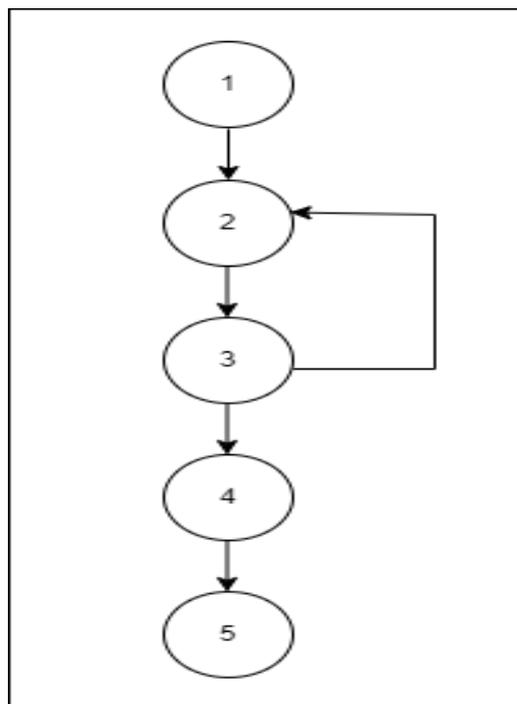
(5) *White Box Testing* Berhasil Login

a. *Flowchart*



Gambar 4. 73 *Flowchart Login Berhasil*

b. Flowgraph



Gambar 4. 74 *Flowgraph Berhasil Login*

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus: } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian:

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1-2-3-2$$

$$\text{Path 2} = 1-2-3-4-5$$

(4) Grafik matriks *username* dan *password* berhasil

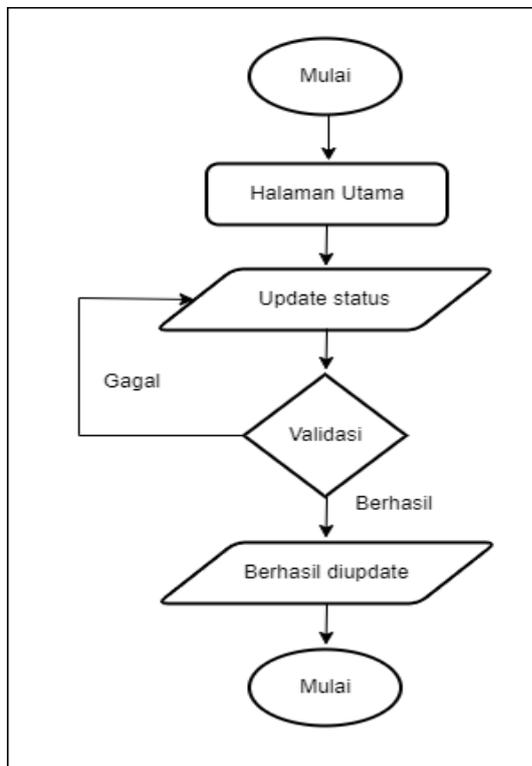
Tabel 4. 40 Grafik Matriks *Username* dan *Password* Berhasil

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$

2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$1 + 1 = 2$

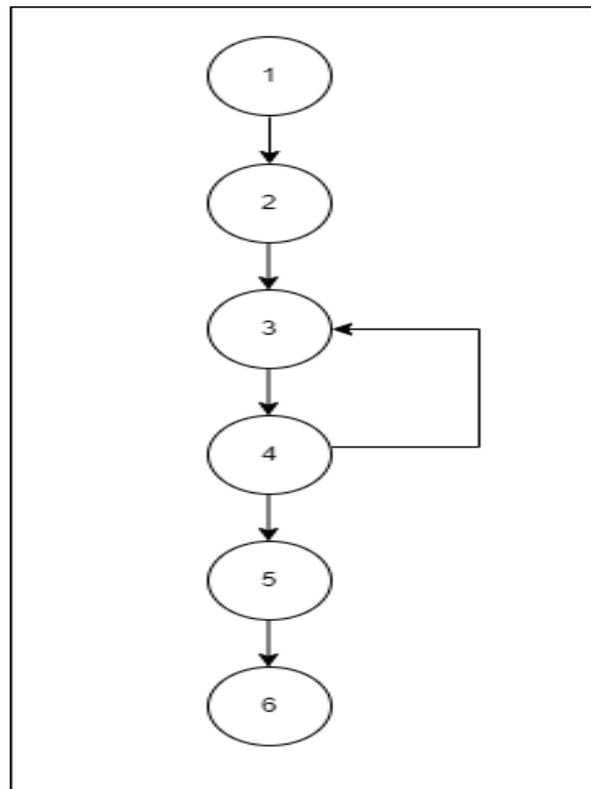
(6) White Box Testing Mengubah Status Pengantaran

a. Flowchart



Gambar 4. 75 Flowchart Berhasil Mengubah Status Pengantaran

b. Flowgraph



Gambar 4. 76 Flowgraph Berhasil Mengubah Status Pengantaran

Berdasarkan gambar di atas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *Region* = 2.

(3) *Independent path* pada *flowgraph* tersebut yakni :

$$\text{Path 1} = 1-2-3-4-3$$

$$\text{Path 2} = 1-2-3-4-5-6$$

(4) Grafik matriks mengubah status pengantaran

Tabel 4. 41 Grafik Matriks Berhasil Mengubah Status Pengantaran

	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

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian, maka dapat ditarik Kesimpulan sebagai berikut:

1. Metode pengumpulan data yang dilakukan yaitu dengan melakukan wawancara dan observasi langsung pada toko es kristal Atika. Beberapa data yang diperoleh peneliti antara lain 3 jenis es kristal dengan kemasan dan harga yang berbeda, 3 karyawan yang melakukan pengantaran, serta 3 lokasi.
2. Hasil uji coba menunjukkan bahwa sistem ini memberikan kemudahan bagi pengguna. Pengguna dapat dengan cepat dan mudah melakukan pemesanan dalam pemilihan jenis es kristal berfungsi dengan baik. Pengguna dapat memilih dan melihat riwayat pesanan tanpa masalah. Proses pembayaran berjalan lancar, namun terdapat sedikit keterlambatan pada opsi pembayaran *online* jika jaringan user dalam gangguan.
3. Dari data yang diperoleh, perbandingan penjualan dan pemesanan es kristal secara *online* yang berjumlah 1.990 bungkus lebih banyak dibandingkan pemesanan secara *offline* yang berjumlah 1.293 bungkus selama kurang lebih 2 bulan.
4. Dari data penjualan *offline* dan *online* pada bulan juni dengan persentase penjualan *offline* 36% dan penjualan *online* 63%. Sedangkan di bulan juli persentase penjualan *offline* sebanyak 44% dan penjualan online sebanyak 55%.

5. Dari data yang didapat dihasilkan persentase pelanggan yang belum bayar sebanyak 50% dan yang sudah bayar sebanyak 49%. Dari persentase tersebut dapat disimpulkan banyaknya pelanggan yang belum bayar dibandingkan pelanggan yang sudah bayar pada bulan juni.
6. Dari data yang didapat dihasilkan persentase pelanggan yang belum bayar sebanyak 72% dan yang sudah bayar sebanyak 27%. Dari persentase tersebut dapat disimpulkan banyaknya pelanggan yang belum bayar dibandingkan pelanggan yang sudah bayar pada bulan juli.

B. Saran

Untuk penelitian selanjutnya disarankan menambah fitur-fitur seperti pengelolaan stok agar pengguna dapat mengetahui stok produk yang tersedia, dan sistem *chatting* untuk mempermudah proses pembayaran *online* pada aplikasi agar pengguna dapat mengetahui stok produk yang tersedia.

DAFTAR PUSTAKA

- Alam, S., & Asrianda, S. (2021). *APLIKASI VOICE TO TEXT (V2T) BERBASIS ANDROID*. 1(3), 1–10.
- Anjani, D. D., Prakasiwi, C., & Windarto, A. P. (2023). Analisis Penerapan Jaringan Saraf Tiruan Backpropagation dalam Memprediksi Penjualan Produk Es Kristal. *Journal of Informatics, Electrical and Electronics Engineering*, 3(1), 153–163. <https://doi.org/10.47065/jieee.v3i1.1610>
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206. <https://doi.org/10.30998/string.v3i2.3048>
- Erni Sri Wahyuni. (2019). Analisis Cara Kerja CRUD dengan Menggunakan Android Studio. *Analisis Cara Kerja CRUD Dengan Menggunakan Android Studio*.
- Fitriyana, F., & Sucipto, A. (2020). Sistem Informasi Penjualan Oleh Sales Marketing Pada Pt Erlangga Mahameru. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 105–110. <https://doi.org/10.33365/jtsi.v1i1.239>
- Irsan, M. (2015). Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan. *Jurnal Penelitian Teknik Informatika*, 1(1), 115–120.
- Kharisma Rizki A. Wijaksono, Muh.Fadli Hasa, E. (2022). *jurnal+kharisma+FIX . 01(01)*, 34–43.
- Mamay Syani, & Nindi Werstantia. (2018). Perancangan Aplikasi Pemesanan Catering Berbasis Mobile Android. *Jurnal Ilmiah Ilmu Dan Teknologi Rekayasa /*, 1(2), 1–10.
- Muhammad, I., Masnur, M., & Syam, A. G. (2021). Aplikasi Qr Code Sebagai Sarana Penyampaian Informasi Pohon Dikebun Raya Jompie. *Jurnal Sintaks Logika*, 1(1), 33–41. <https://doi.org/10.31850/jsilog.v1i1.694>
- Muhammad Saed Novendri, Ade Saputra, C. E. F. (2019). *L e n t e r a d u m a i , . 10*, 46–57.
- P. Eni, & F. Siti. (2021). Perancangan Aplikasi Penjualan Berbasis Android Sebagai Media Pemesanan Pada Distro Online. *Bina Insani ICT Journal*, 8(2), 176–186.

- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. L. (2023). Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box and White Box Testing of Web-Based Parking Information System. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 1–16.
- Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, 1(2), 129–134. <https://doi.org/10.24036/jtein.v1i2.55>
- Santoso, N. R. (2017). *Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)*. 9(1), 84–91.
- Wardhani, K. S. (2014). Pengembangan Sistem Informasi class diagram Secara Digital. *Lambung Pustaka Universitas Negeri Yogyakarta*, 161.
- Yunus, R. M., Sujadi, H., & -, K. (2015). Sistem Keamanan Pesan Dengan Algoritma Rivest Code 6 (Rc-6) Menggunakan Java Pada Smartphone Berbasis Android. *J-Ensitem*, 2(01), 1–10. <https://doi.org/10.31949/j-ensitem.v2i01.45>