

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Perkembangan teknologi di zaman sekarang ini mengalami kemajuan yang sangat pesat. Perkembangan teknologi tersebut ditandai dengan adanya pengolahan dalam bidang pekerjaan yang pada awalnya dikelola secara *offline* kini telah dikelola menggunakan teknologi modern. Baik itu berupa mesin, peralatan digital bahkan teknologi pengolahan yang terkomputerisasi. Dengan teknologi yang semakin canggih dalam hal pernikahanpun sudah banyak yang menggunakan jasa *wedding organizer* yang lebih efektif dalam mencari apa saja yang akan dibutuhkan dalam pernikahan.

Dalam konteks ini, penelitian akan difokuskan pada pengembangan aplikasi *wedding organizer* berbasis Android sebagai solusi untuk mengatasi kendala minimnya informasi dalam pengambilan keputusan terkait resepsi pernikahan. Tujuan utama penelitian ini adalah untuk mengevaluasi efektivitas aplikasi dalam memfasilitasi pelanggan dalam proses penyewaan atau pemesanan perlengkapan pernikahan secara online, serta dampaknya terhadap pengalaman pengguna dan efisiensi dalam persiapan pernikahan.

Toko Rumah Pengantin Dheailova berdiri pada tanggal 27 Januari 2010 berlokasi di Jl Lkr. Lapadde No.2, Lapadde, Kec.Ujung, Kota Parepare, Sulawesi Selatan. Toko Rumah Pengantin Dheailova ini menyediakan berbagai macam model baju mulai dari baju bodo, baju pengantin dekorasi dan juga makeup,

adapun foto *prewedding* dan *make up* yang sudah profesional dan banyak testimonial di Instagram Dheailova, semuanya bisa dipaketkan atau satu penyewaan tergantung kebutuhan *Customer*.

Saat ini, pasti sudah tidak asing saat melihat atau mendatangi pesta pernikahan yang menggunakan jasa *wedding organizer*. *Wedding organizer* adalah salah satu jasa yang membantu calon pengantin dan keluarga dalam merancang moment indah rangkaian acara pesta pernikahan menjadi sangat menarik dan lebih berkesan.

Salah satu penelitian terdahulu yang terkait dengan penelitian ini yaitu pada jurnal berjudul model sistem informasi pemesanan pada Qirani *Wedding Organizer* Berbasis *Web*, sistem aplikasi penyewaan baju berbasis *web* untuk mempermudah *Customer* dalam mencari baju-baju yang ingin disewa dan bisa diakses lewat internet tanpa harus ke lokasi tempat penyewaan terlebih dahulu (Andini 2022)

Berdasarkan penelitian di atas maka peneliti bermaksud melengkapi penelitian sebelumnya yang masih berbasis *web* menjadi aplikasi berbasis *mobile*, dengan alasan di zaman ini semua orang sudah banyak yang memakai *smartphone* jadi dengan adanya penyewaan baju pengantin berbasis *android* nantinya *Customer* dapat dengan mudah menggunakan aplikasi tersebut hanya menginstal aplikasinya di *playstore*, lebih cepat dalam bertransaksi, aman dalam pembayaran dan *modern*.

Berdasarkan pembahasan diatas penulis bermaksud mengangkat judul penelitian “**Aplikasi *Wedding Organizer* Berbasis *Android***” dimana di

harapakan dengan adanya aplikasi *android* ini mampu mempermudah pelanggan dalam hal penyewaan atau pemesanan *wedding* secara *online*.

### **B. Rumusan Masalah**

Berdasarkan latar belakang penelitian yang ditulis di atas, maka rumusan masalah yang akan di jadikan penelitian sebagai berikut :

1. Bagaimana cara membuat aplikasi android *mobile* berbasis informasi dan pemesanan untuk Toko Pengantin Dheailova?
2. Bagaimana agar pelanggan tidak secara langsung mengunjungi tempat untuk melakukan transaksi?

### **C. Batasan Masalah**

Berdasarkan latar belakang dan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah :

1. Aplikasi ini tidak menyediakan jasa *Catering*, Dokumentasi dan Cincin Kawin.
2. Sistem ini dibuat khusus *owner* dan penyewa/pembeli, dan hanya bisa diakses melalui *smartphone android*.
3. Perancangan sistem aplikasi ini hanya membahas tentang proses pemesanan atau penyewaan dan pembayaran transaksi.

### **D. Tujuan Penelitian**

Tujuan dari pembuatan rancangan aplikasi ini adalah sebagai berikut:

1. Membangun dan merancang sebuah aplikasi untuk mempermudah *Customer* atau calon pengantin untuk menyewa atau memesan baju pengantin di toko tersebut.

2. Mengembangkan sistem informasi secara *online* dan melakukan transaksi pembelian atau penyewaan secara online sebagai *alternative* dari sistem yang selama ini menggunakan informasi secara sederhana melalui *chat* teman ke teman dan iklan di media sosial.

### **E. Manfaat Penelitian**

Adapun manfaat yang akan dicapai dari penelitian ini adalah :

1. Bagi Penulis

Manfaat yang didapat bagi peneliti adalah dapat memberi pengetahuan tentang bagaimana membuat sebuah aplikasi berbasis *android*.

2. Bagi Pembaca

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

3. Bagi pengguna

Untuk memudahkan pemesanan wedding *organizer* berbasis *android*.

### **F. 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 penyewaan baju pengantin 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 Baju Pengantin di Toko Rumah Pengantin Dheailova 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**

### **TINJAUAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Wedding organizer**

*Wedding Organizer* sebenarnya sama halnya dengan *Event Organizer* dalam konsep dasarnya sebagai jasa penyelenggara persiapan pernikahan. Sebab, *Wedding Organizer* merupakan bagian dari *Event organizer*. Sesuai dengan namanya maka WO (*Wedding Organizer*) adalah jasa yang pengorganisasian segala aktivitas persiapan pernikahan yang berkaitan dengan kebutuhan dalam suatu pesta pernikahan. Peran penting *wedding organizer* haruslah memberikan layanan yang memenuhi akan kebutuhan kecepatan, kemudahan, dan kepraktisan untuk mengatasi masalah *time efficiency* masyarakat perkotaan. *Event Organizer* sendiri memiliki fungsi secara umum untuk mengkoordinasi, melayani, dan mendukung pihak-pihak tertentu yang terkait persiapan pernikahan acara.

##### **2. Pernikahan**

Dalam islam, pernikahan merupakan ikatan dalam hubungan yang suci untuk hidup bersama dan bahagia serta mencapai tujuan pernikahan yang telah ditetapkan baik dalam urusan duniawi maupun akhirat, Selain itu, hubungan pernikahan adalah fitrah yang harus dan biasa terjadi dalam kehidupan. Dengan pernikahan akan timbulah jalinan hubungan yang harmonis antar pasangan untuk

saling berkasih sayang sehingga keduanya memiliki rumah tangga yang damai. 2 Tidak hanya mengatur mengenai tujuan dan manfaat pernikahan antara dua orang manusia semata, Islam juga mengatur kriteria pasangan yang harus dipilih, mengatur siapa saja yang boleh dan dilarang untuk dinikahi, menjelaskan hukum pernikahan, serta lain sebagainya yang dapat dijadikan pedoman demi mencapai pernikahan yang akan berlabuh pada kehidupan rumah tangga yang baik dan tenteram.(Heri Firmansyah, Hamsah Hudafi, Rizky Syahputra Nasution, 2023).

### **3. 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)

### **4. Perancangan**

Perancangan merupakan dasar utama dalam membuat aplikasi, dengan tujuan memberikan gambaran lengkap dengan jelas kepada *programmer* tentang aplikasi yang akan dibuat. perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa

pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan.(Sari, Siska, & Budiman, 2021)

## 5. Penyewaan

Penyewaan adalah suatu kesepakatan atau persetujuan dimana pihak yang satu menyanggupkan dirinya untuk menyerahkan suatu kebendaan kepada pihak lain, agar pihak ini dapat menikmatinya dalam jangka waktu tertentu, yang mana pihak yang belakang ini sanggup membayarnya. Penyewaan adalah suatu kegiatan yang melayani jasa peminjaman dengan tidak mengabaikan suatu ketentuan atau kesepakatan dan syarat – syarat yang berlaku didalam organisasi tersebut guna mencapai satu tujuan Bersama. (Wahyudi, 2019)

## 6. Android



**Gambar 2. 1** 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, sehingga *android* terus berkembang pesat dari segi teknologi maupun dari segi jumlah *device* yang ada didunia.(Yunus et al., 2015).

## 7. Android studio



**Gambar 2. 2** 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.

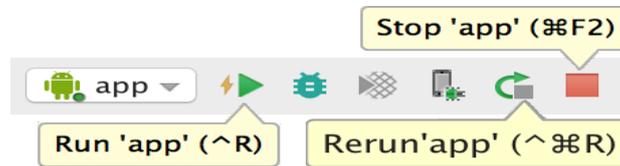
*Android Studio* sendiri dikembangkan berdasarkan *IntelliJ IDEA*, mirip dengan *Eclipse*, disertai dengan plugin ADT (*Android Development Tools*).

*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:

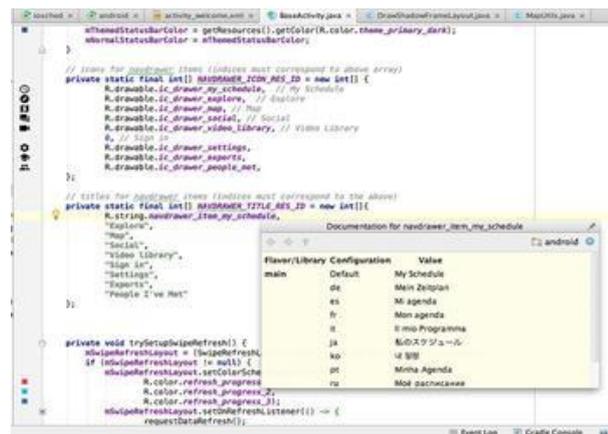
## 1) Instant RUN



**Gambar 2.3 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 2.4 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)

## 8. MySQL



**Gambar 2. 5** 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.

*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. 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*

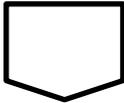
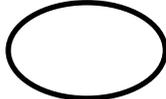
*Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam code dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya 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).

## **10. 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 inialisasi/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.

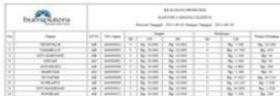
**11. Blackbox testing**

*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.

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

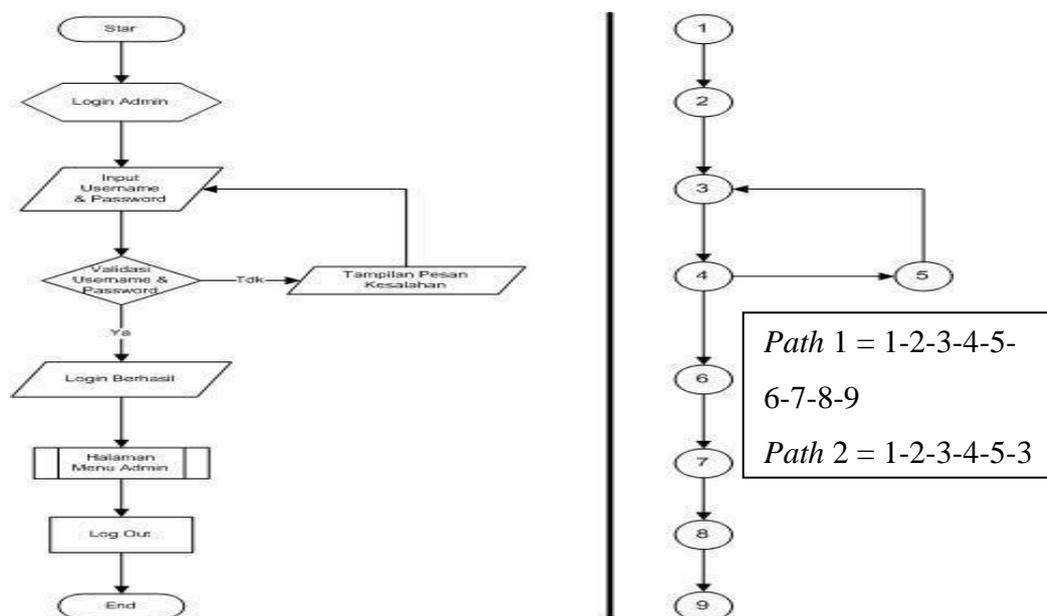
**Gambar 2.7** Contoh *Blackbox Testing*

## 12. *Whitebox 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.

Berikut ini adalah contoh pengujian *white box* pada sistem *login* yang terdapat pada Sistem Informasi laporan Data Statistik Desa Pananrang berbasis *Web*. (Asmil. (2020)), yaitu :



**Gambar 2. 8** Contoh *Whitebox Testing*

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

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

$$E(\text{edge}) = 9, N(\text{node}) = 9$$

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

$$V(G) = 2$$

Jadi jumlah path pada *flowgraph* diatas 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

### 13. 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*)

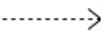
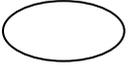
Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

- 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).

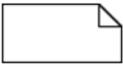
Diagram *use case* menyajikan intraksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun.

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

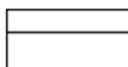
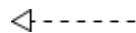
**Tabel 2. 2** Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peranyang 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
3.		<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> ).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</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 <i>actor</i>

Lanjutan **Tabel 2. 2**

NO	GAMBAR	NAMA	KETERANGAN
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

**Tabel 2. 3** Simbol *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 actor
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

Lanjutan Tabel 2. 3

NO	GAMBAR	NAMA	KETERANGAN
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 4 Simbol *Sequence Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang salingberinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi- informasi tentang aktifitas yang terjadi

Tabel 2. 5 Simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki olehsuatu objek.

Lanjutan Tabel 2. 5

NO	GAMBAR	NAMA	KETERANGAN
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 di jalankan dan mencerminkan suatu sumber daya komputasi..
2.		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali

Tabel 2. 6 Simbol *Activity* Diagram

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain

Lanjutan Tabel 2. 6

NO.	GAMBAR	NAMA	KETERANGAN
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

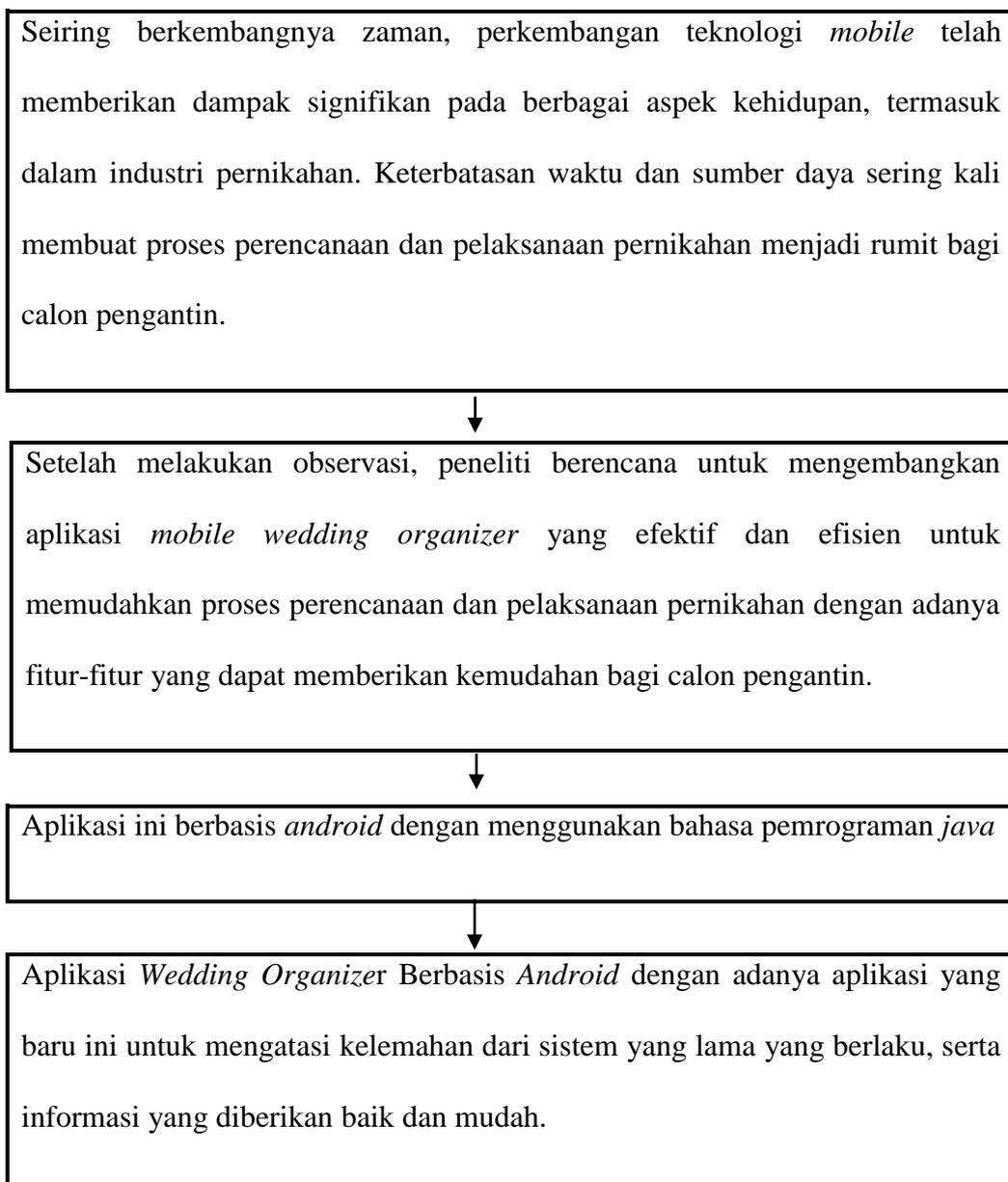
### B. Kajian Hasil Penelitian Terdahulu

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

1. (Andini 2022, umpar) dengan judul penelitian “ model sistem informasi pemesanan pada Qirani *wedding organizer* berbasis *web* “ dalam penelitian ini, aplikasi yang dibuat masih menggunakan web dan aplikasi ini dapat di akses dimana pun dan kapan pun serta mempermudah customer untuk melakukan pemesanan.
2. (Muntari & Masdalipa, n.d.) dengan judul penelitian ” penerapan jasa *wedding organizer* berbasis *web* dikota pagar alam”. Hasil riset yang dilakukan oleh aplikasi ini dapat membantu konsumen atau pengguna mengetahui salon mana saja dikota pagar alam yang memiliki *wedding planner* terbaik dan segala perlengkapan pernikahannya, sehingga memudahkan untuk mencari informasi atau melakukan pemesanan. Acara pernikahan dengan cepat dan andal.

3. (Ginajar & Setiawan, n.d.) dengan judul penelitian “ Perancangan Aplikasi *Wedding Organizer* Berbasis *Android* ”. dalam penelitian ini , Aplikasi *wedding organizer* yang dirancang mempunyai 3 akses *login* yaitu sebagai klien, pihak wo dan admin yang dapat memudahkan melakukan transaksi pemesanan. Aplikasi ini dirancang untuk membantu klien dalam memperoleh informasi seputar paket *wedding* sesuai lokasi daerah. Aplikasi ini dapat memuat informasi semua vendor *wedding organizer* di seluruh Indonesia.

### C. Kerangka Pikir



**Gambar 2.9** Kerangka Pikir

## **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, kuesioner, wawancara mendalam dengan objek penelitian, pengkajian dokumen, hingga fokus *discussion group*.

#### **B. Tempat Dan Waktu Penelitian**

##### 1. Tempat

Beralamat di kota parepare tepatnya di Rumah Pengantin Dheailova Jl. Lkr. Lapadde No.2, Lapadde, Kec.Ujung, Kota Parepare, Sulawesi Selatan.

##### 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 Penelitian**

### **1. Analisis Data**

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 :

### **2. 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.

### **3. Pengumpulan Data**

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

### **4. 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.

## 5. Perancangan

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

## 6. Pengujian

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

## 7. Implementasi

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

### E. Alat dan bahan penelitian

#### 1. Alat

- a. Laptop *Acer Aspire 3 A314-32-C09W*
- b. *Processor N4000*
- c. *RAM 4,00 GB*
- d. *HDD 1000 GB*
- e. *Windows 10*
- f. *Android Studio*
- g. *Java*

#### 2. Bahan

- a. Data foto model baju bodo dan baju pengantin
- b. Data foto *make up*

c. Data Tarif harga

- 1). Baju pengantin = 2.500.000
- 2). Baju bodo = 100.000 1pcs
- 3). Tenda /Dekorasi = 8.000.000
- 4). Make Up = 1.000.000
- 5). Paketan
  - Paket 1 = 10.000.000
  - Paket 2 = 12.000.000

## F. Metode pengujian

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

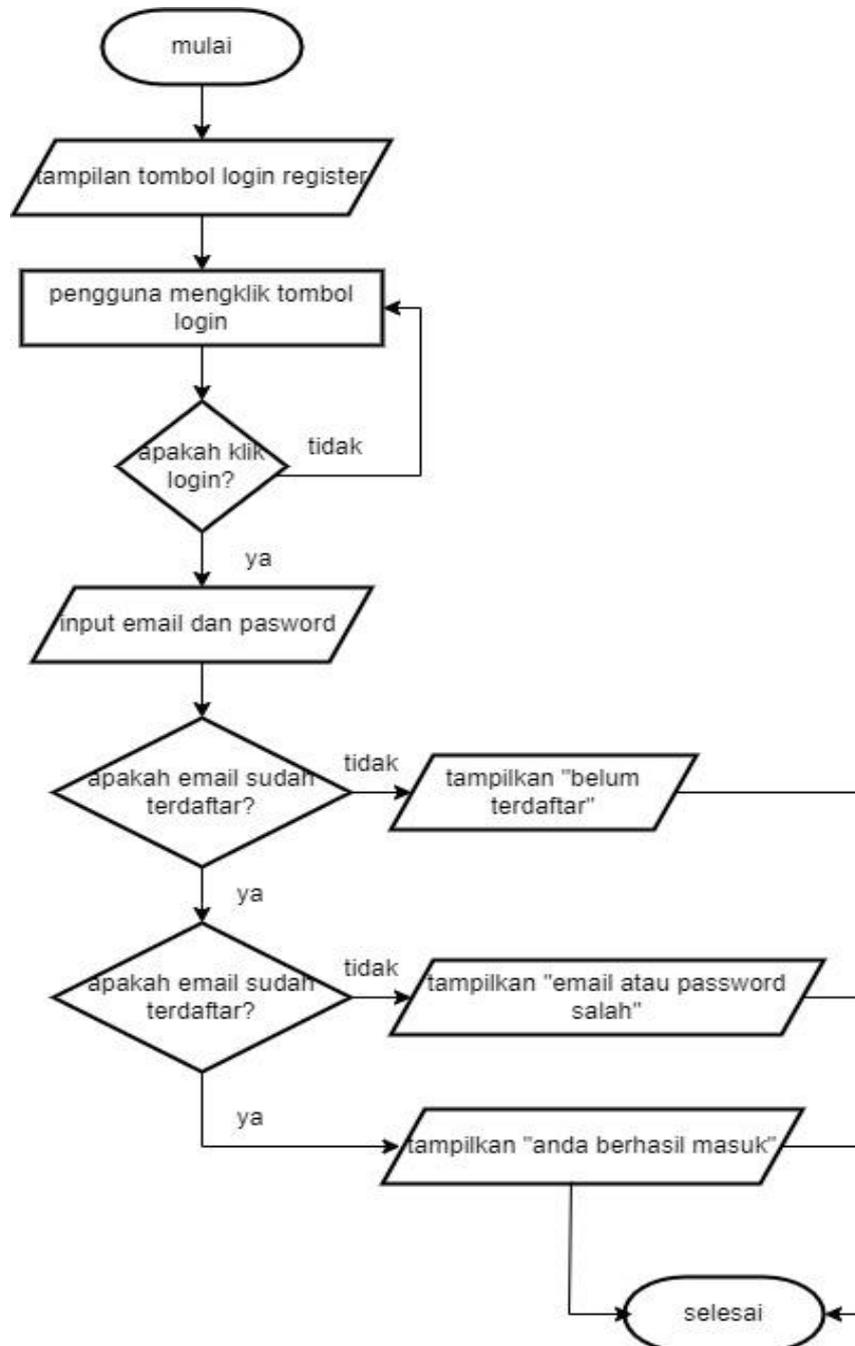
### 1. *Blackbox testing*

*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.

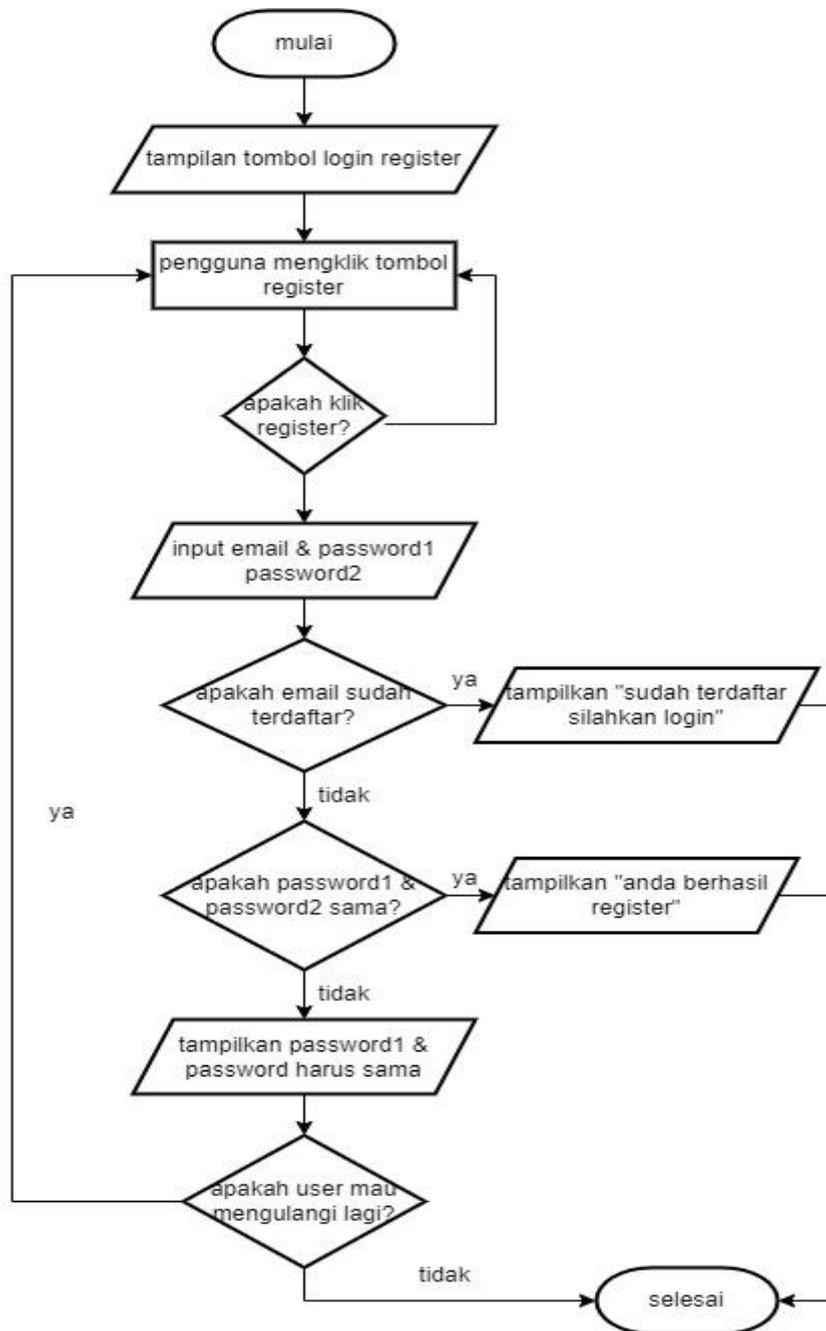
### G. Diagram alir



**Gambar 3.1** Flowchart Login User

Pada gambar diatas 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

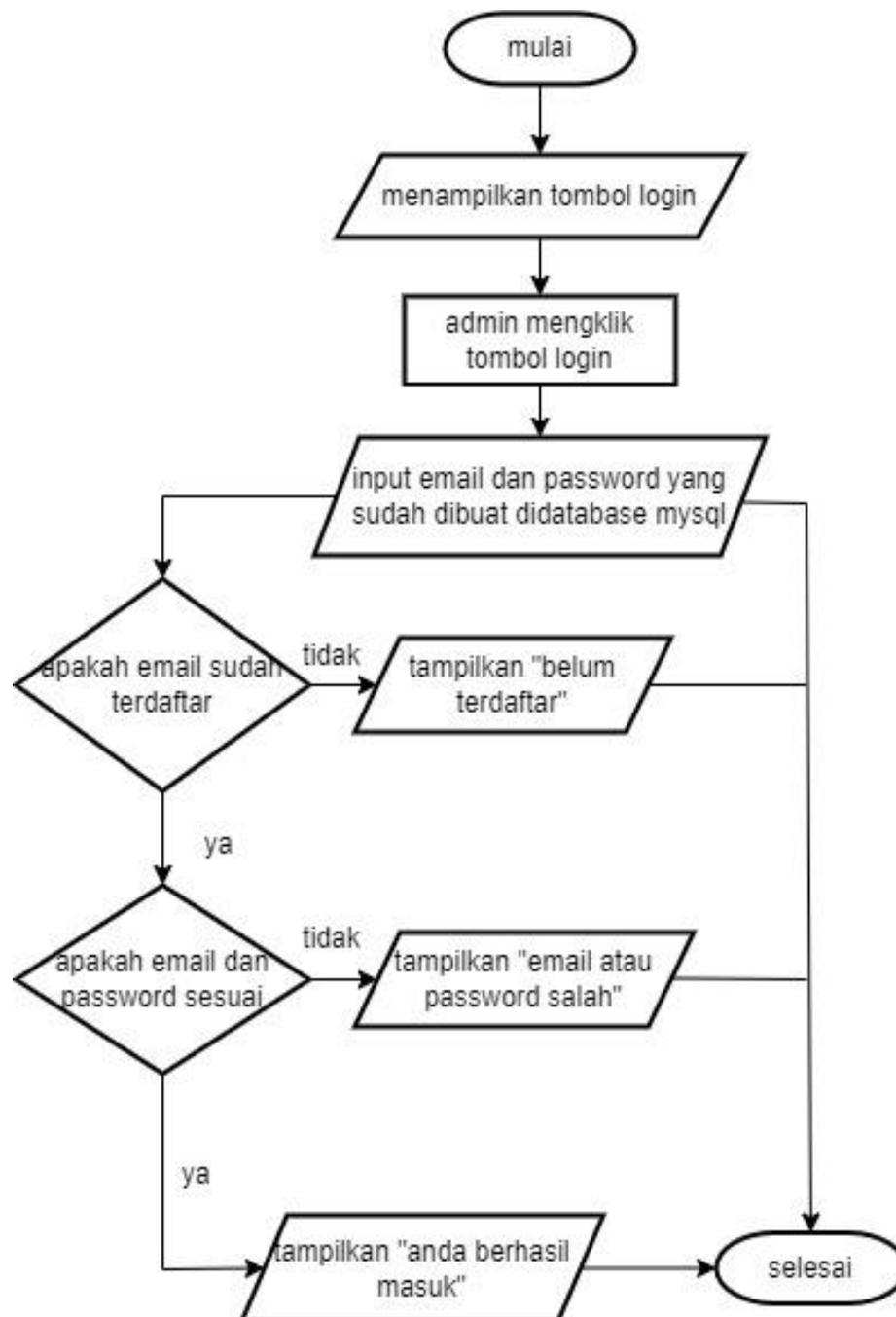
tidak maka kembali ke halaman *login* dan setelah menginput email dan *password* maka *user* akan di tampilkan halaman apakah *email* sudah terdaftar apabila tidak maka akan menampilkan belum terdaftar dan apabila *email* dan *password* belum sesuai maka akan menampilkan *email* dan *password* salah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



**Gambar 3. 2** Flowchart Register user

Pada gambar diatas 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

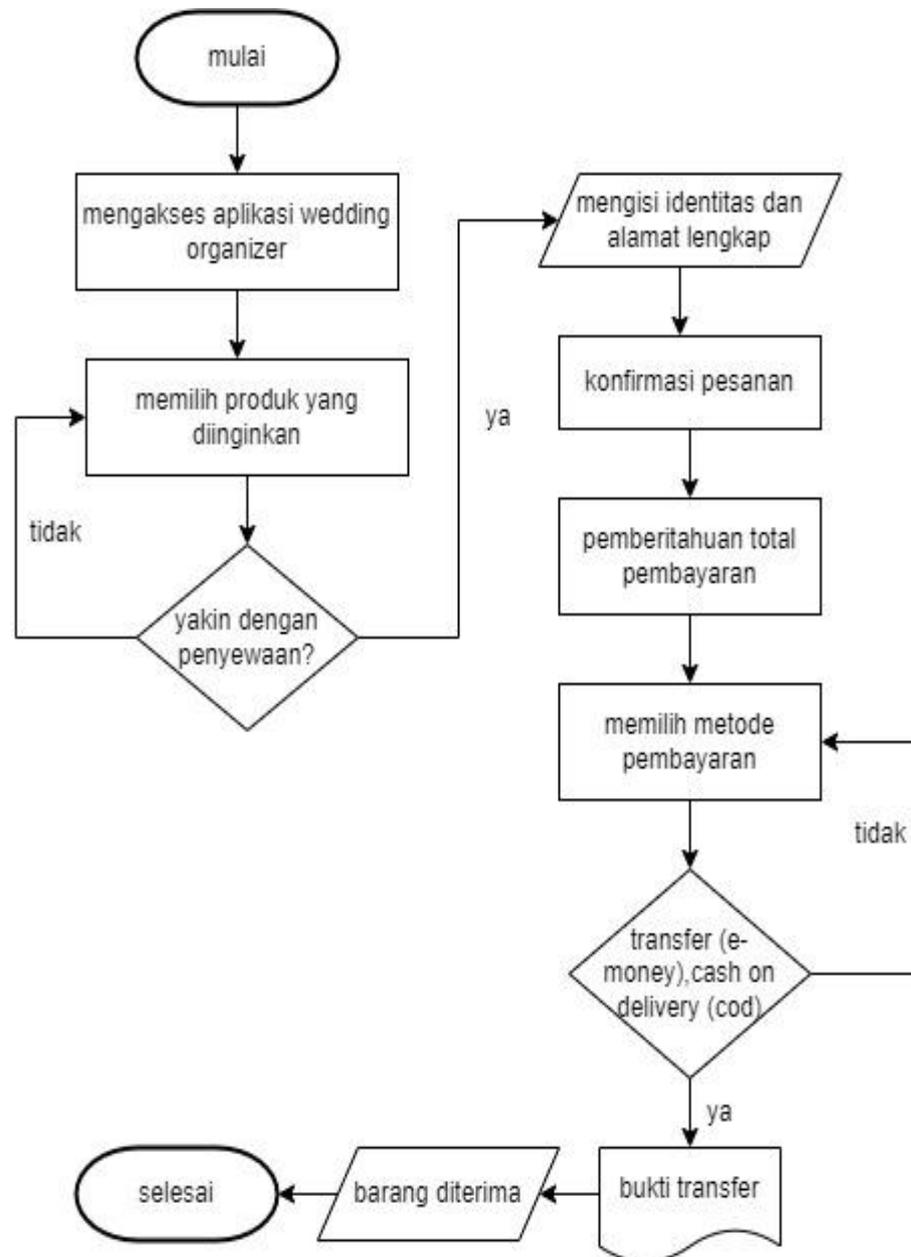
apakah *email* sudah terdaftar dan apabila ya maka akan menampilkan sudah terdaftar silahkan *login* dan apabila tidak maka akan menampilkan *password* satu dan dua harus sama dan apabila ya maka akan menampilkan anda berhasil *register* dan apabila tidak maka akan menampilkan *password* satu dan *password* dua harus sama kemudian akan menampilkan apakah user mau mengulangi lagi apabila maka kembali ke halaman tombol *register* dan apabila tidak maka selesai.



**Gambar 3.3** Flowchart Login Admin

pada gambar diatas menjelaskan mengenai alur dari login *admin* yaitu *admin* akan ditampilkan menu *login* dan mengklik tombol *login* kemudian input email dan *password* yang sudah dibuat di *database mysql* dan akan muncul

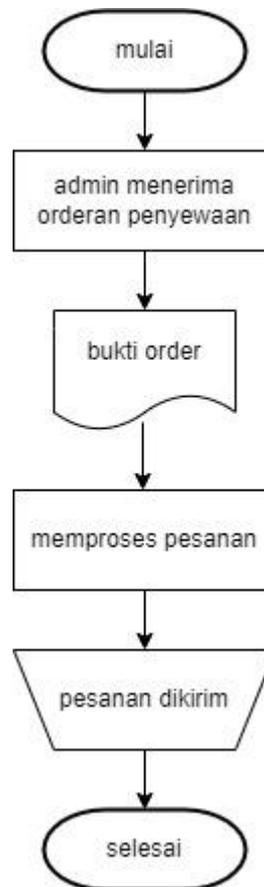
apakah *email* sudah terdaftar apabila tidak maka akan menampilkan belum terdaftar dan apabila ya maka akan menampilkan apakah *email* dan *password* sesuai dan apabila tidak maka akan tampil *email* atau *password* salah dan apabila ya maka akan menampilkan anda berhasil masuk dan selesai.



**Gambar 3. 4** Flowchart Proses Transaksi Costumer/User

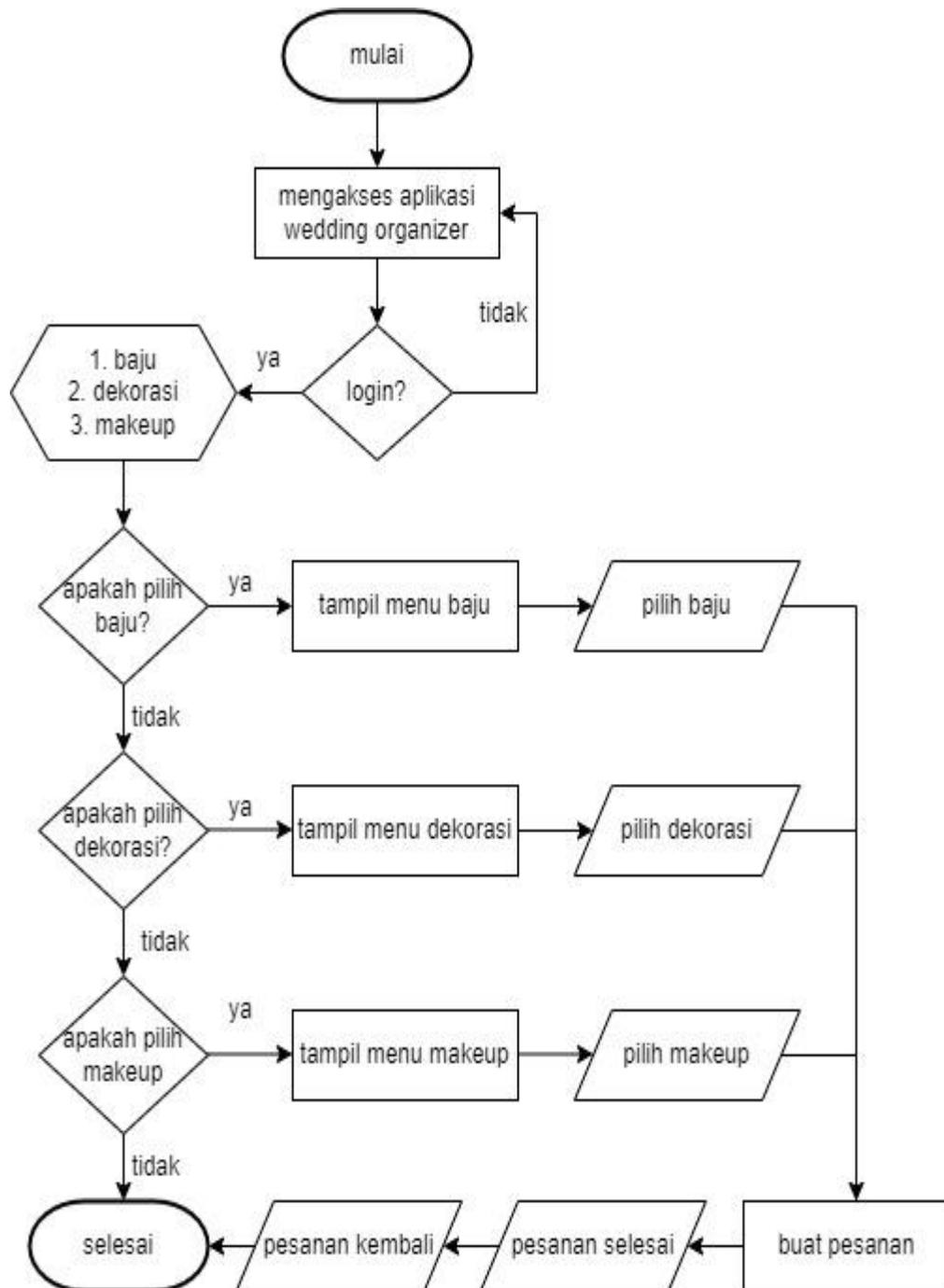
Pada gambar diatas menjelaskan mengenai alur dari proses transaksi yaitu *costomer* akan mengakses wedding organizer kemudian memilih produk yang diinginkan dan apabila yakin dengan penyewaan maka akan mengisi identitas dan alamat lengkap apabila tidak maka kembali ke halaman memilih produk kemudian 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 (e-money)* kemudian menampilkan bukti transfer dan tanda terima kemudian barang di terima dan selesai.



**Gambar 3. 5** *Flowchart* Proses Pengiriman Barang

Pada gambar di atas menjelaskan mengenai alur dari proses pengiriman barang yaitu admin menerima *order* penyewaan kemudian bukti *order* kemudian memproses pesanan kemudian pesanan di kirim dan selesai.



**Gambar 3. 6** Flowchart Memesan Produk

Pada gambar di atas menjelaskan mengenai alur dari aplikasi *wedding organizer*. Ketika pengguna membuka aplikasi maka pengguna akan di tampilkan dengan halaman *login* jika berhasil login maka lanjut ke halaman utama jika

tidak maka kembali ke *login*. kemudian halaman utama dengan 3 opsi pilihan yaitu baju, dekorasi, *make up* dan jika pengguna memilih diantara salah satu opsi tersebut maka akan menampilkan pilihan dengan berbagai macam item didalamnya contoh jika pengguna memilih baju maka akan menampilkan berbagai macam baju diantaranya baju pengantin, mappetuada/lamaran, *prewedding*, dan baju bodo dewasa. Aplikasi ini memberikan pengalaman bagi pengguna dalam mengakses butik pernikahan berbasis *android*.

## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Hasil Penelitian

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

**Tabel 4.1** Hasil Penelitian

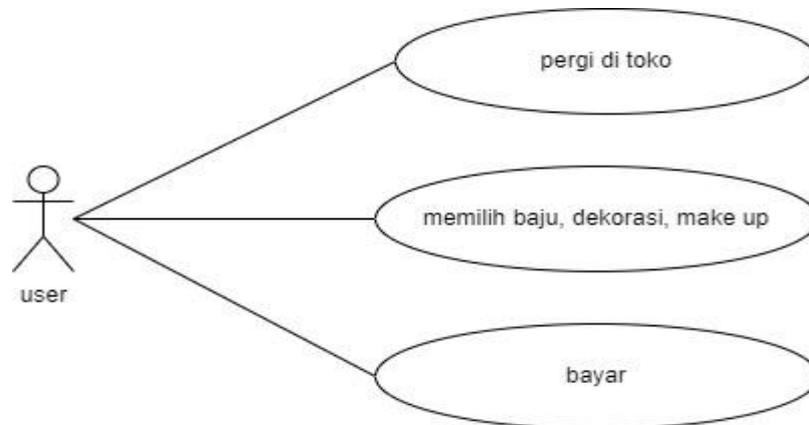
Jenis	Harga
Baju pengantin	2.500.000
Baju bodo	100.000
Dekorasi/tenda	8.000.000
Make up	1.000.000
Paketan	Paket 1 = 10.000.000 Paket 2 = 12.000.000

#### B. Analisis Kebutuhan dan Rancangan

##### 1. *Usecase Diagram*

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

## a) Desain sistem yang berjalan

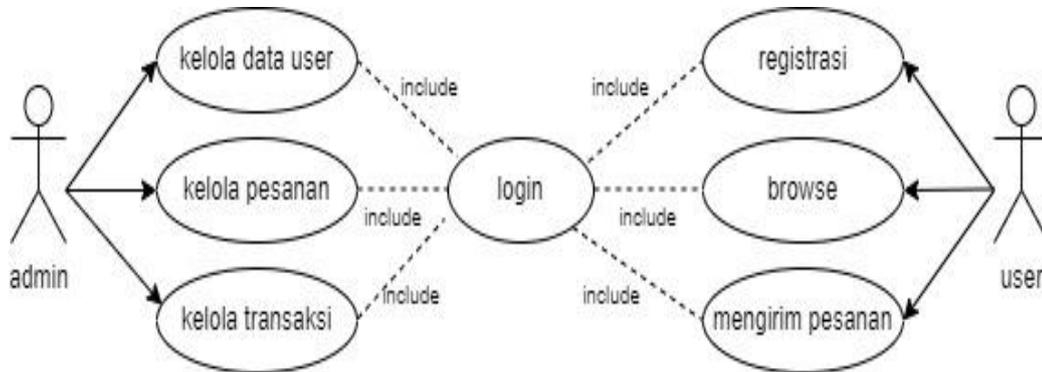
**Gambar 3. 7** Desain Sistem yang Berjalan

Pada gambar di atas sistem yang berjalan pada saat ini, *user* langsung ke toko untuk melihat baju, dekorasi dan *make up* karena belum ada aplikasi khusus yang bisa memudahkan dalam penyewaan/pemesanan.

**Tabel 4. 2** Penjelasan Desain Sistem yang 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
Memilih barang	<i>Use case</i> ini menjelaskan bahwa <i>user</i> memilih baju dekorasi dan <i>make up</i>
Bayar	<i>Use case</i> ini menjelaskan bahwa <i>user</i> melakukan pembayaran

b) Desain sistem yang diusulkan



**Gambar 4. 1** Use Case Diagram Admin dan User

Interaksi antara user dan admin dengan aplikasi dijelaskan dalam skenario *use case* sebagai berikut:

**Tabel 4. 3** Use Case Diagram Admin

Nama use case	deskripsi
<i>Login</i>	<i>Use case</i> ini menjelaskan proses dimana admin login terlebih dahulu
Kelola produk	<i>Use case</i> ini menjelaskan bahwa admin dapat mengelola produk yang ada di dalam aplikasi
Kelola pesanan	<i>Use case</i> ini menjelaskan bahwa admin dapat mengelola pesanan
Kelola transaksi	<i>Use case</i> ini menjelaskan bahwa admin dapat mengelola proses transaksi

**Tabel 4. 4** Use Case Diagram User

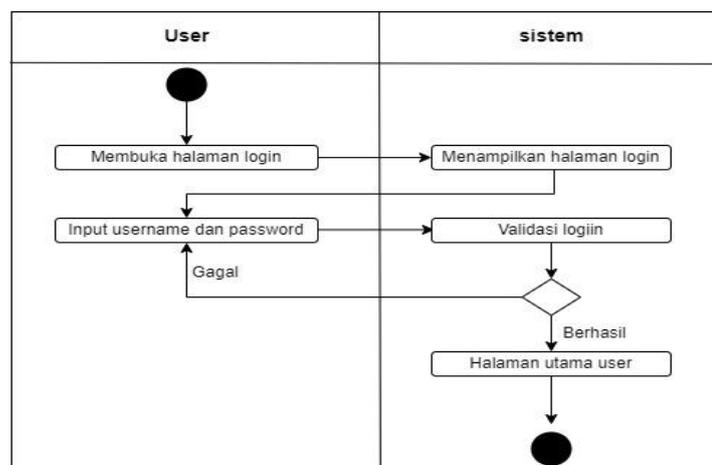
Nama use case	deskripsi
<i>Login</i>	<i>Usecase ini menjelaskan bahwa user akan login</i>
Registrasi	<i>Use case ini menjelaskan bahwa user melakukan registrasi terlebih dahulu</i>
Browse	<i>Usecase ini menjelaskan bahwa user mencari apa yang ingin dipesan/disewa</i>
Kirim pesan	<i>Usecase ini menjelaskan bahwa user dapat mengirim pesannya</i>

## 2. Activity Diagram

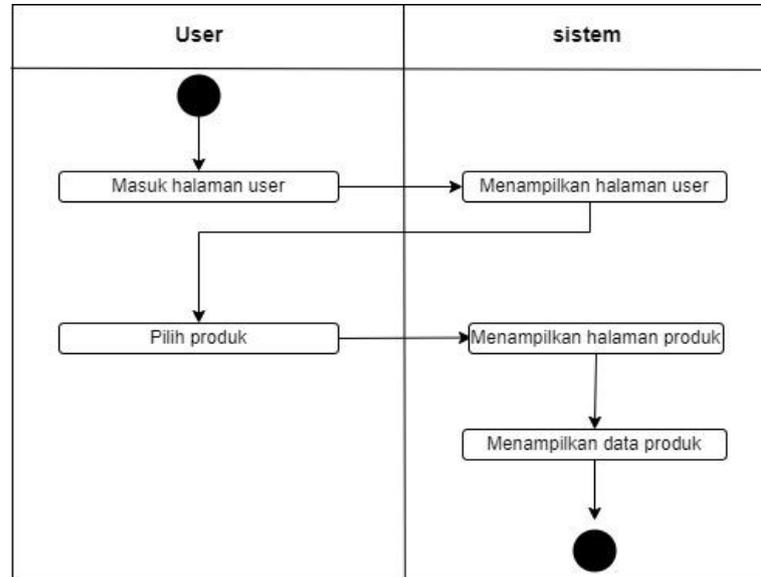
*Activity* diagram merupakan bahasa spesifikasi standar yang dipakai untuk mendokumentasikan, mengspesifikasikan, dan mengembangkan perangkat lunak. Diagram ini menggambarkan berbagai aktivitas dalam aliran proses sebuah sistem.

### a. Activity Diagram User

#### 1) Activity diagram login

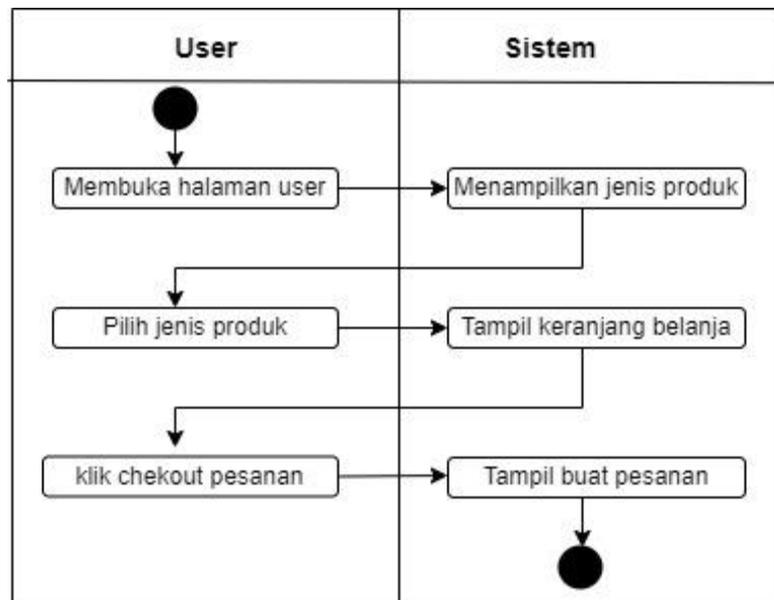
**Gambar 4. 2** Activity Diagram Login

2) *Activity diagram lihat data produk*



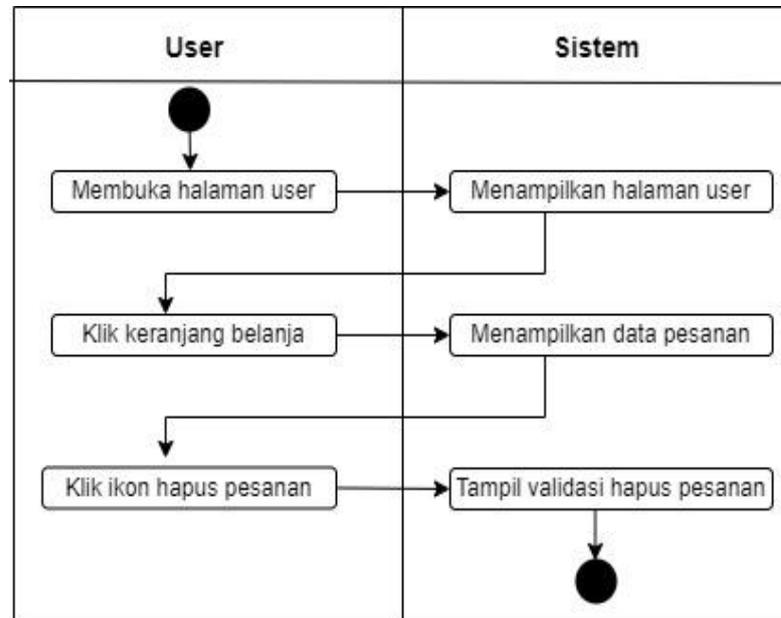
**Gambar 4. 3** *Activity Diagram Lihat Data Produk*

3) *Activity diagram pemesanan*



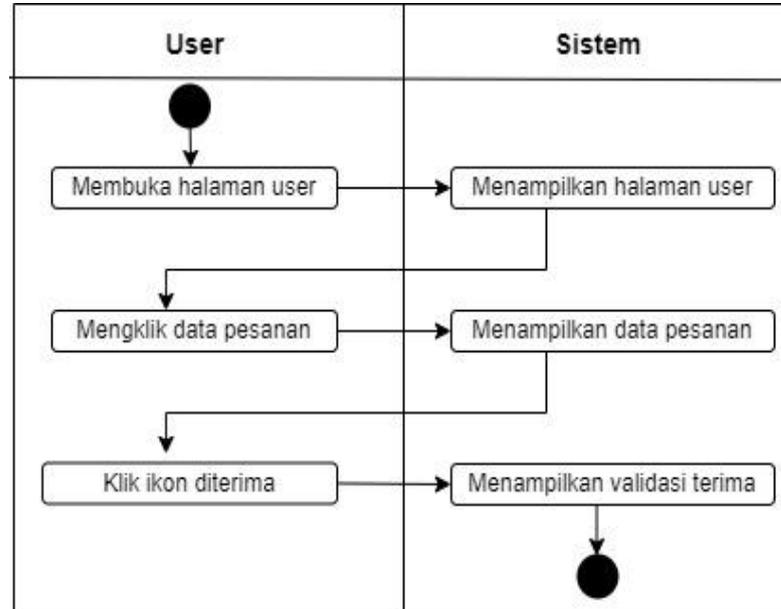
**Gambar 4. 4** *Activity Diagram Pemesanan*

4) *Activity diagram hapus data pesanan*



**Gambar 4. 5** *Activity Diagram Hapus Data Pesanan*

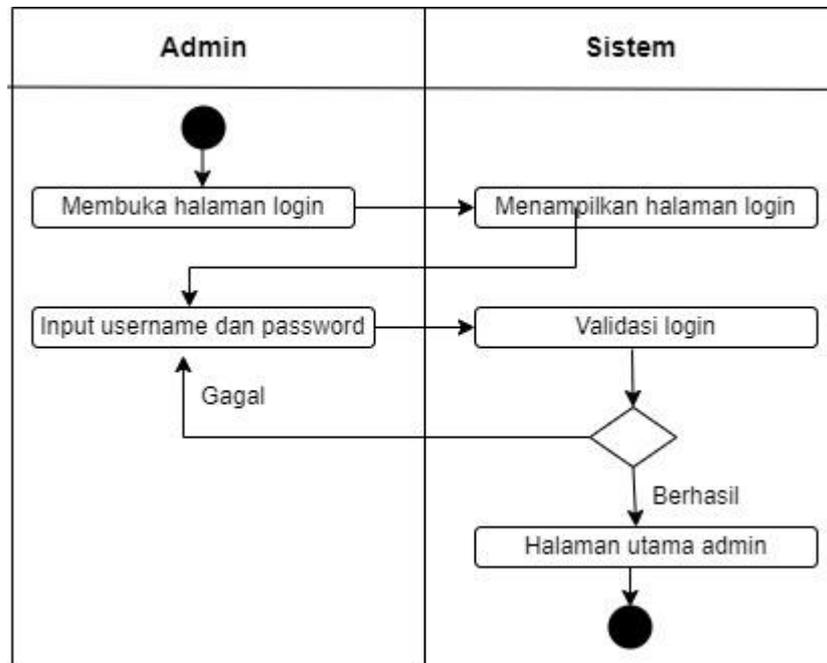
5) *Activity diagram konfirmasi pesanan*



**Gambar 4. 6** *Activity Diagram Konfirmasi Pesanan*

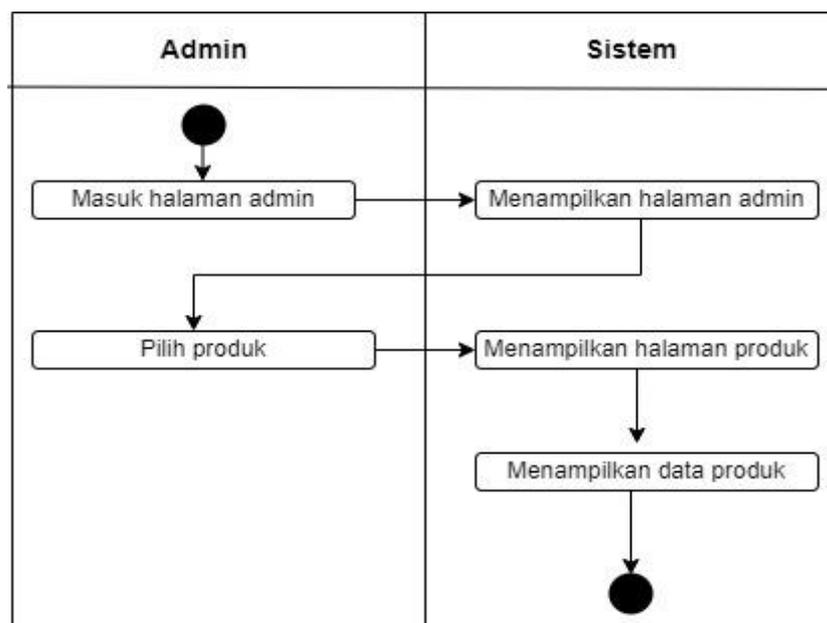
**b. Activity Diagram Admin**

**1) Activity diagram login admin**



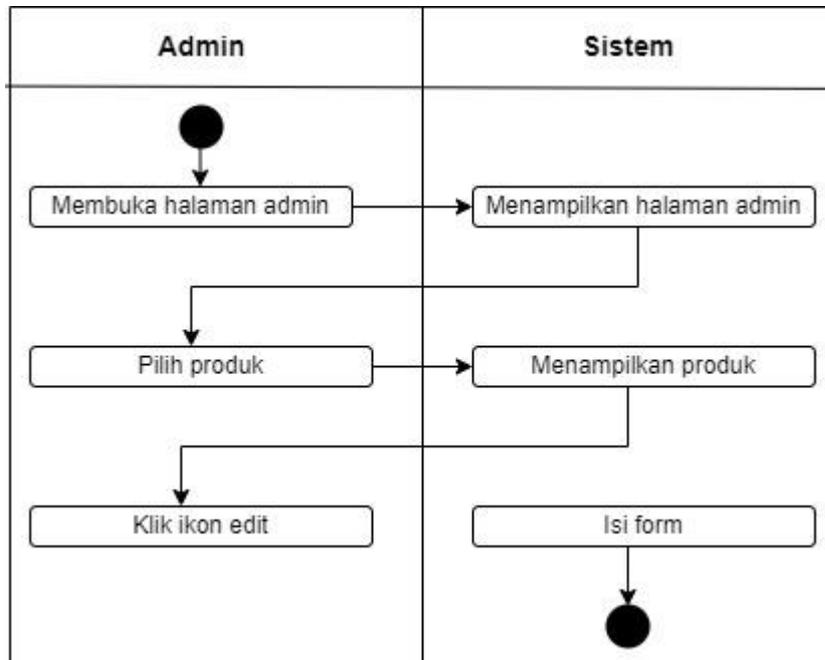
**Gambar 4. 7** Activity Diagram Login Admin

**2) Activity diagram lihat data produk**



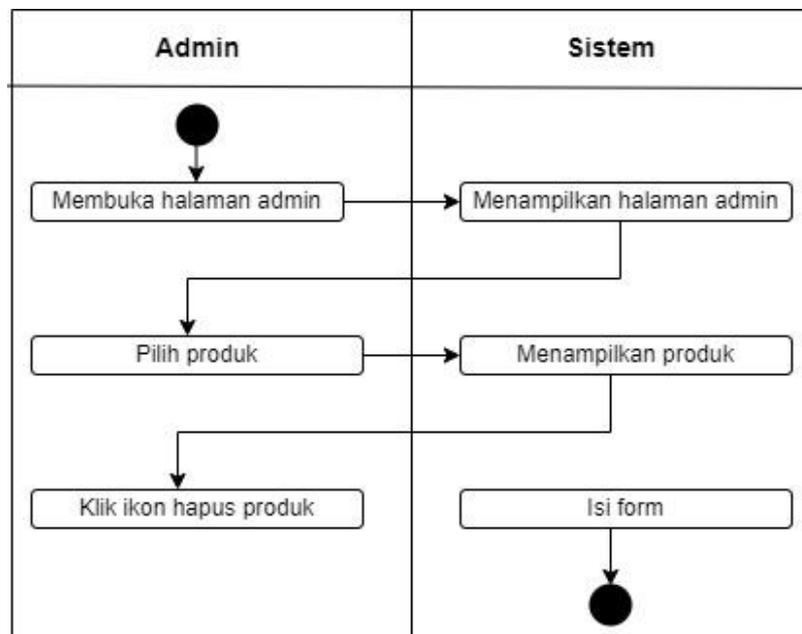
**Gambar 4. 8** Activity Diagram Lihat Data Produk

3) *Activity diagram edit data produk*



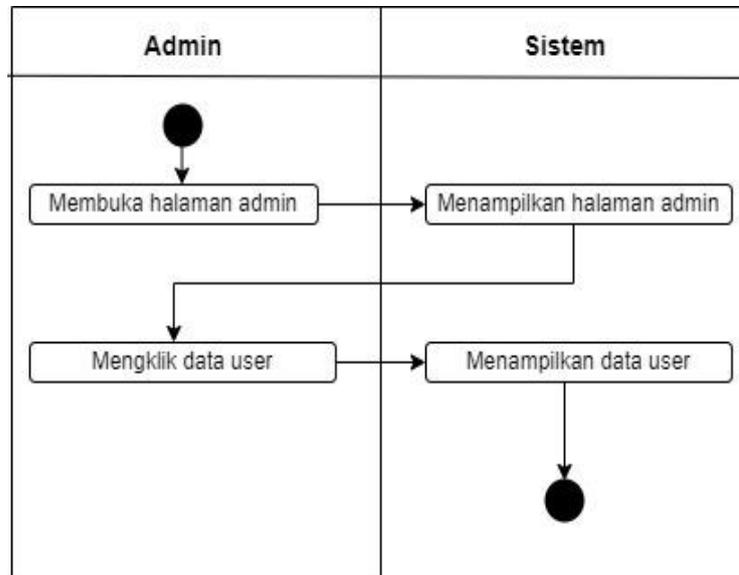
**Gambar 4. 9** *Activity Diagram Edit Data Produk*

4) *Activity diagram hapus data produk*



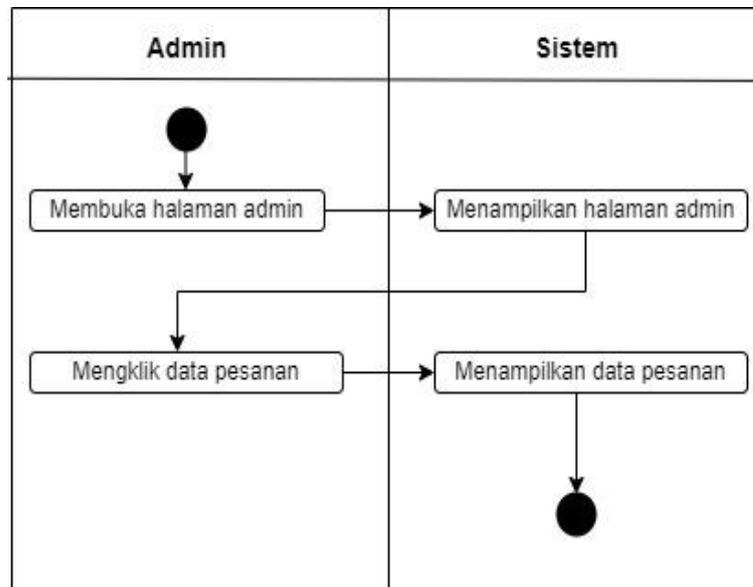
**Gambar 4. 10** *Activity Diagram Hapus Data Produk*

5) *Activity diagram lihat data user*



**Gambar 4. 11** *Activity Diagram Lihat Data User*

6) *Activity diagram lihat data pesanan*



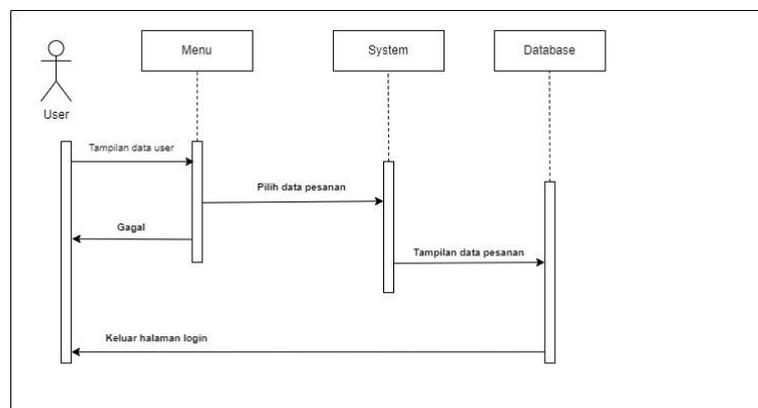
**Gambar 4. 12** *Activity Diagram Lihat Data Pesanan*

### 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.

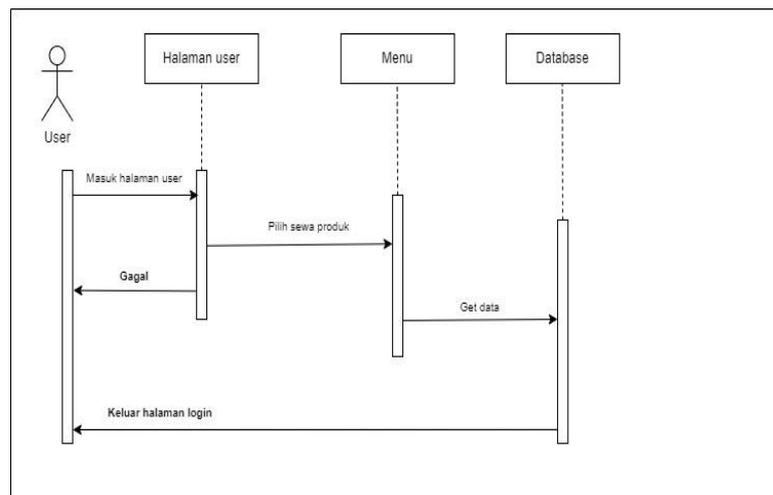
#### a. Sequence diagram user

##### 1) Sequence diagram login



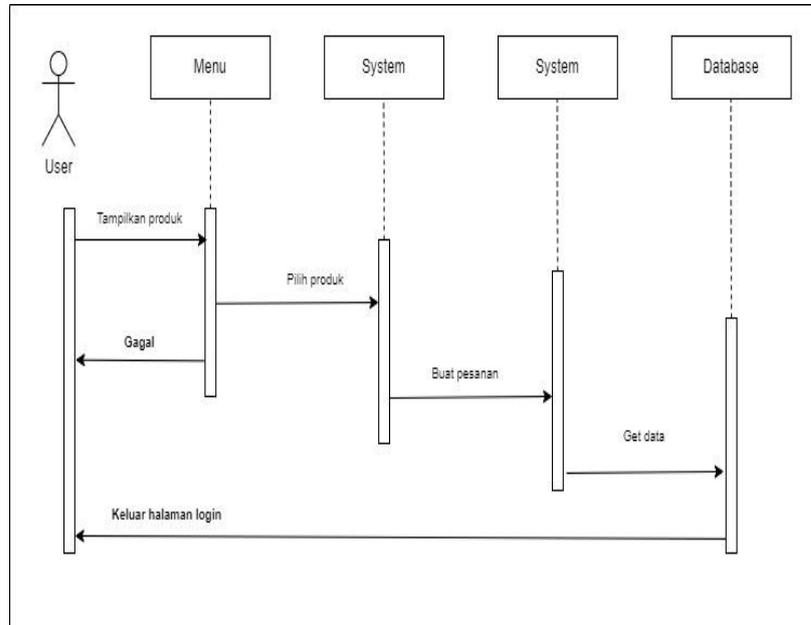
Gambar 4. 13 Sequence Diagram Login

##### 2) Sequence diagram lihat data produk



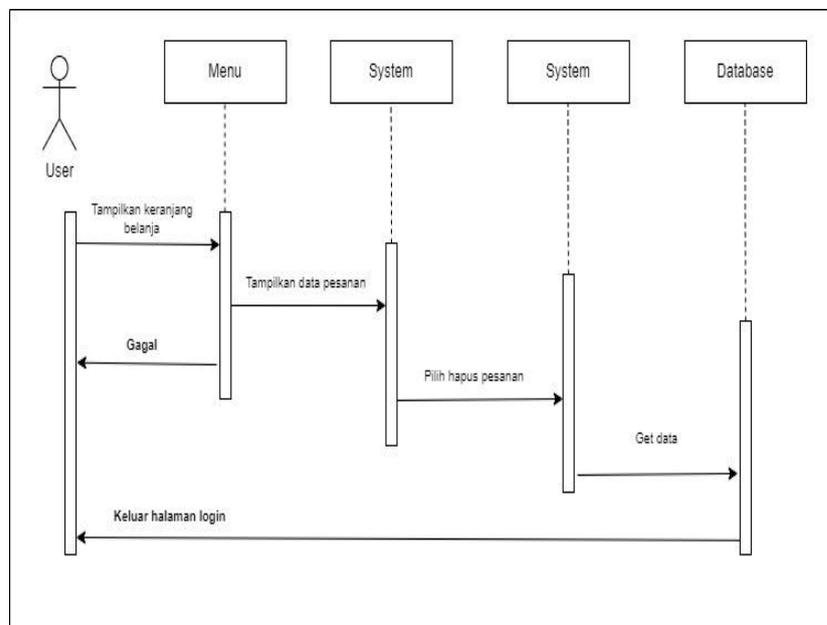
Gambar 4. 14 Sequence Diagram Lihat Data Produk

### 3) *Sequence diagram pemesanan*



**Gambar 4. 15** *Sequence Diagram Pemesanan*

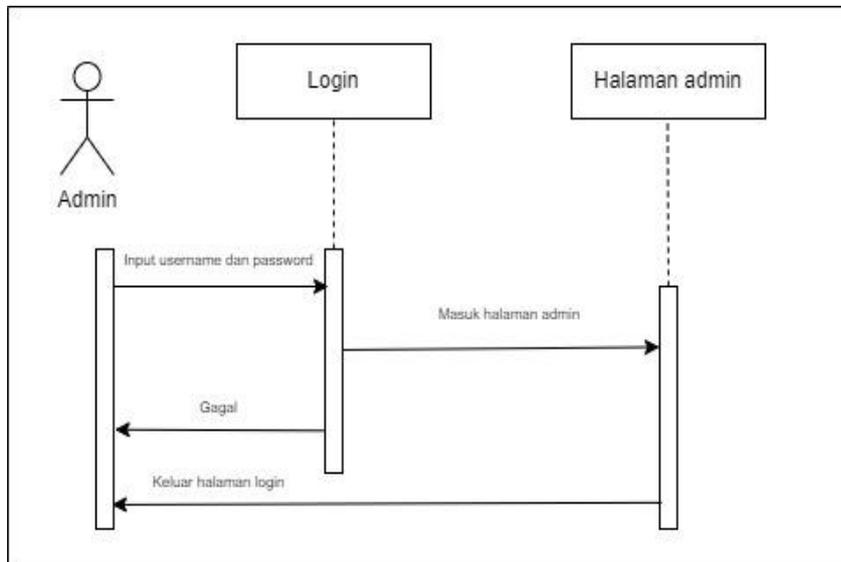
### 4) *Sequence diagram hapus data pesanan*



**Gambar 4. 16** *Sequence Diagram Hapus Data Pesanan*

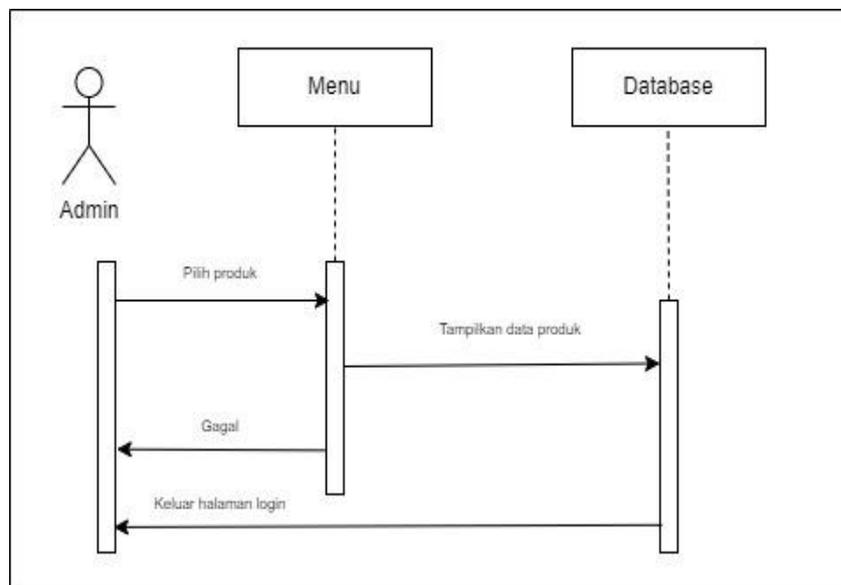
**b. Sequence diagram Admin**

**1) Sequence diagram login**



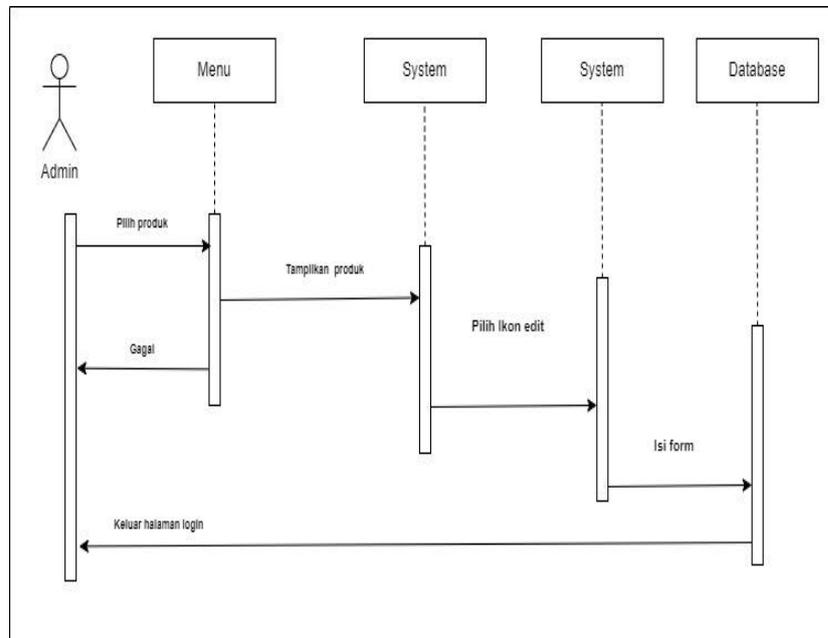
**Gambar 4. 17** Sequence Diagram Login

**2) Sequence diagram lihat data produk**



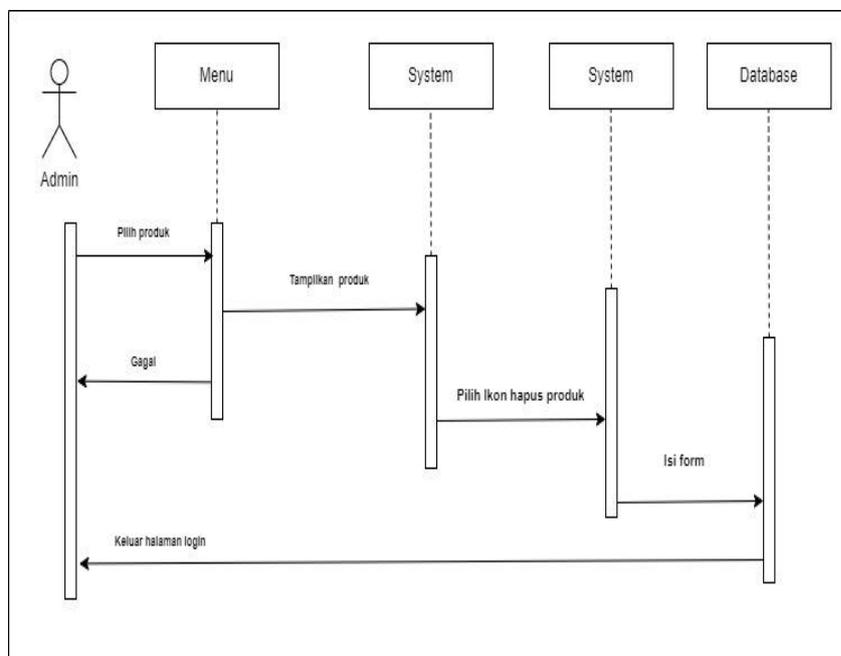
**Gambar 4. 18** Sequence Diagram Lihat Data Produk

### 3) Sequence diagram edit data produk



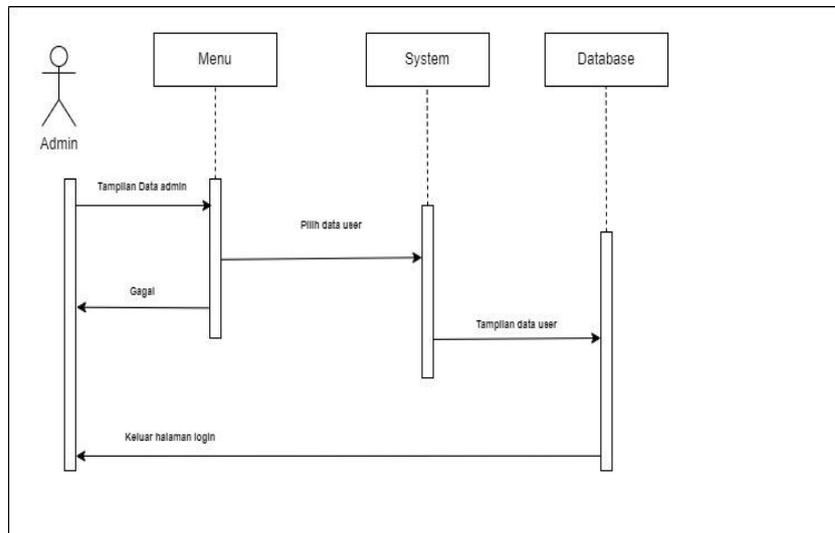
**Gambar 4. 19** Sequence Diagram Edit Data Produk

### 4) Sequence diagram hapus data produk



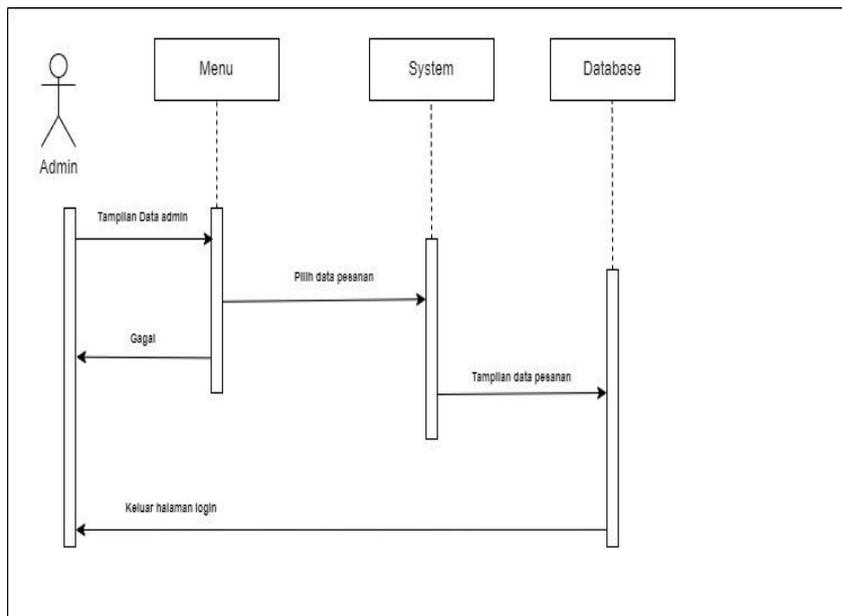
**Gambar 4. 20** Sequence Diagram Hapus Data Produk

5) *Sequence diagram lihat data user*



**Gambar 4. 21** *Sequence Diagram Lihat Data User*

6) *Sequence diagram lihat data pesanan*



**Gambar 4. 22** *Sequence Diagram Lihat Data Pesanan*

### C. Kamus Data

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
tb_kategori	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
tb_keranjang	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
tb_pelanggan	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
tb_produk	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	10	InnoDB	utf8mb4_general_ci	16.0 KB	-
tb_transaksi	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
5 tabel	Jumlah	17	InnoDB	utf8mb4_general_ci	80.0 KB	0 B

**Gambar 4. 23 Database Mysql**

Berikut ini adalah kamus data dari 5 tabel dalam database yang digunakan didalam gambar 4.23 *database mysql* :

**Tabel 4. 5 Tb\_Kategori**

No.	Nama	Jenis	Ukuran	Ket.
1.	Kategori	int	11	PrimaryKey
2.	NamaKategori	varchar	50	

**Tabel 4. 6 Tb\_Keranjang**

No.	Nama	Jenis	Ukuran	Ket.
1.	IdKeranjang	int	11	PrimaryKey
2.	Email	varchar	100	
3.	NoHP	varchar	13	
4.	IdKategori	int	11	
5.	IdProduk	int	11	
6.	Status	varchar	1	

**Tabel 4. 7** Tb\_Pelanggan

No.	Nama	Jenis	Ukuran	Ket.
1.	Email	varchar	100	PrimaryKey
2.	NoHP	varchar	13	
3.	Nama	varchar	50	
4.	Password	varchar	50	
5.	Alamat	text		
6.	OTPNumber	varchar	6	
7.	StatusVerifikasi	varchar	1	
8.	TglRegistrasi	varchar	19	

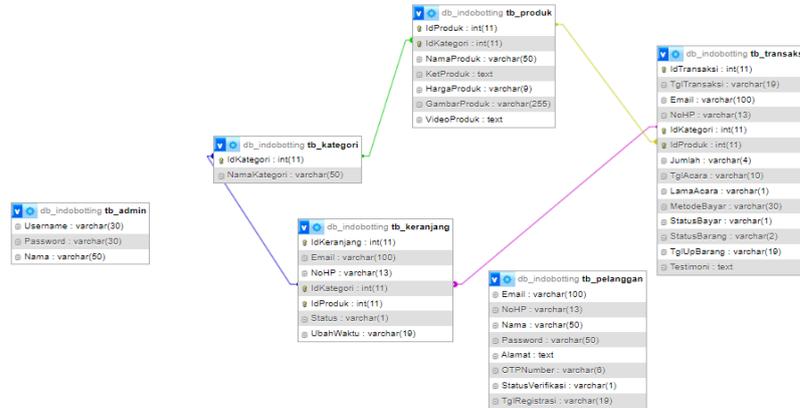
**Tabel 4. 8** Tb\_Produk

No.	Nama	Jenis	Ukuran	Ket.
1.	IdProduk	int	11	PrimaryKey
2.	IdKategori	int	11	
3.	NamaProduk	varchar	50	
4.	KetProduk	text		
5.	HargaProduk	varchar	9	
6.	GambarProduk	varchar	255	
7.	VideoProduk	text		

**Tabel 4. 9** Tb\_Transaksi

<b>No.</b>	<b>Nama</b>	<b>Jenis</b>	<b>Ukuran</b>	<b>Ket.</b>
1.	IdTransaksi	varchar	12	PrimaryKey
2.	TglTransaksi	varchar	19	
3.	Email	varchar	100	
4.	NoHP	varchar	13	
5.	IdKategori	int	11	
6.	IdProduk	int	11	
7.	Jumlah	varchar	4	
8.	TglAcara	varchar	10	
9.	LamaAcara	varchar	1	
10.	MetodeBayar	varchar	30	
11.	StatusBayar	varchar	1	
12.	StatusBarang	varchar	2	
13.	Testimoni	text		

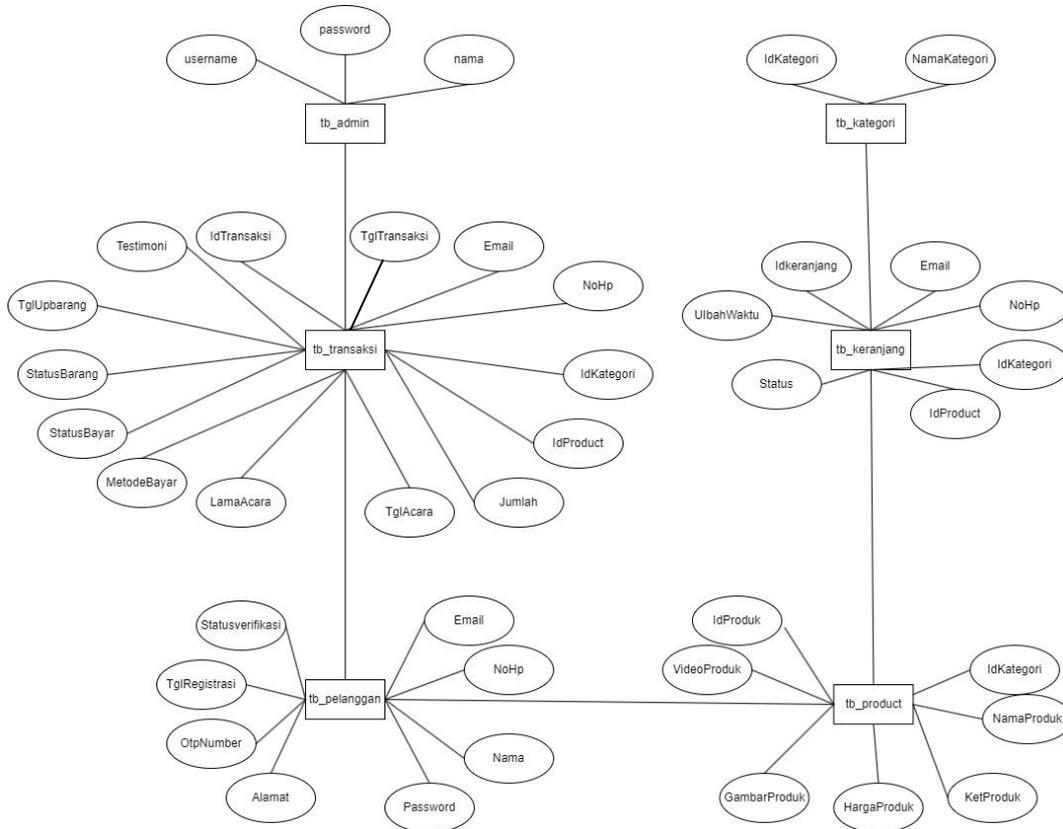
#### D. Relasi Tabel Database



**Gambar 4. 24** Relasi Database

pada gambar 4.24 dijelaskan bagaimana relasi tabel pada database, dimana pada database terdiri dari 6 tabel, yaitu tabel kategori, tabel admin, tabel produk, tabel keranjang, tabel pelanggan dan tabel transaksi. Tabel kategori terdiri dari 2 data yaitu id kategori (sebagai primery key) dan nama kategori. Tabel admin terdiri dari 3 data yaitu username, password dan nama. Tabel produk terdiri dari 7 data yaitu id produk (sebagai primery key), id kategori (sebagai primery key), nama produk, keterangan produk, harga produk, gambar produk dan video produk. Tabel keranjang terdiri dari 7 data yaitu id keranjang (sebagai primery key), email, no hp, id kategori (sebagai primery key), id produk (sebagai primery key), status dan ubah waktu. Tabel pelanggan terdiri dari 8 data yaitu email, no hp, nama, password, alamat, otp number, status verifikasi dan tanggal register. Tabel transaksi terdiri dari 14 data yaitu id transaksi, tgl transaksi, email, no hp, id kategori (sebagai primery key), id produk (sebagai primery key), jumlah, tgl acara, lama acara, metode bayar, status bayar, status barang, tgl up barang dan testimony. Relasi antar tabel berada pada kolom id kategori pada tabel produk dengan kolom id kategori pada tabel kategori dan kolom id kategori pada tabel keranjang dan kolom id kategori pada tabel transaksi, serta pada kolom id produk pada tabel transaksi dengan kolom id produk pada tabel produk.

### E. Diagram ERD



**Gambar 4. 25** Diagram ERD

Pada gambar 4.25 dijelaskan bagaimana diagram erd pada database, dimana pada tabel kategori berisi *field* id kategori, nama kategori. Dan pada tabel keranjang berisi *field* email, no hp, id kategori, id produk, status, ubah waktu, id keranjang. Dan pada tabel produk berisi *field* id kategori, nama produk, keterangan produk, harga produk, gambar produk, video produk, id produk. Dan pada tabel pelanggan berisi *field* email, no hp, nama, password, alamat, otp number, tgl registrasi, status verifikasi. Dan pada tabel transaksi berisi *field* tgl transaksi, email, no hp, id kategori, id produk, jumlah, tgl acara, lama acara, metode bayar, status bayar, status barang, tgl up barang, testimoni, id transaksi. Dan pada tabel admin berisi *field* *username*, *password*, *nama*.

## F. Implementasi *Interface*

### 1. *User*

#### a. Tampilan halaman register user

Merupakan tampilan langkah pertama dalam membuat akun yang terdiri dari nama, email, nomor hp, password dan konfirmasi password kemudian daftar.



16.58 [social icons] 2.00 100% 24G [signal] [battery]

# Wedding Organizer

Lengkapi data dibawah ini !

  
  
  
 [eye icon]  
 [eye icon]  

**Gambar 4. 26** Halaman *Register User*

**b. Tampilan halaman login user**

Merupakan tampilan login setelah membuat akun yang dimana pengguna memasukkan email dan password yang sesuai.



16.58 [Facebook] [Instagram] [WhatsApp] 2.00 KB/S Vo LTE 4G 48

# Wedding Organizer

Masuk menggunakan Email / No. HP

**Masuk**

Belum punya akun ? [Klik Disini](#)

[Menu] [Home] [Back]

**Gambar 4. 27** Halaman *Login User*

### c. Tampilan daftar menu utama

Merupakan tampilan daftar menu dalam wedding organizer yang terdiri dari paketan, baju pengantin, baju bodo, makeup, dan tenda/dekorasi.



**Gambar 4. 28** Menu Utama

#### d. Tampilan menu paketan

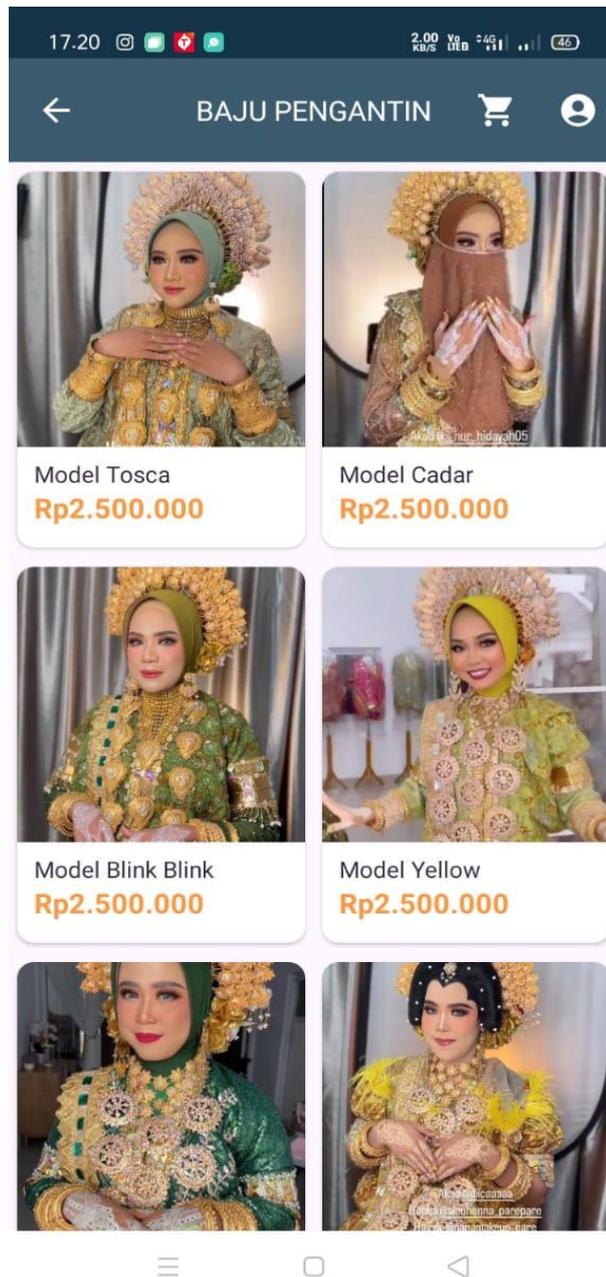
Merupakan tampilan menu jika user ingin memilih paketan yang terdiri dari paket 1 dan paket 2.



**Gambar 4. 29** Menu Paketan

**e. Tampilan menu baju pengantin**

Merupakan tampilan menu yang berisi beberapa baju pengantin yang akan dipilih user.



**Gambar 4. 30** Menu Baju Pengantin

### f. Tampilan menu baju bodo

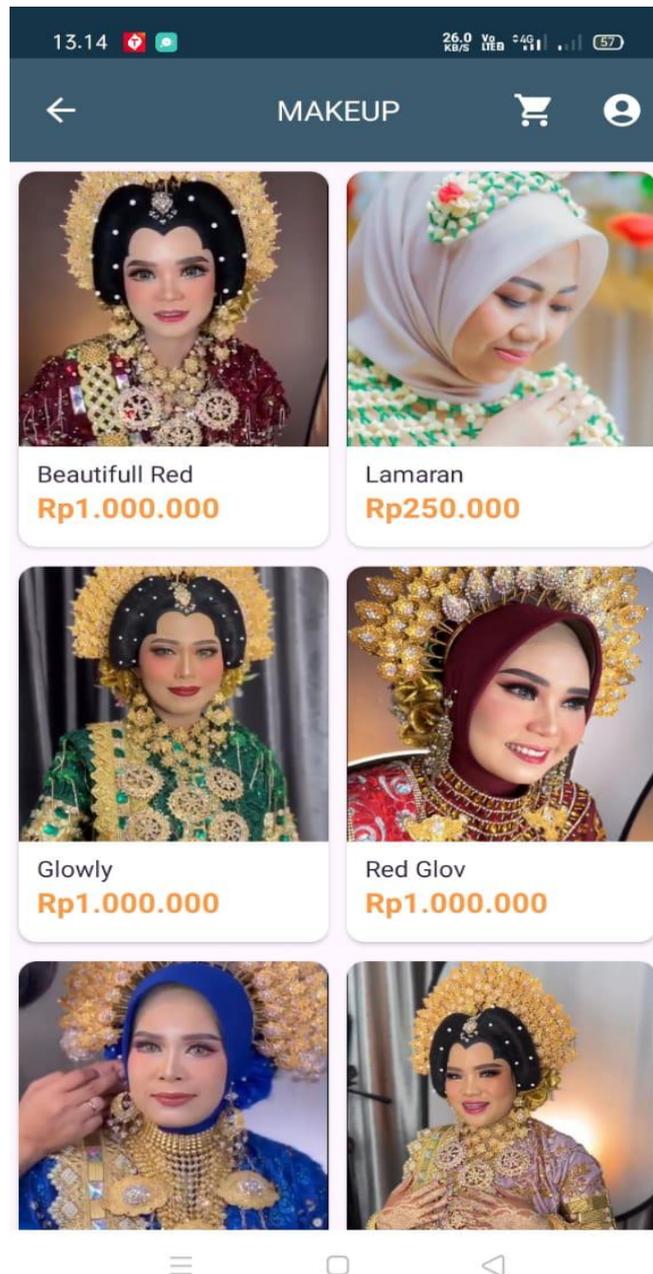
Merupakan tampilan menu yang berisi beberapa baju bodo yang akan dipilih user.



**Gambar 4. 31** Menu Baju Bodo

**g. Tampilan menu make up**

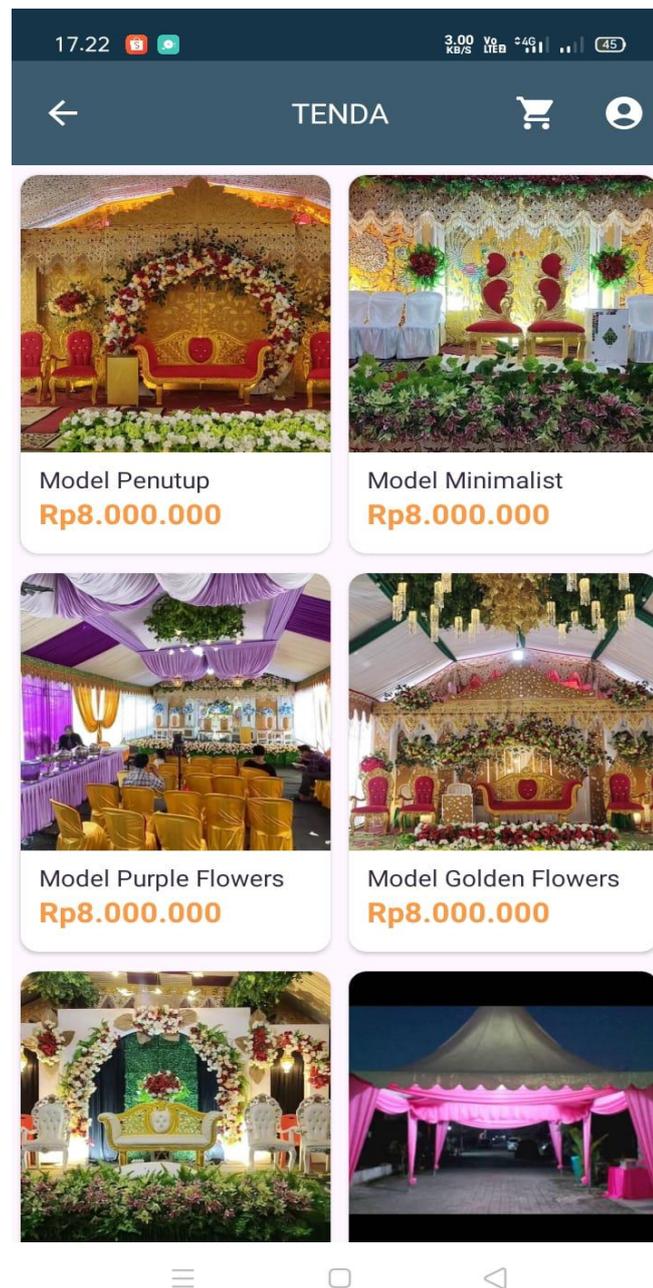
Merupakan tampilan menu yang berisi beberapa makeup yang akan dipilih user.



**Gambar 4. 32** Menu *Makeup*

### h. Tampilan menu tenda/dekorasi

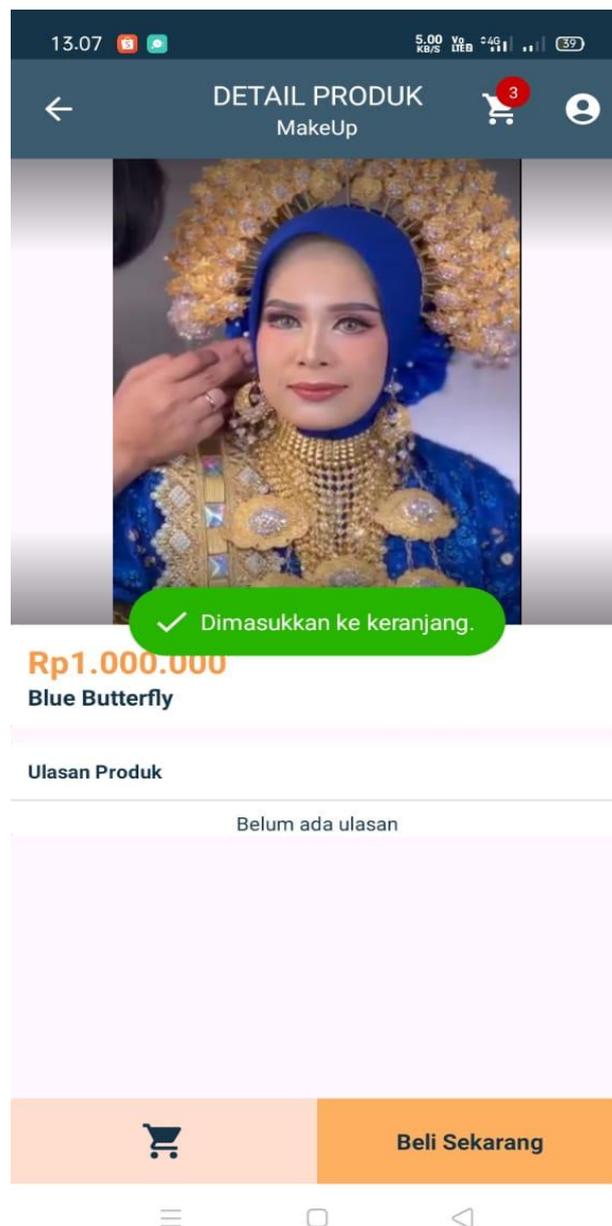
Merupakan tampilan menu yang berisi beberapa tenda/dekorasi yang akan dipilih user.



Gambar 4. 33 Menu Tenda/Dekorasi

**i. Tampilan menu buat pesanan**

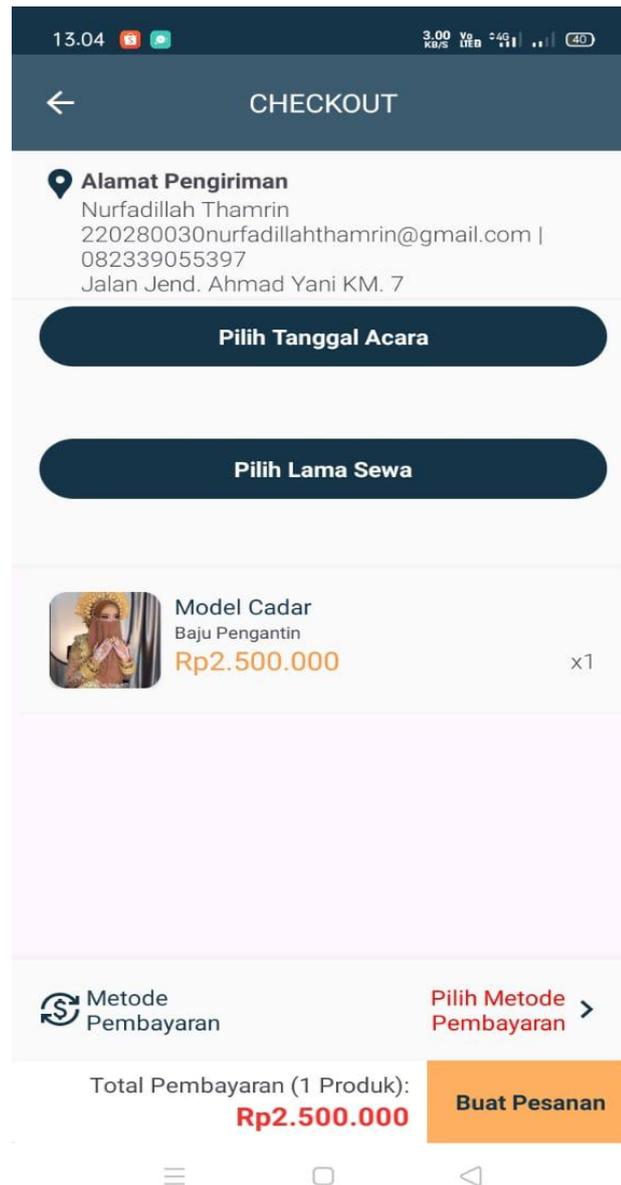
Merupakan tampilan item yang dipilih user untuk dimasukkan ke keranjang atau bisa langsung akses fitur beli sekarang.



**Gambar 4. 34** Menu Buat Pesanan

### j. Tampilan menu checkout

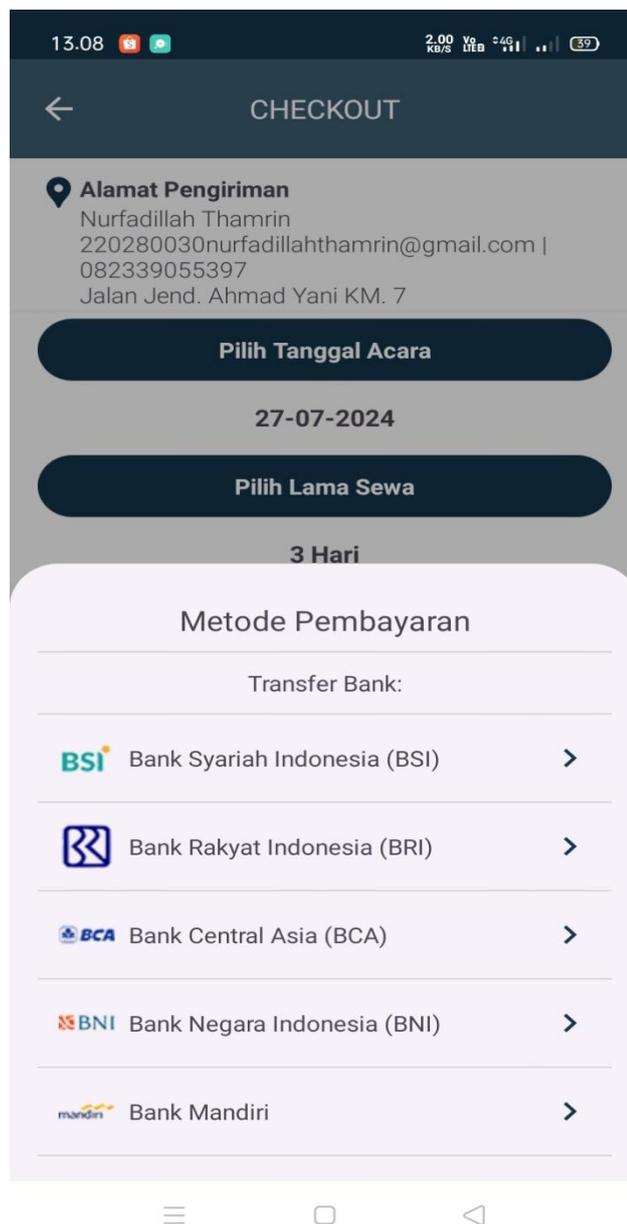
Merupakan tampilan checkout yang dimana user bisa memilih tanggal acara dan lama acara.



Gambar 4. 35 Menu *Checkout*

### k. Tampilan menu metode pembayaran

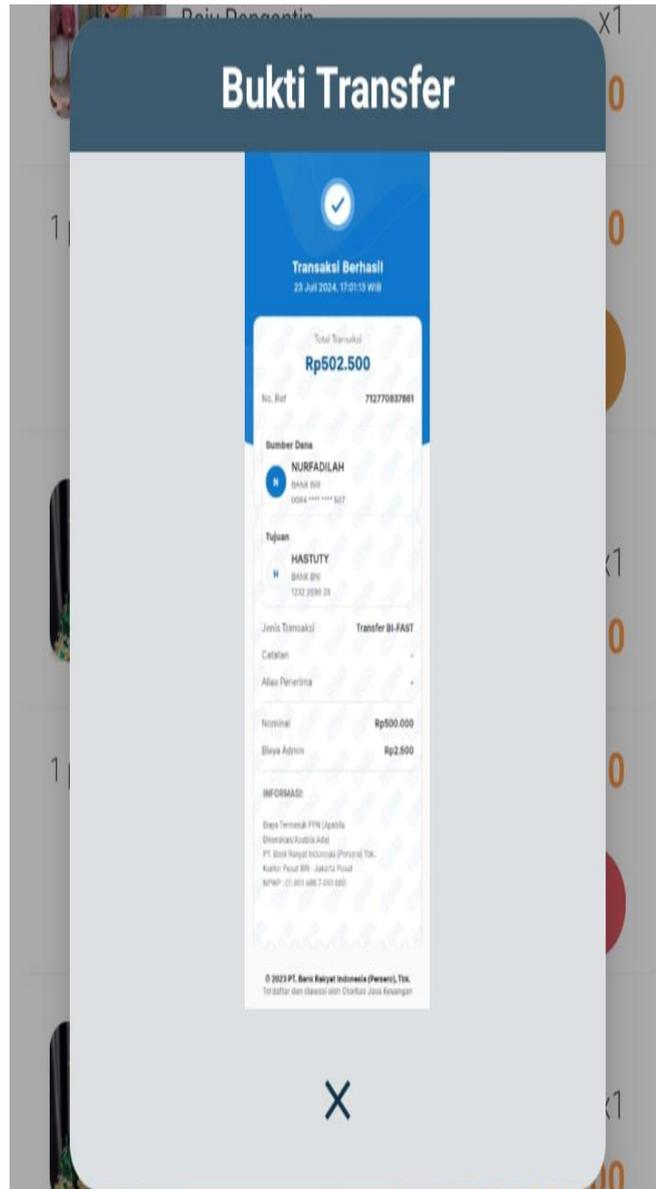
Merupakan tampilan apabila ingin memilih salah satu metode pembayaran yang terdiri dari beberapa bank.



**Gambar 4. 36** Metode Pembayaran

### I. Tampilan menu bukti pembayaran

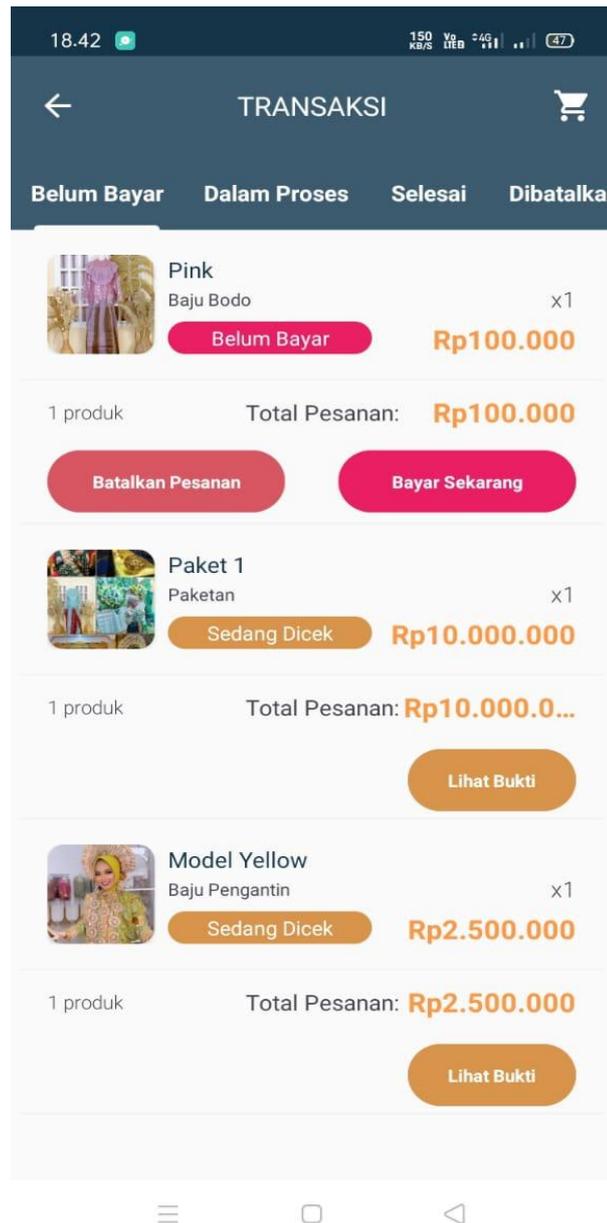
Merupakan tampilan menu yang akan digunakan user untuk mengupload bukti pembayaran yang terdiri dari nama pengirim dan yang menerima, serta jenis transaksi, nominal dan biaya admin.



Gambar 4. 37 Menu Bukti Bayar

### m. Tampilan menu transaksi belum bayar

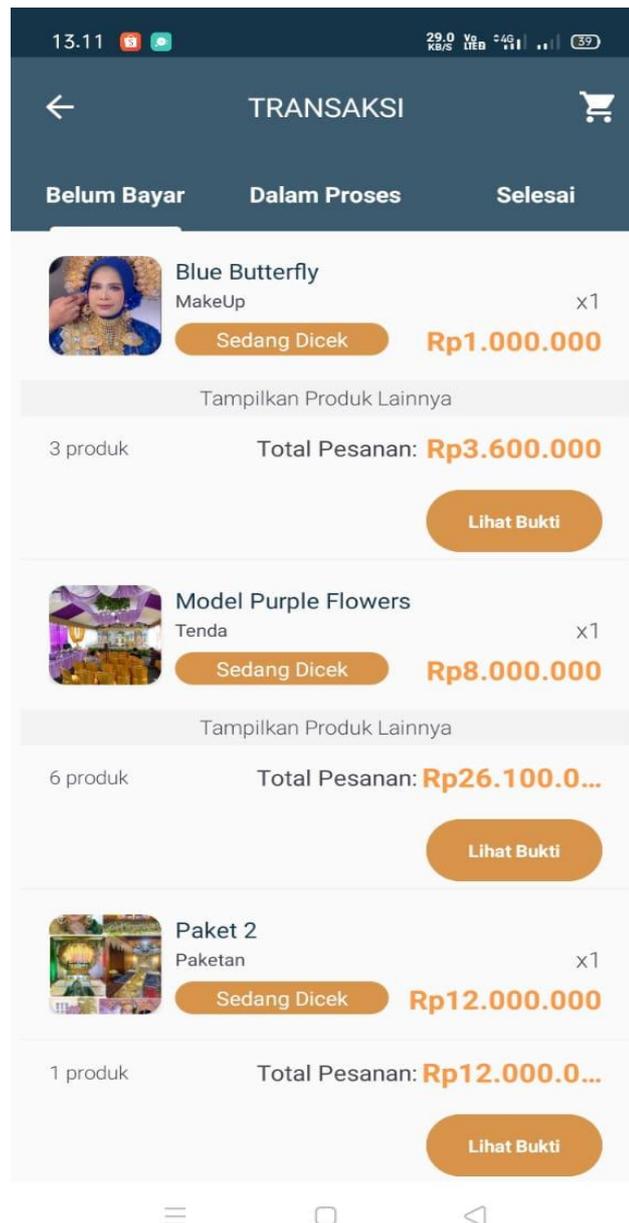
Merupakan tampilan item yang sudah di pesan namun belum dibayar dan terdapat form batalkan pesanan apabila tidak jadi memesan.



**Gambar 4. 38** Menu Belum Bayar

#### n. Tampilan menu transaksi sedang di cek

Merupakan tampilan item apabila admin sedang mengecek pesanan.



**Gambar 4. 39** Transaksi sedang dicek

**o. Tampilan menu transaksi sedang di proses**

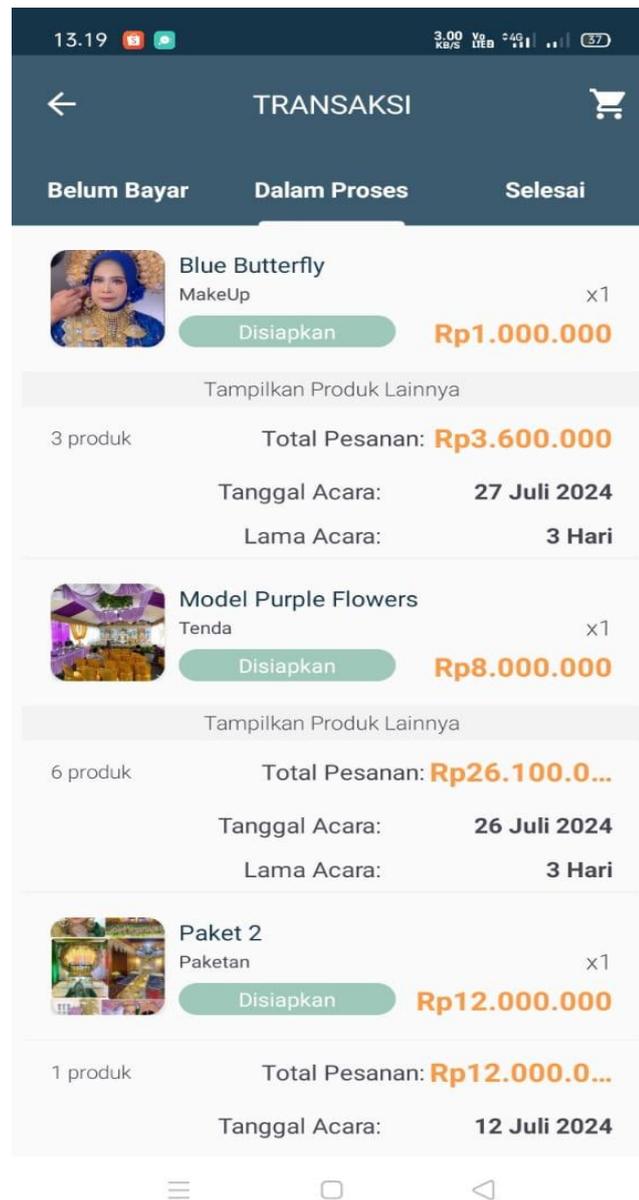
Merupakan tampilan item apabila pesanan sedang diproses atau dikemas untuk dikirim ke lokasi.



**Gambar 4. 40** Sedang diproses

**p. Tampilan menu transaksi disiapkan**

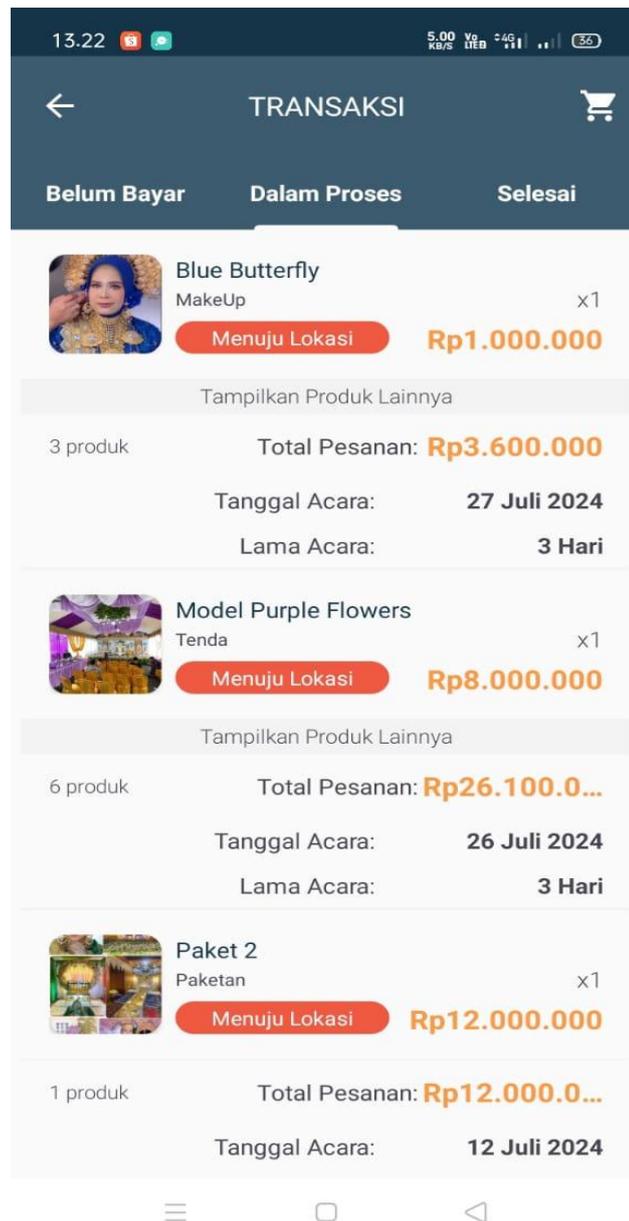
Merupakan tampilan item apabila pesanan sudah dikemas dan akan dikirim.



**Gambar 4. 41** Disiapkan

**q. Tampilan menu transaksi menuju lokasi**

Merupakan tampilan item apabila pesanan sedang dalam perjalanan menuju ke lokasi.



**Gambar 4. 42 Menuju Lokasi**

**r. Tampilan menu transaksi diterima/dipasang**

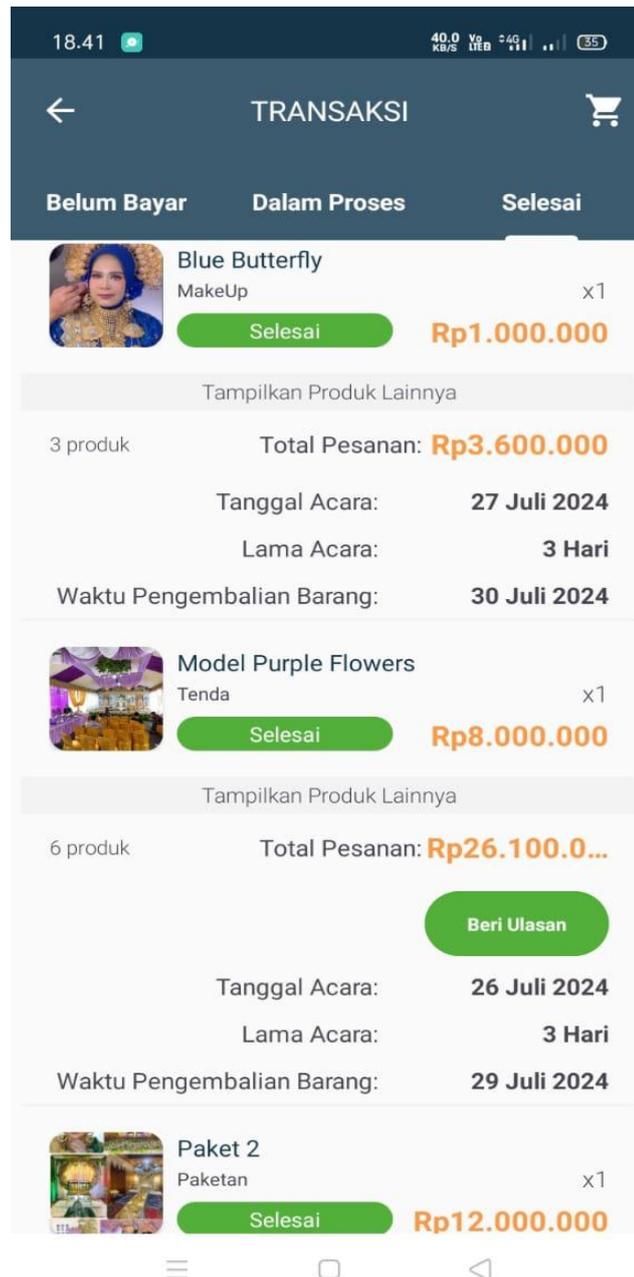
Merupakan tampilan item apabila pesanan sudah sampai dan pesanan sedang dipasang.



**Gambar 4. 43** Pesanan diterima dan dipasang

### s. Tampilan menu transaksi selesai

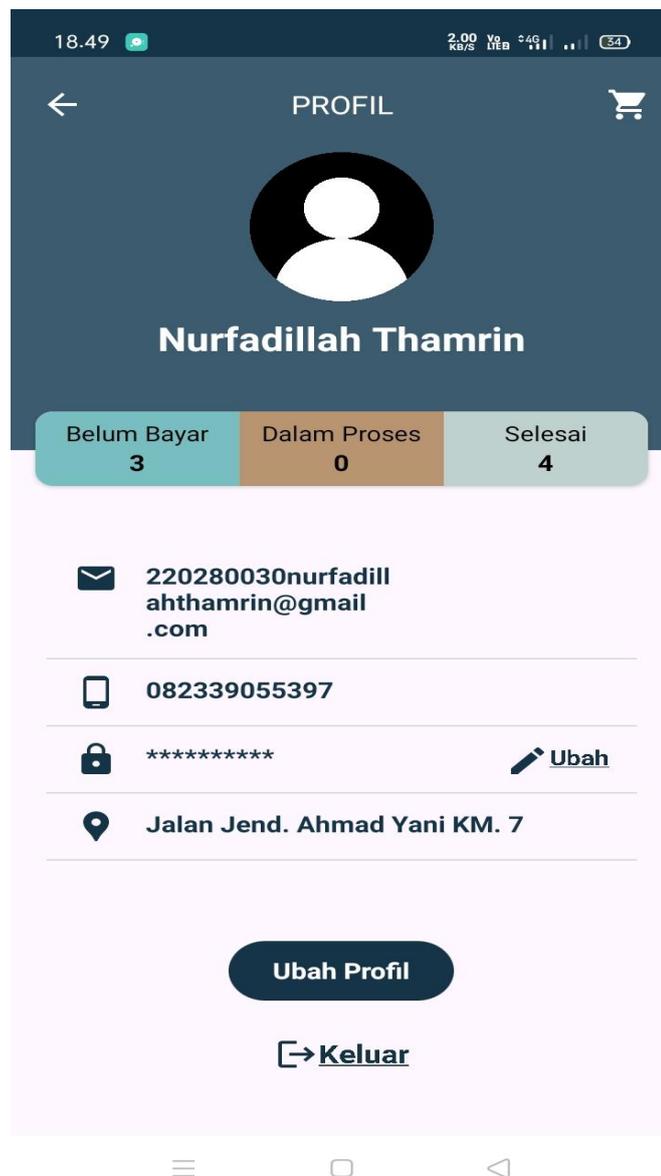
Merupakan tampilan item apabila pesanan sudah selesai di pakai dan sudah dikembalikan.



Gambar 4. 44 Pesanan Selesai

**t. Tampilan menu ubah profil**

Merupakan tampilan profil user apabila ingin mengubah profil yang terdiri dari email, no telepon, password, dan alamat serta dapat melihat informasi pesanan yang terdiri dari belum bayar, dalam proses, dan selesai.

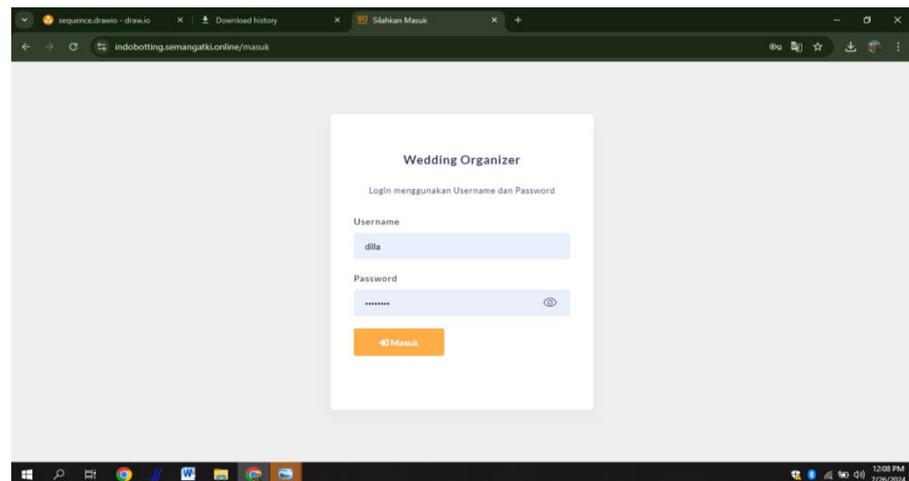


**Gambar 4. 45** Ubah Profil

## 2. Admin

### a. Tampilan halaman *login* admin

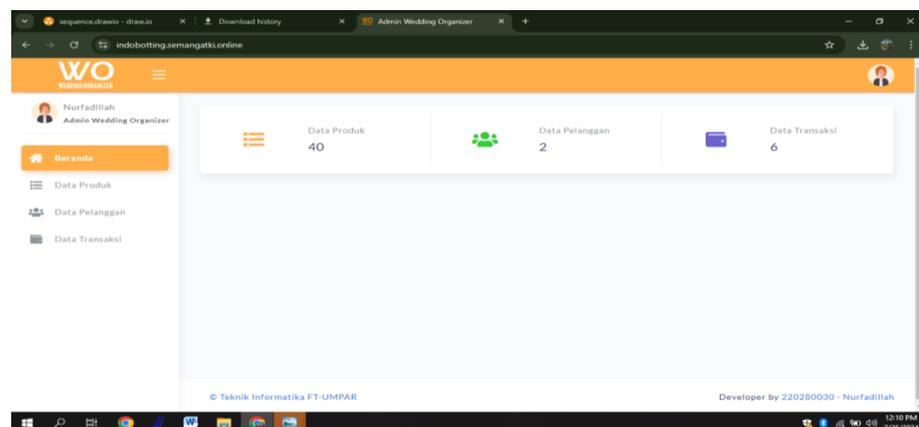
Merupakan tampilan halaman *login* untuk admin agar bisa masuk yang terdiri dari username dan password.



Gambar 4. 46 Halaman *Login*

### b. Tampilan halaman beranda admin

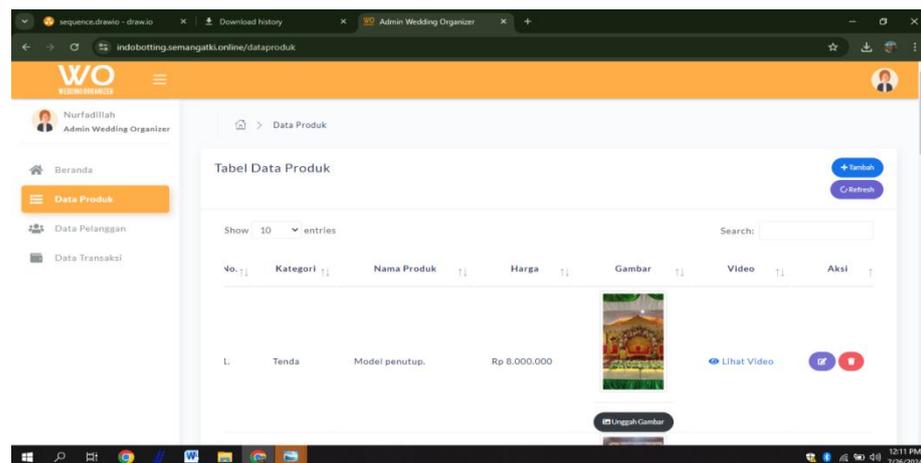
Merupakan tampilan halaman beranda/*dashboard* dari admin yang terdiri dari data produk, data pelanggan, dan data transaksi.



Gambar 4. 47 Halaman Beranda Admin

### c. Tampilan Halaman data produk

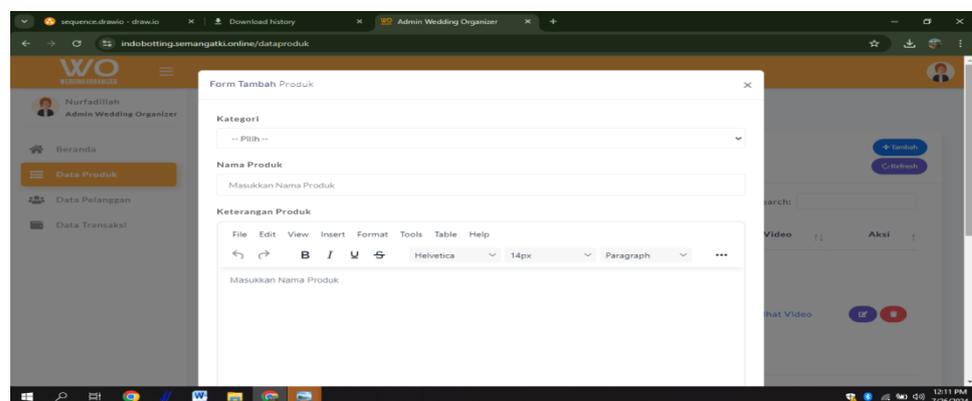
Merupakan tampilan halaman data produk admin, pada halaman ini terdapat kolom kategori, nama barang, harga, gambar, video, dan aksi dimana didalam kolom aksi terdapat dua ikon yaitu edit dan hapus.



**Gambar 4. 48** Halaman Data Produk

### d. Tampilan Halaman tambah produk

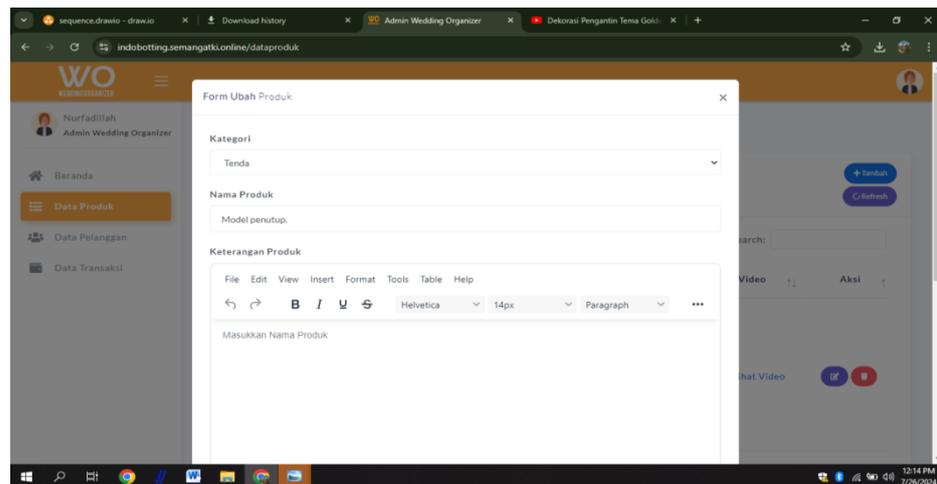
merupakan tampilan halaman admin apabila ingin menambah produk yang terdiri dari kategori, nama produk, keterangan produk, harga dan video.



**Gambar 4. 49** Halaman Tambah Produk

### e. Tampilan Halaman ikon ubah produk

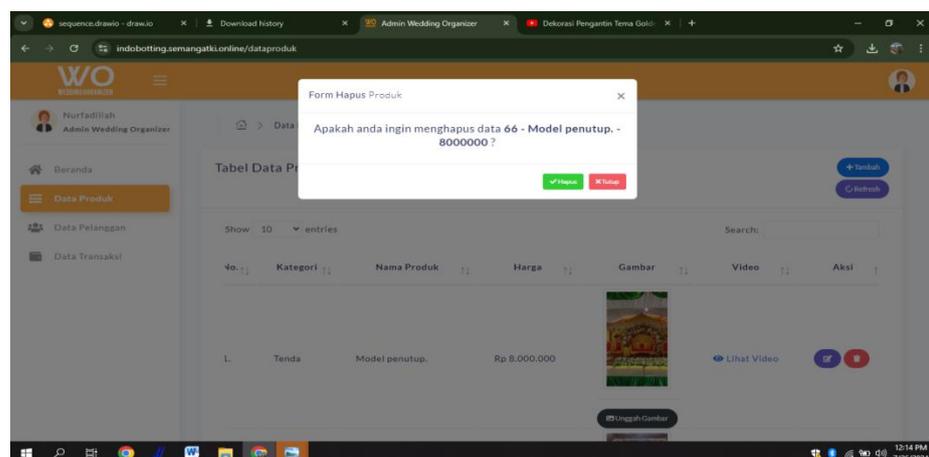
Merupakan tampilan halaman apabila admin mengubah produk yang terdiri dari kategori, nama produk, keterangan produk, harga dan video.



**Gambar 4. 50** Ikon Ubah Produk

### f. Tampilan halaman ikon hapus data produk

Merupakan tampilan halaman apabila admin akan menghapus produk, pada halaman ini akan muncul apakah anda ingin menghapus data jika ya maka klik hapus jika tidak maka klik tutup.



**Gambar 4. 51** Ikon Hapus Data Produk

### g. Tampilan halaman Data pengguna

Merupakan tampilan halaman data-data pengguna, yaitu pada halaman ini terdapat kolom status akun, email, no.hp/wa, nama, password, alamat dan tanggal registrasi.

No.	Status Akun	Email	No. HP/WA	Nama/WA	Pass
1.	Aktif	220280030nurfadillahthamrin@gmail.com	082339055397	Nurfadillah Thamrin	Dill...
2.	Aktif	220280097akmal@gmail.com	085258532600	Akmal	qwe
3.	Aktif	220280026yusriani@gmail.com	085340508590	Yusriani	anik
4.	Aktif	220280052sittifatimahazzahrad@gmail.com	085462335961	Fatimah	123
5.	Aktif	djihanfadillah03@gmail.com	08533364948	Jihan	jihan
6.	Aktif	220280010sittiaisyahabullah@gmail.com	087866863334		

Gambar 4. 52 Halaman Pengguna Sudah Daftar

### h. Tampilan halaman data transaksi

Merupakan tampilan halaman data transaksi yang terdiri dari kolom id transaksi, tanggal transaksi, email, jumlah produk, quantity, total harga, metode bayar, acara, berlangsung selama, status bayar, status barang dan aksi serta informasi mengenai jalannya produk yang sudah user pesan dan sudah mengupload data bukti pembayaran.

No.	ID Transaksi	Tanggal Transaksi	Email	Jumlah Produk
1.	202407270009	2024-07-27 18:24:42	220280030nurfadillahthamrin@gmail.com	3
2.	202407270008	2024-07-27 18:21:21	220280010sittiasyahabullah@gmail.com	

Acara	Berlangsung Selama	Waktu Pengembalian	Status Bayar	Status Barang	Aksi
27 Juli 2024	3 Hari	29 Juli 2024	Sudah Bayar	Selesai	Print

**Gambar 4. 53** Halaman Data Transaksi

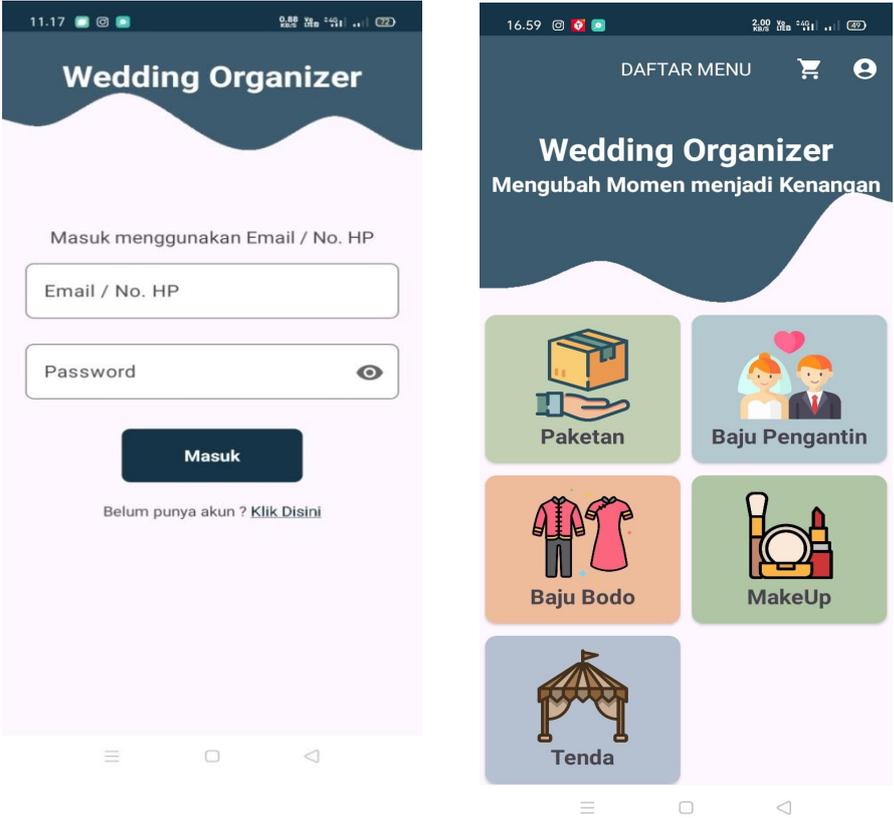
## G. Pengujian Sistem

### 1. Blackbox testing

#### a. User

##### a) Blackbox testing berhasil login

**Tabel 4.10** Blackbox Testing Berhasil Login

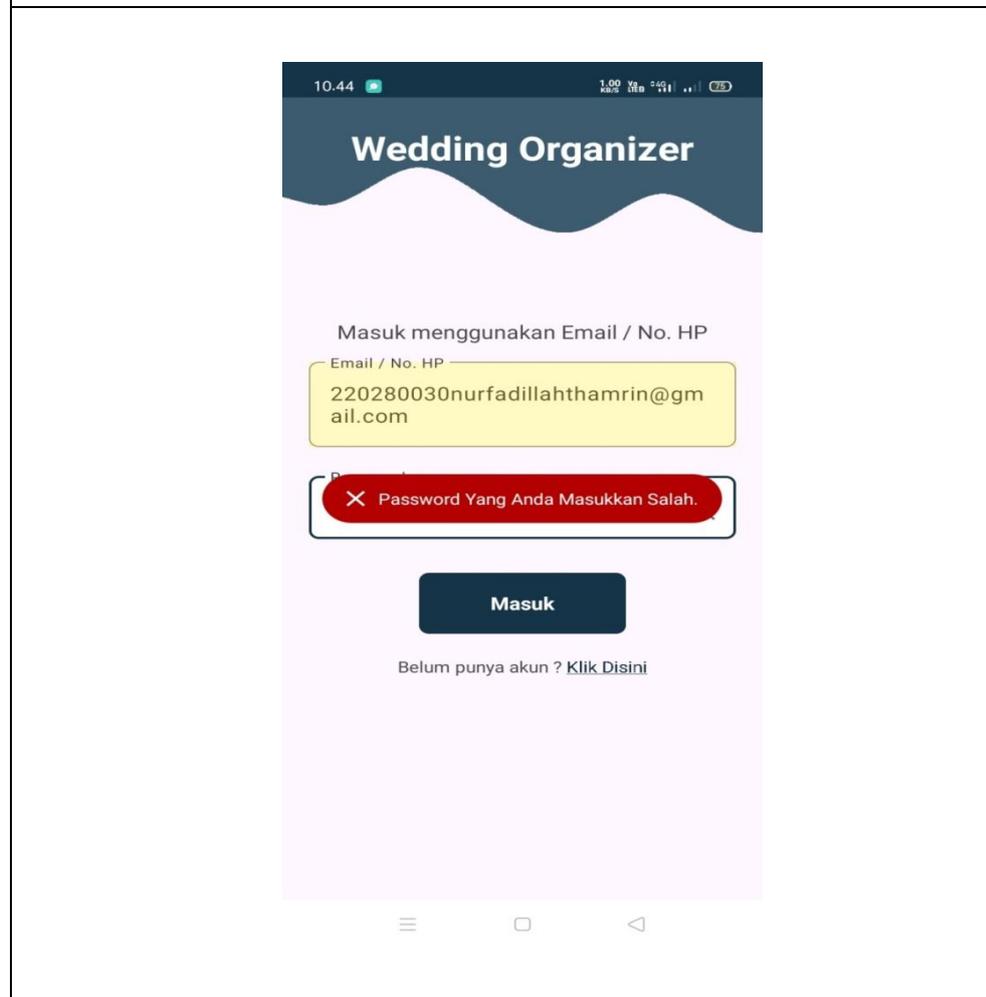
Tes Faktor	Hasil	Keterangan
User memasukkan <i>email</i> dan <i>password</i> yang benar	✓	Berhasil, ketika <i>email</i> atau <i>password</i> sesuai maka tampil halaman daftar menu
<b>Screenshot</b>		
		

**b) Blackbox testing kesalahan email dan password**

**Tabel 4. 11** *Blackbox Testing Kesalahan Email/Password*

Tes Faktor	Hasil	Keterangan
User memasukkan email/password yang salah	✓	Berhasil, sistem akan menampilkan pesan "password yang anda masukkan salah"

**Screenshot**



c) *Blackbox testing* tidak berhasil memasukkan pesanan**Tabel 4. 12** *Blackbox Testing* Tidak Berhasil Masukkan Pesanan

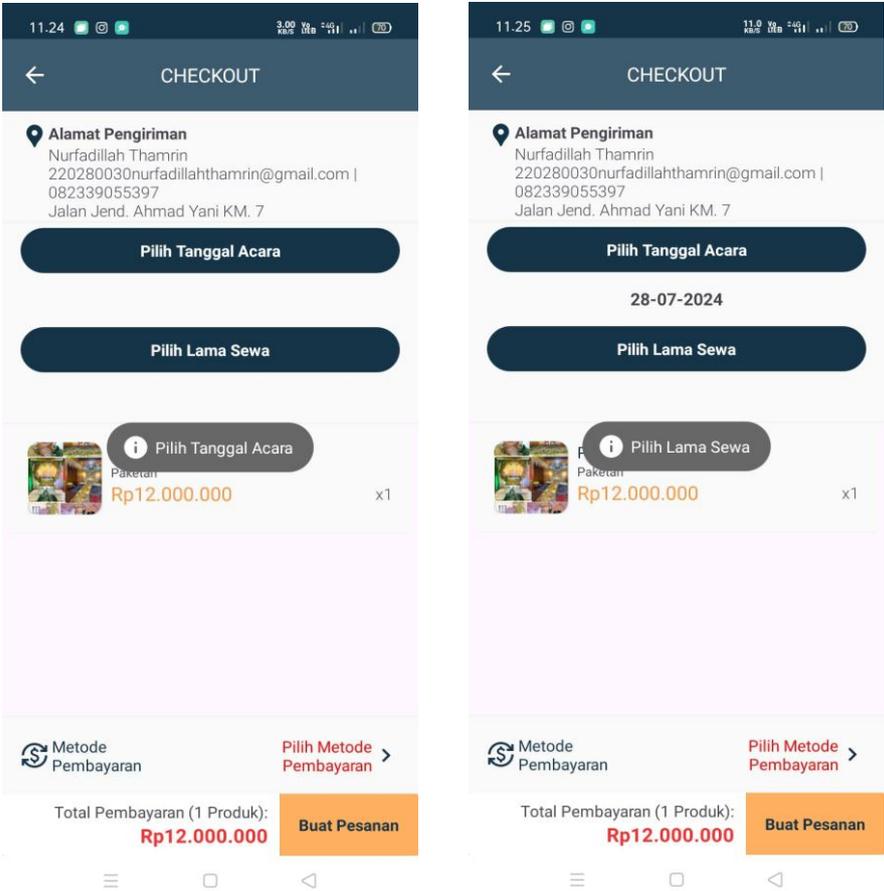
Tes Faktor	Hasil	Keterangan
User tidak memilih produk	✓	Berhasil, sistem akan menampilkan pesan “produk belum dipilih”
<i>Screenshot</i>		
		

d) *Blackbox testing* berhasil memasukkan pesanan**Tabel 4. 13** *Blackbox Testing* Berhasil Memasukkan Pesanan

Tes Faktor	Hasil	Keterangan
User memilih produk	✓	Berhasil, sistem akan menampilkan “produk pesanan telah dipilih ”

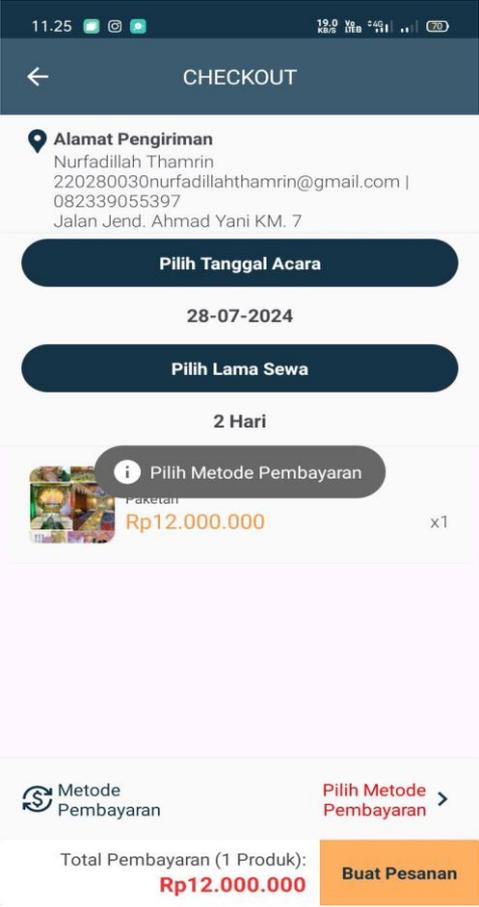
*Screenshot*

e) *Blackbox testing* tidak mengisi tanggal acara dan lama acara**Tabel 4. 14** *Blackbox Testing* Tidak Mengisi Tanggal dan Lama Acara

Tes Faktor	Hasil	Keterangan
User tidak mengisi tanggal acara dan lama acara	✓	Berhasil, sistem akan menampilkan “ pilih tanggal acara dan lama acara”
<i>Screenshot</i>		
		

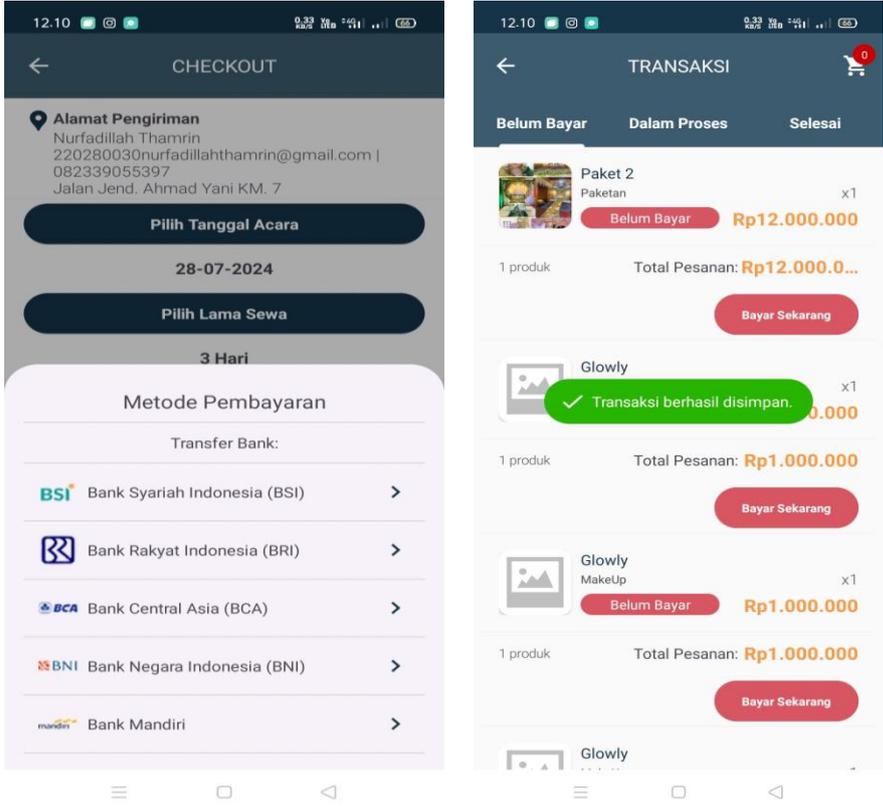
f) **Blackbox testing tidak memilih metode pembayaran**

**Tabel 4. 15** *Blackbox Testing Tidak Memilih Metode Pembayaran*

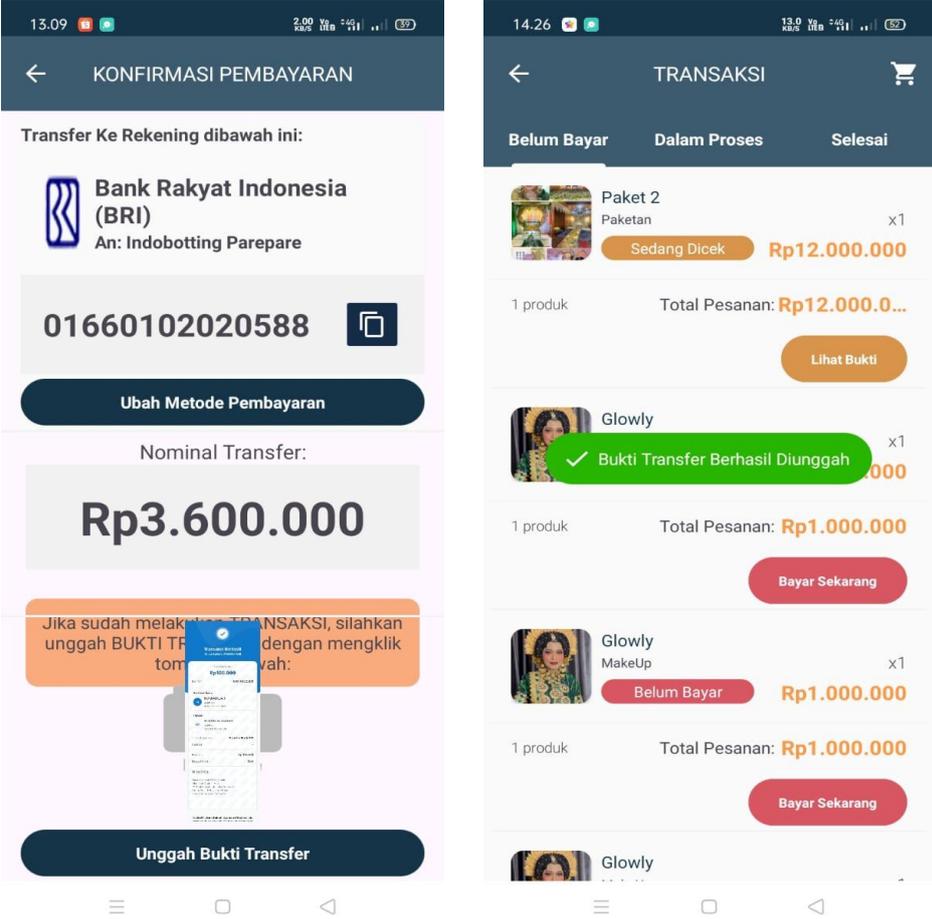
Tes Faktor	Hasil	Keterangan
User tidak memilih metode pembayaran	✓	Berhasil, sistem akan menampilkan “pilih metode pembayaran”
<b>Screenshot</b>		
		

**g) Blackbox testing berhasil memilih tanggal acara, lama acara dan metode pembayaran**

**Tabel 4. 16 Blackbox Testing Berhasil Memilih Tanggal Acara, Lama Acara dan Metode pembayaran**

Tes Faktor	Hasil	Keterangan
User memilih tanggal acara, lama sewa dan metode pembayaran	✓	Berhasil, sistem akan menampilkan “transaksi berhasil disimpan”
<i>Screenshot</i>		
		

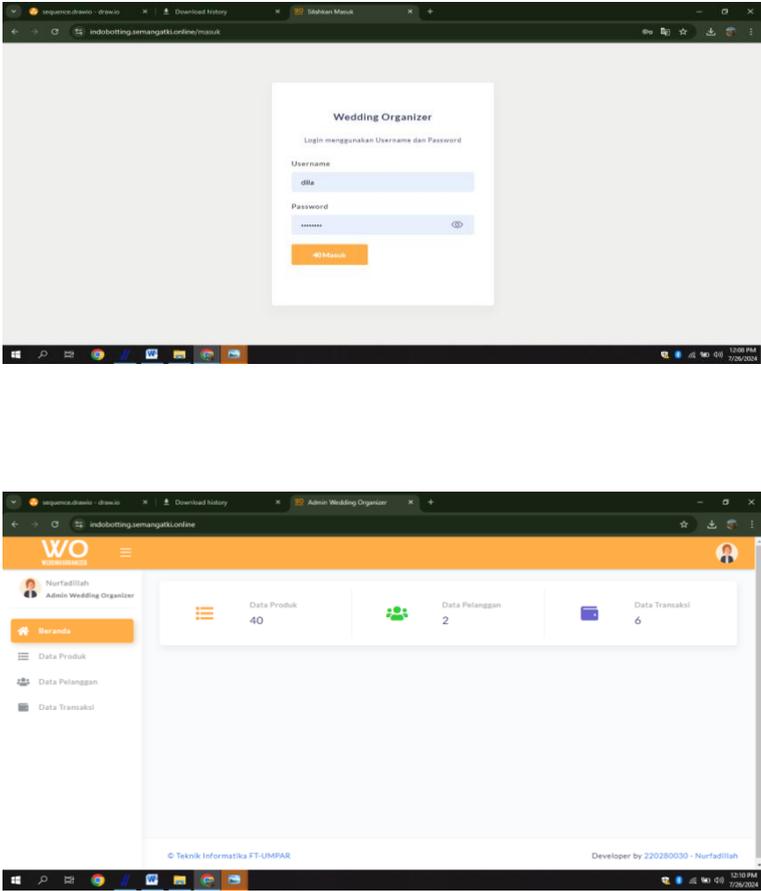
h) *Blackbox testing* tampilan transaksi sudah bayarTabel 4. 17 *Blackbox Testing* Tampilan Transaksi Sudah Bayar

Tes Faktor	Hasil	Keterangan
User memilih unggah bayar	✓	Berhasil, sistem akan menampilkan “bukti transfer sudah bayar”
<i>Screenshot</i>		
		

## b. Admin

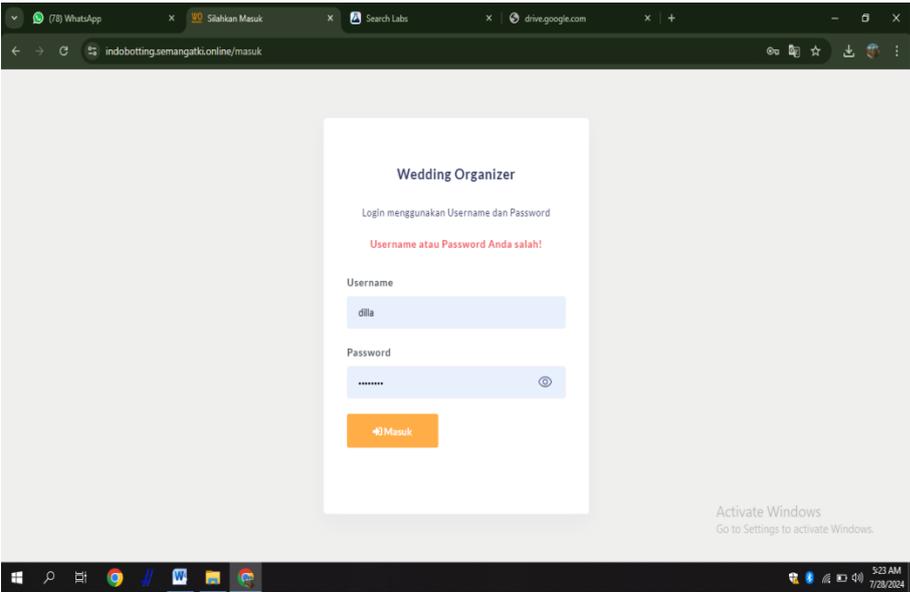
### a) *Blackbox testing login admin*

**Tabel 4. 18** *Blackbox Testing Login Admin*

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

b) **Blackbox testing** kesalahan username dan password

**Tabel 4. 19** *Blackbox Testing Kesalahan Username dan Password*

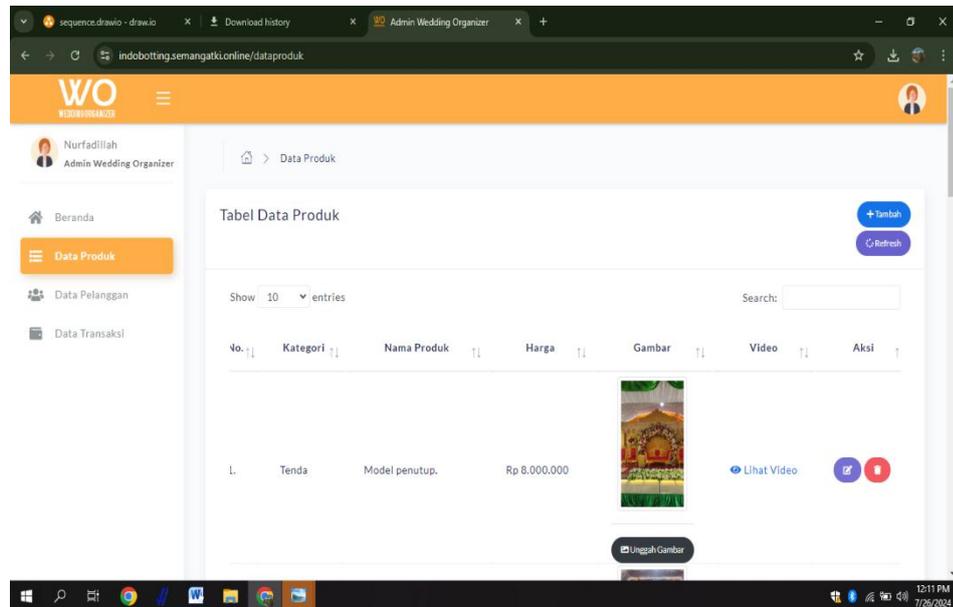
Tes Faktor	Hasil	Keterangan
Admin memasukkan <i>username</i> dan <i>password</i> yang tidak sesuai	✓	Berhasil, sistem akan menampilkan “ <i>username</i> atau <i>password</i> anda salah”
<b>Screenshot</b>		
		

c) *Blackbox testing* lihat data produk

**Tabel 4. 20** *Blackbox Testing* Lihat Data Produk

Tes Faktor	Hasil	Keterangan
Admin mengklik data produk	✓	Berhasil, sistem akan menampilkan “halaman data produk”

*Screenshot*

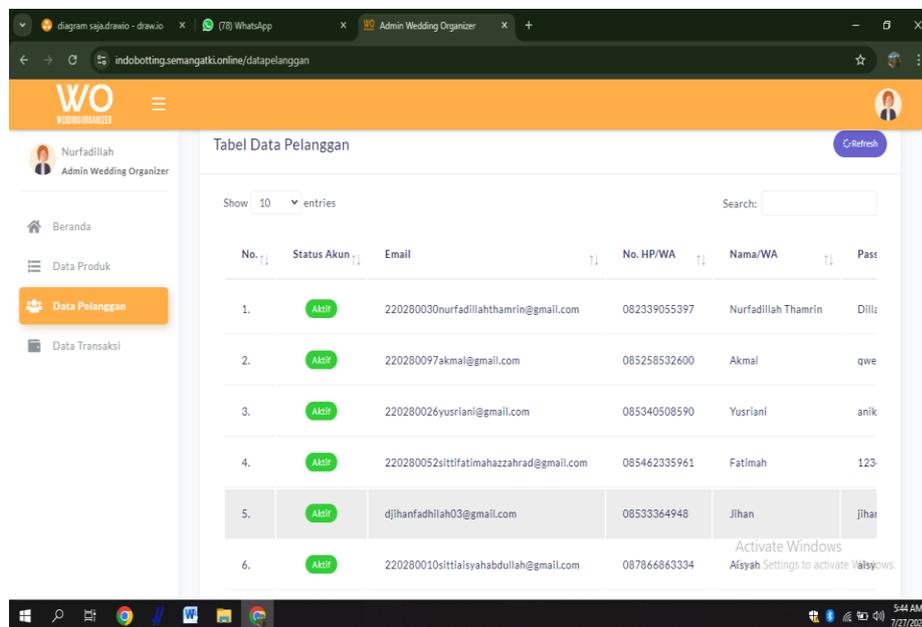


d) *Blackbox testing* lihat data pengguna

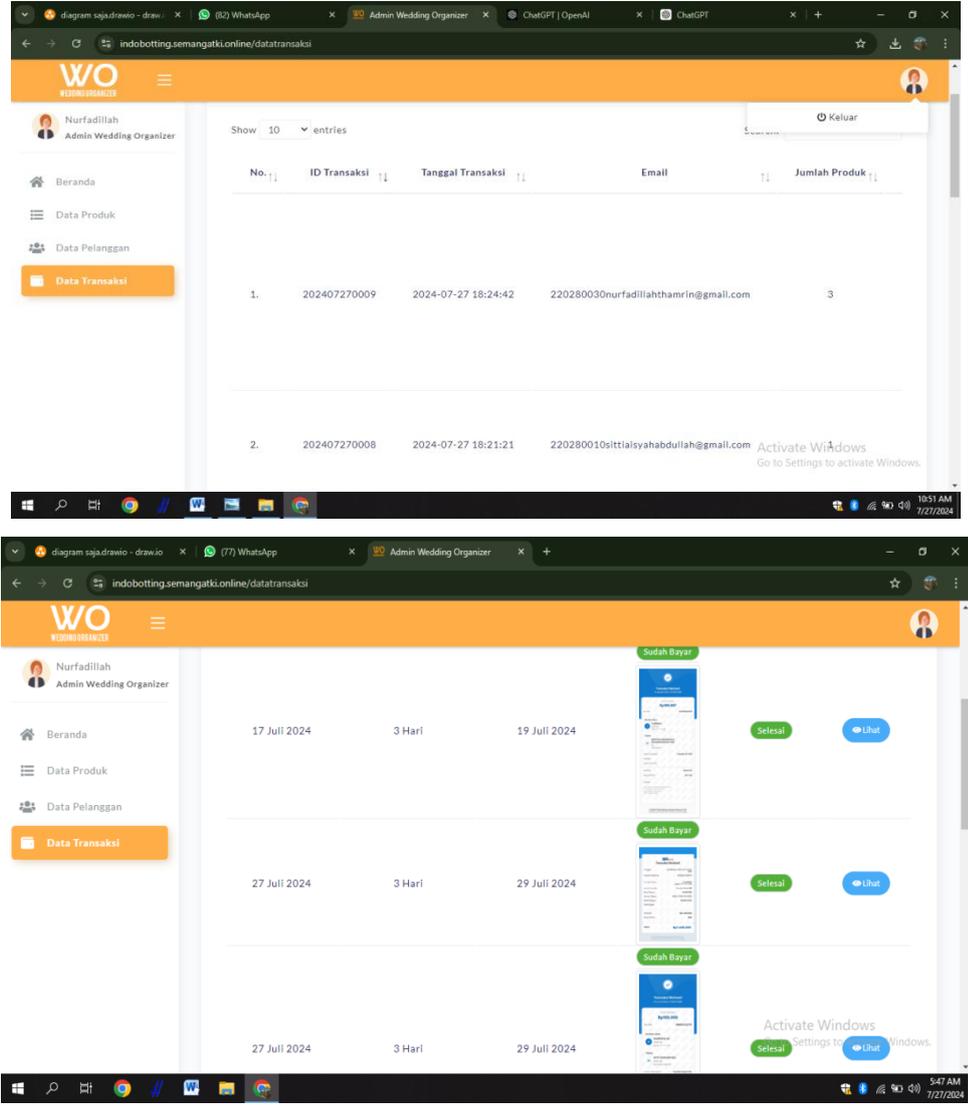
**Tabel 4. 21** *Blackbox Testing* Lihat Data Pengguna

Tes Faktor	Hasil	Keterangan
Admin mengklik data pelanggan	✓	Berhasil, sistem akan menampilkan “halaman data pengguna”

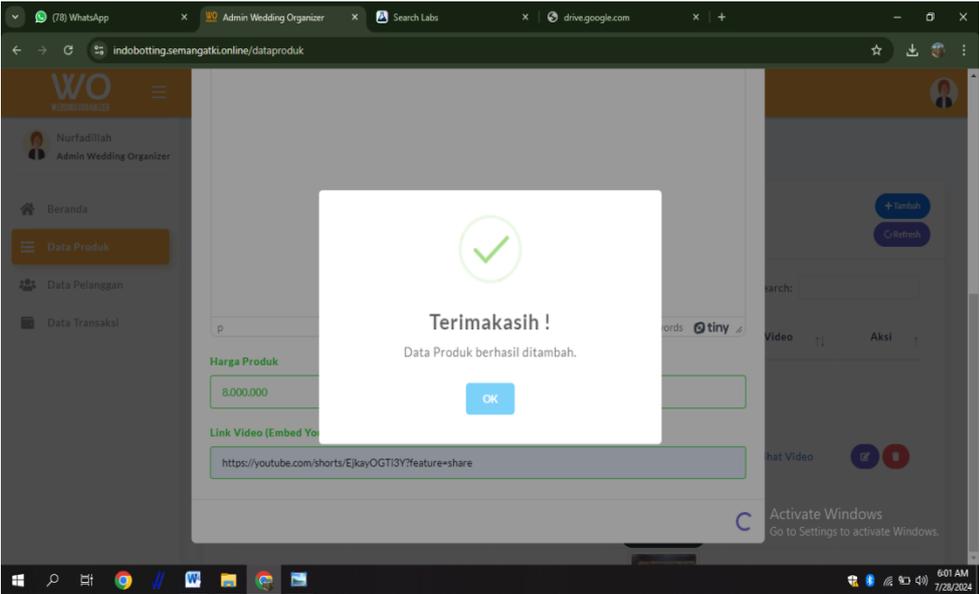
*Screenshot*



e) *Blackbox testing* lihat data transaksiTabel 4. 22 *Blackbox Testing* Lihat Data Transaksi

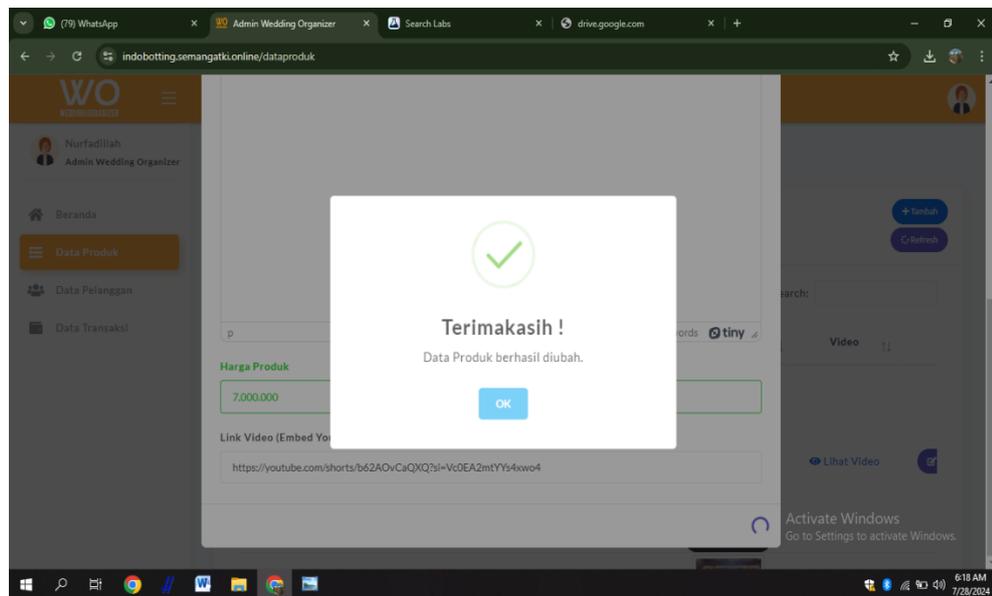
Tes Faktor	Hasil	Keterangan
Admin mengklik data transaksi	✓	Berhasil, sistem akan menampilkan “halaman data transaksi”
<i>Screenshot</i>		
		

f) *Blackbox testing* tambah dataTabel 4. 23 *Blackbox Testing* Tambah Data

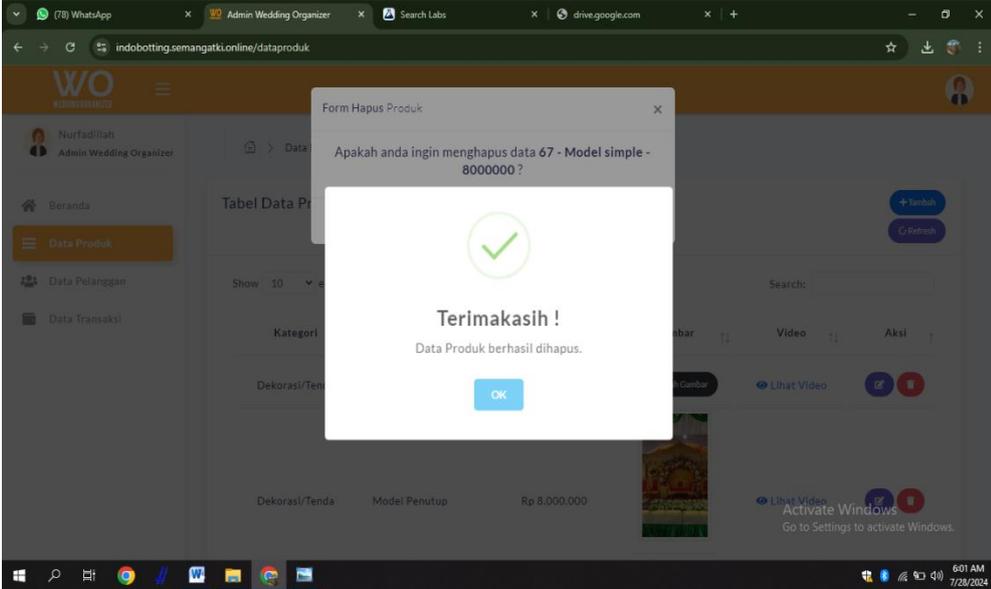
Tes Faktor	Hasil	Keterangan
Admin mengklik tambah data	✓	Berhasil, sistem akan menampilkan “data produk berhasil ditambah”
<i>Screenshot</i>		
 <p>The screenshot shows a web browser window with the URL 'indobotting.semangatkonline/dataproduk'. A modal dialog box is displayed in the center with a green checkmark icon and the text 'Terimakasih! Data Produk berhasil ditambah.' Below the message is an 'OK' button. The background shows a form with the following fields: 'Harga Produk' with the value '8.000.000', and 'Link Video (Embed Ya)' with the value 'https://youtube.com/shorts/EjkayOGT13?feature=share'. The browser's taskbar at the bottom shows the time as 6:01 AM on 7/28/2024.</p>		

g) *Blackbox testing edit data*Tabel 4. 24 *Blackbox Testing Edit Data*

Tes Faktor	Hasil	Keterangan
Admin mengklik edit data	✓	Berhasil, sistem akan menampilkan “data produk berhasil diubah”

*Screenshot*

h) *Blackbox testing hapus data*Tabel 4. 25 *Blackbox Testing Hapus Data*

Tes Faktor	Hasil	Keterangan
Admin mengklik hapus data	✓	Berhasil, sistem akan menampilkan “data produk berhasil dihapus”
<i>Screenshot</i>		
 <p>The screenshot shows a web browser window with the URL 'indobotting.semangatki.online/dataproduk'. A modal dialog titled 'Form Hapus Produk' is displayed in the center. The dialog contains the text: 'Apakah anda ingin menghapus data 67 - Model simple - 8000000?'. Below this text is a large green checkmark icon and the message 'Terimakasih! Data Produk berhasil dihapus.' with an 'OK' button. The background of the browser shows a dashboard with various menu items like 'Beranda', 'Data Produk', 'Data Pelanggan', and 'Data Transaksi'.</p>		

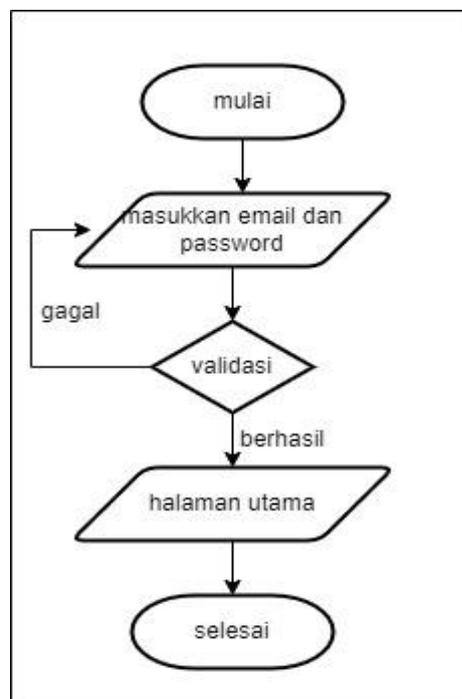
## 2. whitebox testing

Pada metode pengujian ini akan ditampilkan *flowchart* dan *flowgraph* dari sistem yang telah dibuat. Berikut dibawah ini merupakan hasil pengujian menggunakan metode *white box testing*.

### a. user

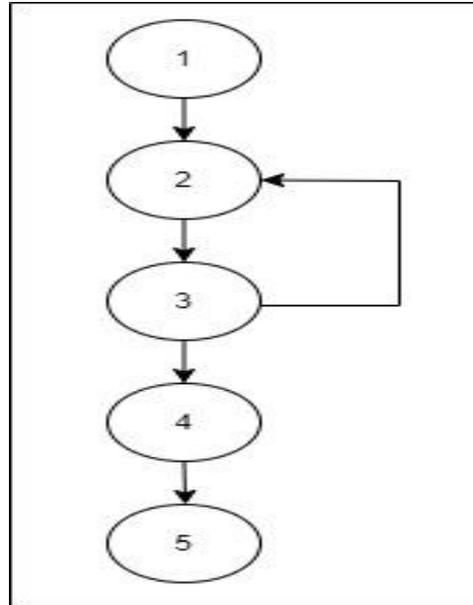
#### 1) *white box testing login berhasil*

##### a) *Flowchart*



**Gambar 4. 54** *Flowchart Login Berhasil*

**b) Flowgraph**



**Gambar 4. 55** *Flowgraph Login Berhasil*

Berdasarkan gambar diatas, 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

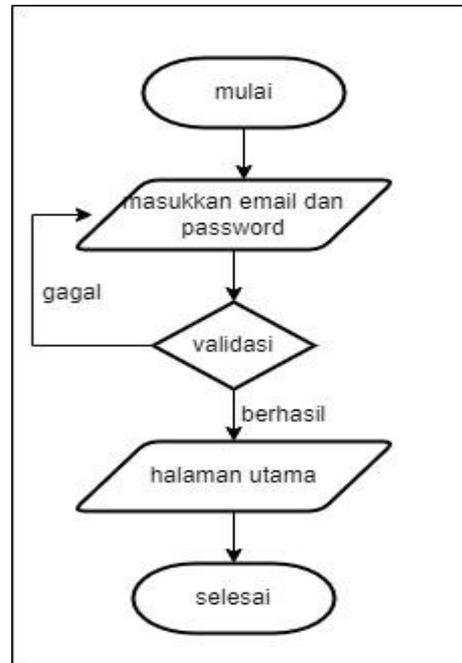
(4) *Grafik matriks email dan password* berhasil

**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

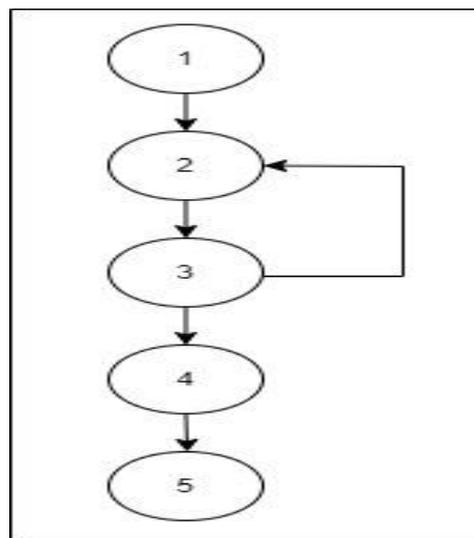
2) *White box testing* kesalahan *email* dan *password*

a) *flowchart*



**Gambar 4. 56** Flowchart Kesalahan *Email* dan *Password*

b) *Flowgraph*



**Gambar 4. 57** Flowgraph Kesalahan *Email* dan *Password*

Berdasarkan gambar diatas, 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

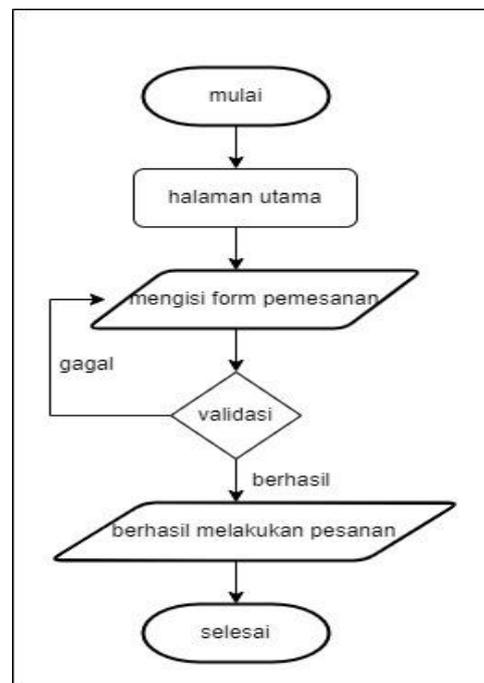
(4) *Grafik matriks* kesalahan *email* dan *password*

**Tabel 4. 27** Grafik Matriks Kesalahan Email dan Password

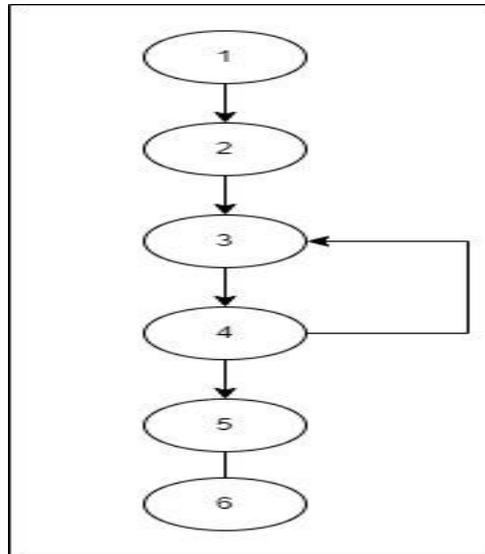
	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0
3		1		1		2-1=1
4					1	1-1=0
5						0
SUM (E +1)						1+1=2

### 3) White box testing melakukan pemesanan

#### a) flowchart

**Gambar 4. 58** Flowchart Melakukan Pemesanan

**b) Flowgraph**



**Gambar 4. 59** *Flowgraph* Melakukan Pemesanan

Berdasarkan gambar diatas, 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 :

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

$$= 6 - 6 + 2$$

$$= 2$$

(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* melakukan pesanan

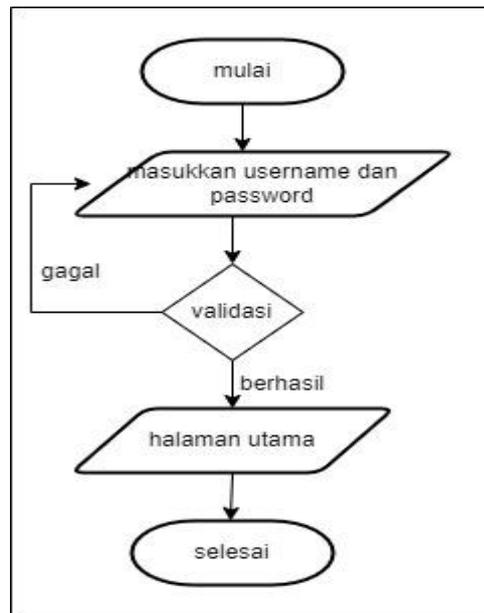
**Tabel 4. 28** *Grafik Matriks* Melakukan Pesanan

	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

*b. Admin*

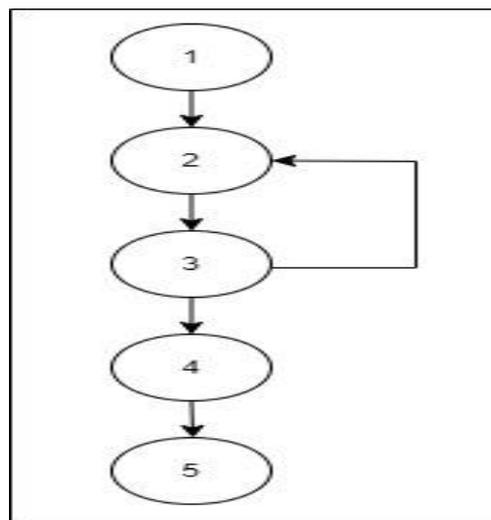
1) *White box testing login berhasil*

a) *Flowchart*



**Gambar 4. 60** *Flowchart Login Berhasil*

b) *Flowgraph*



**Gambar 4. 61** *Flowgraph Login Berhasil*

Berdasarkan gambar diatas, 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

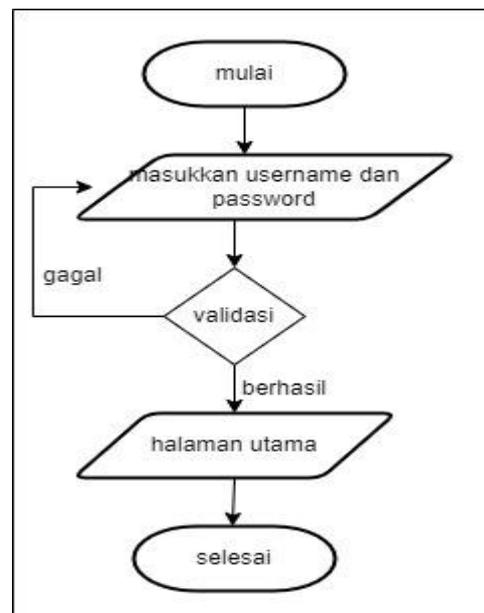
(4) *Grafik matriks username* dan *password* berhasil

**Tabel 4. 29** Grafik Matriks Login Berhasil

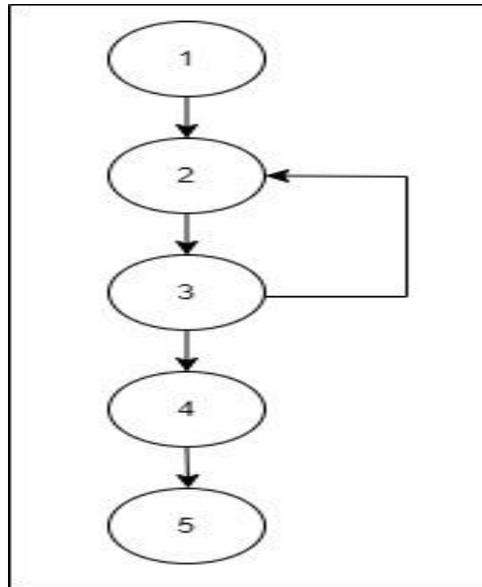
	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0
3		1		1		2-1=1
4					1	1-1=0
5						0
SUM (E +1)						1+1=2

## 2) White box testing kesalahan username dan password

### a. Flowchart

**Gambar 4. 62** Flowchart Kesalahan Username dan Password

***b. Flowgraph***



**Gambar 4. 63** *Flowgraph* Kesalahan *Username* dan *Password*

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut :

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

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 :

*Path* 1 = 1-2-3-2

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

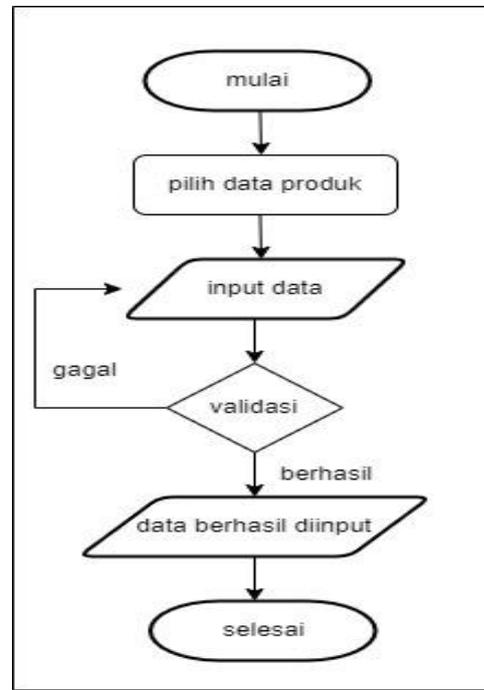
(4) *Grafik matriks* kesalahan *username* dan *password*

**Tabel 4. 30** *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

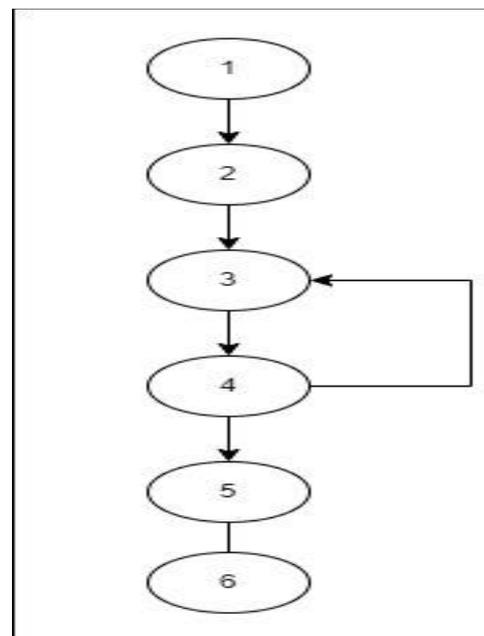
### 3) *White box testing* tambah data

#### a. *flowchart*



Gambar 4. 64 *Flowchart* Tambah Data

#### b. *flowgraph*



Gambar 4. 65 *Flowgraph* Tambah Data

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut :

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

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

(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

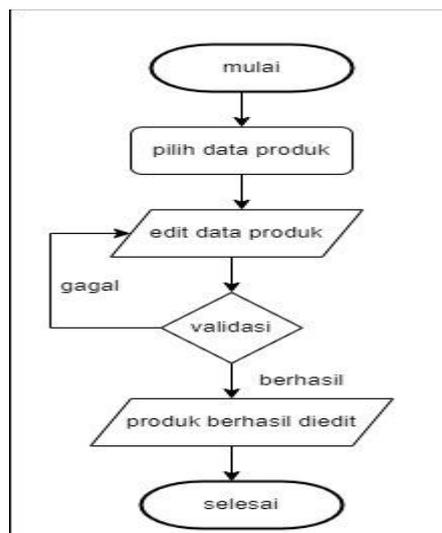
(4) *Grafik matriks* tambah data

**Tabel 4. 31** Grafik Matriks Tambah Data

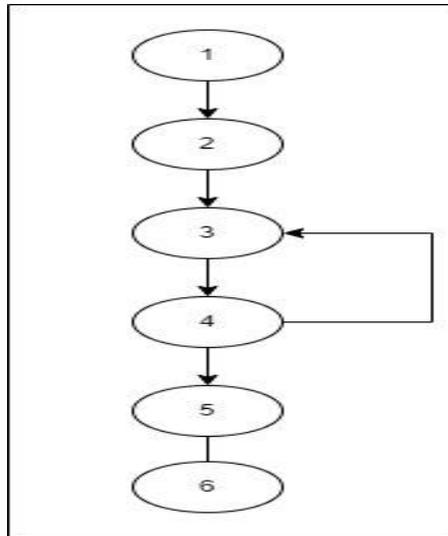
	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3				1			1-1=0
4			1		1		2-1=1
5						1	1-1=0
6							0
SUM (E+1)							1+1=2

#### 4) White box testing edit data

##### a. Flowchart

**Gambar 4. 66** Flowchart Edit Data

**b. Flowgraph**



**Gambar 4. 67** Flowgraph Edit Data

Berdasarkan gambar diatas, 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 :

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

$$= 6 - 6 + 2$$

$$= 2$$

(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*

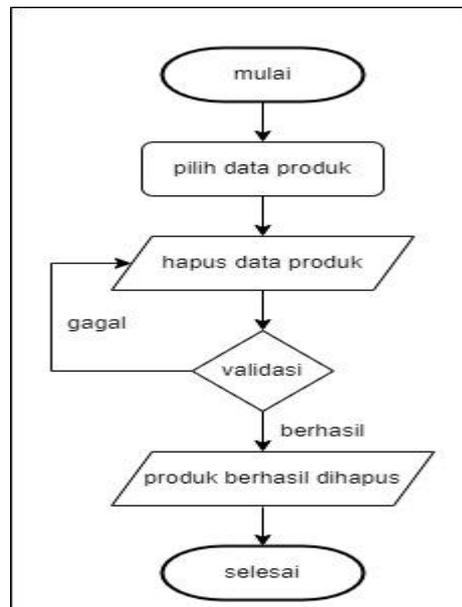
(4) *Grafik matriks edit data*

**Tabel 4. 32** *Grafik Matriks Edit 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

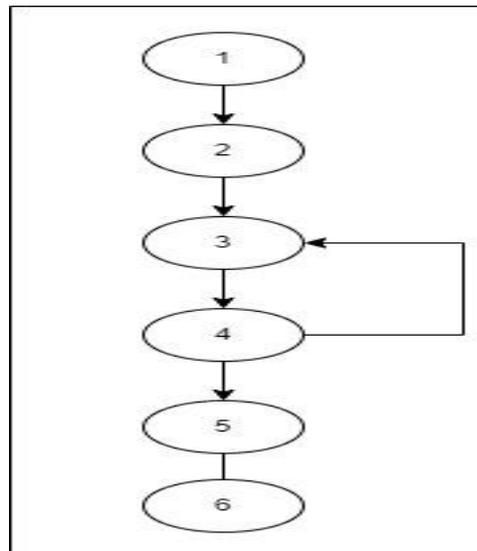
## 5) White box testing hapus data

### a. Flowchart



Gambar 4. 68 Flowchart Hapus Data

### b. Flowgraph



Gambar 4. 69 Flowgraph Hapus Data

Berdasarkan gambar diatas, 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 :

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

$$= 6 - 6 + 2$$

$$= 2$$

(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

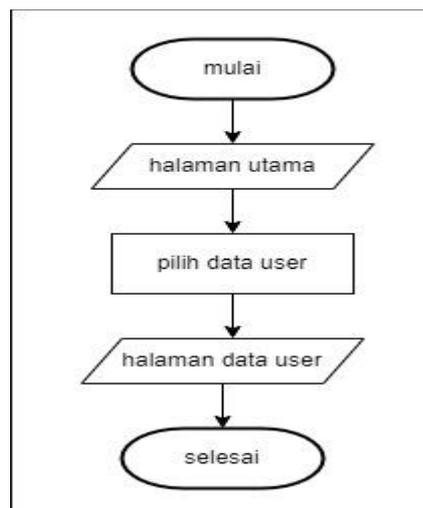
(4) *Grafik matriks* hapus data

**Tabel 4.33** Grafik Matriks Hapus 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

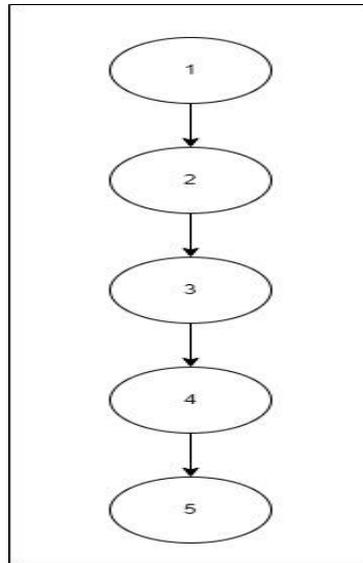
6) *White box testing lihat data user*

a. *Flowchart*



**Gambar 4.70** *Flowchart* Lihat Data User

**b. Flowgraph**



**Gambar 4. 71** Flowgraph Lihat Data User

Berdasarkan gambar diatas, 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 = 1-2-3-4-5

(4) *Grafik matriks* lihat data *user*

**Tabel 4.34** *Grafik Matriks Lihat Data User*

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

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

1. Berhasil dibuat sebuah aplikasi wedding organizer berbasis android menggunakan database MYSQL.
2. Aplikasi wedding organizer berbasis android ini dibuat dengan menggunakan bahasa pemrograman java dan mysql sebagai database management system, telah melalui pengujian dengan baik menggunakan metode blackbox testing dan whitebox testing.
3. Aplikasi wedding organizer yang mempunyai 2 akses login yaitu user dan admin dimana user dapat melihat produk-produk yang ada di wedding organizer sedangkan admin dapat melihat data produk, data pelanggan, dan data transaksi. aplikasi ini membantu user dalam memilih paket wedding.
4. Aplikasi wedding organizer ini mempunyai 5 menu yaitu menu paketan yang terdiri dari paket 1 dan paket 2 dan menu per item yang terdiri dari baju pengantin, dekorasi, makeup dan baju bodo. serta menggunakan metode pembayaran e-money/via bank dan cod yaitu jika user/customer memilih e-money maka user akan mengupload bukti pembayaran di form bukti bayar.

## **B. Saran**

penulis berharap untuk mengembangkan aplikasi ini dikemudian hari karena aplikasi ini masih memiliki fungsi yang sangat terbatas dan disarankan agar bisa menambahkan beberapa data seperti chatering, sistem chatting untuk mempermudah proses pembayaran online.

## DAFTAR PUSTAKA

- Daffa, R. F., & Giatika Chrisnawati. (2022). *Jurnal Pendidikan dan Konseling*, 4, 4386–4396.
- Erni Sri Wahyuni. (2019). Analisis Cara Kerja CRUD dengan Menggunakan Android Studio.
- Ginangjar, H., & Setiawan, R. (n.d.). *Perancangan Aplikasi*, (1), 155–162.
- Heri Firmansyah, Hamsah Hudafi, Rizky Syahputra Nasution, (2023). (2023). No Title, 9(1), 32–44.
- Irsan, M. (2015). Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan. *Jurnal Penelitian Teknik\_Informatika*,1(1),115–120.
- Michelle Soeharto, Maureen Nuradhi, 2019. (2019). DENGAN PENDEKATAN SENSE OF PLACE, 5.
- Monica, N., Sarkum, S., & Purnama, I. (2018). Aplikasi Data Mahasiswa Berbasis Android : Studi Pada Sekolah Tinggi Ilmu Ekonomi Labuhanbatu, 3(1), 43–53.
- Muhammad Saed Novendri, Ade Saputra, C. E. F. (2019). *L e n t e r a d u m a i* , 10, 46–57.
- Muntari, S., & Masdalipa, R. (n.d.). Aplikasi Jasa Wedding Organizer Berbasis Web Di Kota Pagar Alam, (75), 68–78.
- Nahlah, Amiruddin, A. R. (2017). PERANCANGAN WEBSITE E-COMMERCE BERBASIS PHP DAN MYSQL PADA, 2017, 11–16.
- Ramadhan, R. F., & Mukhaiyar, R. (2020). *Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi*. 1(2), 129–134.
- 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.
- Sari, I. P., Siska, S. T., & Budiman, A. (2021). Perancangan Aplikasi Pelayanan Gangguan Tv Kabel Berbasis Web Dan Sms Gateway, 1(1), 20–28.
- Wahyudi, J. (2019). MOBIL BERBASIS ANDROID, 2(1).

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>