BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi informasi saat ini mengalami perubahan yang pesat. Manusia telah mampu membangun aplikasi modern yang bermanfaat untuk masyarakat agar dapat mempermudah banyak hal, terutama dalam melakukan pemasaran produk secara digitalisasi. Pemasaran produk secara digitalisasi bisa disebut dengan e-marketplace karena terjadinya pertemuan antara penjual dan pembeli secara online atau virtual. Pertemuan tersebut berupa transaksi antara penjualan dan pembeli dengan menggunakan platform yang disediakan oleh operator marketplace.

Wedding organizer adalah event organizer yang membantu dalam persiapan dan pelaksanaan pesta pernikahan, menawarkan berbagai macam paket pernikahan yang bisa dipilih diantaranya gedung, pelaminan, dekorasi, make up, baju, catering, dokumentasi dan hiburan yang dibutuhkan dalam pernikahan tersebut. Seiring berjalannya waktu, penggunaan jasa wedding organizer sangat membantu calon pengantin untuk mempersiapkan hari bahagianya, apalagi calon pengantin yang tidak punya banyak waktu untuk mempersiapkan hari bahagianya dan masih banyak kebutuhan yang diperlukan untuk menyiapkan sebuah pernikahan.

Pemesanan paket pernikahan di wedding organizer khususnya di Parepare masih dilakukan dengan cara yang manual. Seperti calon pengantin harus datang

terlebih dahulu ke tempat wedding organizer untuk mengetahui apa saja jasa yang ditawarkan oleh event organizer yang dikunjungi dan melakukan pemesanan paket. Masih manualnya pemesanan untuk acara pernikahan ini membuat pelanggan merasa kesulitan dalam melakukan pemesanan disebabkan tingkat kesibukan masyarakat sekarang dan jarak daerah. Saat ini hampir semua kalangan masyarakat telah memakai smartphone dan sebagian besar berbasis android, melihat peluang itu maka diciptakan suatu sistem yang dapat melakukan pemesanan wedding organizer secara online atau digital sehingga membuat kemudahan dalam melakukan pemesanan, yang tidak terikat oleh waktu dan jarak, dalam sistem ini pelanggaan dapat memilih konsep apa yang dipakai seperti gedung, pelaminan dan dekorasi, baju pengantin, make up, dokumentasi, catering, dan hiburan. Pada sistem ini, nantinya juga ada pilihan paket pernikahan sebagai contoh untuk para pengantin mencari paket yang sesuai berdasarkan bentuk konsep pernikahan yang diinginkan.

Berdasarkan uraian diatas, penulis membuat penelitian dengan judul "Aplikasi E-Marketplace Wedding Organizer berbasis Android" agar dapat membantu masyarakat khususnya di daerah Parepare untuk melakukan pemesanan Wedding Organizer kapan dan dimana calon pengantin berada tanpa harus mengunjungi kantor event organizer.

B. Rumusan Masalah

Dengan mengacu pada latar belakang yang sudah di jelaskan di atas maka disusunlah rumusan masalah yang akan dibahas dalam penelitian ini adalah

bagaimana merancang sebuah aplikasi E-Marketplace agar dapat memudahkan pemesanan bagi konsumen dan memudahkan pihak Wedding Organizer dalam melakukan pemasaran secara online?

C. Tujuan Penelitian

Berdasarkan permasalahan yang telah dirumuskan, maka tujuan peneitian ini adalah:

1. Membuat aplikasi yang dapat memudahkan pihak pengguna jasa dalam memilih wedding organizer yang mereka inginkan.
2. Membuat aplikasi e-marketplace di Parepare untuk membantu pengelola wedding organizer dalam mengelola pemesanan.

D. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Sistem pembayaran harus lunas tidak bisa membayar dengan mengansur.
2. Sistem penjualan yang sebelumnya secara manual perlu diubah ke sistem berbasis *mobile.*
3. Pelanggan tidak diperbolehkan untuk membantalkan pemesananya dengan alasan apapun.
4. Penjualan melalui aplikasi berbasis android ini dapat dilakukan oleh pelanggan atau calon pembeli.

E. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Bagi dunia akedemik

Dapat memberikan suatu referensi yang berguna bagi dunia akademis khususnya dalam penellitian yang akan dilaksanakan oleh para peneliti yang akan datang dalam hal e-marketplace.

2. Bagi Pelanggan

Memberikan pilihan paket WO yang lebih banyak kepada pelanggan, Pelanggan tidak perlu datang ke tempat setiap WO dan Pelanggana dapat merencanakan anggaran biaya pernikahan sejak dini.

3. Bagi Pihak Wedding Organizer

Memudahkan proses bertransaksi dan membantu WO memasarkan usaha mereka secara online serta dapat memantau setiap data vendor dan venue secara online.

4. Bagi penulis

Menambah pengetahuan dan wawasan serta mengembangkan daya nalar dalam perancangan teknologi berbasis *android*, dan sebagai salah satu syarat untuk jadi sarjana jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Parepare.

F. Sistematika Penulisan

Struktur penulisan penelitian ini adalah:

BAB I : PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan dari pembuatan proposal ini.

BAB II : TINJAUAN PUSTAKA

Berisi tentang penelitian kajian teori, kajian hasil penelitian dan hasil tinjauan penelitianyang mendukung pembuatan skripsiini.

BAB III : METODE PENELITIAN

Berisi tentang letak lokasi serta waktu, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian, dan analisasistem.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Berisi tentang gambaran umum, analisis, meliputi pendefenisian dan pemodelan sistem dalam bentuk *use case, class* diagram, *sequence* diagram serta desain *database*, dan pengujian *system* dengan *whitbox* dan *blackbox.*

BAB V : PENUTUP

Berisi kesimpulan yang dapat di ambil dari penulisan akhir ini dan saran-saran pengembangannya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Usaha Pada Sektor Pernikahan

Badan usaha pada sektor jasa seputar pernikahan sangat beragam, namun secara garis besar dapat dikategorikan ke dalam tiga hal utama yaitu:

a. Wedding Organizer

Wedding organizer adalah sebuah layanan servis yang diberikan kepada calon pengantin secara menyeluruh dari mulai perencanaan konsep acara, penganggaran dana, sampai bekerja sama dengan bermacam vendor dengan tepat sesuai dengan permintaan. Tugas salah satu wedding organizer adalah menyediakan panitia pernikahan, yang juga harus memikirkan penempatan posisi keluarga dalam kepanitiaan tersebut. Masalah yang muncul dalam mengelola panitia keluarga biasanya adalah pemilihan lokasi, biaya, bagaimana kronologis jalannya acara dan lain-lain. Pada intinya peran sebuah WO adalah mensukseskan acara pernikahan dengan mengurus hampir seluruh hal sedetail mungkin. (Rinaldi, Apsari Susanti, 2019).

b. Wedding Vendor

Wedding vendor merupakan badan usaha di bidang pernikahan yang lebih terfokus pada penjualan atau pengembangan produk atau jasa mereka sendiri, vendor-vendor ini secara garis besar dapat dikategorikan sebagai berikut:

1) Bridal

Dari bahasa Inggris yaitu bride yang artinya pengantin, bridal maksudnya adalah usaha khusus untuk mengurus hal-hal yang terkait penampilan serta berbagai aksesoris pengantin dan orang tua. Seperti jas pria, gaun wanita, sepatu, make-up maupun hairdo dan lain-lain.

2) Dokumentasi

Dokumentasi yang dimaksud dalam wedding documentation ini adalah media foto dan video pada umumnya. Berikut istilah yang biasa digunakan dalam wedding documentation:

a) Pre-wedding

Merupakan sesi foto yang dapat dilakukan indoor maupun outdoor dengan tujuan foto yang diambil pada saat prewedding dapat dicetak besar dan dipajang pada acara di hari pernikahan dan juga dapat digunakan sebagai gambar pada kartu undangan pernikahan.

b) Full Documentation

Rekaman seluruh acara pernikahan dari awal sampai akhir secara kronologis. Hasilnya pun kurang artistik dan durasi nya bisa mencapai 3 jam.

c) Cinematic Video

Memiliki hasil video yang artistik, karena menggunakan kamera yang lebih bagus lalu di buat dan di edit dengan teknis yang lebih kreatif dan terencana serta ada musik yang mengiringi video tersebut. Oleh sebab itu cinematic video ini memiliki harga yang relatif lebih mahal dibandingkan full

documentation. Dengan durasi 1 hingga 5 menit, cinematic video dapat dibuat untuk pre-wedding maupun weddingnya sendiri.

d) Trailer

Adalah video pendek berdurasi 15 detik hingga 2 menit yang biasanya dicuplik dari keseluruhan video. Dengan tujuan sebagai undangan (invitation), sebagai pembuka video documentation dan lain-lain.

e) Same-day Edit

Video kilat dengan durasi yang beragam dan berisi hasil acara sepanjang hari yang selesai di edit pada hari itu juga, dengan tujuan ditampilkan pada acara keesokan harinya.

f) Photo Booth

Merupakan bagian dokumentasi bertipe dekorasi yang lebih interaktif dengan tamu undangan karena fungsinya yang memang ditujukan sebagai latar untuk berfoto.

3) Dekorasi

Mengurus hal-hal yang berkaitan dengan dekorasi, panggung, dekorasi kamar pengantin, tenda, lampu dan tata cahaya, bungabungaan dan lain-lain

4) Catering

Bagian makanan dan minuman, yang biasanya sudah disediakan satu paket oleh pihak penyedia venue (gedung, ballroom hotel, taman dan lain-lain) serta juga beberapa menawarkan jajanan kecil (sweet corners)

5) Wedding Cake

Wedding cake atau kue pernikahan mulai menjadi gaya hidup masyarakat Indonesia dimana makanan khusus ini umumnya digunakan sebagai penghias yang nantinya dibelah oleh kedua pengantin.

6) Wedding Invitation

Undangan pernikahan biasanya berbentuk media cetak namun sekarang mengundang kerabat dan saudara menggunakan undangan yang berbentuk media digital sedang banyak digandrungi, misalnya dalam bentuk cinematic video berdurasi sekitar 1 menit.

7) Other Wedding Vendor

Bagian dan lain-lain ini untuk vendor lain yang tidak masuk pada bagian kategori di atas, seperti honeymoon, rental mobil dan lainlain.

c. Wedding Venue

Wedding adalah istilah yang digunakan untuk menyebut tempat di selenggarakan pernikahan. Contohnya rumah, lapangan terbuka, hall gedung, ballroom hotel, taman dan lain-lain. Pihak penyedia tempat seperti ini biasanya sudah memiliki katering mereka masing-masing.

2. Sistem Informasi

Sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen, atau sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen.

Menurut definisi sistem informasi sebelumnya, sistem informasi memiliki hubungan dengan sistem data saat ini di satu sisi dan sistem aktivitas di sisi lain.

Definisi Sistem Informasi Menurut para ahli :

a. Menurut John F.Nash (1995:8) Sistem informasi adalah kombinasi dari orang-orang, fasilitas atau teknologi, media, prosedur-prosedur, dan pengendalian yang mempunyai maksud untuk menata jaringan komunikasi yang penting, proses-proses dan transaksi-transaksi tertentu secara teratur, membantu manajemen dan para pemakain internal dan eksternal, dan menyediakan suatu dasar pengambilan keputusan yang tepat.
b. Menurut Robert A. Leitch (1999:11), sistem informasi adalah suatu sistem yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

3. Pengertian *E-Commerce*

Dedi Riswandi. (2019, p.1) menyatakan bahwa penjualan elektronik (*e-commerce*) seperti TokoBagus, Tokopedia, dan Blibli memiliki model bisnis yang berbeda. Karena model bisnis yang berbeda itulah mereka tidak dapat dibandingkan satu dengan yang lainnya sebagai saingan.

a. Jenis *E-Commerce*

Dan di Indonesia terdapat lima model bisnis yang paling banyak diusung oleh pelaku bisnis e-commerce yaitu, iklan baris, market place, shopping mall, toko online dan media sosial.

1) Classifieds/Listing/Iklan

baris Model bisnis e-commerce paling sederhana yang cocok digunakan di negara berkembang. Indikator model bisnis ini:

a) Website yang bersangkutan tidak memfasilitasi kegiatan transaksi online dan penjual (individual) dapat menjual barang kapan saja, dimana saja secara gratis.

b) Contoh: Tokobagus, Berniaga, dan OLX.

c) Metode transaksi favorit: *Cash on delivery (*COD).

d) Cara mencari uang: Iklan premium.

e) Jenis penjual: Situs iklan baris seperti ini cocok bagi penjual yang hanya ingin menjual sekali-kali saja, seperti barang bekas atau barang yang stok nya sedikit.

2) Marketplace C2C (*Customer to Customer*)

Model bisnis dimana website yang bersangkutan tidak hanya membantu mempromosikan barang dagangan saja, tapi juga memfasilitasi transaksi uang secara online. Indikator model bisnis ini:

a) Seluruh transaksi online harus difasilitasi oleh website yang bersangkutan

b) Bisa digunakan oleh penjual individu

c) Kegiatan jual beli di website marketplace harus menggunakan fasilitas transaksi online seperti layanan escrow atau rekening pihak ketiga untuk menjamin keamanan transaksi. Penjual hanya akan menerima uang pembayaran setelah barang diterima oleh pembeli. Selama barang belum

sampai, uang akan disimpan di rekening pihak ketiga. Apabila transaksi gagal, maka uang akan dikembalikan ke tangan pembeli.

d) Contoh: Tokopedia, Bukalapak, Lamido, Blanja dan Elevenia.

e) Cara mencari uang: Layanan penjual premium, iklan premium, dan komisi dari setiap transaksi.

f) Jenis penjual: Cocok bagi penjual yang lebih serius dalam berjualan online. Biasanya sang penjual memiliki jumlah stok barang yang cukup besar dan mungkin sudah memiliki toko fisik.

3) Shopping mall

Model bisnis ini mirip sekali dengan marketplace, tapi penjual yang bisa berjualan disana haruslah penjual atau brand ternama karena proses verifikasi yang ketat. Satu-satunya situs online shopping mall yang beroperasi di Indonesia adalah Blibli.

a) Cara mencari uang: Komisi dari penjual.

4) Toko online B2C (*Business to Consumer*)

Model bisnis ini cukup sederhana, yakni sebuah toko online dengan alamat website (domain) sendiri dimana penjual memiliki stok produk dan menjualnya secara online kepada pembeli.

a) Beberapa contoh toko online di Indonesia ialah Bhinneka, Lazada Indonesia, BerryBenka, dan Tiket.com.

b) Keuntungan: Memiliki kebebasan penuh (merubah jenis tampilan sesuai keinginan, dapat membuat blog untuk memperkuat SEO toko online anda dan lain-lain).

c) Bagi yang tertarik membuka toko online secara mudah, dapat coba menggunakan *Shopify,* Jejualan, *Pixtem, Jarvis Store,* dan *Klakat.*

d) Cara mencari uang: Berjualan barang demi mendapatkan profit.

e) Jenis penjual: Cocok bagi mereka yang serius berjualan online dan siap mengalokasikan sumber daya mereka untuk mengelola situs mereka sendiri.

5) Toko online di media sosial

Banyak penjual di Indonesia yang menggunakan situs media sosial seperti Facebook dan Instagram untuk mempromosikan barang dagangan mereka. Membuat toko online di media sosial sangatlah mudah, sederhana, dan asiknya gratis. Tidak hanya media sosial, aplikasi chatting pun juga sering digunakan sebagai media jual beli barang seperti LINE, WhatsApp, BBM dan lain-lain.

a) Jenis penjual: Penjual yang ingin memiliki toko online sendiri tapi tidak ingin repot.

4. Android

Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka atau sering juga disebut open source bagi para pengembang (*developer*) untuk membangun aplikasi ciptaan mereka sendiri. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open *Handset Alliance,* konsorsium dari 34 perusahaan piranti keras,

piranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile* dan Nvidia.

a. Arsitektur *Android*

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

1) *Applications* dan *Widgets* adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdaat aplikasi inti termasuk *klien email*, program sms, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java*.
2) *Applications Frameworks* adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *Android,* karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.
3) *Libraries, libraries* ini adalah *layer* di mana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.
4) *AndroidRun Time, layer* yang membuat aplikasi *Android* dapat dijalankan di mana prosesnya menggunakan Implementasi *Linux.Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.

5) *Linux Karnel* adalah *layer* di mana inti dari operating sistem dari *Android* itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi *Android* lainnya.

b. Komponen *android*

Ada enam jenis komponen pada aplikasi *Android,* yaitu:

1) *Activities,* suatu *activities* akan menyajikan *user interface (UI)* kepadapengguna, sehingga pengguna dapat melakukan interaksi untuk menjalankan fungsi tertentu. Sebuah aplikasi *Android* bisa jadi hanya memiliki satu *activity,* tetapi umumnya aplikasi memiliki banyak *Activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.
2) *Service, Service* tidak memiliki *Graphic User Interface (GUI),* tetapi *service* berjalan secara *background* untuk melakukan operasi-operasi yang longrunning (proses yang memakan waktu cukup lama) atau melakukan operasi untuk proses *remote.*
3) *Broadcast Reciever*, *broadcast reciever* berfungsi menerima dan bereaksi untuk menyiapkan notifikasi. *Broadcast Reciever* tidak memiliki *User Interface (UI)* tapi memiliki sebuah *Activity* untuk merespon informasi yang mereka terima atau kepada pengguna. *Broadcast receiver* hanyalah pintu gerbang menuju komponen lain dan memang dirancang untuk hanya melakukan kerja seminimal mungkin.
4) *Content Provider*, *content provider* membuat kumpulan aplikasi data secara *spesifik* sehingga bisa digunakan oleh aplikasi lain. *Content*

providers juga berguna untuk membaca dan menulis data yang berstatus *private* dan tidak dibagikan ke suatu aplikasi. (Safaat, 2018).

c. Kelebihan Sistem Operasi Android

Adapun beberapa kelebihan sistem operasi android adalah sebagai berikut:

1) Lengkap (Complete Platform)

Android merupakan sistem operasi yang aman dan banyak menyediakan tools yang berguna untuk membangun sebuah aplikasi yang kemudian aplikasi tersebut dapat lebih dikembangkan lagi oleh para developer.

2) Terbuka

Platform Android yang bersifat open source menjadikan sistem operasi ini mudah dikembnagkan oleh para developer karena bersifat terbuka.

3) Gratis (Free Platform)

Developer dengan bebas bisa mengembangkan, mendistribusikan dan memperdagangkan sistem operasi Android tanpa harus membayar royalty untuk mendapatkan license.

d. Fitur-fitur Android

Beberapa fitur-fitur android yang paling penting adalah sebagai berikut:

1) Framework Aplikasi: memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.

2) Dalvik Virtual Machine: virtual machine yang dioptimalkan untuk Perangkat mobile.

3) Grafik: grafik 2D dan grafik 3D yang didasarkan pada library OpenGL.

4) SQLite: sebagai pengganti dari database pada android.

5) Mendukung Media: audio dan video dengan berbagai format.

6) Lingkungan pengembangan yang kaya, termasuk emulator, peralatan debugging, dan plugin untuk Eclipse IDE.

7) Bluetooth, EDGE, 3G, 4G dan WiFi.

8) Mendukung berbagai macam sensor dan GPS.

5. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA . Selain sebagai editor kode dan fitur *developer IntelliJ* yang l, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android, seperti (Muhammad Yusuf Fadillah, 2021**)**:

a) Sistem build berbasis *Gradle* yang *fleksibel*

b) *Emulator* yang cepat dan kaya fitur

Lingkungan terpadu tempat bisa mengembangkan aplikasi untuk semua perangkat Android

c) Terapkan Perubahan untuk melakukan push pada perubahan kode dan resourceke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi

d) Template kode dan integrasi GitHub untuk membantu membuat fitur aplikasiumum dan mengimpor kode sampel

e) Framework dan alat pengujian yang lengkap

f) Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalahlainnya

g) Dukungan C++ dan NDK

h) Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*

6. *Famework*

Sistem operasi Android didukung oleh banyak framework untuk mengembangkan aplikasi. Aplikasi android secara native dapat dikembangkan dengan bahasa java atau C, sedangkan framework-nya menggunakan editor Eclipse dan SDK Android. Selain itu, android bisa juga dikembangkan menggunakan framework lain seperti:

a) Basic4android: Bahasa pemrograman Basic.

b) Adobe AIR: Bahasa pemrograman *ActionScript.*

c) phoneGap: Bahasa Pemrograman HTML5 dan *JavaScript.*

d) PHP for Android: Bahasa pemrograman PHP.

7. *Kotlin*

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan *pemrograman fungsional*. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk backend. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi adalah sebagai berikut:

a) Dapat mengatasi *NullPointerException* yang umumnya terdapat pada Java.

b) Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.

c) Mudah dipelajari.

d) Dukungan IDE untuk mempermudah dalam pemrograman.

8. *Javascript*

JavaScript adalah salah satu bahasa pemrograman yang paling banyak digunakan dalam kurun waktu dua puluh tahun ini. Bahkan, bahasa pemrograman ini merupakan salah satu yang paling utama bagi web *developer*, Alasan mengapa *JavaScript* menjadi salah satu bahasa pemrograman yang populer adalah kemudahan proses belajar dan penggunaannya. Banyak *developer* yang bahkan akhirnya memilih *JavaScript* sebagai bahasa pemrograman terbaik. Bahasa pemrograman lainnya hanya diperlukan jika developer menginginkan sesuatu yang lebih spesifik. (Awaluddin,2022).

Biasanya, Anda bisa menyematkan *JavaScript* langsung ke halaman website atau mengarahkannya melalui file .js sendiri. *JavaScript* merupakan bahasa *client-side***,** yang berarti proses pengunduhan dan pemrosesan *script* terjadi di perangkat pengunjung situs Anda. Sedangkan untuk bahasa pemrograman *server-side***,** proses tersebut terjadi pada *server* sebelum bahasa pemrograman mengirimkan file ke pengunjung situs.

Perlu Anda ketahui, sebagian web browser juga menawarkan kesempatan bagi *user* untuk menonaktifkan *JavaScript*. Jadi, sebaiknya cari tahu apa yang terjadi pada *event* yang Anda unduh ke perangkat yang bahkan tidak

mendukungnya.

10. MySQL

Beberapa pengertian *MySQL* menurut para ahli:

a) Menurut Raharjo (2018:21), "*MySQL* merupakan RDBMS (atau *server database)* yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*".

b) Menurut Kadir (2020:2), "*MySQL* adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source.* (Adi Nugroho.2020).

11. Metode Pengembangan Perangkat Lunak

a. Metode Pengumpulan Data

Pengumpulan data bertujuan untuk melengkapi dan mendukung materi dalam uraian pembahasan. Pengumpulan data dilakukan guna mendapatkan data yang real sesuai refrensi yang diperlukan. Metode ini meliputi studi pustaka, observasi dan wawancara.

b. Prototype

Metode yang digunakan dalam pengembangan sistem ini adalah prototype, yaitu metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna. Model prototype merupakan salah satu model SDLC yang mempunyai ciri khas sebagai model proses evolusioner. Prototype sendiri bertujuan agar pengguna dapat memahami

alur proses sistem dengan tampilan dan simulasi yang terlihat siap digunakan. Model prototype dapat dilihat pada gambar 2.1 dibawah ini:



Gambar 2 1 *Prototype* Paradigma

Tahap-tahap pada model prototype diatas adalah sebagai berikut:

1) Mendengarkan Pelanggan Tahap ini dilakukan pengumpulan data untuk memenuhi kebutuhan sistem dengan cara mendengar keluhan dari pelanggan. Analisis kebutuhan pemakai digunakan untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi melalui studi pustaka, observasi, wawancara.
2) Merancang dan Membuat Prototype Tahap ini melakukan pembuatan dan perancangan sistem prototype serta perancangan antarmuka (*User Interface*). Prototype yang akan dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.
3) Uji Coba Prototype Tahap ini prototype yang telah dirancang akan diuji coba. Kemudian akan dievaluasi kekurangan-kekurangan apa saja yang terdapat

pada sistem. Pengembang kemudian memperbaiki sistem sesuai kekurangan sampai dihasilkan sistem yang benar-benar dikehendaki oleh pengguna (Melisa Adhari, 2021).

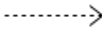
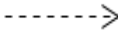
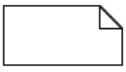
c. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap seb uah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

d. *Use Case Diagram*

Diagram *Use Case* menjelaskan kapabilitas sistem yang diharapkan. Penekanannya adalah pada "apa" yang dilakukan sistem daripada "bagaimana". Sebuah Use Case menggambarkan interaksi aktor dan sistem. Kasus penggunaan adalah pekerjaan tertentu, seperti masuk ke sistem atau membuat daftar belanja. Aktor adalah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan tugas tertentu. Di antara simbol-simbol *Use Case Diagram* adalah:

Tabel 2. 1 Simbol *Usecase* Diagram

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

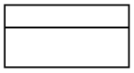
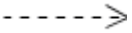
Lanjutan Tabel 2. 1

e. *Class Diagram*

class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *class*, atributnya, metode, dan hubungan antar objek. *Class diagram* disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen.

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2. 2 Simbol *Class* Diagram

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

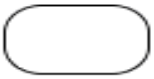
Lanjutan Tabel 2. 2

f. *Activity Diagram*

Menyatakan bahwa diagram aktivitas atau *activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2 3 Simbol *Activity* Diagram

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

g. *Sequence Diagram*

Menyatakan bahwa sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasi komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu obyek-obyek tersebut kemudian diurutkan dari kiri kekanan, aktor yang menginisialisasi interaksi biasanya ditaruh di paling kiri dari diagram.

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2. 4 Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

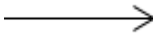
h. *StateChartDiagram*

*Statechar*t diagram merupakan model perilaku yang dinamis dari class secara individual maupun beberapa bentuk dari obyek. Semua itu menunjukkan deretan dari state yang dilakukan obyek melalui event yang menyebabkan sebuah transisi dari satu ke aktifitas yang lain, dan beberapa aksi yang menghasilkan dari satu state atau aktifitas yang berubah. Statechart diagram fokus pada state dan secara bentuk digunakan untuk memodelkan tahapan yang nampak terpotong-potong dari sebuah aktivitas yang kontinyu dari obyek. Diagram ini memiliki hubungaan yang tetrtutup dengan activity diagram. Activity diagram fokus pada aktifitas dan pada pemodelan deretan aktivitas dari proses.

Adapun simbol-simbol *StateChart Diagram* antara lain :

Tabel 2. 5 Simbol *StateChart* Diagram

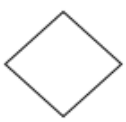
No.	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.

No.	GAMBAR	NAMA	KETERANGAN
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

i. *Flowchart*

Awaluddin. (2022). *Flowchart* (diagram alir) dapat digunakan sebagai alternative untuk menyajikan algoritma. *Flowchart* adalah bentuk penyajian grafis yang menggambarkan solusi langkah demi langkah terhadap suatu masalah.

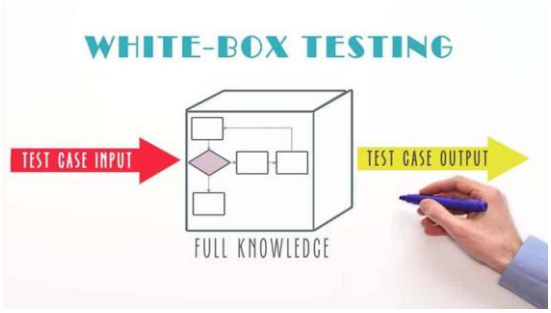
Tabel 2. 6 Simbol Standar *Flowchart*

Simbol	Nama dan Funsi
	***Terminator***(Tanda mulai atau selesai), menyatakan titik awal dan titik akhir diagram alir.
	**Proses** (Tanda komputasi), menyatakan sebarang proses.
	**Keputusan** (Pengambilan keputusan), digunakan untuk pengambilan keputusan. Dalam hal ini yang ada didalam symbol ini berupa suatu pertanyaan yang jawabannya berupa Ya atau Tidak.
	**Proses Terdefinisi** (Tanda prosedur atau fungsi subalgoritma), menyatakan prosedur lain yang telah didiagramalirkan.
	**Konektor** (Tanda penghubung), digunakan untuk menghubungkan ke berbagai bagian dalam alir diagram.

12. Metode Pengujian

a. *White Box*

White box testing adalah suatu pengujian oleh *software* atau aplikasi dengan melihat modul untuk menganalisis kode program ada yang salah atau tidak. Bila suatu modul yang diproduksi tidak memenuhi syarat, kode akan dikompilasi ulang dan periksa lagi hingga mencapai apa yang diharapkan dalam arti bahwa *white box* testing menguji dengan cara melihat pure code dari suatu perangkat lunak atau aplikasi yang diuji tanpa mempedulikan tampilan atau UI dari aplikasi tersebut.



Gambar 2. 2 *White Box Testing*

1) Teknik

Pada white box testing ada beberapa teknik yang digunakan. Berikut akan kami sajikan :

a) *Basis path testing* adalah metode yang memungkinkan perancang tast case untuk membuat pengukuran kompleksitas logika dari rancangan prosedural dan menggunakan pengukuran untuk mendefinisikan himpunan basis dari jalur eksekusi.

b) *Flow graph* merupakan notasi sederhana guna mempresentasi control flow.

c) *Cyclomatic complexity* yang digunakan guna mengetahui jumlah path yang perlu dicari. Di mana *cyclomatic complexity* merupakan metric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika program.

d) *Branch coverage* merupakan pengujian yang didesain agar masing-masing branch code yang paling tidak diuji satu kali.

e) *Loop testing* merupakan testing pada berbagai looping/perulangan yang terdapat pada program, seperti *for, while, dan do-while*. Dengan pengujian ini Anda bisa mengecek kondisi perulangan, sudahkah berjalan dengan benar ataupun tidak.

f) *Condition coverage* merupakan teknik pengujian yang bertujuan guna melakukan uji semua kode untuk menghasilkan value FALSE atau *TRUE*. *Teste*r bisa memastikan software dalam beroperasi dan menghasilkan output sesuai *input user*.

g) *Multiple condition coverage* merupakan teknik yang digunakan guna menguji semua kombinasi beragam kode yang digunakan dalam berbagi kondisi.

h) *Statement coverage* merupakan teknik yang minimal 1 kali dilakukan guna menguji masing-masing statement pada *software*. Anda bisa mengetahui beberapa code yang error dengan pengujian ini sehingga bisa segera memperbaikinya.

2) Trik Pengujian

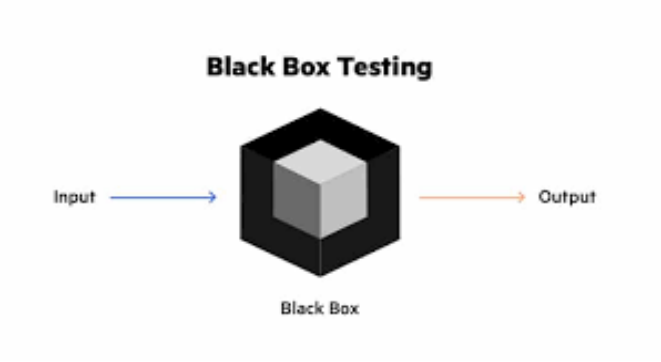
White box testing merupakan pengujian dengan dasar pada pengecekan di dalam detail desain dari suatu rancangan pemrograman untuk membagi pengujian. Dalam membuat pengujian white box maka tester perlu memiliki pemahaman terkait pemrograman. Sebab untuk memahami kode sumber yang dipelajari dan harus memiliki pandangan global tentang fungsi aplikasi serta elemen yang dirancangnya. Metode white test ini dapat diterapkan untuk tes integrasi, tes unit dan tes sistem.

3) Contoh *White Box*

Contoh *white box* testing menggunakan teknik *cyclomatic*, di mana pada dasarnya dengan cara menghitung jumlah siklomatik yaitu menghitung jumlah conditional I predikat dan tambahkan 1. Misalnya terdapat lima *conditional* maka *cyclomaticnya* adalah 6. Berikut cara menghitungnya :

a) 1-2-3-4-5-10 yaitu suatu properti yang dimiliki orang lain, tidak mempunyai uang untuk sewa
b) 1-2-3-4-6-10 suatu properti yang dimiliki orang lain dan membayar sewa
c) 1-2-3-10 yaitu properti yang dimiliki oleh pemain
d) 1-2-7-10 yaitu properti yang tersedia dan tidak memiliki cukup uang
e) 1-2-7-8-10 yaitu properti yang tersedia, memiliki uang dan tidak ingin membelinya
f) 1-2-7-8-9-10 yaitu properti yang tersedia, memiliki uang dan membelinya. (Sumber : *https://itbox.id/blog/white-box-testing-adalah/*)

b. *Black Box*



Gambar 2. 3 *Black Box Testing*

Pengujian *black box* adalah pengujian yang dilakukan semata-mata dengan memantau hasil eksekusi melalui data pengujian dan memvalidasi fungsionalitas perangkat lunak. Jadi, mirip dengan melihat kotak hitam, kita hanya bisa melihat tampilan luarnya saja dan tidak tahu apa yang ada di balik bungkus hitamnya. Mirip dengan pengujian *black box*, hanya mengevaluasi tampilan luar (antarmuka) dan fungsionalitasnya. Tanpa mengetahui apa yang terjadi dalam proses detail (hanya mengetahui input dan output).

1) Jenis Pengujian

Pada suatu black box testing terdapat tiga jenis pengujian, berikut akan kami jelaskan :

a) *Functional testing*

Merupakan suatu proses pengujian pada fitur spesifik atau fungsi pada suatu software. Pengujian ini bisa dilakukan guna memastikan bahwa user tidak dapat masuk pada aplikasi yang diuji tanpa informasi sehingga program tersebut terjaga keamanannya.

b) *Non functional testing*

Memiliki fungsi yang tidak kalah dari functional testing, di mana Non functional testing merupakan suatu pengujian pada beberapa aspek tambahan non fungsional. Bila functional testing menguji software menjalankan suatu tugas atau peringkat.

c) *Regression testing*

Jenis ini merupakan pengujian yang dilakukan pada software untuk mengamati apakah terjadinya kemunduran atau regresi ketika software diperbaharui. Dalam regression test dilakukan pengecekan aspek non fungsional maupun fungsional.

2) Teknik Menguji *Black Box* Dengan *Software*

Tulisan di atas sudah menjelaskan apa itu black box testing dan contohnya. Namun *black box* testing memiliki beberapa teknik dalam menguji suatu software. Berikut akan kami jelaskan :

a*) Equivalence partitioning* merupakan cara kerja dengan melakukan pembagian atau partition menjadi beberapa partisi dari input data.

b) *Fuzzing merupakan* teknik untuk mencari gangguan atau bug dari software dengan cara injeksi data yang terbilang sesi semi otomatis atau cacat.

c) *Boundary value analysis* merupakan teknik yang lebih fokus pada *boundary* di mana jika ada error dari sisi atau luar baik minimum maupun maksimum pada pada software yang didapat.

d) *Orthogonal array testing* merupakan teknik yang digunakan jika input domain yang relatif ukurannya kecil tetapi cukup berat jika digunakan dalam skala besar.

e) *Cause effect graph* merupakan teknik yang digunakan di mana graphic sebagai pacuannya. Dalam teknik ini di mana graphic menggambarkan relasi antara penyebab dan efek dari error tersebut.

f) *State transition* merupakan testing yang digunakan untuk mengecek apakah kondisi dari navigasi dan mesin dalam bentuk grafik. (Sumber : *https://itbox.id/blog/black-box-testing-adalah/).*

B. Kajian Hasil Penelitian Terdahulu

Dalam penelitian ini, mengacu pada penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan. Berikut beberapa hasil penelitian yang relevan atau berhubungan dengan penelitian yang dijadikan bahan telaah bagi penelitian yang akan dilakukan, antara lain :

1. Penelitian yang dilakukan oleh: Muh Jayadi (2021), Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Parepare yang berjudul "Aplikasi pemesanan menu pada warkop sharelock berbasis android". Aplikasi coffee shop berbasis android yang banyak dipakai saat ini di gunakan untuk kegiatan pemesanan menu dan melakukan pembayaran. Seperti yang terjadi saat ini di salah satu warkop yang ada di Kota Parepare tidak lain adalah warkop shearlock masih menggunakan pemesanan yang manual, sehingga membuat penulis memunculkan ide-ide untuk mengadakan pengembangan di warkop tersebut, salah satunya ialah dalam hal pemesanan menu yang di buat dengan berbasis android.

2. Penelitian yang dilakukan oleh: M. Shandy Putra (2019) Program Studi Ilmu Komputer, Fakultas *Teknologi informasi*, yang berjudul "Aplikasi Penyedia Jasa Online Berbasis Mobile Android" Layanan penyedia jasa merupakan layanan yang

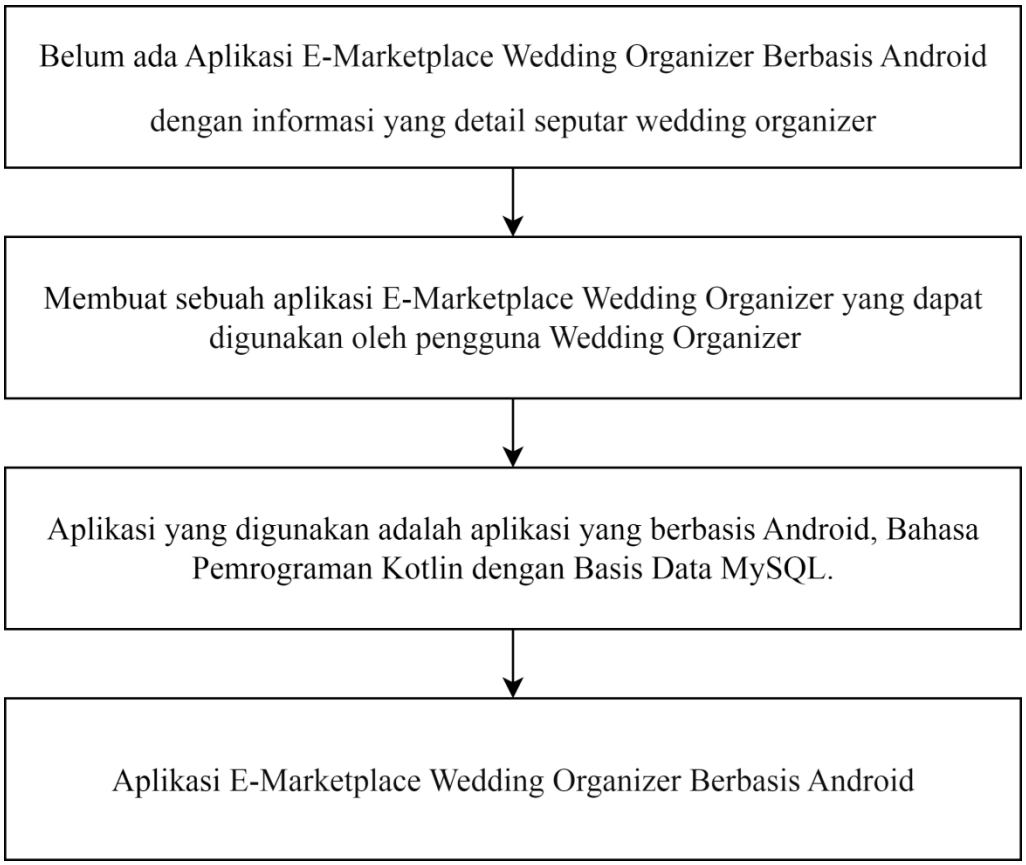
sangat penting dan sangat dibutuhkan saat ini. Keinginan orang untuk membuat, membeli, menjual, dan memperbaiki sesuatu membuat penyedia jasa sangat dibutuhkan. Banyaknya penyedia jasa saat ini membuat kita terkadang bingung untuk memilih dan menentukan penyedia jasa mana yang dapat membantu dan cocok dengan keinginan kita sebagai pengguna jasa.

Perbandingan dari penelitian sebelumnya adalah Dari penelitian Muh Jayadi (2021). Aplikasi pemesanan menu pada warkop sharelock berbasis android, Merupakan aplikasi pemesanan atau penyedia menu, sama halnya dengan penelitian M. Shandy Putra (2019) Aplikasi Penyedia Jasa Online Berbasis Mobile Android, juga merupakan aplikasi penyedia bedanya hanya di objek dan aplikasi yang ingin dirancang adalah aplikasi *e-marketplace wedding organizer* berbasis android.

Kesimpulan dari penelitian sebelumnya dan penelitian yang akan dibuat sama-sama aplikasi pemesanan atau penyedia yang dapat meng-*order* dan menerima *order*-an barang dan jasa sesuai dengan kategori yang tersedia di dalam aplikasi.

C. Kerangka Pikir

Untuk memahami kerangka pikir penelitian ini dapat dilihat dibawah ini.



Gambar 2. 4 Kerangka Pikir

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kualitatif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan Skripsi ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat.

Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

B. Waktu dan Tempat Penelitian

Adapun waktu dan tempat penelitian yang dilaksanakan pada bulan Juli tahun 2024, di Nippita Wedding Organizer Kota parepare, dimana proses pengambilan data yang akan dimasukkan dalam aplikasi nantinya.

Tabel 3. 1 Jadwal Penelitian

No	UraianKegiatan	Bulan 2024	
		Juli	Agustus
1	*Studi Literatur*		
2	Perancangan aplikasi		
3	Pembuatan aplikasi		
4	Pengujian		

C. Alat dan Bahan Penelitian

Dalam pelaksanaan penelitian ini digunakan beberapa alat dan bahan yang terdiri dari Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*).Perangkat keras yang digunakan terdiri dari Laptop ASUS. Perangkat lunak yang digunakan adalah *Windows 10 Home Single Language*, *Android Studio, Kotlin, dan MySQL*. Sedangkan bahan penelitian berupa data-data yang berasal dari hasil pengumpulan data yang telah dilakukan oleh peneliti yang nantinya akan dianalisis lebih lanjut sebagai landasan untuk merancang sebuah sistem.

D. Tahap Penelitian

Tahap penelitian dilakukan dengan cara :

1. Persiapan Penelitian

Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel tentang topik penelitian serta software yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini penelitian melakukan observasi dengan peninjauan dan juga studi literatur.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem dan juga data proposal yang diterapkan berdasarkan data yang didapatkan saat melakukan pengumpulan data, kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi e-marketplace wedding organizer yang ingin dibuat berdasarakan alternatif pemecahan masalah.

5. Pengujian

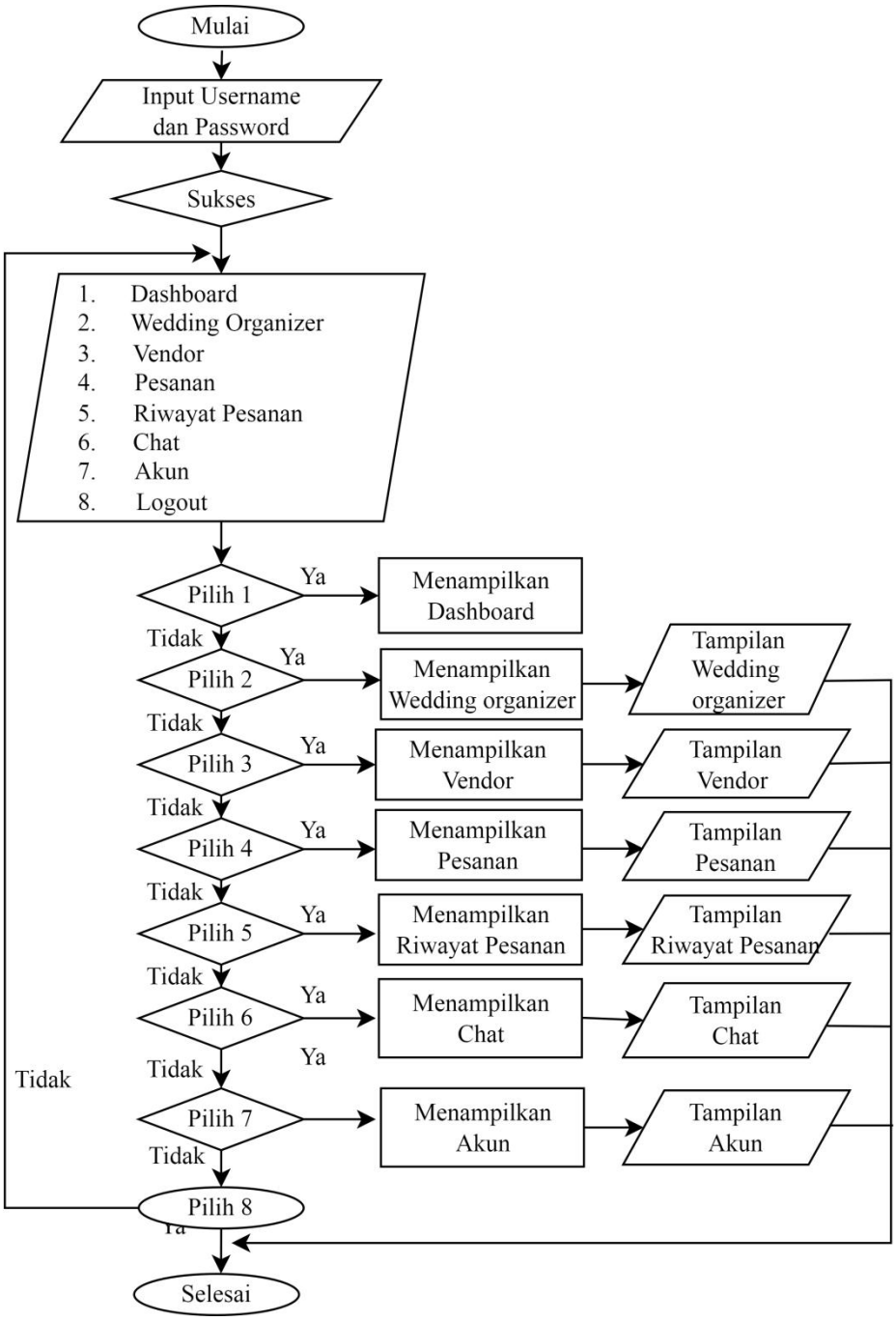
Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan aplikasi.Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

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

E. Analisis Sistem

1. Flowchart Aplikasi



Gambar 3. 1 *Flowchart* Aplikasi

Pada gambar 3.1 yaitu *flowchart* dimana pada suatu *flowchart* pastinya diawali dengan mulai atau start. Kemudian menampilkan tampilan awal aplikasi e-marketplace , apabila sudah punya akun maka lansung masuk ke halaman utama aplikasi, setelah masuk pengguna dapat cari WO, apabila WO yang diinginkan ketemu maka dapat memilih WO dan melakukan pemesanan, setelah melakukan pemesanan lanjut ke pembayaran dan tentunya data pemesanan tersebut tersimpan, kemudian selesai.

2. Sistem Yang Berjalan

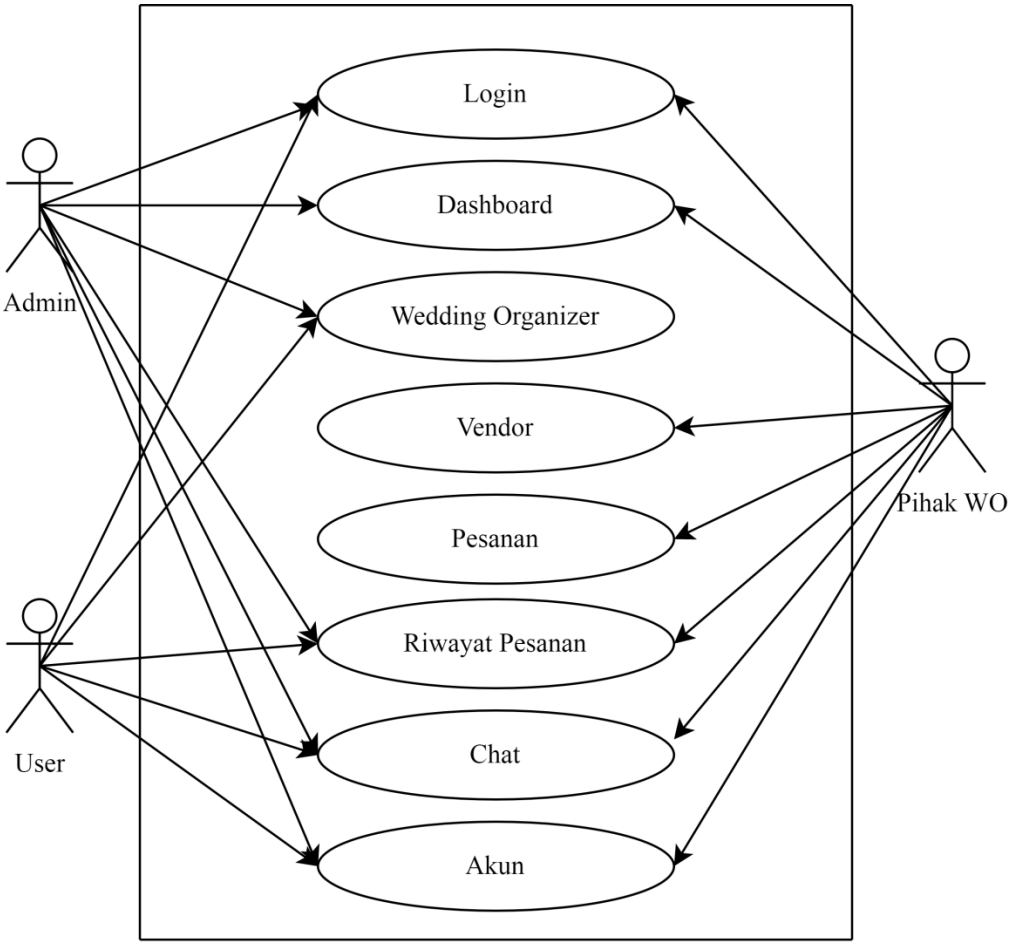
Sistem saat ini digambarkan sebagai dekonstruksi sistem lengkap menjadi bagian-bagian komponennya dengan tujuan mendeteksi dan menganalisis kesulitan dan hambatan yang muncul dalam persyaratan yang diharapkan akan diajukan.



Gambar 3. 2. Sistem Yang Berjalan

Pada gambar 3.2 menggambarkan tentang proses pemesanan WO yang kebanyakan berjalan sekarang ini yaitu masih menggunakan cara lama dengan mendatangi tempat untuk memesan WO dang melakukan transaksi secara *face to face* atau tatap muka langsung.

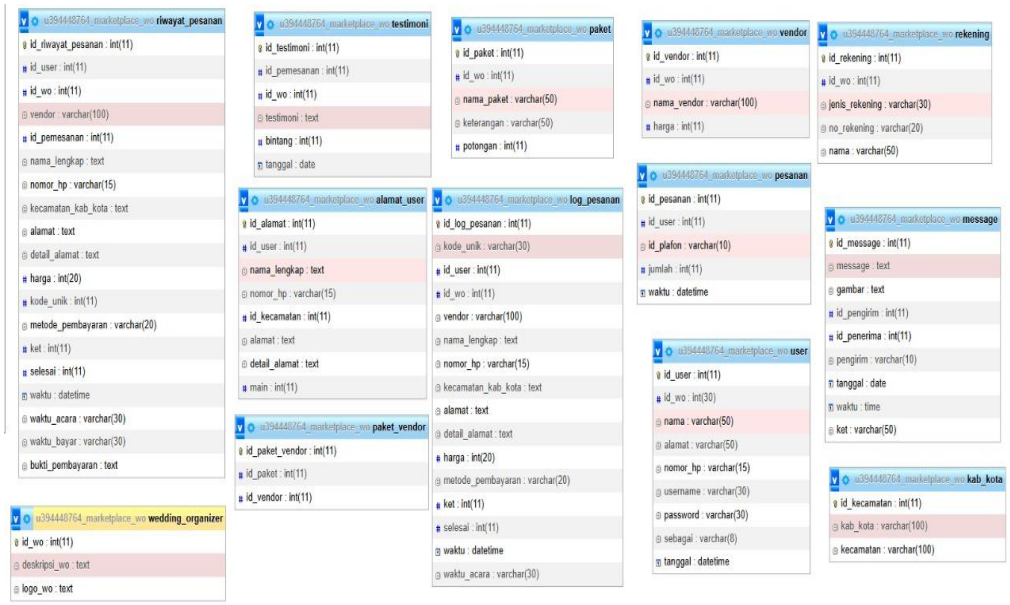
3. Sistem Yang Diusulkan



Gambar 3. 3. Sistem Yang Dusulkan

Keterangan gambar 3.3. Pada gambar diatas Mempunyai tiga aktor Admin, Pihak WO serta Pengguna jasa WO disini Admin login kedalam aplikasi, admin dapat mengakses menu Kelola WO, Kelola user, Kelola review, Kelola menu, Kelola sub menu, Kelola akses. Kemudian Pihak WO dapat login dan dapat mengakses menu Kelola data vendor dan venue, Kelola transaksi pemesanan WO dan Melihat review sedangkan pengguna jasa WO setelah login dapat mengakses menu pilih WO, Pesan WO dan Memberikan komentar.

4. Relasi Antar Tabel



Gambar 3.4 Relasi Antar Tabel

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Selama melakukan penelitian di Kantor Nipitta Organizer Jalan lingkar tassio, Kelurahan galung maloang, Kecamatan bacukiki, Kota Parepare, Sulawesi Selatan, dalam membangun Aplikasi E-Marketplace Wedding Organizer Berbasis Android, didapatkanlah hasil penelitian antara lain. *Pre event, Job descriptio*n dan *Pricing guide.*

Tabel 4 1. *Pre Event*

No	Keterangan *Pre-Event*
1	Diskusi & Merencanakankonsep, waktusertatempat acara Bersama calon & keluarga inti pengantin
2	Memberikan rekomendasi vendor & talent pengisi acara sesuai dengan konsep & budget acara pernikahan
3	Menjadi Narahubung antara vendor serta pengisi acara sebelum & sesudah acara
4	Mengawasi vendor saat loading agar sesuai dengan kontrak antara calon pengantin dan vendor

Tabel 4 2. Job Description

No	*Job Description*	Keterangan
1	*Leader*	Akan berkordinasi langsung dengan calon pengantin & keluarga serta berkomunikasi langsung dengan seluruh vendor & pengisi acara terkait
2	*Braid Ass*	Yang akan mendampingi dan membantu segala kebutuhan pengantin pada saat acara, dimulai dari persiapan Makeu-up hingga seluruh rangkaian acara pernikahan selesai.
3	*Groom Ass*	Yang akan mendampingi dan membantu segala kebutuhan pengantin Pria & Keluarga pada saat acara, dimulai dari persiapan sebelum keberangkatan hingga seluruh rangkaian acara pernikahan selesai.

Lanjutan Tabel 4.2 *Job Description*

4	*Pic Entertaint*	PIC entertaint yang akan berkordinasi & bertanggung jawab terhadap semua talent & rundown pada hari H
5	*Pic Souvenir*	Bekerjasama dengan team Pagar ayu & bertanggung jawab atas perhitungan jumlah souvenir & system pembagiannya
6	*Pic Stage*	Bertanggung jawab dan memastikan area pelaminan agar tidak menumpuk saat acara serta membantu tim FG saat foto session di pelaminan.
7	*Pic Cathering*	PIC CATERING/FOOD STALL Berkordinasi langsung dengan vendor catering & keluarga mengenai pengaturan prasmanan, kue & minuman pada saat hari H.
8	*Pic Vip*	PIC VIP area bekerja sama dengan pihak keluarga (Among Tamu) atas area VIP.
9	*Pic Fotografer*	Bekerjasama dengan tim FG dalam hal seluruh sesi pegambilan gambar & seluruh kebutuhan untuk sesinya.

Tabel 4 3. Pricing Guide

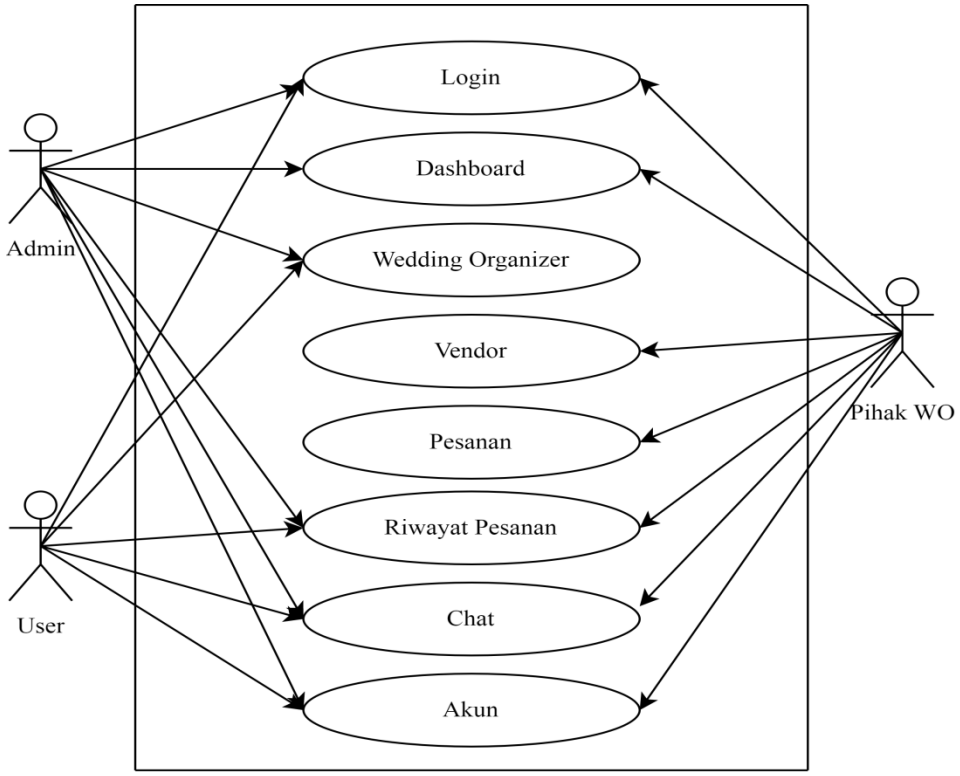
No	Nama Paket	Harga	Keterangan
1	Basic Package (Akad & Resepsi)	7.000.000	-10 CREW PROFESIONAL - Free Undangan Website - Free Filter Instagram - Free Akomodasi
2	Premium Package (Mapacci, Akad & Resepsi)	8.500.000	- 10 - 12 CREW PROFESIONAL - Free Undangan Website - Free Sesi Siraman - Free 12 BOX SESERAHAN
3	Extra Package (All Prosesi)	12.000.000	- 10 - 12 CREW PROFESIONAL - Free Undangan Website - Free Sesi Siraman - Free 12 BOX SESERAHAN

B. Pembahasan

1. Analisis Aliran Data UML

a. *Use Case Diagram*

Use Case Diagram atau diagram *use case* merupakan pemodelan untuk memodelkan kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi. *Use case* juga dapat digunakan untuk merepresentasikan interaksi yang terjadi antara aktor dengan proses sistem yang dibuat. Pada perancangan *use case* juga terdapat skenario, yaitu langkah yang menerangkan urutan kejadian antara pengguna dengan sistem. Diagram *use case* pada Aplikasi ini digambarkan seperti berikut ini:



Gambar 4 1. *Use Case Diagram.*

b. Tabel *Use Case*

Penjabaran use case diagram di atas didefinisikan atas definisi aktor dan definisi use case. Definisi aktor sebagai berikut :

Tabel 4 1. Definisi Aktor dalam Sistem.

No	Aktor	Keterangan
1	*User*	*User* merupakan aktor yang berperan sebagai pengguna aplikasi Wedding Organizer.
2	Admin	Admin merupakan aktor yang memiliki hak akses sebagai pengelola aplikasi.
3	Pihak Wedding Organizer	Pihak Wedding Organizer merupakan aktor yang berperan sebagai pengguna aplikasi Wedding Organizer dengan akses yang lebih banyak dari pada admin dan user.

Sedangkan untuk penjelasan definisi setiap use case sebagai berikut:

Tabel 4 2. Definisi Use Case Sistem

No	*Use Case*	Keterangan
1	Login	Menampilkan halaman login aplikasi
2	Dashboard	Menampilkan halaman utama aplikasi
3	Wedding organizer	Menampilkan halaman Wedding organizer
4	Vendor	Menampilkan halaman Vendor
5	Pesanan	Menampilkan halaman Pesanan
6	Riwayat pesanan	Manampilkan halaman Riwayat pesanan
7	Chat	Manampilkan halaman Chat
8	Akun	Menampilkan halaman Akun

Dalam use case yang telah didefinisikan di atas, terdapat beberapa use case yang memiliki alur proses yang hampir sama, sehingga dalam penjabaran

selanjutnya akan didefinisikan secara ringkas untuk memudahkan pemahaman proses tiap use case. Berikut adalah skenario use case yang telah didefinisikan sebagai berikut :

1) Use Case Login

Aktor : Admin, Pihak Wo dan User

Deskripsi : Sebuah kegiatan untuk masuk kedalam aplikasi.

Pre-condition : Admin, Pihak Wo dan user berada di halaman login.

Post-condition : -

Tabel 4 3. Skenario *Use Case Login*

Aksi Aktor	Reaksi Sistem
1. Masuk kehalaman login dan Memasukkan Username dan Password.	
	2. Masuk ke halaman utama aplikasi

2) Use Case Home.

Aktor : Admin, Pihak Wo dan User

Deskripsi : Sebuah kegiatan untuk menampilkan menu-menu aplikasi.

Pre-condition : Admin, Pihak Wo dan user berada di halaman utama.

Post-condition : - .

Tabel 4 4. Skenario *Use Case Home*.

Aksi Aktor	Reaksi Sistem
1. Masuk kehalaman utama aplikasi	
	2. Menampilkan menu-menu aplikasi.

3) Use Case Wedding Organizer

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk menampilkan wedding organizer.

Pre-condition : Admin dan user berada di halaman wedding organizer.

Post-condition : -

Tabel 4 5. Skenario *Use Case* Wedding Organizer

Aksi Aktor	Reaksi Sistem
1. Memilih menu Wedding Organizer	
	2. Menampilkan form Wedding Organizer

4) Skenario Use Case Vendor

Aktor : Pihak Wo

Deskripsi : Sebuah kegiatan untuk melihat vendor.

Pre-condition : Pihak Wo berada di halaman vendor

Post-condition : -

Tabel 4 6. Skenario Use Case Vendor

Aksi Aktor	Reaksi Sistem
1. Menekan menu Vendor	
	2. Menampilkan data Vendor

5) Use Case Pesanan

Aktor : Pihak Wo

Deskripsi : Sebuah kegiatan untuk melihat pesanan.

Pre-condition : Pihak Wo berada di halaman pesanan .

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 7. Skenario Use Case Pesanan

Aksi Aktor	Reaksi Sistem
1. Menekan menu pesanan	
	2. Menampilkan menu pesanan

6) Use Case Riwayat Pesanan

Aktor : Admin, Pihak Wo danUser

Deskripsi : Sebuah kegiatan untuk melihat riwayat pesanan.

Pre-condition : Admin, Pihak Wo danUser berada di halaman riwayat pesanan

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 8. Skenario Use Case Pengaduan Bencana

Aksi Aktor	Reaksi Sistem
1.Menekan menu riwayat pesanan	
	1. Menampilkan menu riwayat pesanan

7) Use Case Chat

Aktor : Admin, Pihak Wo dan User

Deskripsi : Sebuah kegiatan untuk melihat chat.

Pre-condition : Admin, Pihak Wo danUser berada di halaman chat.

Post-condition : -

Tabel 4 9. Skenario Use Case Chat

Aksi Aktor	Reaksi Sistem
1.Menekan menu chat	
	2. Menampilkan menu Chat

8) Use Case Akun

Aktor : Admin, Pihak Wo dan User

Deskripsi : Sebuah kegiatan untuk melihat chat.

Pre-condition : Admin, Pihak Wo danUser berada di halaman chat.

Post-condition : -

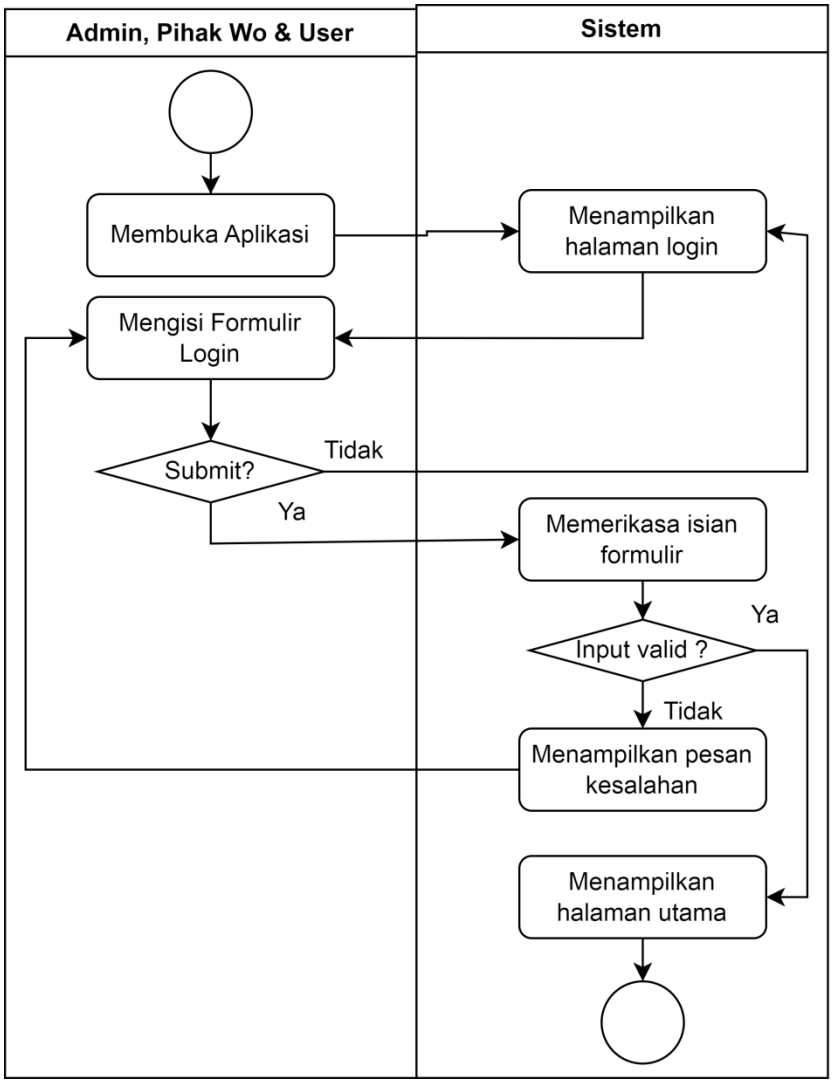
Tabel 4 4. Skenario Use Case Akun

Aksi Aktor	Reaksi Sistem
1.Menekan menu akun	
	2. Menampilkan menu akun

2. Activity Diagram

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah system.

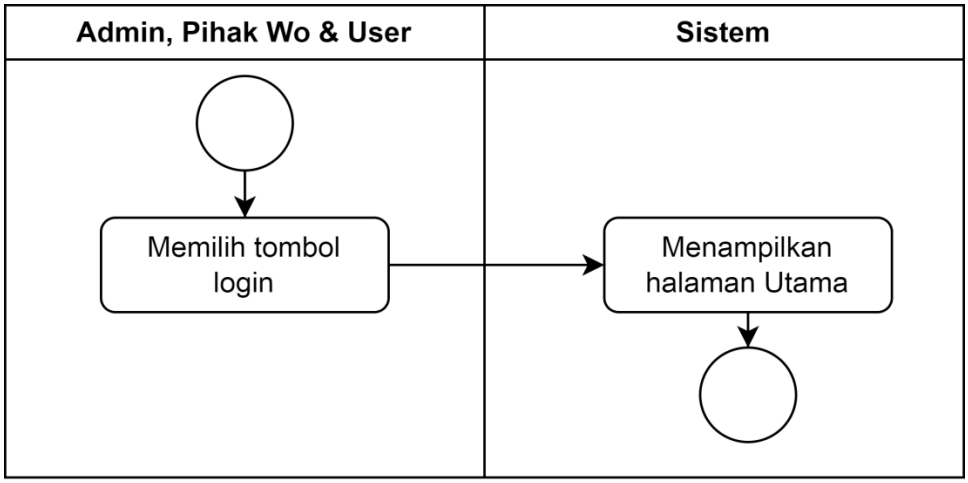
a. ***Activity Diagram Login***



Gambar 4 2. *Activity Diagram Login*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas login. Proses dimulai dengan admin, pihak wo atau user memilih tombol login, kemudian sistem menampilkan form isian yang kemudian admin atau user akan mengisi form isian. Ketika sudah disimpan, sistem akan memperbarui data di database yang akan mengecek data yang dikirim, jika terdapat kesalahan, akan menampilkan pesan kesalahan, jika tidak, sistem menampilkan halaman utama.

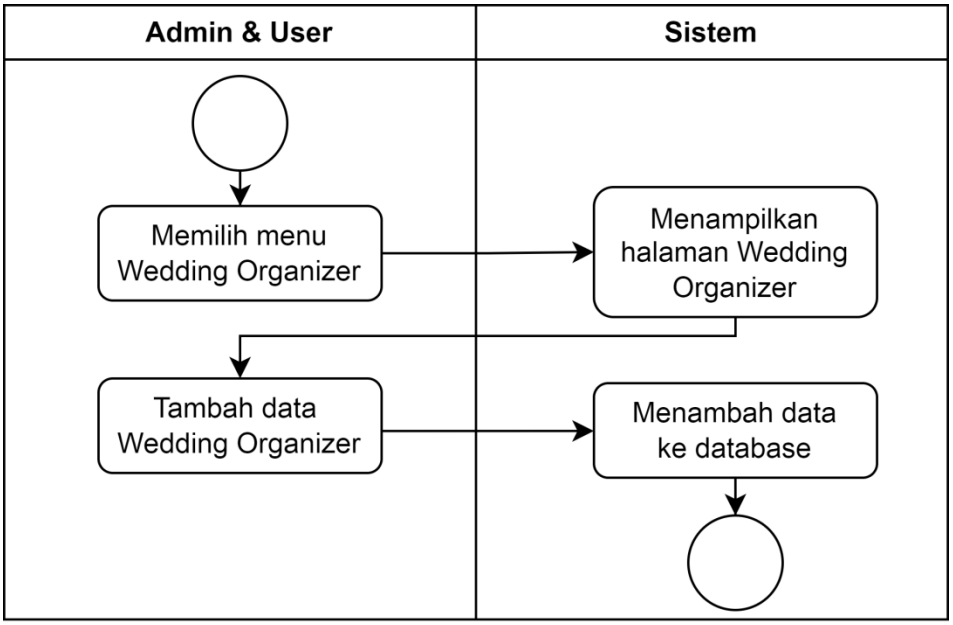
b. ***Activity Diagram Home***



Gambar 4 3. Activity Diagram Home

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas Home. Proses dimulai dengan admin, Pihak Wo atau *User* memilih tombol login, kemudian system berhasil menampilkan halaman utama aplikasi.

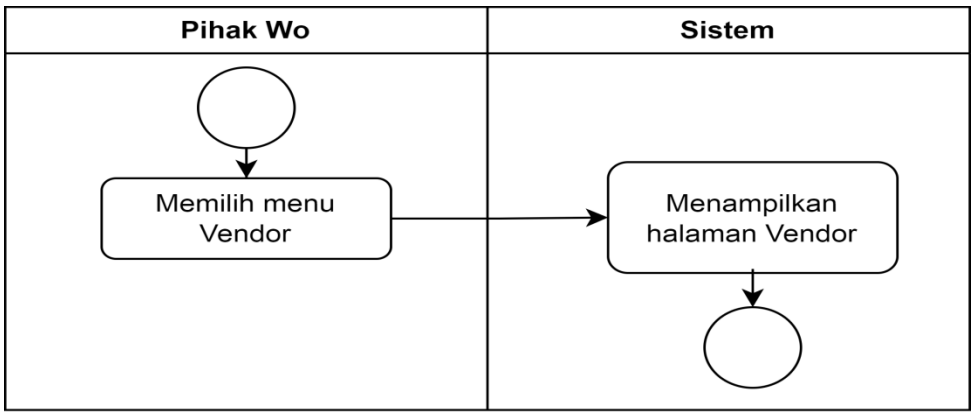
c. ***Activity*** **Diagram Wedding Organizer**



Gambar 4 4. *Activity Diagram* Wedding Organizer

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas wedding organizer Proses dimulai dengan admin maupun user memilih tombol wedding organizer, kemudian sistem menampilkan halaman wedding organizer. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

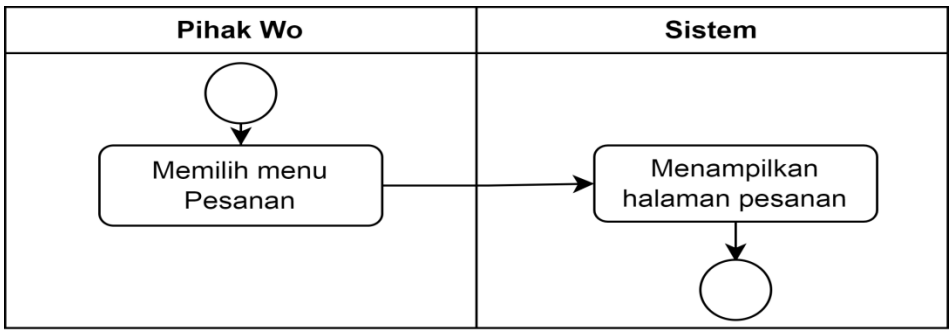
d. *Activity* Diagram Vendor



Gambar 4 5. *Activity Diagram* Vendor

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas vendor. Proses dimulai dengan pihak Wo memilih menu vendor, kemudian sistem menampilkan halaman vendor.

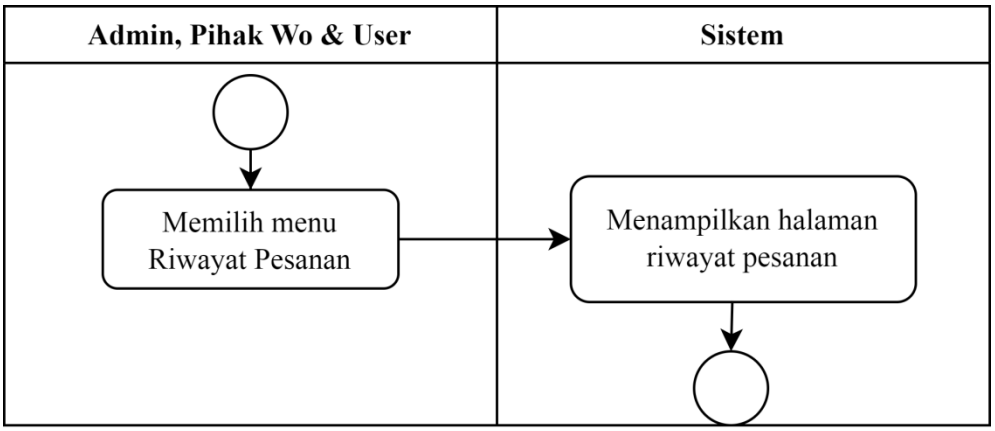
e. *Activity* Diagram Pesanan



Gambar 4 6. *Activity* Diagram Pesanan

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas pesanan. Proses dimulai dengan pihak Wo memilih menu pesanan, kemudian sistem menampilkan halaman pesanan.

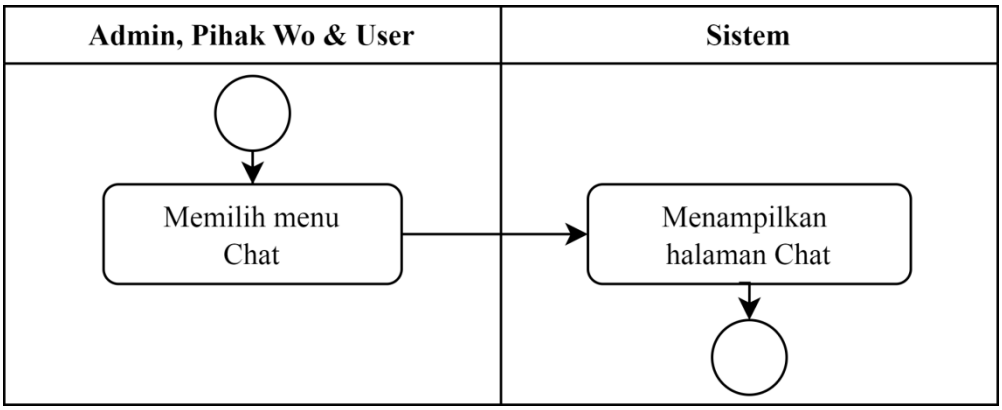
f. *Activity* Diagram Riwayat Pesanan



Gambar 4 7. *Activity* Diagram Riwayat Pesanan

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas riwayat pesanan. Proses dimulai dengan admin, pihak Wo dan user memilih menu riwayat pesanan, kemudian sistem menampilkan halaman riwayat pesanan.

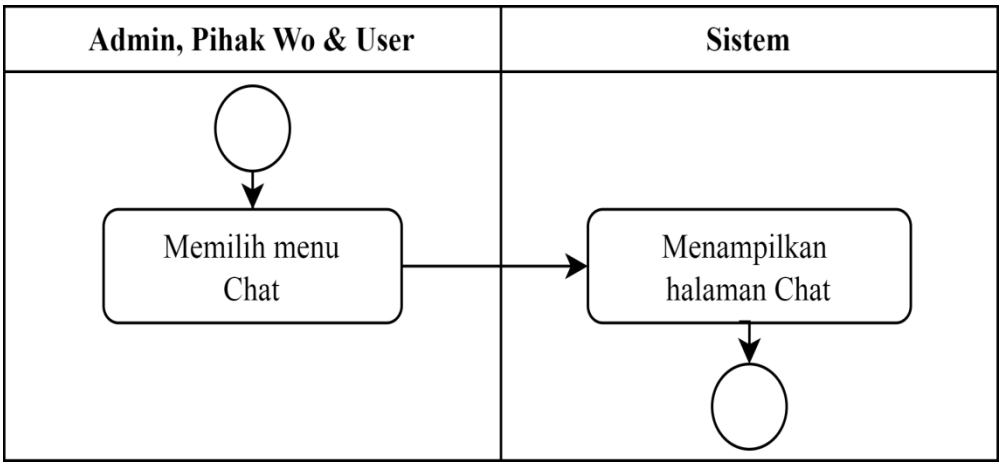
g. *Activity* Diagram Chat



Gambar 4 8. *Activity* Diagram Chat

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas chat. Proses dimulai dengan admin, pihak Wo dan user memilih menu chat, kemudian sistem menampilkan halaman chat.

h. *Activity* Diagram Akun



Gambar 4 9. *Activity* Diagram Akun

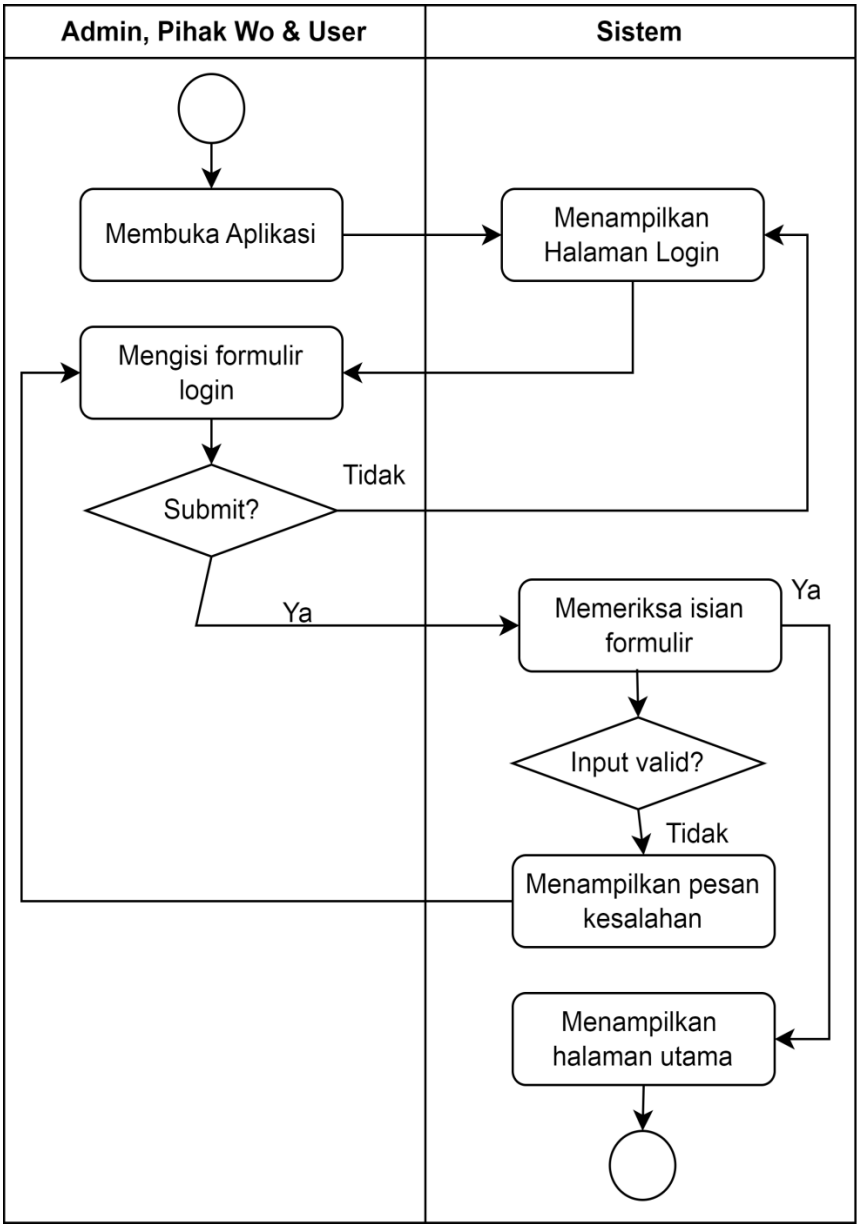
Diagram aktivitas di atas menerangkan alur proses untuk aktivitas akun. Proses dimulai dengan admin, pihak Wo dan user memilih menu akun, kemudian sistem menampilkan halaman akun.

3. Sequence Diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripisikan waktu hidup objek dan message yang dikirim dan diterima antar objek. Proses menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta method yang dimiliki kelas yang diinisialisasi menjadi objek yang sudah tergambar dalam class diagram.

Berdasarkan desain use case, terdapat beberapa use case yang prosesnya hampir sama satu sama lain. Untuk mempermudah pembaha-san proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada aplikasi wedding organizer.

a. *Sequence Diagram*: *Login*

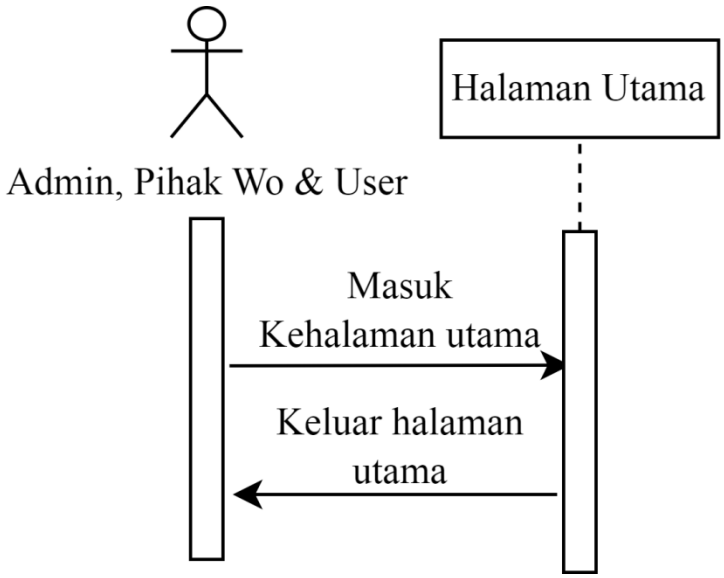


Gambar 4 10. *Sequence* Diagram login

Diagram Sequence diagram login Admin, Pihak Wo dan user merupakan penggambaran proses login yang dilakukan oleh admin, pihak wo dan *user*. Bila proses login benar admin, pihak wo

dan user bisa masuk ke dalam halaman utama sebaliknya bila proses login gagal akan kembali ke menu *login*.

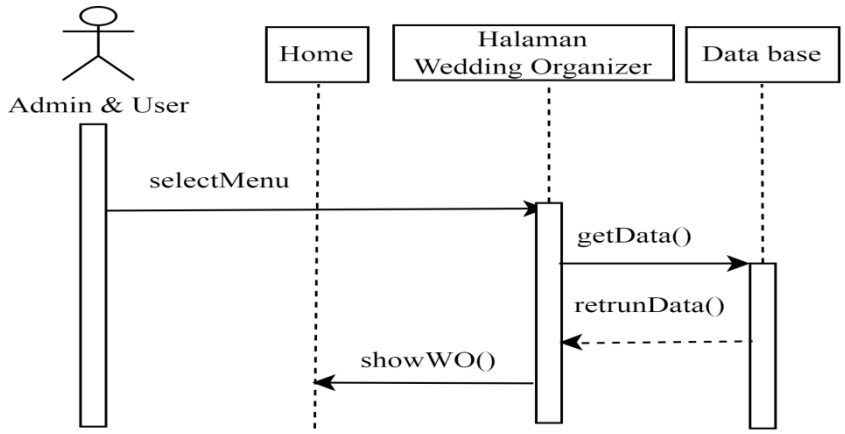
b. *Sequence Diagram*: Home



Gambar 4 11. *Sequence* Diagram Home

Diagram Sequence diagram home Admin dan user merupakan penggambaran proses masuk ke halaman utama. Bila proses login benar admin dan user bisa masuk ke dalam halaman utama dan menampilkan menu-menu yang ada pada aplikasi.

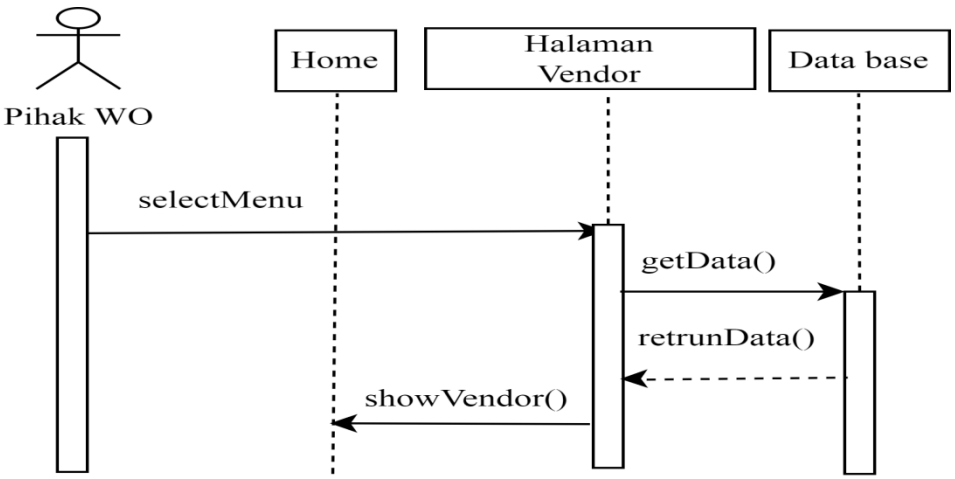
c. *Sequence Diagram* Wedding Organizer



Gambar 4 12. *Sequence* Diagram Home

Diagram sekuen di atas menerangkan alur proses untuk aktivitas wedding organizer. Admin dan user memilih data yang akan ditampilkan pada halaman wedding organizer. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showWeddingOrganizer() akan menampilkan Wedding Organizer.

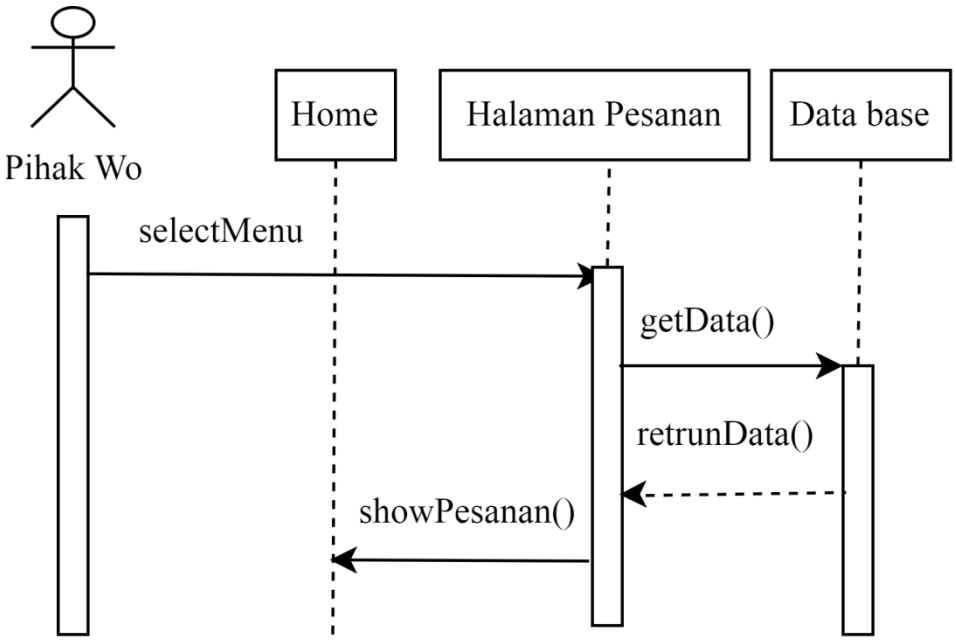
d. *Sequence Diagram* Vendor



Gambar 4 13. *Sequence* Diagram Vendor

Diagram sekuen di atas menerangkan alur proses untuk aktivitas vendor. pihak Wo memilih data yang akan ditampilkan pada halaman vendor, kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showVendor() akan menampilkan vendor.

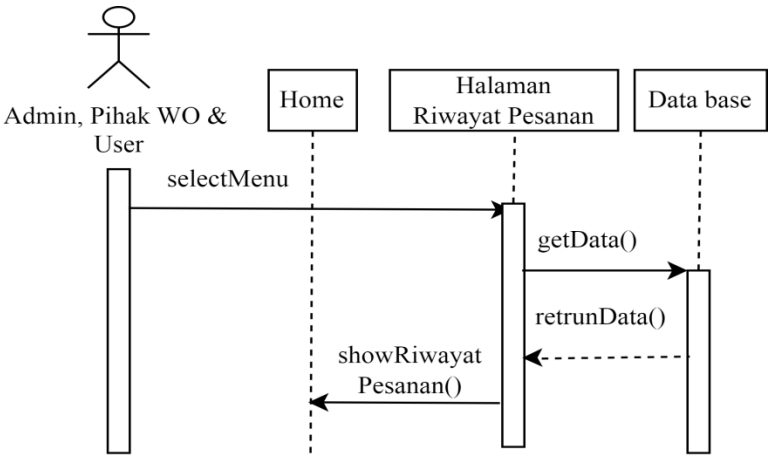
e. *Sequence Diagram* Pesanan



Gambar 4 14. *Sequence* Diagram Pesanan

Diagram sekuen di atas menerangkan alur proses untuk aktivitas pesanan. Pihak Wo memilih data yang akan ditampilkan pada halaman pesanan, kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showPesanan() akan menampilkan pesanan.

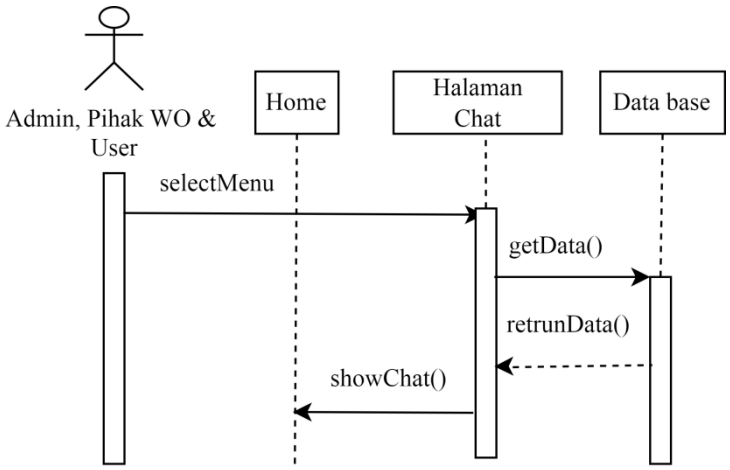
f. *Sequence Diagram* Riwayat Pesanan



Gambar 4 15. *Sequence* Diagram Riwayat Pesanan

Diagram sekuen di atas menerangkan alur proses untuk aktivitas riwayat pesanan. Admin, Pihak Wo dan User memilih data yang akan ditampilkan pada halaman riwayat pesanan, kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showRiwayatPesanan() akan menampilkan riwayat pesanan.

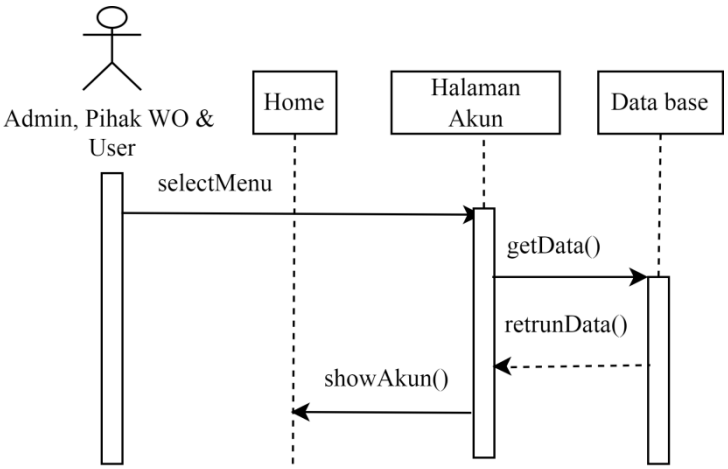
g. *Sequence Diagram* Chat



Gambar 4 16. *Sequence* Diagram Chat

Diagram sekuen di atas menerangkan alur proses untuk aktivitas chat. Admin, Pihak Wo dan User memilih data yang akan ditampilkan pada halaman chat, kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showChat() akan menampilkan chat.

h. *Sequence Diagram* Akun



Gambar 4 17. *Sequence* Diagram Akun

Diagram sekuen di atas menerangkan alur proses untuk aktivitas akun. Admin, Pihak Wo dan User memilih data yang akan ditampilkan pada halaman chat, kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showAkun() akan menampilkan akun.

4. Kamus Data

Kamus data merupakan kumpulan data yang digunakan serta dihasilkan oleh perangkat lunak. Berikut ini daftar seluruh data yang akan digunakan dan dihasilkan oleh perangkat lunak sistem marketplace wedding organizer.

Tabel 4.11. marketplace_wo
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_user	int	11
2	id_wo	int	11
3	nama	Varchar	30
4	alamat	Varchar	100
5	nomor_hp	Varchar	15
6	username	Varchar	30
7	password	Varchar	30
8	sebagai	Varchar	8

Tabel 4. 12 *wedding organizer*
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_wo	int	11
2	deskripsi_wo	text	-
3	logo_wo	text	-

Tabel 4. 13 riwayat Pesanan
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_riwayat_pesanan	int	11
2	id_user	int	11
3	id_wo	int	11
4	vendor	Varchar	100
5	id_pesanan	int	11
6	nama_lengkap	text	-
7	nomor_hp	Varchar	15
8	kecamatan_kab_kota	text	-
9	alamat	text	-
10	detail_alamat	text	-

Lanjutan Tabel

11	harga	int	20
12	metode_pembayaran	Varchar	20
13	ket	text	-
14	selesai	text	-
15	waktu	datatime	-
16	waktu_acara	Varchar	30
17	waktu_bayar	Varchar	30

Tabel 4. 14. testimoni
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_testimoni	int	11
2	id_pesanan	int	11
3	id_wo	int	11
4	Testimony	text	-
5	bintang	int	11
6	tanggal	date	-

Tabel 4. 15. alamat_user
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_alamat	int	11
2	id_user	int	11
3	nama_lengkap	text	-
4	nomor_hp	Varchar	15
5	id_kecamatan	int	11
6	alamat	text	-
7	detail_alamat	text	-
8	main	int	11

Tabel 4. 16. Vendor
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id_vendor	int	11
2	id_wo	int	11
3	nama_vendor	Varchar	100
4	harga	int	11

Tabel 4. 17. Message
Primary key
Foreign key

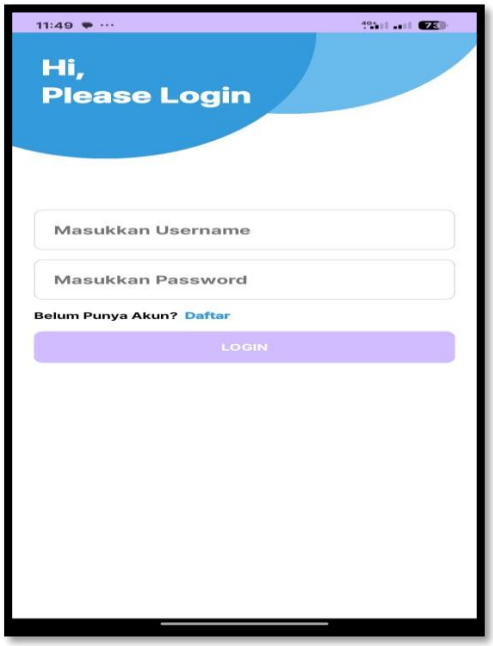
No	Nama Field	Tipe Data	Ukuran
1	id_message	int	11
2	message	text	-
3	gambar	text	-
4	id_pengirim	int	11
5	id_penerima	int	11
6	pengirim	Varchar	10
7	tanggal	date	-
8	waktu	time	-
9	ket	Varchar	50

5. Perancangan Sistem

a. Halaman Admin

1) Halaman Login

Berikut adalah tampilan login pada Aplikasi E-Marketplace Wedding Organizer. Halaman menu login merupakan tampilan yang ada pada sistem, yang digunakan untuk masuk ke halaman selanjutnya dengan memasukkan *username* dan *password*.



Gambar 4 18. Halaman Login

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.login
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import
com.abcd.vian_marketplaceweddingorganizer.data.model.UsersModel
importcom.abcd.vian_marketplaceweddingorganizer.databinding.Act
ivityLoginBindingimportcom.abcd.vian_marketplaceweddingorganize
r.ui.activity.admin.AdminMainActivityimportcom.abcd.vian_market
placeweddingorganizer.ui.activity.register.RegisterActivitympor
tcom.abcd.vian_marketplaceweddingorganizer.ui.activity.user.mai
n.MainActivityimportcom.abcd.vian_marketplaceweddingorganizer.u
i.activity.wo.main.WeddingOrganizerMainActivityimportcom.abcd.v
ian_marketplaceweddingorganizer.utils.LoadingAlertDialog.
```

2) Halaman Dashboard

Berikut adalah tampilan dashboard pada Aplikasi E-Marketplace Wedding Organizer. Halaman dashboard dalam aplikasi ini berfungsi sebagai pusat informasi utama dan titik navigasi bagi pengguna.



Gambar 4 19. Halaman Home

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import androidx.navigation.compose.*
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            DashboardApp()
        }
    }
}
@Composable
fun DashboardApp() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination =
"dashboard") {
        composable("dashboard") {
DashboardScreen(navController) }
        composable("profile") { ProfileScreen(navController) }
    }
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Welcome to Dashboard", fontSize =
24.sp, fontWeight = FontWeight.Bold)
            Spacer(modifier = Modifier.height(20.dp))
            Button(onClick = {
navController.navigate("profile") }) {
                Text("Go to Profile")
            }
        }
    }
}
```

3) Halaman Wedding Organizer

Berikut adalah tampilan wedding organizer pada Aplikasi E-Marketplace Wedding Organizer. Halaman wedding organizer dalam aplikasi ini berfungsi untuk menampilkan daftar wedding organizer, memberikan akses ke fitur-fitur tertentu, serta menyediakan informasi detail terkait setiap wedding organizer.



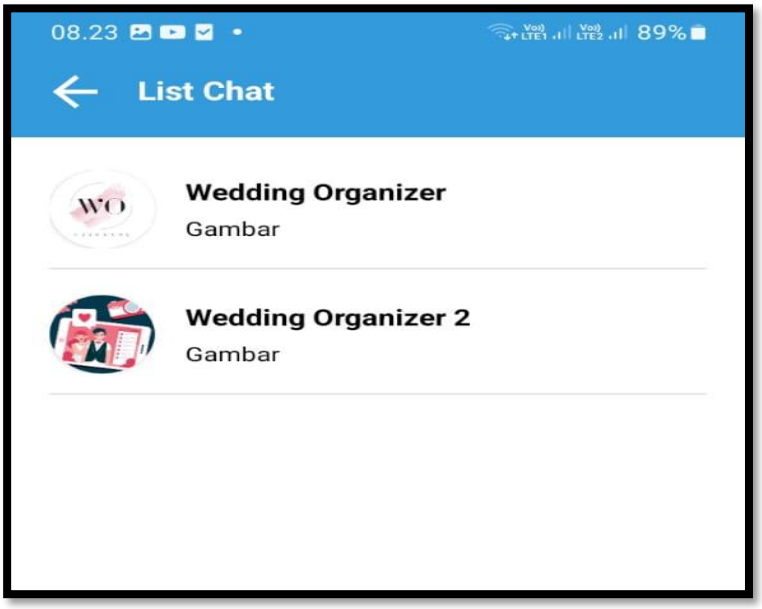
Gambar 4 20. Halaman Wedding Organizer

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            WeddingOrganizerApp()
        }
    }
}
data class WeddingPackage(val name: String, val price: String,
val imageRes: Int)
@Composable
fun WeddingOrganizerApp() {
    val packages = listOf(
        WeddingPackage("Paket   Silver",   "Rp   10.000.000",
android.R.drawable.ic_menu_gallery),
        WeddingPackage("Paket    Gold",    "Rp    20.000.000",
android.R.drawable.ic_menu_gallery),
        WeddingPackage("Paket   Platinum",   "Rp   30.000.000",
android.R.drawable.ic_menu_gallery)
    )

    Scaffold(
        topBar = {
            TopAppBar(title = { Text("Wedding Organizer") })
        }
    ) { padding ->
        LazyColumn(modifier = Modifier.padding(padding)) {
            items(packages) { weddingPackage ->
                PackageItem(weddingPackage)
            }
        }
    }
}
```

4) Halaman Chat

Berikut adalah tampilan chat pada Aplikasi E-Marketplace Wedding Organizer. Halaman chat dalam aplikasi ini berfungsi untuk menampilkan halaman chat.

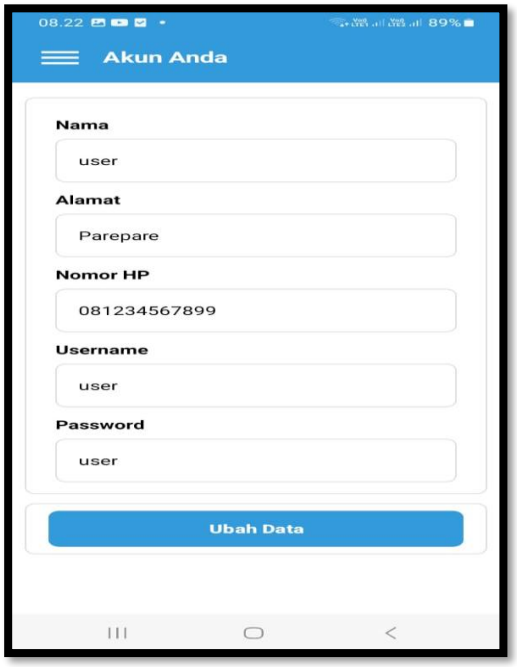


Gambar 4 21. Halaman Chat

```
package com.abcd.vian_marketplaceweddingorganizer.ui.activity.
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.util.Log
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
```

5) Halaman Akun

Berikut adalah tampilan akun pada Aplikasi E-Marketplace Wedding Organizer. Halaman akun dalam aplikasi ini berfungsi untuk menampilkan data akun.



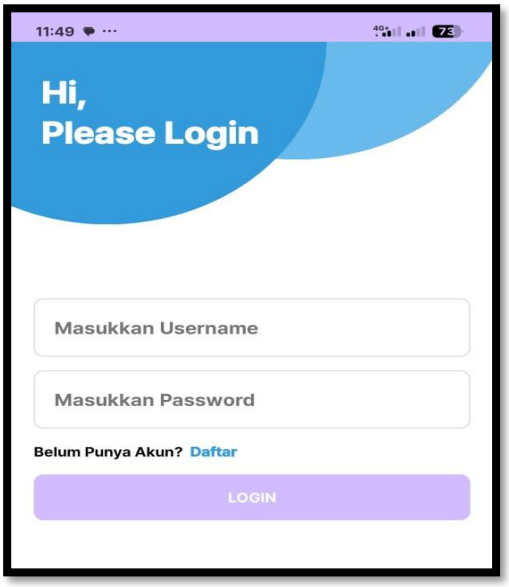
Gambar 4 22. Halaman Akun

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.akun
import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.widget.Toast
import androidx.activity.viewModels
```

b. Halaman User

1) Halaman Login

Berikut adalah tampilan login pada Aplikasi E-Marketplace Wedding Organizer. Halaman menu login merupakan tampilan yang ada pada sistem, yang digunakan untuk masuk ke halaman selanjutnya dengan memasukkan *username* dan *password*.

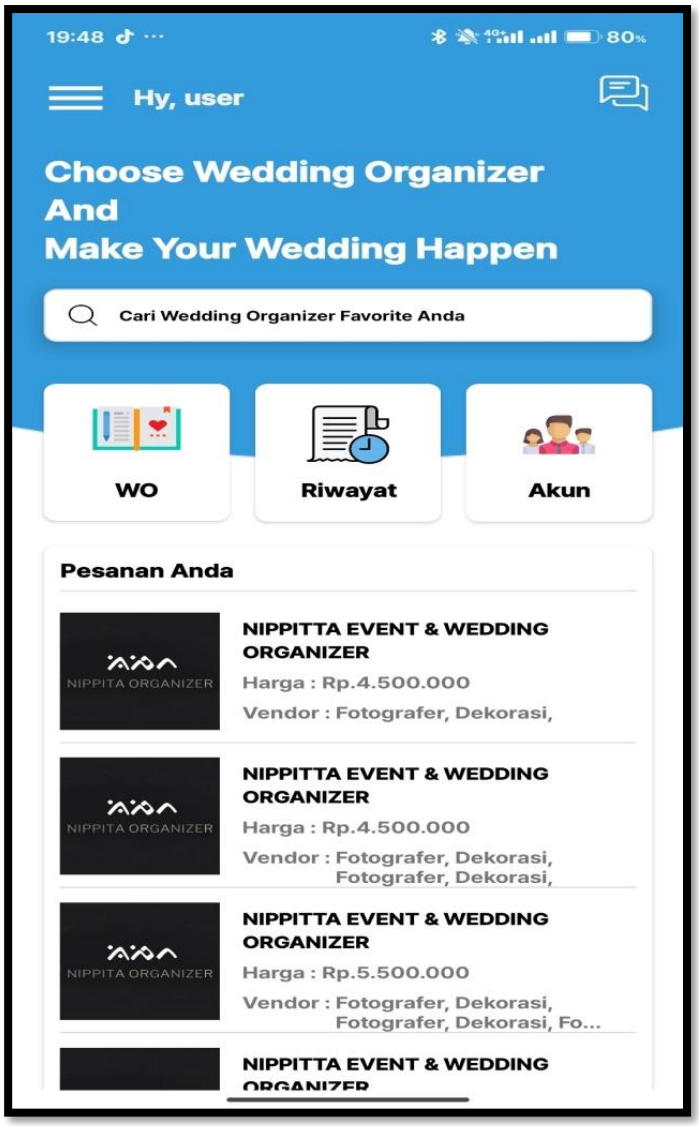


Gambar 4 23. Halaman Login

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.login
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import
com.abcd.vian_marketplaceweddingorganizer.data.model.UsersModel
importcom.abcd.vian_marketplaceweddingorganizer.databinding.Act
ivityLoginBindingimportcom.abcd.vian_marketplaceweddingorganize
r.ui.activity.admin.AdminMainActivityimportcom.abcd.vian_market
placeweddingorganizer.ui.activity.register.RegisterActivitympor
tcom.abcd.vian_marketplaceweddingorganizer.ui.activity.user.mai
n.
```

2) Halaman Dashboard

Berikut adalah tampilan dashboard pada Aplikasi E-Marketplace Wedding Organizer. Halaman dashboard dalam aplikasi ini berfungsi sebagai pusat informasi utama dan titik navigasi bagi pengguna.



Gambar 4 24. Halaman Dasboard

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import androidx.navigation.compose.*
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            DashboardApp()
        }
    }
}
@Composable
fun DashboardApp() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination =
"dashboard") {
        composable("dashboard") {
DashboardScreen(navController) }
        composable("profile") { ProfileScreen(navController) }
    }
}
@Composable
fun DashboardScreen(navController: NavController) {
    Scaffold(
        topBar = {
            TopAppBar(title = { Text("Dashboard") },
backgroundColor = Color.Blue, contentColor = Color.White)
        }
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp),
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Welcome to Dashboard", fontSize =
24.sp, fontWeight = FontWeight.Bold)
            Spacer(modifier = Modifier.height(20.dp))
            Button(onClick = {
```

3) Halaman Wedding Organizer

Berikut adalah tampilan wedding organizer pada Aplikasi E-Marketplace Wedding Organizer. Halaman wedding organizer dalam aplikasi ini berfungsi untuk menampilkan daftar wedding organizer, memberikan akses ke fitur-fitur tertentu, serta menyediakan informasi detail terkait setiap wedding organizer.



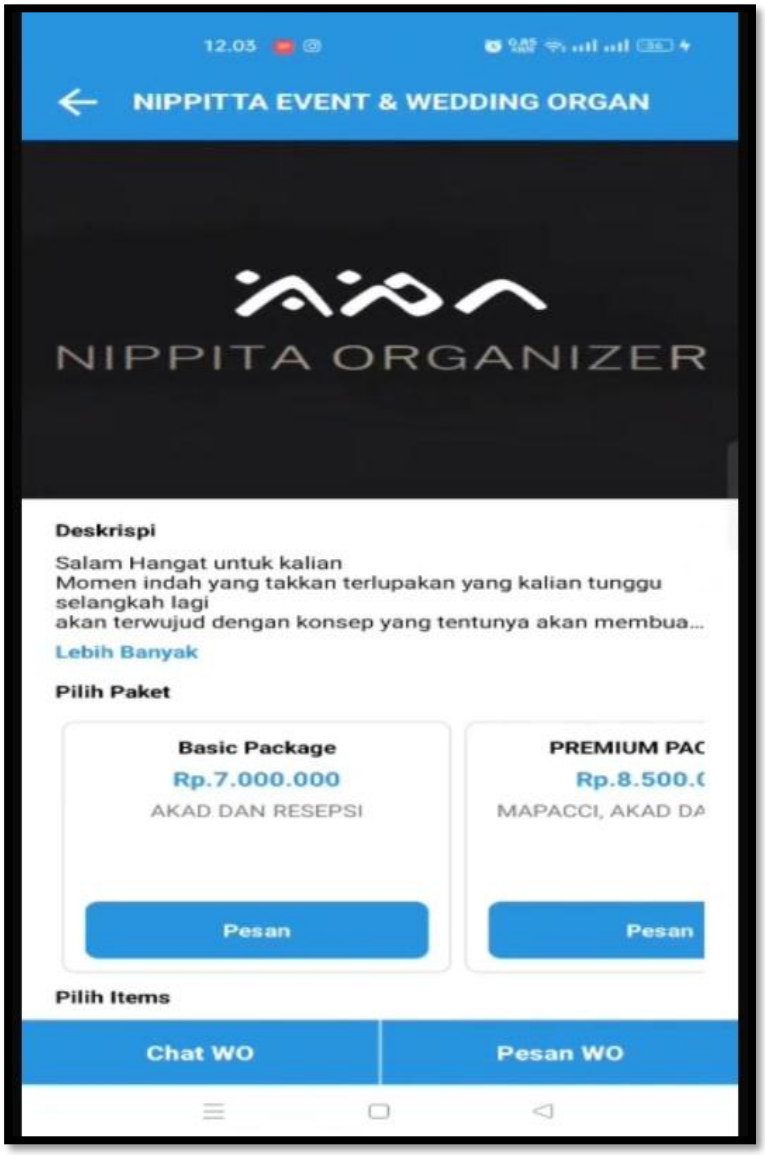
Gambar 4 25. Halaman Wedding Organizer

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            WeddingOrganizerApp()
        }
    }
}
data class WeddingPackage(val name: String, val price: String,
val imageRes: Int)
@Composable
fun WeddingOrganizerApp() {
    val packages = listOf(
        WeddingPackage("Paket   Silver",   "Rp   10.000.000",
android.R.drawable.ic_menu_gallery),
        WeddingPackage("Paket    Gold",    "Rp    20.000.000",
android.R.drawable.ic_menu_gallery),
        WeddingPackage("Paket   Platinum",   "Rp   30.000.000",
android.R.drawable.ic_menu_gallery)
    )

    Scaffold(
        topBar = {
            TopAppBar(title = { Text("Wedding Organizer") })
        }
    ) { padding ->
        LazyColumn(modifier = Modifier.padding(padding)) {
            items(packages) { weddingPackage ->
                PackageItem(weddingPackage)
            }
        }
    }
}
```

4) Halaman Pilih Paket

Berikut adalah tampilan pilih paket pada Aplikasi E-Marketplace Wedding Organizer. Halaman pilih paket dalam aplikasi ini berfungsi untuk menampilkan daftar pilih paket.



Gambar 4 26. Halaman Pilih Paket

```
package.com.abcd.vian_marketplaceweddingorganizer.ui.activity.u
ser.wedding_organizer
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
importcom.abcd.vian_marketplaceweddingorganizer.adapter.

com.abcd.vian_marketplaceweddingorganizer.utils.KontrolNavigati
onDrawer
import
com.abcd.vian_marketplaceweddingorganizer.utils.LoadingAlertDia
log
import
com.abcd.vian_marketplaceweddingorganizer.utils.OnClickItem
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import com.bumptech.glide.Glide
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject
@AndroidEntryPoint
class WeddingOrganizerActivity : AppCompatActivity() {
    private           lateinit           var           binding:
ActivityWeddingOrganizerBinding
    private      lateinit      var      kontrolNavigationDrawer:
KontrolNavigationDrawer
    private      lateinit      var      sharedPreferencesLogin:
SharedPreferencesLogin
    private   val   viewModel:   WeddingOrganizerViewModel   by
viewModels()
    private lateinit var plafonAdapter: WeddingOrganizerAdapter
    @Inject
    lateinit var loading: LoadingAlertDialog
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding                                                =
ActivityWeddingOrganizerBinding.inflate(layoutInflater)
        setContentView(binding.root)
        setNavitagionDrawer()
        setSharedPreferencesLogin()
        setButton()
        fetchWeddingOrganizer()
        getWeddingOrganizer()
    }
    private fun setButton() {
        binding.apply {
        }
    }
```

5) Halaman Pesan Sekarang

Berikut adalah tampilan pesan sekarang pada Aplikasi E-Marketplace Wedding Organizer. Halaman pesan sekarang dalam aplikasi ini berfungsi untuk menampilkan alamat pengirim, Metode pembayaran, Tanggal acara, Kode unit pembayaran serta total pembayaran.



Gambar 4 27. Halaman Pesan Sekarang

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.paym
ent
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Toast
import androidx.activity.result.ActivityResultLauncher
import
androidx.activity.result.contract.ActivityResultContracts
import androidx.activity.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
import
com.abcd.vian_marketplaceweddingorganizer.adapter.ListPaymentVe
ndorAdapter
import
com.abcd.vian_marketplaceweddingorganizer.data.model.AlamatMode
l
import
com.abcd.vian_marketplaceweddingorganizer.data.model.ResponseMo
del
import
com.abcd.vian_marketplaceweddingorganizer.data.model.VendorMode
l
import
com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityP
aymentBinding
import
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.addr
ess.ChooseAddressActivity
import
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.main
.MainActivity
import com.abcd.vian_marketplaceweddingorganizer.utils.Constant
import com.abcd.vian_marketplaceweddingorganizer.utils.KataAcak
import
com.abcd.vian_marketplaceweddingorganizer.utils.KonversiRupiah
import
com.abcd.vian_marketplaceweddingorganizer.utils.LoadingAlertDia
log
import
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.TanggalDanWaktu
import
```

6) Halaman Detail Pesanan

Berikut adalah tampilan detail pesanan pada Aplikasi E-Marketplace Wedding Organizer. Halaman detail pesanan dalam aplikasi ini berfungsi untuk menampilkan detail pesanan serta dapat mengupload bukti pembayaran.



Gambar 4 28. Halaman Detail Pesanan

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.paym
ent
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Toast
import androidx.activity.result.ActivityResultLauncher
import
androidx.activity.result.contract.ActivityResultContracts
import androidx.activity.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
com.abcd.vian_marketplaceweddingorganizer.adapter.ListPaymentVe
ndorAdapt
import
com.abcd.vian_marketplaceweddingorganizer.data.model.AlamatMode
l
importcom.abcd.vian_marketplaceweddingorganizer.data.model.Reke
ningModel
com.abcd.vian_marketplaceweddingorganizer.data.model.ResponseMo
del
import
com.abcd.vian_marketplaceweddingorganizer.data.model.VendorMode
l
com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityP
aymentBind
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.addr
ess.ChooseAd
com.abcd.vian_marketplaceweddingorganizer.ui.activity.user.main
.MainActivit
import com.abcd.vian_marketplaceweddingorganizer.utils.Constant
import com.abcd.vian_marketplaceweddingorganizer.utils.KataAcak
import
com.abcd.vian_marketplaceweddingorganizer.utils.KonversiRupiah
import
com.abcd.vian_marketplaceweddingorganizer.utils.LoadingAlertDia
log
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.TanggalDanWaktu
import
com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import com.midtrans.sdk.uikit.api.model.CustomerDetails
import com.midtrans.sdk.uikit.api.model.ItemDetails
import com.midtrans.sdk.uikit.api.model.SnapTransactionDetail
import com.midtrans.sdk.uikit.external.UiKitApi
import com.midtrans.sdk.uikit.internal.util.UiKitConstants
import dagger.hilt.android.AndroidEntryPoint
import java.util.UUID
```

7) Halaman Riwayat Pesanan

Berikut adalah tampilan pesanan pada Aplikasi E-Marketplace Wedding Organizer. Halaman riwayat pesanan dalam aplikasi ini berfungsi untuk menampilkan daftar riwayat pesanan.

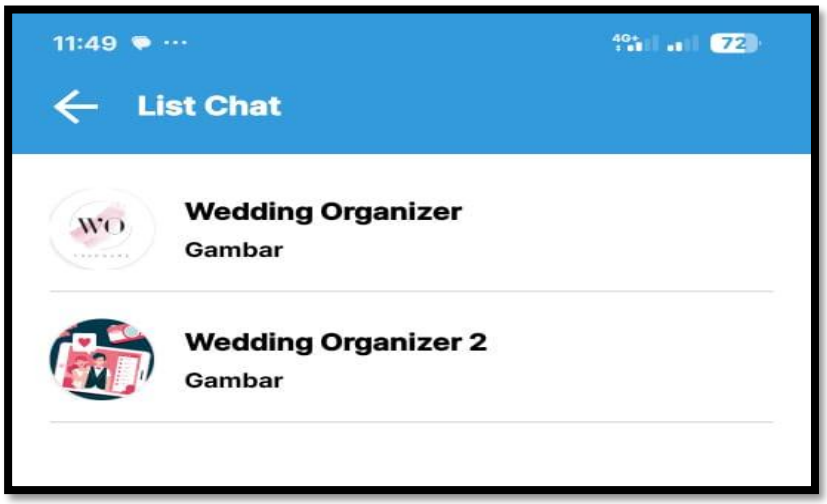


Gambar 4 29. Halaman Riwayat Pesanan

```
package.com.abcd.vian_marketplaceweddingorganizer.ui.activity.w
o.riwayat_pesanan.list
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
import
com.abcd.vian_marketplaceweddingorganizer.adapter.wo.WeddingOrg
anizerPesananListAdapterimportcom.abcd.vian_marketplaceweddingo
rganizer.data.model.RiwayatPesananListModelimportcom.abcd.vian_
marketplaceweddingorganizer.databinding.ActivityWeddingOrganize
rRiwayatPesananBinding
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.pesana
n.detail.WeddingOrganizerPesananDetailActivity
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.pesana
n.list.WeddingOrganizerPesananViewModel
com.abcd.vian_marketplaceweddingorganizer.utils.KontrolNavigati
onDrawer
import
com.abcd.vian_marketplaceweddingorganizer.utils.OnClickItem
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
@AndroidEntryPoint
class          WeddingOrganizerRiwayatPesananActivity          :
AppCompatActivity() {
ActivityWeddingOrganizerRiwayatPesananBinding
    private      lateinit      var      kontrolNavigationDrawer:
KontrolNavigationDrawer
    private      lateinit      var      sharedPreferencesLogin:
SharedPreferencesLogin
    private                    val                    viewModel:
WeddingOrganizerRiwayatPesananViewModel by viewModels()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

8) Halaman Chat

Berikut adalah tampilan chat pada Aplikasi E-Marketplace Wedding Organizer. Halaman chat dalam aplikasi ini berfungsi untuk menampilkan halaman chat.



Gambar 4 30. Halaman Chat

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.chat.chat
import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.util.Log
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.LinearLayoutManager
import
com.abcd.vian_marketplaceweddingorganizer.adapter.ChatWeddingOr
ganizerAdapter
    private fun postMessage(
        idPengirim: Int,
        idPenerima: Int,
        message: String,
        pengirim: String,
    ){
        viewModel.postMessage(message, idPengirim, idPenerima,
pengirim)
    }
    private fun getResponseMessage(){

viewModel.getMessage().observe(this@ChatWeddingOrganizerActivit
y){result->
            when(result){
                is UIState.Loading->{}
                is UIState.Success->
setSuccessPostMessage(result.data)
                is UIState.Failure->
setFailurePostMessage(result.message)
            }
        }
    }
```

9) Halaman Akun

Berikut adalah tampilan akun pada Aplikasi E-Marketplace Wedding Organizer. Halaman akun dalam aplikasi ini berfungsi untuk menampilkan data akun.



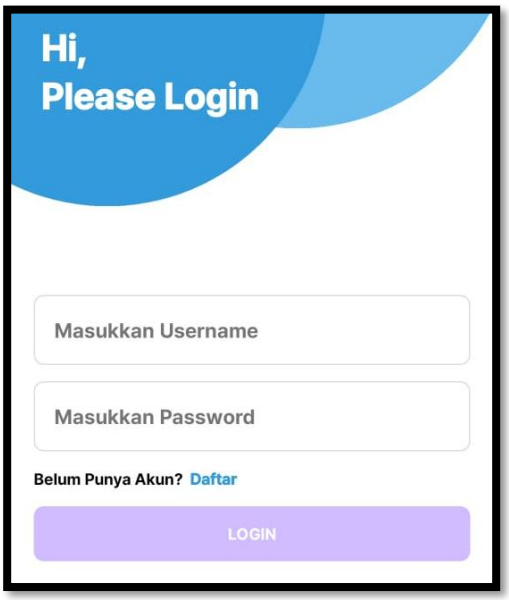
Gambar 4 31. Halaman Akun

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.akun
import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
```

c. Halaman Pihak Wedding Organizer

1) Halaman Login

Berikut adalah tampilan login pada Aplikasi E-Marketplace Wedding Organizer. Halaman menu login merupakan tampilan yang ada pada sistem, yang digunakan untuk masuk ke halaman selanjutnya dengan memasukkan *username* dan *password*.

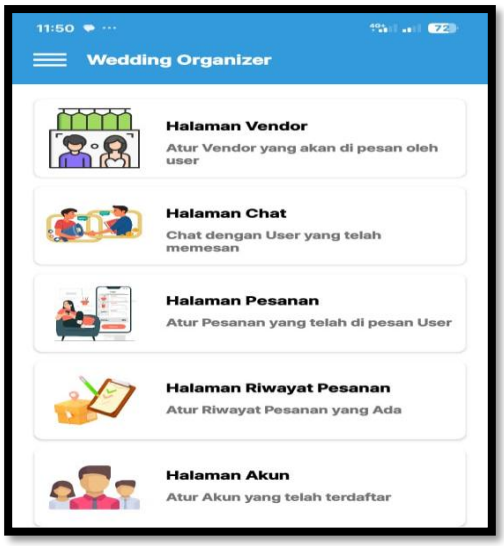


Gambar 4 32. Halaman Login

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.login
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import
com.abcd.vian_marketplaceweddingorganizer.data.model.UsersModel
import
com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityL
oginBinding
```

2) Halaman Dashboard

Berikut adalah tampilan Dashboard pada aplikasi E-Marketplace Wedding Organizer. Halaman dashboard dalam aplikasi ini berfungsi sebagai pusat informasi utama dan titik navigasi bagi pengguna.



Gambar 4 31. Halaman Dashboard

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent
```

3) Halaman Vendor

Berikut adalah tampilan Vendor pada Aplikasi E-Marketplace Wedding Organizer. Halaman vendor dalam aplikasi ini berfungsi untuk menampilkan daftar vendor, memberikan akses ke fitur-fitur tertentu, serta menyediakan informasi detail terkait setiap vendor.

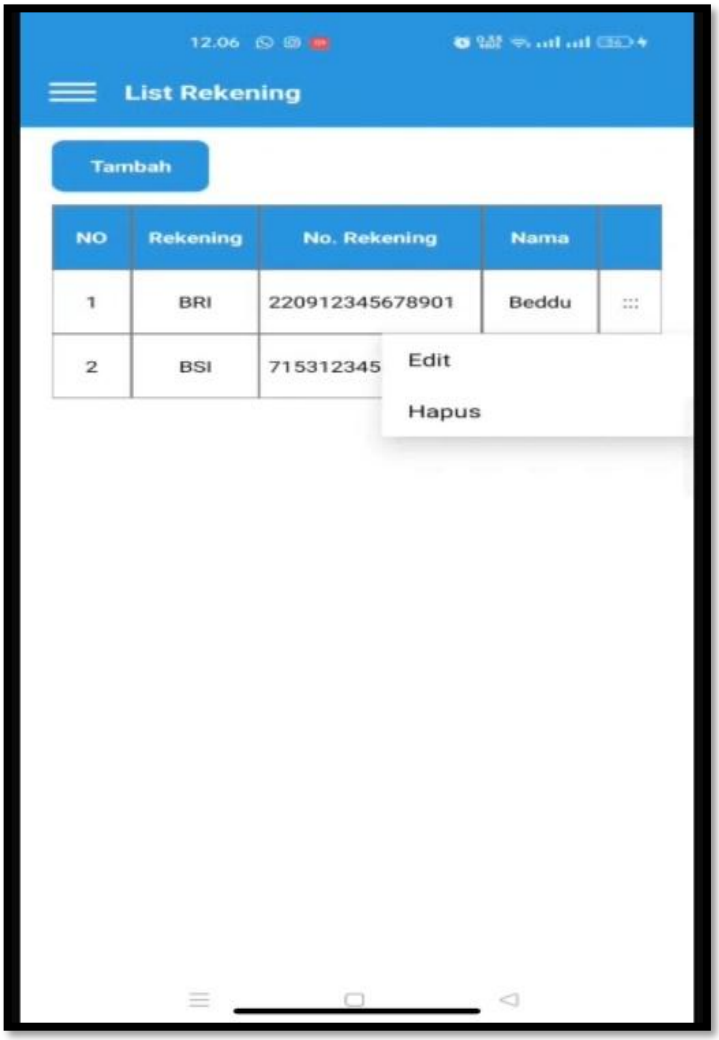


Gambar 4 32. Halaman Vendor

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.vendor
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
import
com.abcd.vian_marketplaceweddingorganizer.adapter.wo.WeddingOrg
anizerVendorAdapter
import
com.abcd.vian_marketplaceweddingorganizer.data.model.ResponseMo
del
import
com.abcd.vian_marketplaceweddingorganizer.data.model.VendorMode
l
import
com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityW
eddingOrganizerVendorBinding
import
com.abcd.vian_marketplaceweddingorganizer.databinding.AlertDial
ogKeteranganBinding
import
com.abcd.vian_marketplaceweddingorganizer.databinding.AlertDial
ogKonfirmasiBinding
import
com.abcd.vian_marketplaceweddingorganizer.databinding.AlertDial
ogWoVendorBinding
import
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.main.W
eddingOrganizerMainActivity
import
com.abcd.vian_marketplaceweddingorganizer.utils.KontrolNavigati
onDrawer
import
com.abcd.vian_marketplaceweddingorganizer.utils.LoadingAlertDia
log
import
com.abcd.vian_marketplaceweddingorganizer.utils.OnClickItem
import
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import java.util.ArrayList
import javax.inject.Inject
```

4) Halaman List Rekening

Berikut adalah tampilan list rekening pada Aplikasi E-Marketplace Wedding Organizer. Halaman list rekening dalam aplikasi ini berfungsi untuk menampilkan list rekening.

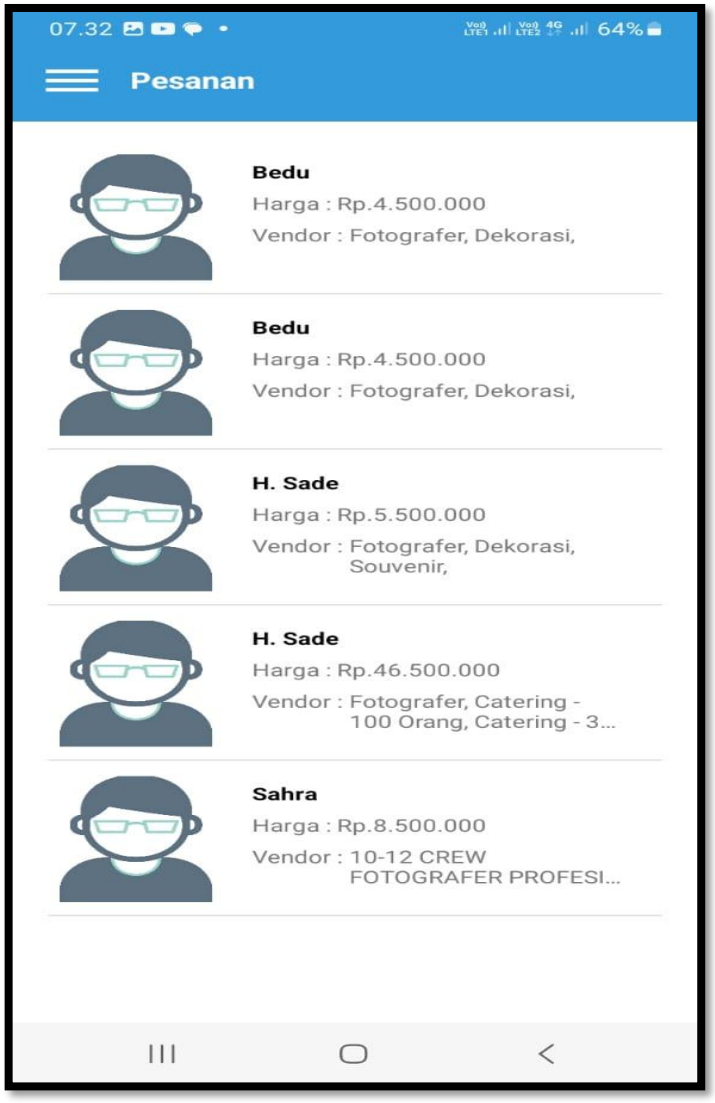


Gambar 4 35. Halaman List Rekening

```
package
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.reken
in
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
@AndroidEntryPoint
class WeddingOrganizerRekeningActivity : AppCompatActivity() {
    private           lateinit           var           binding:
ActivityWeddingOrganizerRekeningBinding
    private     lateinit     var     kontrolNavigationDrawer:
KontrolNavigationDrawer
    private     lateinit     var     sharedPreferencesLogin:
SharedPreferencesLogin
    private val viewModel: WeddingOrganizerRekeningViewModel by
viewModels()
    @Inject
    lateinit var loading: LoadingAlertDialog
    private           lateinit           var           adapter:
WeddingOrganizerRekeningAdapter
    private var idWo = 0
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState) @AndroidEntryPoint
class WeddingOrganizerPesananActivity : AppCompatActivity() {
    private           lateinit           var           binding:
ActivityWeddingOrganizerPesananBinding
    private     lateinit     var     kontrolNavigationDrawer:
KontrolNavigationDrawer
    private     lateinit     var     sharedPreferencesLogin:
SharedPreferencesLogin
    private val viewModel: WeddingOrganizerPesananViewModel by
viewModels()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding                                                =
ActivityWeddingOrganizerPesananBinding.inflate(layoutInflater)
        setContentView(binding.root)
        setStartShimmer()
        setKontrolNavigationDrawer()
```

5) Halaman Pesanan

Berikut adalah tampilan pesanan pada Aplikasi E-Marketplace Wedding Organizer. Halaman pesanan dalam aplikasi ini berfungsi untuk menampilkan pesanan, memberikan akses ke fitur-fitur tertentu, serta menyediakan informasi tentang pesanan.



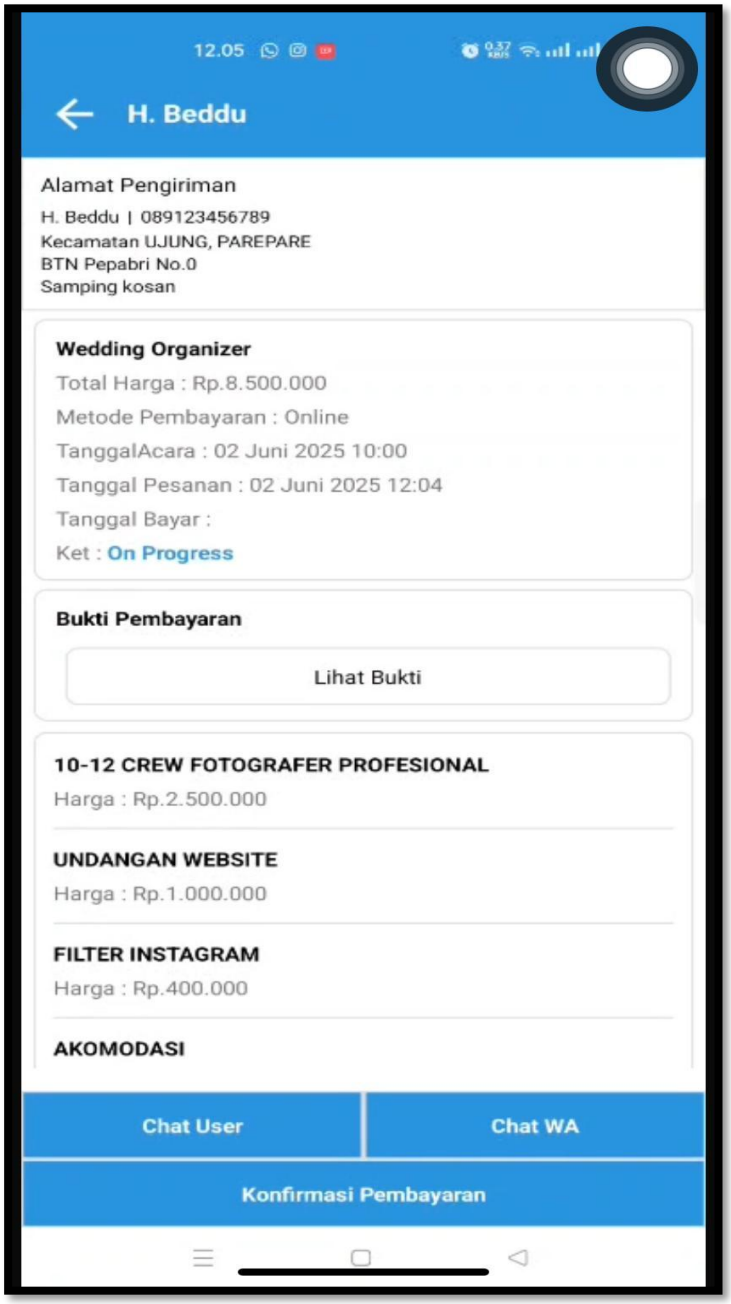
Gambar 4 36. Halaman Pesanan

```
package.com.abcd.vian_marketplaceweddingorganizer.ui.activitywo
.pesanan.list
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
import
com.abcd.vian_marketplaceweddingorganizer.adapter.wo.WeddingOrg
anizerPesananListAdapter
import
com.abcd.vian_marketplaceweddingorganizer.data.model.RiwayatPes
ananListModel
import
com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityW
eddingOrganizerPesananBinding
import
com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.pesana
n.detail.WeddingOrganizerPesananDetailActivity
import
com.abcd.vian_marketplaceweddingorganizer.utils.KontrolNavigati
onDrawer
import
com.abcd.vian_marketplaceweddingorganizer.utils.OnClickItem
import
com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferenc
esLogin
import
com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import                    dagger.hilt.android.AndroidEntryPoint
setSharedPreferencesLogin()
        fetchPesanan()
        getPesanan()
    }
    private fun setKontrolNavigationDrawer() {
        binding.apply {
            drawerView.tvTitle.text = "Pesanan"
            kontrolNavigationDrawer                              =
KontrolNavigationDrawer(this@WeddingOrganizerPesananActivity)
            kontrolNavigationDrawer.cekSebagai(navView)

kontrolNavigationDrawer.onClickItemNavigationDrawer(navView,
drawerLayoutMain,                        drawerView.ivNavDrawer,
this@WeddingOrganizerPesananActivity)
        }
    }
    private fun setSharedPreferencesLogin() {
        sharedPreferencesLogin                                   =
SharedPreferencesLogin(this@WeddingOrganizerPesananActivity)
```

6) Halaman Konfirmasi Pembayaran

Berikut adalah tampilan konfirmasi pembayaran pada Aplikasi E-Marketplace Wedding Organizer. Halaman konfirmasi pembayaran dalam aplikasi ini berfungsi untuk menampilkan konfirmasi pembayaran.



Gambar 4 33. Halaman Konfirmasi Pembayaran

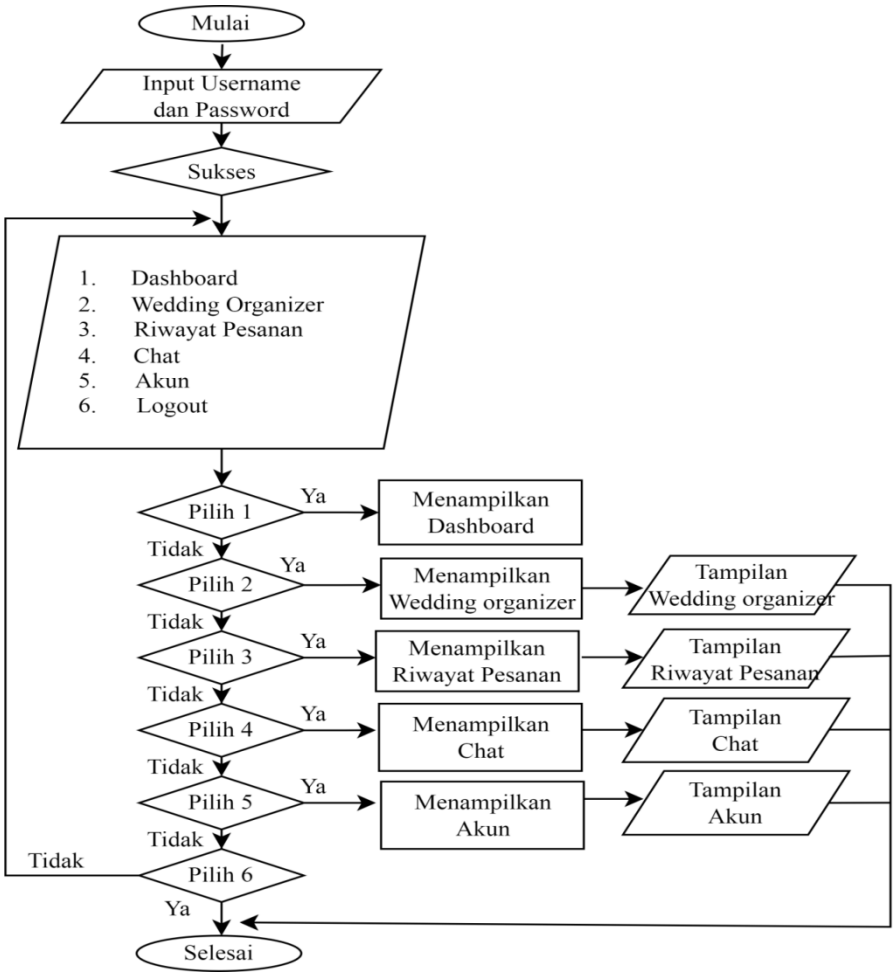
```
package.com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.riwayat_pesanan.list
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.vian_marketplaceweddingorganizer.R
import com.abcd.vian_marketplaceweddingorganizer.adapter.wo.WeddingOrganizerPesananListAdapter
import com.abcd.vian_marketplaceweddingorganizer.data.model.RiwayatPesananListModel
import com.abcd.vian_marketplaceweddingorganizer.databinding.ActivityWeddingOrganizerRiwayatPesananBinding
import com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.pesanan.detail.WeddingOrganizerPesananDetailActivity
import com.abcd.vian_marketplaceweddingorganizer.ui.activity.wo.pesanan.list.WeddingOrganizerPesananViewModel
import com.abcd.vian_marketplaceweddingorganizer.utils.KontrolNavigationDrawer
import com.abcd.vian_marketplaceweddingorganizer.utils.OnClickItem
import com.abcd.vian_marketplaceweddingorganizer.utils.SharedPreferencesLogin
import com.abcd.vian_marketplaceweddingorganizer.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
@AndroidEntryPoint
class WeddingOrganizerRiwayatPesananActivity : AppCompatActivity() {
    private lateinit var binding: ActivityWeddingOrganizerRiwayatPesananBinding
    private lateinit var kontrolNavigationDrawer: KontrolNavigationDrawer
    private lateinit var sharedPreferencesLogin: SharedPreferencesLogin
    private val viewModel: WeddingOrganizerRiwayatPesananViewModel by viewModels()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityWeddingOrganizerRiwayatPesananBinding.inflate(layoutInflater)
        setContentView(binding.root)
        setStartShimmer()
        setKontrolNavigationDrawer()
        setSharedPreferencesLogin()
```

6. Pengujian Sistem

a. *WhiteBox*

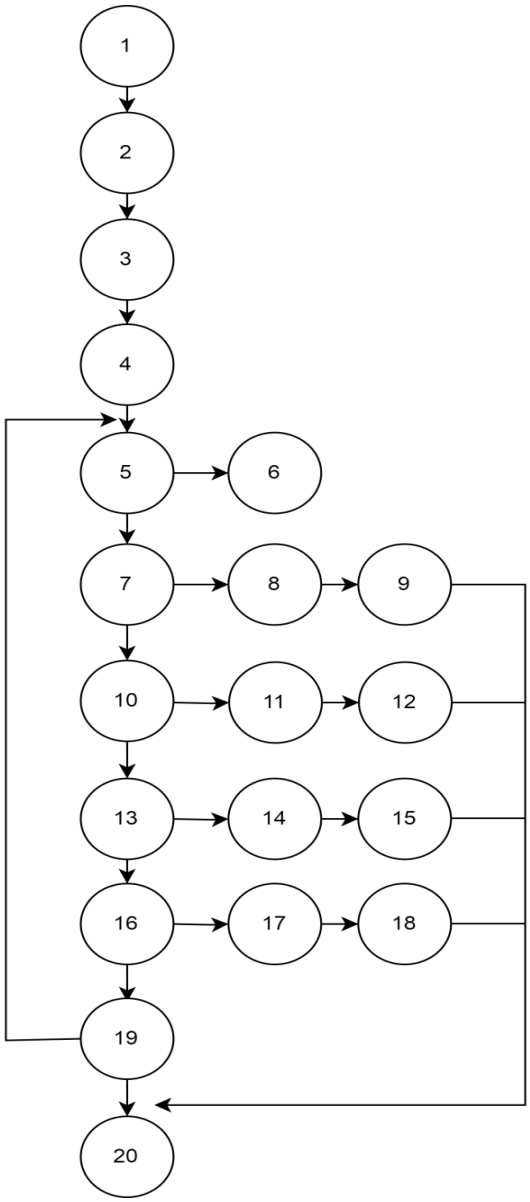
Pengujian aplikasi dilakukan dengan cara pengujian *WhiteBox*. *WhiteBox* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi, yang bertentangan dengan fungsinya. Dalam pengujian kotak putih, perspektif internal sistem digunakan untuk merancang kasus uji.

1) Flowchart dan *Flowgraph* Aktivitas *Admin* dan *User*



Gambar 4 34. *Flowchart* Aktivitas *Admin* dan *User*

Dari Flowchart yang digunakan untuk pengujian perangkat lunak, maka ditentukan Flowgraph sebagai berikut:



Gambar 4. 39. *Flowgraph* Aktivitas Admin dan User

Dari *Flowgraph* aktivitas *admin* dan *user* diatas dapat dilakukan proses perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

Diketahui :

N = 16 E = 20 R = 2

Penyelasaian :

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

= (20 – 16) + 2 = 6

Independent Path :

Path 1 = 1 – 2 – 3 – 4 – 5 –6

Path 2 = 1 – 2 – 3 – 5 – 7 – 8 –9 – 20

Path 3 = 1 – 2 – 3 – 5 – 7 – 10 –11 –12– 20

Path 4 = 1 – 2 – 3 – 5 – 7 – 10 –13 –14 – 15 –20

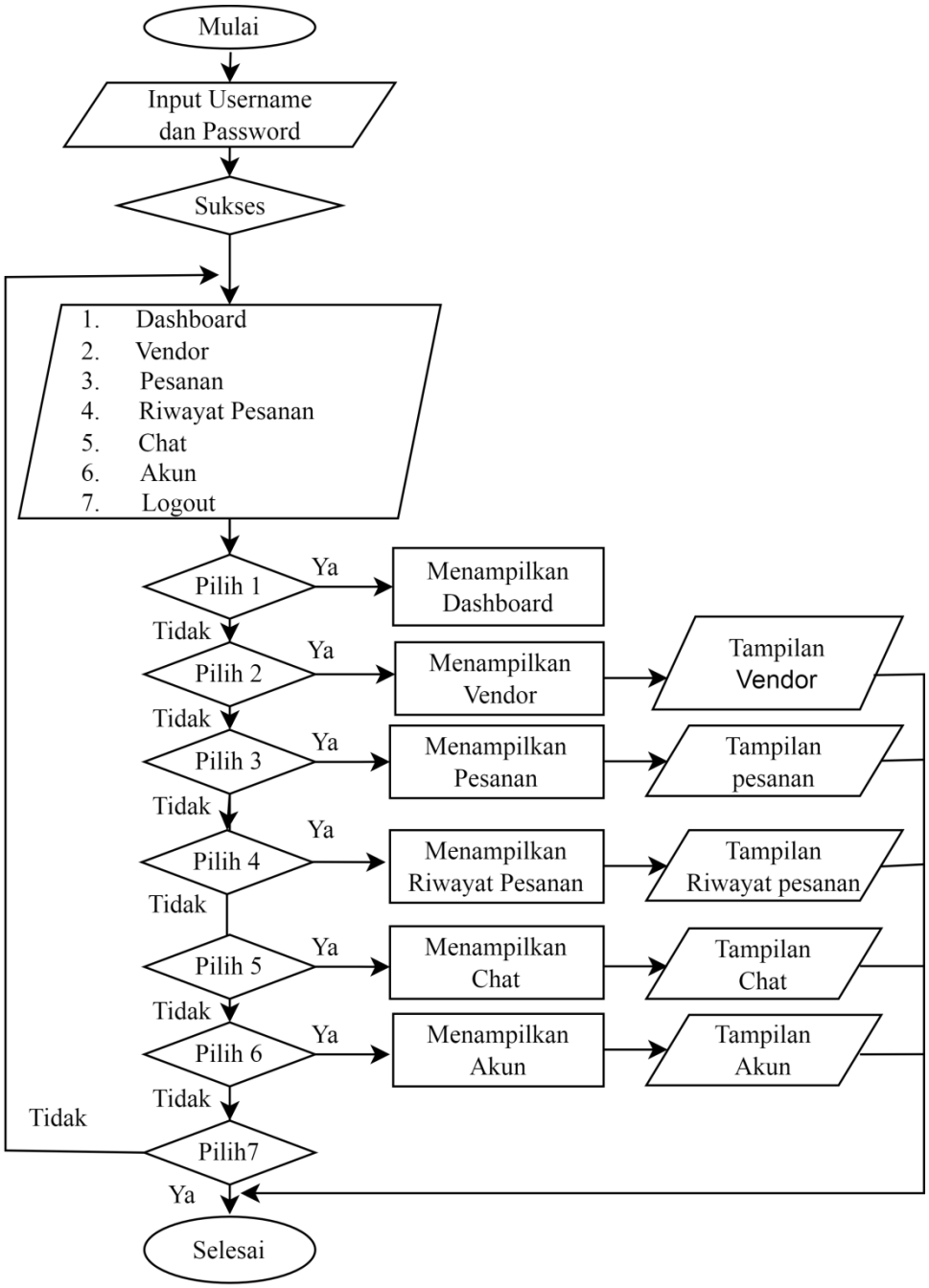
Path 5 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 17– 18 –20

Path 6 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 20

Tabel 4 18. *White Box* Perhitungan *menu Admin* dan *User*

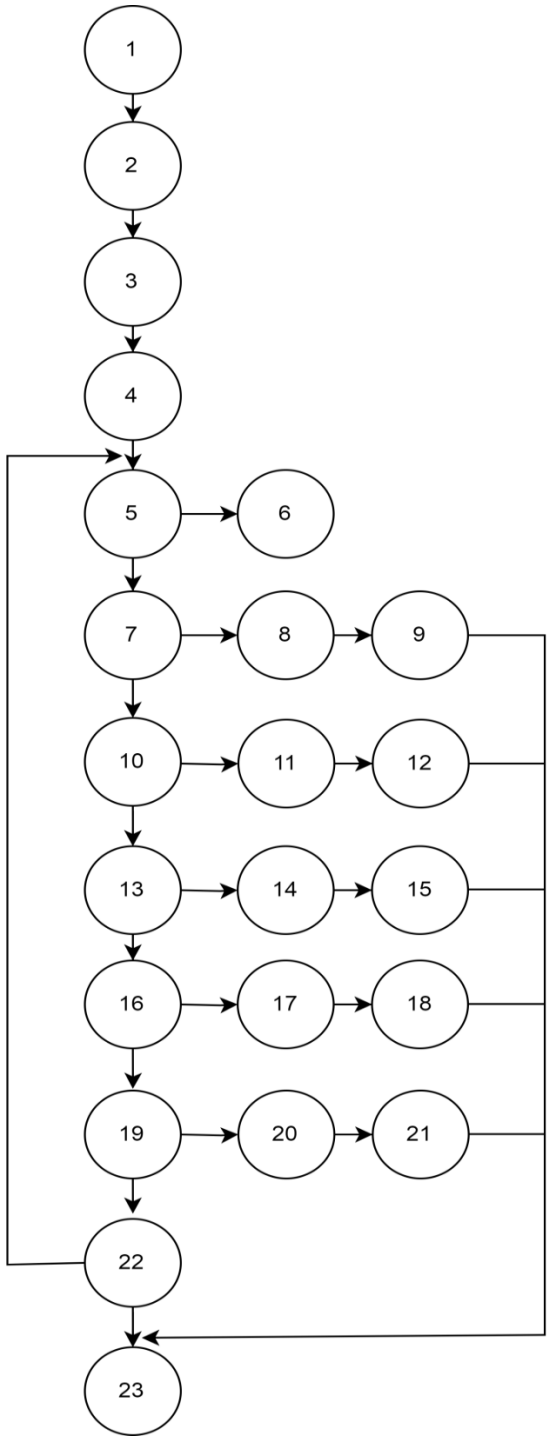
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	E-1
1		1																			1-1=0
2			1																		1-1=0
3				1																	1-1=0
4					1																1-1=0
5						1	1														2-1=1
6																					0
7								1		1											2-1=1
8									1												1-1=0
9																					1-1=0
10											1		1								2-1=1
11												1									1-1=0
12																					1-1=0
13														1		1					2-1=1
14															1						1-1=0
15																					1-1=0
16																	1		1		2-1=1
17																		1			1-1=0
18																					1-1=0
19																				1	2-1=1
20																					0
SUM(E+1)																					5+1=6

2) *Flowchart* dan *Flowgraph* Aktivitas Pihak WO



Gambar 4. 40. *Flowchart* Aktivitas Pihak Wo

Dari *Flowchart* yang digunakan *untuk* pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut:



Gambar 4. 41. *Flowgraph* Aktivitas Pihak Wo

Dari *Flowgraph* aktivitas *user* diatas dapat dilakukan proses perhitungan sebagai berikut:

b) Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

Diketahui :

N = 18 E = 23 R = 2

Penyelasaian :

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

= (23 – 18) + 2

= 7

Independent Path :

Path 1 = 1 – 2 – 3 – 4 – 5 –6

Path 2 = 1 – 2 – 3 – 5 – 7 – 8 –9 – 23

Path 3 = 1 – 2 – 3 – 5 – 7 – 10 –11 –12– 23

Path 4 = 1 – 2 – 3 – 5 – 7 – 10 –13 –14 – 15 –23

Path 5 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 17– 18 –23

Path 6 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 20 –21–23

Path 7 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 22 –23

Tabel 4 5. *White Box* Perhitungan *menu* Pihak Wo

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	E-1
1		1																						1-1=0
2			1																					1-1=0
3				1																				1-1=0
4					1																			1-1=0
5						1	1																	2-1=1
6																								0
7								1		1														2-1=1
8									1															1-1=0
9																							1	1-1=0
10											1		1											2-1=1
11												1												1-1=0
12																							1	1-1=0
13														1		1								2-1=1
14															1									1-1=0

Lanjutan Tabel

15																					1	1-1=0
16																1		1				2-1=1
17																	1					1-1=0
18																					1	1-1=0
19																			1		1	2-1=1
20																				1		1-1=0
21																					1	1-1=0
22																					1	1-1=0
23																					0	0
SUM(E+1)																						6+1=7

b. *Black box*

Pengujian *BlackBox* digunakan untuk menguji sistem. *BlackBox* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi tanpa mempelajari struktur atau cara kerja internalnya. Pendekatan pengujian ini berlaku untuk hampir semua tingkat pengujian perangkat lunak, termasuk unit, integrasi, sistem, dan penerimaan.

1) Black Box Admin

Tabel 4 20. *BlackBox* Halaman Utama Admin

Test Faktor	Hasil	Kesimpulan
Jika memasukkan username dan password dengan benar.	✓	Informasi, tampil halaman utama *admin*.
Screen Shot		
		

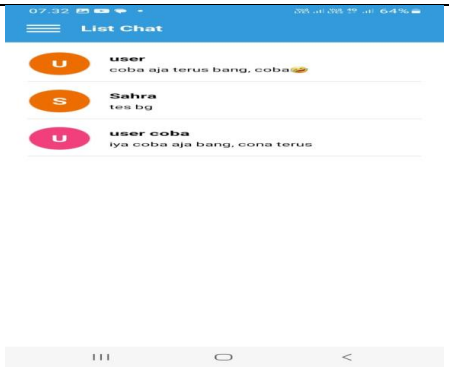
Tabel 4 21. *BlackBox Form* Wedding Organizer

Test Faktor	Hasil	Kesimpulan
Jika memilih menu wedding organizer pada halaman utama.	✓	Informasi, tampil form wedding organizer.
Screen Shot		
		

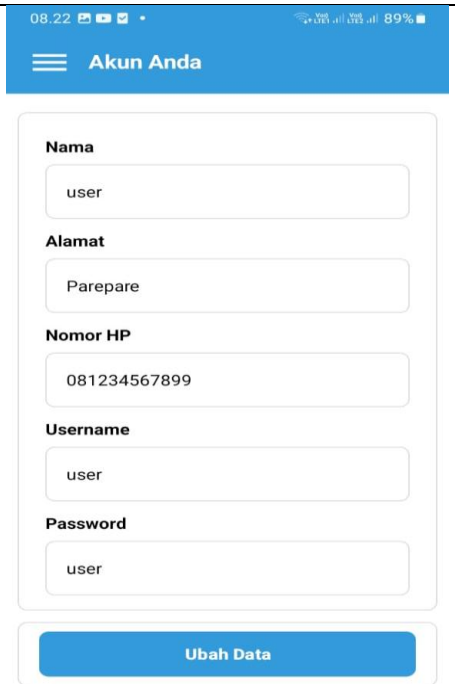
Tabel 4 22. *BlackBox Form* Riwayat Pesanan

Test Faktor	Hasil	Kesimpulan
Jika memilih menu riwayat pesanan pada halaman utama.	✓	Informasi, tampil form riwayat pesanan.
Screen Shot		
		

Tabel 4 23. *BlackBox Form* Chat

Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada halaman utama.	✓	Informasi, tampil form chat.
Screen Shot		
		

Tabel 4 6. *BlackBox Form* Akun

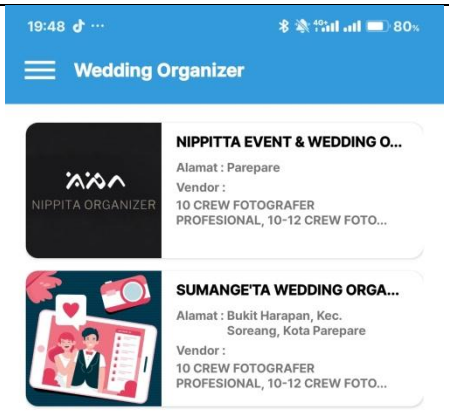
Test Faktor	Hasil	Kesimpulan
Jika memilih menu akun pada halaman utama.	✓	Informasi, tampil form akun
Screen Shot		
		

2) *BlackBox User*

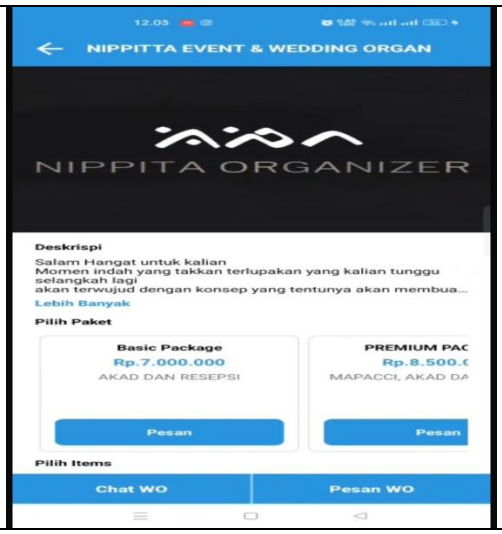
Tabel 4 7. *BlackBox* Halaman Utama

Test Faktor	Hasil	Kesimpulan
Jika memasukkan username dan password dengan benar.	✓	Informasi, tampil halaman utama *user*.
Screen Shot		
		

Tabel 4 8. *BlackBox Form* Wedding Organizer

Test Faktor	Hasil	Kesimpulan
Jika memilih menu wedding organizer pada halaman utama.	✓	Informasi, tampil form wedding organizer.
Screen Shot		
		

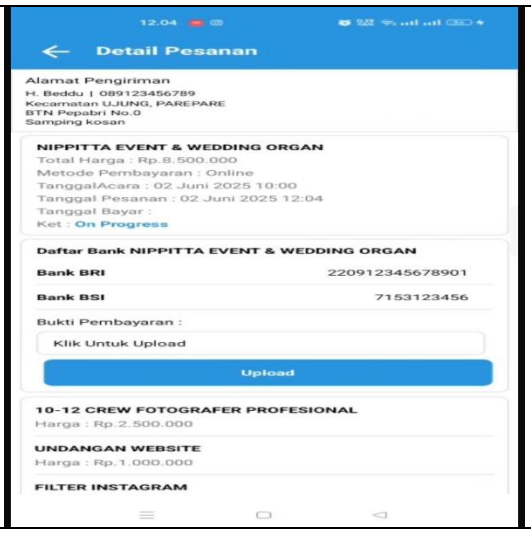
Tabel 4 9. *BlackBox Form* Pilih Paket

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Pilih Paket pada halaman utama.	✓	Informasi, tampil form Pilih Paket.
Screen Shot		
		

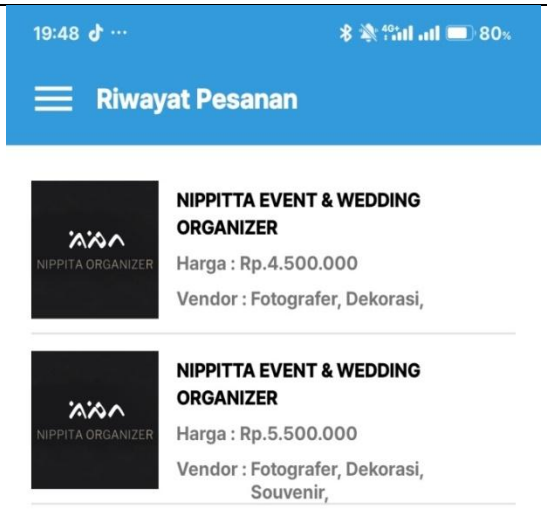
Tabel 4 10. *BlackBox Form* Pesan Sekarang

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Pesan Sekarang pada halaman utama.	✓	Informasi, tampil form Pesan Sekarang.
Screen Shot		
		

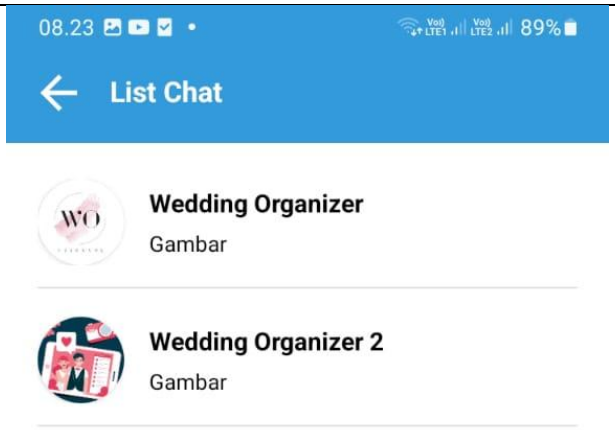
Tabel 4 11. *BlackBox Form* Detail Pesanan

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Detail Pesanan pada halaman utama.	✓	Informasi, tampil form Detail Pesanan.
Screen Shot		
		

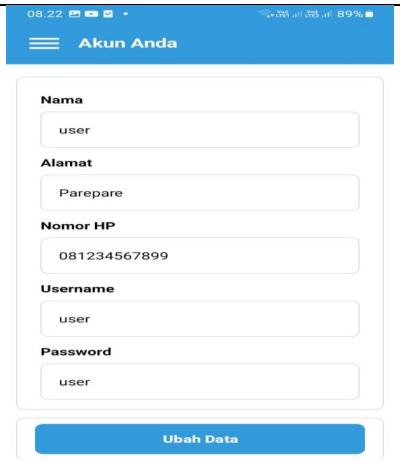
Tabel 4 30. *BlackBox Form* Riwayat Pesanan

Test Faktor	Hasil	Kesimpulan
Jika memilih menu riwayat pesanan pada halaman utama.	✓	Informasi, tampil form riwayat pesanan.
Screen Shot		
		

Tabel 4 31. *BlackBox Form* Chat

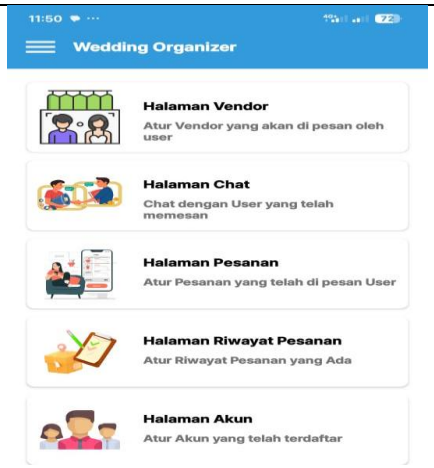
Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada halaman utama.	✓	Informasi, tampil form chat.
Screen Shot		
		

Tabel 4 32. *BlackBox Form* Akun

Test Faktor	Hasil	Kesimpulan
Jika memilih menu akun pada halaman utama.	✓	Informasi, tampil form akun
Screen Shot		
		

3) *BlackBox* Pihak Wo

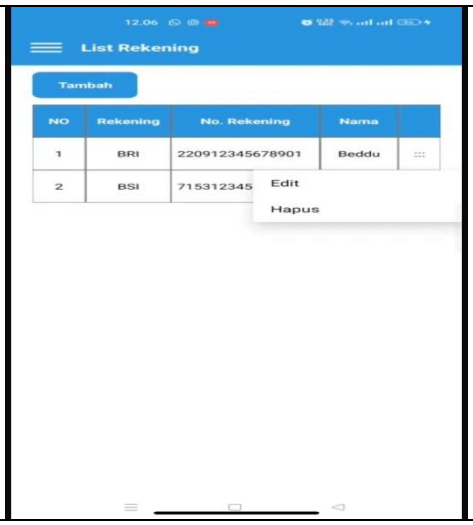
Tabel 4 33. *BlackBox* Halaman Utama

Test Faktor	Hasil	Kesimpulan
Jika memasukkan username dan password dengan benar.	✓	Informasi, tampil halaman utama pihak wo.
Screen Shot		
		

Tabel 4 12. *BlackBox Form* Vendor

Test Faktor	Hasil	Kesimpulan
Jika memilih menu vendor pada halaman utama.	✓	Informasi, tampil form vendor.
Screen Shot		
		

Tabel 4 13. *BlackBox Form* List Rekening

Test Faktor	Hasil	Kesimpulan
Jika memilih menu list rekening pada halaman utama.	✓	Informasi, tampil form list rekening.
Screen Shot		
		

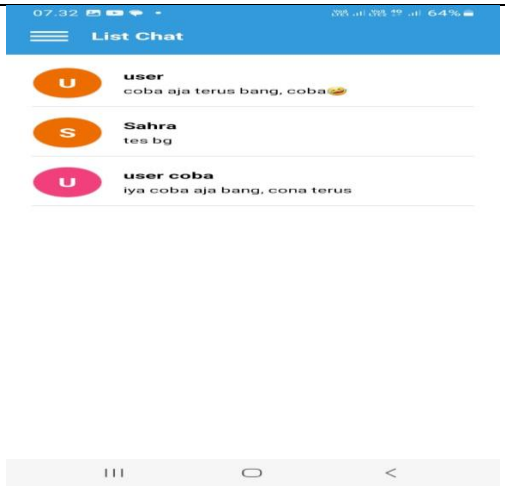
Tabel 4 14. *BlackBox Form* Pesanan

Test Faktor	Hasil	Kesimpulan
Jika memilih menu pesanan pada halaman utama.	✓	Informasi, tampil form pesanan.
Screen Shot		
		

Tabel 4 37. *BlackBox Form* Konfirmasi Pembayaran

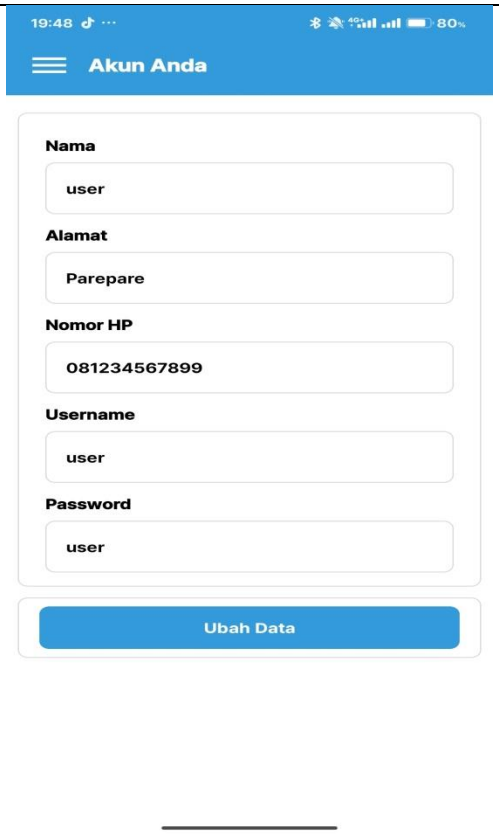
Test Faktor	Hasil	Kesimpulan
Jika memilih menu konfirmasi pembayaran pada halaman utama.	✓	Informasi, tampil form konfirmasi pembayaran.
Screen Shot		
		

Tabel 4 38. *BlackBox Form* Chat

Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada halaman utama.	✓	Informasi, tampil form chat.
Screen Shot		
		

Tabel 4 39. *BlackBox Form* Akun

Test Faktor	Hasil	Kesimpulan
Jika memilih menu akun pada halaman utama.	✓	Informasi, tampil form akun
Screen Shot		



d. *Maintenance*

Pada Tahap ini sistem yang telah dibangun sudah berhasil untuk digunakan dan dilakukan pemeliharaan sistem termasuk memperbaiki perubahan sistem kembali atau update sistem sesuai dengan kebutuhan perusahaan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian aplikasi e-marketplace wedding organizer berbasis android, dapat disimpulkan Aplikasi ini telah berhasil dibuat berbasis android dengan fitur utama pihak wedding organizer dapat mengelola vendor, pesanan, serta komunikasi dengan pengguna. Dengan adanya fitur vendor, wedding organizer dapat merekomendasikan berbagai vendor yang bekerja sama, sehingga memudahkan calon pengantin dalam memilih penyedia layanan yang sesuai. Wedding organizer dapat melihat daftar pesanan masuk melalui menu pesanan, serta dapat melakukan pemantauan terhadap pesanan yang telah selesai melalui fitur riwayat pesanan. Hal ini membantu wedding organizer dalam mengelola data pemesanan dengan lebih terstruktur dan efisien. Aplikasi ini dilengkapi dengan fitur chat, yang memungkinkan komunikasi langsung antara wedding organizer dan calon pelanggan. Dengan adanya fitur ini, calon pelanggan dapat bertanya dan berdiskusi mengenai layanan yang ditawarkan secara lebih cepat dan praktis. Masyarakat atau calon pengguna aplikasi dapat dengan mudah mencari wedding organizer melalui menu wedding organizer. Di dalam menu ini, pengguna dapat melihat informasi wedding organizer, termasuk daftar vendor yang direkomendasikan, sehingga membantu dalam proses perencanaan pernikahan yang lebih sistematis.

B. Saran

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Peningkatan Fitur Pencarian dan Filter: Diperlukan pengembangan fitur pencarian yang lebih spesifik dan filter yang lebih lengkap agar pengguna dapat dengan mudah menemukan layanan yang sesuai dengan kebutuhan dan anggaran mereka.
2. Integrasi Notifikasi: Penambahan fitur notifikasi real-time untuk update status pesanan, promosi, atau penawaran khusus akan meningkatkan interaksi dan keterlibatan pengguna.
3. Optimasi Performa Aplikasi: Optimasi kecepatan loading aplikasi dan pengurangan bug perlu dilakukan secara berkala agar pengalaman pengguna tetap optimal..
4. Fitur Chat dan Konsultasi Online: Menambahkan fitur chat langsung dengan vendor atau konsultasi online dapat meningkatkan kualitas layanan dan membantu calon pengantin dalam merencanakan pernikahan mereka dengan lebih baik.

DAFTAR PUSTAKA

Adi Nugroho.(2020), *Rekayasa Peragkat Lunak Menggunakan UML dan Java, Pustaka Belajar.* Yogyakarta.

Anggraeni, Yunaeti E., Irviani, R. (2017). Pengantar Sistem Informasi. Yogyakarta: CV. Andi Offset.

Awaluddin. (2022). *Simbol simbol pada Flowchart dan penjelasannya*. *Fakultas Teknik. Universitas Muhammdiyah Parepare.*

Dedi Riswandi. (2019), transaksi *on-line (e-commerce)* : peluang dan tantangan dalam perspektif ekonomi islam, Jurnal Econetica, Vol.1 No.1.

Endra, Aprilita. (2018). *Pengertian HTML (Hypertext Markup Language)* .

Gilang Ramadhan. (2023). *Contoh USE CASE Diagram Legkap Penjelasan, Dicoding Intren*. (Online), https://www.dicoding.com/blog/contoh-use-case-diagram.

Kolter. (2022), Pengertian jasa, IBI KKG, Institut Bisnis dan Informatika Kwik Kian Gie, Vol. 1, No. 2.

Midtrans. (2023). Payment Gateway Indonesia. Diakses dari https://midtrans.com

M. Shandy Putra. (2019), Aplikasi Penyedia Jasa Online Berbasis Mobile Android, SKRIPSI, Jurusan Ilmu Komputer.

Muh Jayadi. (2021), Aplikasi pemesanan menu pada warkop sharelock berbasis android. JURNAL SINTAKS LOGIKA Vol. 1 No. 3.

Rinaldi, Apsari Susanti, (2019), Perbandingan Analisa Pemilihan Vendor Trucking Menggunakan meetode AHP dan TOPSIS pada PT. YUSHAR PUTERA JAYA, TEKINFO Vol. 20, No. 2, Oktober 2019.

Rukin, (2019). Metodologi penelitian kualitatif. Takalar : Yayasan Ahmar Cendikia Indonesia.

Safaat, Gargenta, Listiaji, (2018), pemrograman aplikasi mobile berbasis android, Perpustakaan RI**,** Bandung.

Suryana, Nugroho. (2020). Pengenalan *MySQL, Xampp* Dan *Database*. *Journal of Educational Learning and Innovation*, Volume 2 Number, Halaman 42-72.