

BAB I

PENDAHULUAN

A. Latar Belakang

Kebutuhan manusia akan teknologi dan media komunikasi terpadu di era digital ini terus mengalami peningkatan. Diantaranya adalah kebutuhan akan industri kreatif. Industri ini tidak hanya menawarkan produk atau jasa entertainment saja, namun juga digunakan sebagai sarana visualisasi suatu objek untuk membangun persepsi masyarakat yang di harapkan terhadap objek tersebut. Salah satu dari sekian banyak bagian dari industri kreatif ini adalah industri fotografi.

Dalam hitungan tahun, perkembangan dunia fotografi sangat maju pesat. Kemajuan itu ditandai dengan semakin banyak bermunculannya kamera dengan kecanggihan yang menawarkan segala bentuk kemudahan. Era fotografi digital memudahkan fotografer untuk memahami dunia fotografi lebih luas lagi. Namun dengan segala kemudahan itu, untuk menghasilkan foto yang berkualitas, tidak semua orang bisa melakukannya. Anak kecil pun bisa memotret hanya dengan menekan tombol *shutter*. Untuk menghasilkan sebuah gambar yang bagus, perlu ketajaman insting seorang fotografer dalam memadukan *skill*, teknologi, dan juga nilai seni, agar foto yang dihasilkan bisa dikomunikasikan dengan orang yang melihat foto tersebut.

Perkembangan teknologi yang begitu pesat menjadikan kebutuhan akan jasa fotografi semakin meningkat. Kini hampir segala bentuk *advertising*,

penjualan, publikasi khusus, *event* seperti *wedding event*, dan banyak lainnya membutuhkan jasa fotografi dengan kualitas yang baik. Begitu strategisnya kehadiran seorang fotografer, maka terbukalah kesempatan bagi semua orang yang ingin menekuni dunia fotografi sebagai profesi. Saat ini banyak sekali *freelancer* yang mendalami dunia fotografi serta semakin banyak pula perusahaan atau komunitas yang menyediakan jasa fotografi *professional*.

UKM Fokus merupakan salah satu Organisasi Kampus Universitas Muhammadiyah Parepare fotografer komersial *professional* yang bekerja sama untuk membangun sebuah *brand image* dengan kualitas yang baik terhadap jasa fotografi yang mereka tawarkan. UKM Fokus memiliki spesialisasi di bidang fotografi *wedding*, *pre-wedding*, *potrait*, dan *catalogue*. Penyediaan jasa fotografi oleh UKM Fokus terdapat pada kota Ajatappareng. Sedangkan untuk diluar ajatappareng yaitu kota Mkassar. Organisasi ini telah memiliki ribuan *portofolio* yang dibuat di dalam maupun di luar dari wilayah penyediaan jasa mereka tersebut.

Proses bisnis yang terdapat pada organisasi UKM Fokus tidak luput dari berbagai permasalahan-permasalahan yang terjadi di lapangan. Proses pemesanan saat ini yang dilakukan *client* yaitu dengan menggunakan media *social messenger* atau *email* ke akun resmi UKM Fokus. Pihak organisasi pun tidak dapat secara langsung menerima tawaran proyek yang datang dari setiap *client*, namun harus melakukan proses koordinasi terlebih dahulu dengan fotografer anggota organisasi yang berada di *region* atau kota yang di pesan

oleh *client* perihal jadwal pemotretan. Apabila fotografer yang dihubungi oleh *Project Manager* tersebut memiliki kegiatan lain pada tanggal yang di pesan oleh *Client*, maka *Project Manager* akan mencoba menghubungi fotografer anggota lainnya yang dapat mengerjakan proyek tersebut. Proses koordinasi ini pun dilakukan dengan menggunakan media *social messenger* atau *email*. Adanya pembatas-pembatas wilayah penyediaan jasa ini menyebabkan banyak waktu dan *resource* yang terbuang hanya untuk melakukan proses koordinasi serta menimbulkan rasa kurang nyaman yang dirasakan oleh aktor-aktor yang terlibat di dalam proses bisnis organisasi UKM Fokus.

Maka solusi untuk mengatasi permasalahan di atas adalah dengan membangun sistem *inFormasi* yang dapat menangani proses pemesanan jasa fotografi dan koordinasi antara aktor-aktor yang terlibat dalam proses bisnis komunitas UKM Fokus berbasis *Android*. Sistem dibangun berbasis *android* agar mempermudah dalam mendapatkan akses *inFormasi* secara cepat, tepat dan akurat. Berdasarkan atas gambaran tersebut, maka penulis tertarik untuk menjadikan permasalahan tersebut sebagai bahan penelitian dalam skripsi dengan memilih judul “**Aplikasi Pemesanan Jasa Fotografi dan Videografi berbasis Mobile**”.

B. Rumusan Masalah

Berdasarkan pemaparan pada bagian latar belakang masalah, maka permasalahan yang akan dikaji pada penelitian ini adalah sebagai berikut :

Bagaimana membuat Aplikasi Pemesanan Jasa Fotografi dan Videografi berbasis Mobile?

C. Batasan Masalah

Untuk mendapatkan sasaran sistem *inFormasi* yang tepat dan tujuan dari penelitian dapat tercapai, penulis membatasi ruang lingkup sistem *inFormasi* yang akan dirancang ini adalah sebagai berikut:

1. Aplikasi yang dibangun hanya untuk digunakan di UKM Fokus.
2. Sistem *inFormasi* ini hanya membahas mengenai sistem pemesanan jasa foto dan Video pada UKM Fokus.
3. Pembayaran pemesanan jasa fotografi dan videografi pada UKM Fokus menggunakan pembayaran sistem COD.
4. Data yang diolah dalam aplikasi pemesanan jasa fotografi ini berupa data data *client*, data fotografer, dan data proyek yang akan di kerjakan.
5. *InFormasi* yang dihasilkan berupa data proyek yang *valid* beserta fotografer yang mengerjakannya sebagai acuan bagi *fotografer* dan project manager dalam bentuk tampilan di halaman website, *invoice* bukti pembayaran client dalam bentuk file PDF, dan laporan periodik untuk ketua komunitas dalam bentuk file PDF.

D. Tujuan Penelitian

Adapun Adapun Tujuan dari Penelitian ini adalah sebagai barikut:

1. Untuk mengintegrasikan sistem pemesanan jasa fotografi dan videografi yang sedang berjalan pada UKM Fokus.
2. Untuk membuat sistem *inFormasi* pemesanan jasa fotografi dan videografi agar mempermudah proses bisnis pada UKM Fokus.

3. Untuk memadukan proses koordinasi antar *Project Manager* dan fotografi dan videografi yang terlibat dalam proses bisnis pada fotografi dan videografi.
4. Untuk dapat menentukan jadwal sesi pemotretan dengan cepat oleh pihak UKM Fokus.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Penelitian Sebelumnya

Penelitian relevan berisikan penelitian yang terdahulu kemudian dicari bahan penelitian yang ada dan relevan dengan penelitian ini karena sangat berguna untuk mendukung kajian teori yang dikemukakan, sehingga dapat digunakan sebagai landasan pada penyusunan kerangka pikir. Beberapa hasil penelitian yang relevan dengan penelitian ini adalah penelitian yang dilakukan oleh:

1. Andi Pratama Putra (2021) dengan judul penelitian “Aplikasi Pemesanan Personal Fotografer Berbasis Perangkat Bergerak”. Fotografi menurut KBBI adalah seni dan penghasilan gambar dan cahaya pada film atau permukaan yang dipekatkan. Umumnya fotografer memasarkan jasa layanan melalui media sosial (Instagram, Whatsapp, dan media sosial lainnya) atau secara langsung dengan komunikasi personal atau penyebaran pamflet, dan setiap proses pemesanan dilakukan dengan komunikasi melalui aplikasi pesan atau dengan telepon. Pada umumnya pengguna layanan fotografi menemukan fotografer mereka melalui Instagram dan dari rekomendasi teman yang telah menggunakan jasa fotografer sebelumnya. Penulis menyimpulkan beberapa masalah dari sisi pelanggan, yaitu tidak jelasnya harga atau keuntungan paket yang ditawarkan fotografer, kurangnya portofolio fotografer untuk dilihat oleh calon pengguna layanan fotografi, sulit dalam membanding fotografer satu dengan yang lain karena

pengguna layanan harus menghubungi fotografer satu per satu untuk membandingkan mana yang terbaik, serta susah mendapatkan fotografer di tanggal tertentu karena jadwal fotografer sudah penuh pada tanggal tersebut.

2. Ozy Prayogi (2018) dalam penelitiannya yang berjudul " sistem inFormasi pemesanan jasa foto dokumentasi Dan video *shooting* pada *ozy photography* berbasis Web dan sms *gateway*" Tujuan dari penelitian ini adalah untuk menghasilkan sistem inFormasi pemesanan jasa foto dokumentasi dan video shooting pada *ozy photography* berbasis web dan *sms gateway* sehingga dapat mempermudah konsumen dalam proses pendaftaran dan pemesanan, mempermudah *Admin* dalam proses pengolahan data sehingga proses pembuatan laporan bisa lebih cepat dan dapat membantu dalam mengirimkan promo melalui sms gateway.
3. Dimas Septa Eka Aditya (2019) dalam penelitiannya yang berjudul "Portal Jasa Fotografi Dan *Videografi Online Berbasis Web Responsif*" Portal Jasa Fotografi dan *Videografi Online Berbasis Web Responsif* merupakan sebuah aplikasi yang digunakan untuk menyediakan layanan bagi para fotografer dan videografer untuk memasarkan dan mengelola jasanya. Jasa fotografi dan videografi sering kali dibutuhkan untuk mendokumentasikan sebuah acara maupun hal-hal yang dianggap sangat penting seperti acara pernikahan, khitanan, *prewedding*, acara wisuda, acara perpisahan, *workshop*, foto keluarga dan lain-lain. Dengan aplikasi ini orang-orang yang akan lebih mudah menemukan jasa fotografi dan videografi dengan harga dan service yang sesuai keinginan mereka. Penggunaan aplikasi ini diharapkan dapat mempermudah bagi penyedia jasa dan orang yang akan memesan menentukan kapan

service dilakukan agar tidak bertabrakan jadwal pemesanan yang lain. Aplikasi ini dirancang menggunakan UML (*Unified Modelling Language*) dan dibuat menggunakan bahasa pemrograman PHP dan MySQL untuk mengolah databasenya, yang mana diharapkan dapat membantu dan mempermudah pihak yang bersangkutan.

B. JAVA

1. Pengertian Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program . Java juga mendukung sumber daya internet yang saat ini populer, yaitu *World Wide Web* atau yang sering disebut web. Java juga mendukung aplikasi klien/server, baik dalam jaringan lokal (LAN) maupun jaringan berskala luas (WAN).

Java dikembangkan oleh Sun *Microsystems* pada Agustus 2021, dengan nama Oak. Konon Oak adalah pohon semacam jati yang terlihat dari jendela tempat pembuatnya, *James Gosling*, bekerja. Ada yang mengatakan bahwa Oak adalah singkatan "*Object Application Kernel*", tetapi ada juga yang menyatakan hal itu muncul setelah nama Oak diberikan pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java.

Dalam sejumlah literatur disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C, C++, Object-C, *SmallTalk*, dan Common LISP. Selain itu Java juga dilengkapi dengan unsur keamanan. Yang tak kalah penting adalah bahwa Java menambahkan unsur paradigma pemrograman yang sederhana.

Dengan kata lain, Java merupakan bahasa pemrograman berorientasi objek (suatu model pengembangan perangkat lunak yang saat ini populer). Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas untuk membentuk suatu objek. Sejumlah kelas sudah tersedia dan Anda dapat menggunakannya dengan mudah, dan bahkan dapat mengembangkannya dengan mudah, serta dapat mengembangkannya lebih jauh melalui konsep pewarisan. Pewarisan adalah sifat yang ada pada bahasa pemrograman berorientasi objek, yang memungkinkan sifat-sifat suatu objek diturunkan dengan mudah ke objek lain. (Abdul Kadir; 2019)

2. Karakteristik Java

a. Sederhana

Bahasa pemrograman Java menggunakan sintaks yang mirip dengan bahasa C++ namun sintaks pada Java telah banyak diperbaiki, terutama dengan menghilangkan *pointer* yang rumit dan *multipleinheritance*. Java juga menggunakan *automatic memory allocation* dan *garbage collection*.

b. Berorientasi Objek

Java merupakan bahasa pemrograman berorientasi objek yang memungkinkan program untuk dibuat secara modular dan digunakan kembali.

c. Terdistribusi

Java dibuat untuk memudahkan distribusi aplikasi dengan adanya *networking libraries* yang terintegrasi dalam Java.

d. *Interpreted*

Program Java dijalankan menggunakan program *Interpreter*, yaitu Java *Virtual Machine* (JVM). Hal ini menyebabkan source code Java yang telah dikompilasi menjadi *bytecodes* dapat dijalankan pada berbagai *platForm*.

e. *Robust*

Java mempunyai *reliabilitas* yang tinggi. *Kompiler* pada Java mempunyai kemampuan mendeteksi *error* yang lebih baik dibandingkan bahasa pemrograman yang lain. Java mempunyai *Runtime Exception Handling* untuk membantu mengatasi *error* pada pemrograman.

f. *Secure*

Sebagai bahasa pemrograman aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga agar aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

3. Cara Kerja

Program Java bersifat tidak bergantung pada *platForm*, artinya Java dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi. Ketidakbergantungan terhadap *platForm* sering dinyatakan dengan istilah portabilitas. Yang menarik, tingkat portabilitas Java tidak hanya sebatas program sumber (*source code*), melainkan juga pada tingkat kode biner yang disebut *bytecode*. Kode yang disebut *bytecode* dapat dijalankan pada berbagai sistem operasi karena kode ini berbeda dengan kode mesin. Kode mesin sangat bergantung pada *platForm*, sedangkan *bytecode* dapat dimengerti oleh semua *platForm* yang telah dilengkapi dengan interpreter Java. Mengingat bahwa hasil kompilasi Java dapat

dijalankan pada sebarang sistem operasi ataupun prosesor, Java sering dikatakan bersikap netral terhadap arsitektur komputer.

C. *Smartphone*

Smartphone adalah perangkat telekomunikasi serbaguna. *Smartphone* atau yang disebut juga piranti pintar dewasa ini menjadi trend yang mewabah di Indonesia, digemarinya *Smartphone* ini juga bukan tanpa alasan, tapi karena feature yang ditawarkan sangat menarik dan mengubah hobi pengguna untuk *browsing*, *chating* dan semacamnya yang awalnya dilakukan secara statis, sekarang dengan *smartphone* bisa dilakukan dengan mobile atau bergerak.

Seperti halnya pengguna atau *User* yang semakin cerdas memilih perangkat pintarnya, vendor pun bersaing menawarkan *feature* dan *interface* yang menarik dan mudah untuk penggunanya, mereka menawarkan *OS (operating system)* dan aplikasi yang memanjakan penggunanya, RIM menawarkan OS *Blackberry*, *Apple* dengan *Iphonenya*, Samsung, Sony, Huawei, dan beberapa vendor lainnya mengadopsi *Android* sebagai OS nya, ada juga yang memakai jasa *Windows Phone* untuk OS , seperti yang dilakukan oleh Nokia. (Nazruddin Safaat H, 2019)

D. *Android*

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh *Android Inc.* Dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan

telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. (Naruddin Safaat H, 2012)

Android menyediakan *platForm* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Posel pertama yang menggunakan *Android* sebagai OSnya adalah *HTC Dream* yang dirilis pada 22 Oktober 2008. Dan hingga kini setidaknya ada sudah ada 8 jenis versi OS *Android* yang sudah beredar di dunia telepon seluler. Jenis-jenis *Android* tersebut antara lain:

1. *Android versi 1.1*

Android versi 1.1 di rilis pada 9 Maret 2009 oleh Google. *Android* versi ini dilengkapi disupport oleh Google Mail Service dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. *Android versi 1.5 Cup Cake*

Android Cup Cake di rilis pada pertengahan Mei 2009, masih oleh Google Inc. *Android* ini dilengkapi software development kit dengan berbagai pembaharuan termasuk penambahan beberapa fitur antara lain yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube, upload gambar ke Picasa langsung dari telepon, serta mendapat dukungan Bluetooth A2DP.

3. *Android versi 1.6 Donut*

Android Donut di rilis pada September 2009 menampilkan proses pencarian yang lebih baik dibandingkan versi-versi sebelumnya. Selain itu *Android Donut* memiliki fitur-fitur tambahan seperti galeri yang memungkinkan pengguna untuk

memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech*. Android Donut juga dilengkapi baterai indikator, dan kontrol applet VPN.

4. *Android versi 2.0/2.1 Eclair*

Android Eclair dirilis pada 3 Desember 2009. Perubahan yang ada antara lain adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan *Bluetooth* 2.1. Android Eclair merupakan Android pertama yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan *User interface*.

5. *Android versi 2.2 Froyo (Frozen Yogurt)*

Android Froyo dirilis pada 20 Mei 2012. Android versi ini memiliki kecepatan kinerja dan aplikasi 2 sampai 5 kali dari versi-versi sebelumnya. Selain itu ada penambahan fitur-fitur baru seperti dukungan Adobe Flash 10.1, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. *Android versi 2.3 Gingerbread*

Android Gingerbread di rilis pada 6 Desember 2010. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (*User Interface*) didesain ulang, dukungan *Format* video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan

kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. *Android versi 3.0/3.1 Honeycomb*

Android *Honeycomb* di rilis pada awal 2012. Merupakan versi Android yang dirancang khusus untuk device dengan layar besar seperti Tablet PC. Fitur baru yang ada pada *Android Honeycomb* antara lain yaitu dukungan terhadap prosesor multicore dan grafis dengan hardware acceleration. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. Tablet pertama yang memakai *Honeycomb* adalah tablet *Motorola Xoom* yang dirilis bulan Februari 2011. Selain itu sebuah perangkat keras produksi Asus bernama Eee Pad *TransFormer* juga menggunakan OS Android *honeycomb* dan diharapkan akan masuk ke pasaran Indonesia pada Mei 2011.

8. *Android versi 4.0 ICS (Ice Cream Sandwich)*

Android *Ice Cream Sandwich* diumumkan secara resmi pada 10 Mei 2011 di ajang *Google I/O Developer Conference (San Francisco)*, pihak Google mengklaim *Android Ice Cream Sandwich* akan dapat digunakan baik di smartphone ataupun tablet. *Android Ice Cream Sandwich* membawa fitur *Honeycomb* untuk smartphone serta ada penambahan fitur baru seperti membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi *inFormasi* dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung *Galaxy Nexus*.

9. *Android versi 4.1 Jelly Bean*

Android Jelly Bean juga diluncurkan pada acara Google I/O 10 Mei 2011 yang lalu. Android versi ini membawa sejumlah keunggulan dan fitur baru, diantaranya peningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Versi ini juga dilengkapi *Google Now* yang dapat memberikan inFormasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui inFormasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 pertama kali digunakan dalam produk tablet Asus, yakni Google Nexus 7.

10. *Android Versi 4.4 Kitkat*

Sistem operasi *Android* terbaru yang akan melanjutkan sistem operasi Android seri 4.1, 4.2 dan 4.3 *Jelly Bean* ini mempunyai nama *KitKat*. Pemberian nama *KitKat* sendiri cukup mengejutkan bagi sejumlah kalangan, pasalnya jauh sebelumnya rumor telah beredar luas bahwa sistem operasi penerus *Jelly Bean* mempunyai nama *Key Lime Pie*.

Pemberian nama *KitKat* sendiri memang mengejutkan, pasalnya *KitKat* merupakan nama sebuah produk bukan nama sebuah makanan umum yang biasa digunakan *Google*. Namun alasan *Google* menggunakan nama *KitKat* adalah produk ini merupakan coklat batangan kesukaan seorang Kepala Teknik *Android Google Hiroshi Lockheimer*, ungkap juru bicara *Google*.

Yang menarik dari hal ini adalah tidak terjadi adanya pertukaran uang di dalam penamaan Android *KitKat* yang dilakukan oleh pihak *Google* dan *Nestle*. Namun, elemen penting yang terjadi di dalam pemberian nama ini adalah produk

KitKat adalah adanya produk *KitKat* sebanyak 50 juta yang ada di 19 negara akan memiliki branding *Android* dan menawarkan pembeli untuk mendapatkan kesempatan memenangkan *Google Play Gift Card* dan *Nexus 7*.

11. *Android lollipop (versi 5.0)*

Google kembali meluncurkan versi android terbarunya diberi nama Android lollipop atau disingkat dengan android L. Beberapa media terkemuka seperti tech times menuturkan bahwa android lollipop ini akan dirilis, memiliki fitur super canggih dan sudah tertanam dalam tablet nexus 9. Banyak sekali fitur baru dalam android lollipop.

12. *Android marshmallow (versi 6.0)*

Google kembali merilis versi teranyar dari si robot hijau versi android 6.0 ini dinamakan dengan android marshmallow. Pengembangan android telah menanamkan berbagai fitur baru nan canggih dalam versi ini termasuk juga penyempurnaan fitur dari versi sebelumnya.

13. *android nougat (versi 7.0)*

Versi selanjutnya adalah android versi 7.0 atau disebut juga dengan android nougat. Versi nougat ini resmi dirilis pada tanggal 22 agustus 2016 berbagai fitur pada android nougat juga telah disempurnakan termasuk juga penambahan fitur dan fungsi.

E. Eclipse

Eclipse adalah sebuah IDE(*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platForm

(*platForm-independent*). *Eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. (Wahana Komputer, 2019)

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini.

F. Android Developer Tools

Android Developer Tools (ADT) merupakan *Plugin* untuk *Eclipse* yang menyediakan lingkungan pengembangan untuk membangun aplikasi Android. ADT adalah Java IDE penuh dengan fitur-fitur canggih untuk membantu Anda membangun, menguji, debug, dan paket aplikasi Android. (Nazruddin Safaat H, 2019)

G. XML

XML(*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML(*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet. XML didesain untuk mampu menyimpan data secara ringkas dan

mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan inFormasi. (Wahana Komputer, 2019)

H. JRE (*Java Runtime Environment*)

Java Runtime Environment merupakan perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dengan versi JDK yang digunakan untuk membangun aplikasi agar aplikasi dapat berjalan sesuai dengan yang diharapkan. (Eko Kurniawan K, 2019)

I. JDK (*Java Development Kit*)

Java Development Kit (JDK) adalah sebuah produk yang dikembangkan oleh Oracle yang ditujukan untuk para developer java. Sejak Java, diperkenalkan JDK merupakan *Java Software Development Kit* (SDK) yang paling serig digunakan Pada tanggal 17 November 2006, Sun mengatakan akan merilis JDK sebagai opensource menggunakan lisensi GNU *General Public License* (GPL), artinya software ini sekarang menjadi free software. Pada tanggal 8 Mei 2007, Sun mengontribusikan *source code*-nya ke OpenJDK.

JDK juga berisi paket *Java Runtime Environment* yang komplit, biasaya disebut private run time karena biasanya dipisah dari JRE reguler dan dilengkapi dengan konten tambahan, yaitu terdiri dari Java Virtual Machine dan semua class library yang ada di *environment* produk. Selain itu termasuk juga library tambahan yang berguna untuk *developer*, misalnya library *internationalization* dan library IDL.

Salinan dari JDK juga berisi beberapa contoh program yang menjelaskan penerapan porsi dari API Java. *Form* JDK merupakan bagian dari *software development kit* (SDK). Untuk Java SE, EE, dan ME, Sun mengatakan bahwa JDK

merupakan bagian dari SDK yang memiliki tanggung jawab untuk menuliskan dan menggabungkan program java yang sedang berjalan. Bagian SDK lainnya antara lain *application server, debugger*, dan dokumentasi. (Wahana Komputer; 2019)

J. Android studio

Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan *google* pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut Menggunakan *Gradle-based build system* yang fleksibel.

1. Bisa mem-*build* multiple APK .
2. *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.
3. Built-in support untuk *Google Cloud PlatForm*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
4. *Import library* langsung dari *Maven repository*

Ketika awal saya berkenalan dengan Android, saya menggunakan Eclipse sebagai IDE untuk membuat aplikasi Android. Jika dibandingkan dengan Android Studio memang dari sisi build lebih baik dibandingkan Eclipse, karena Android Studio menggunakan Gradle. Ditambah lagi berbeda dengan Eclipse, kita tidak perlu

lagi dipusingkan dengan *dependencies package* pada Android Studio. Satu hal tambahan lagi yang membuat Android Studio unggul adalah dukungan layout xml editor secara visual yang jauh lebih baik daripada Eclipse. Walaupun begitu, Android Studio saat ini masih dalam tahap beta dan belum mempunyai dukungan untuk NDK/Native Development Kit.

a. Android Studio Sebagai Code Editor Cerdas

Inti dari Android Studio adalah editor kode cerdas mampu *code completion* dengan cerdas, *refactoring*, dan analisis *code*. Editor yang baik membantu Anda menjadi pengembang aplikasi Android lebih produktif.

b. Tersedia template dan bisa integrasi GitHub

New project wizards membuatnya lebih mudah dari sebelumnya untuk memulai sebuah project baru, memulai project menggunakan kode template untuk pola seperti navigasi laci dan melihat halaman, dan bahkan mengimpor contoh kode Google dari GitHub.

c. *Multi-screen app development*

Membangun aplikasi untuk ponsel Android, tablet, Android *Wear*, Android TV, Android Auto dan *Google Glass*. Dengan Project Android View baru dan modul dukungan di Android Studio, lebih mudah untuk mengelola project aplikasi dan sumber daya.

d. *Virtual devices for all shapes and sizes*

Android Studio hadir pra-dikonfigurasi dengan emulator gambar yang optimal. Diperbarui dan efisien *Virtual Device Manager* memberikan profil perangkat yang telah ditetapkan untuk perangkat Android umum.

e. *Android builds evolved, with Gradle*

Membuat beberapa APK untuk aplikasi Android Anda dengan fitur yang berbeda menggunakan project yang sama. Mengelola aplikasi dependensi dengan Maven. Membangun APK dari Android Studio atau *command line*.

f. *System Requirements Android Studio*

1). Windows

1. Microsoft® Windows® 10/8/7/Vista (32 or 64-bit)
2. 2 GB RAM minimum, 4 GB RAM recommended
3. 400 MB hard disk space
4. *At least 1 GB for Android SDK, emulator system images, and caches*
5. 1280 x 800 minimum screen resolution
6. Java Development Kit (JDK) 7
7. *Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality*

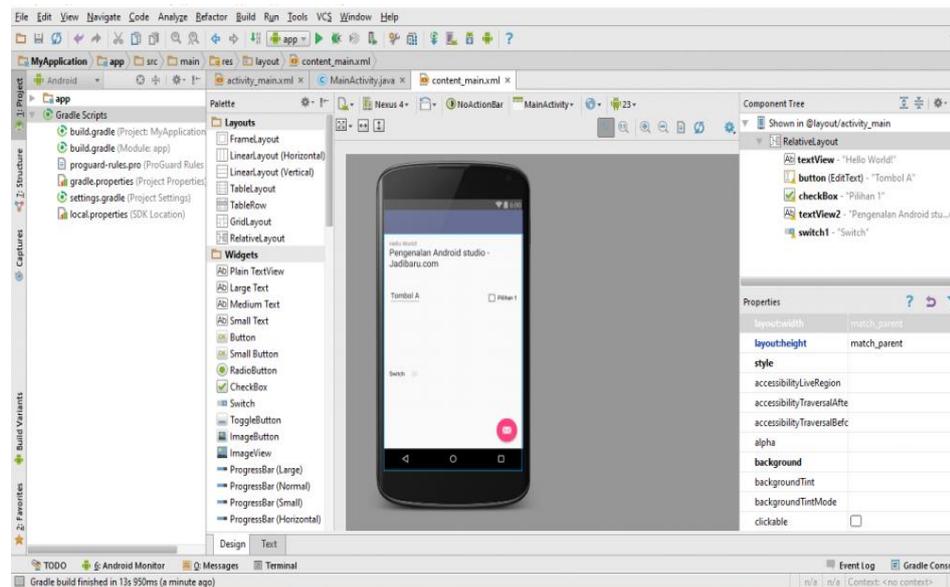
2) Mac OS X

1. Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
2. 2 GB RAM minimum, 4 GB RAM recommended
3. 400 MB hard disk space
4. At least 1 GB for Android SDK, emulator system images, and caches
5. 1280 x 800 minimum screen resolution
6. Java Runtime Environment (JRE) 6
7. Java Development Kit (JDK) 7
8. *Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality*

On Mac OS, run Android Studio with Java Runtime Environment (JRE) 6 for optimized font rendering. You can then configure your project to use Java Development Kit (JDK) 6 or JDK 7.

3) Linux

1. GNOME or KDE desktop
2. GNU C Library (glibc) 2.15 or later
3. 2 GB RAM minimum, 4 GB RAM *recommended*
4. 400 MB hard disk space
5. *At least 1 GB for Android SDK, emulator system images, and caches*
6. 1280 x 800 minimum screen resolution
7. Oracle® Java Development Kit (JDK) 7



Gambar 2 1. Android Studio

K. Tesseract Engine

1. Pengertian

Tesseract merupakan sebuah *library open source* untuk mengenali karakter OCR (*Optical Character recognition*) yang pada awalnya di kembangkan oleh Hawlet-Packard antara tahun 1985 dan 1995.

Tesseract tidak pernah dikomersialkan. Keakuratannya diuji di *The Fourt Annual Test of OCR Accurasi* yang diselenggarakan pada tahun 1995 di Universitas Nevada Las Vegas' *InFormation Science Research Institute*. Namun pada saat itu pengembangan *tesseract* telah berhenti. (Digib Unila,2011)

2. Cara Kerja

Gambaran mengenai cara kerja Tesseract secara singkat adalah:

- 1). Garis-garis dianalisis dan disimpan
- 2). Garis-garis berkumpul bersama sebagai himpunan

- 3). Himpunan tersebut akan disusun dalam baris teks
- 4). Baris teks akan dipecah menjadi kata-kata
- 5). Langkah pertama proses pengenalan mencoba untuk mengenali tiap kata pada urutannya
- 6). Kata yang tepat harus melewati proses penyesuaian oleh *adaptive trainer*
- 7). Pembelajaran oleh *adaptive trainer* berlangsung pada tahap kedua yang mencoba mengenali kata-kata yang tidak dikenali secara tepat pada tahap pertama
- 8). Proses fuzzy diselesaikan dan teks diperiksa *small caps*
- 9). Teks digital dikeluarkan

Selama proses tersebut berlangsung, *tesseract* menggunakan:

- Algoritma untuk mendeteksi *skewed pages*.
- Algoritma untuk mendeteksi kata-kata proporsional dan non proporsional (proporsional merupakan sebuah kata di mana semua huruf yang lebar yang sama).
- Algoritma untuk mencacah karakter bergabung dan untuk menghubungkan karakter yang rusak.
- Linguistik analisis untuk mengidentifikasi kata yang paling mungkin dibentuk oleh sekelompok karakter.
- Dua karakter pengklasifikasi: classifier statis, dan sebuah classifier yang adaptif menggunakan data pelatihan, dan yang lebih baik dalam membedakan antara atas dan huruf kecil.

Tesseract memiliki akurasi yang tinggi jika teks yang diterjemahkan berasal dari bahasa yang *support* untuk bahasa Inggris, 8 komponen yang digunakan:

1. *General Words Wordlist* (tessdata/eng.word-dawg).
2. *Frequent Word Wordlist* (tessdata/eng.freq-dawg).
3. *User Wordlist* (tessdata/eng.User-words).
4. *Index for Character Set* (tessdata/eng.inttemp).
5. Box file for use in locating characters in the training file (tessdata/eng.normproto).
6. Box file for use in locating characters in the training file³⁰ (tessdata/eng.pffmtable).
7. *Language's Character Set* (tessdata/eng.unicharset).
8. *Character Cluster Disambiguator for 'm' and 'rn', for instance.* (tessdata/eng.DangAmbigs).

L. UML(*Unified Modelling Language*)

UML (*Unified Modelling Language*) menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

a. Sesuatu (*things*)

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

1. *Structural things*, bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu.
3. *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem-subsistem.
4. *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi (*relationship*)

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

1. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemn independent akan mempengaruhi elemen yang bergantung padanya.
2. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

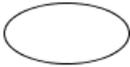
3. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
 4. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.
- c. Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).
 - d. diagram *use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai

Adapun simbol-simbol *Use case* Diagram antara lain :

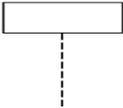
Tabel 2.1 Simbol *Use case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

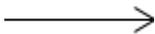
NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

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

Tabel 2.3 Simbol *Sequence Diagram*

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

Tabel 2.4 Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

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

Tabel 2.5 Simbol *Actifity Diagram*

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

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif 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 Februari tahun 2023, di UKM Fokus Universitas Muhammadiyah Parepare, dimana proses pengambilan data yang akan dimasukkan dalam aplikasi nantinya

NO	Uraian Kegiatan	Bulan 2024				
		Februari	Maret	Maret	April	April
1	Studi Literatur					
2	Perancangan aplikasi					
3	Pembuatan aplikasi					
4	Pengujian					

C. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui beberapa cara yaitu :

1. Analisis Data

Menganalisa data-data yang sebelumnya telah dikumpulkan.

2. Perancangan Program

Sebagai pedoman dalam penulisan program atau kode-kode agar berjalan sesuai rencana.

3. Uji Coba Program

Pengujian program dilakukan untuk memastikan bahwa program yang dibuat dapat berjalan dengan baik.

4. Evaluasi

Sistem yang telah selesai dibangun perlu adanya evaluasi untuk menemukan kelemahan yang terdapat pada program yang telah dibangun tadi, yang nantinya bisa digunakan sebagai acuan untuk memperbaiki program sehingga lebih sempurna.

D. Jenis Data

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

a. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

b. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, website dan jurnal.

E. Tahapan Penelitian

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

1. Persiapan Penelitian

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

2. Pengumpulan Data

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

3. Analisis

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

4. Perancangan

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

5. Pengujian

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

6. Implementasi

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

F. Metode Pengujian Aplikasi

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *typical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik.

Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

Uji coba *White Box Testing* merupakan metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian *White Box Testing* ini merupakan peringatan ketika *User* menginputkan password *User* yang salah, untuk kesalahan semacam ini akan memberikan suatu inFormasi kepada *User* mengenai kesalahan yang di lakukan.

2. *Black Box*

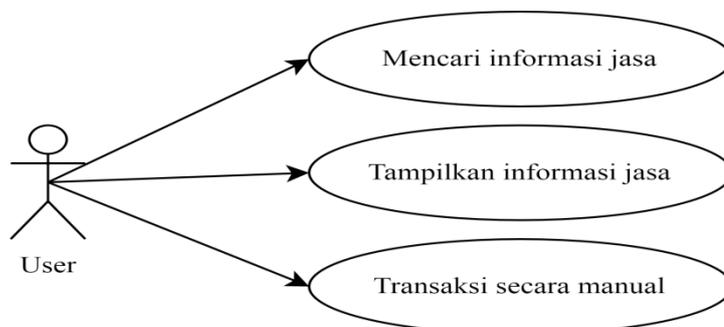
Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika *User* belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan inFormasi kepada *User* karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

G. Metode Pengujian

Metode pengujian yang dilakukan pada penelitian ini menggunakan blackbox yang dipakai untuk mengetahui kinerja aplikasi pada saat melakukan proses *Tour Guide* dan mengetahui berfungsinya semua aplikasi yang telah dibuat, *blackbox* dalam metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja.

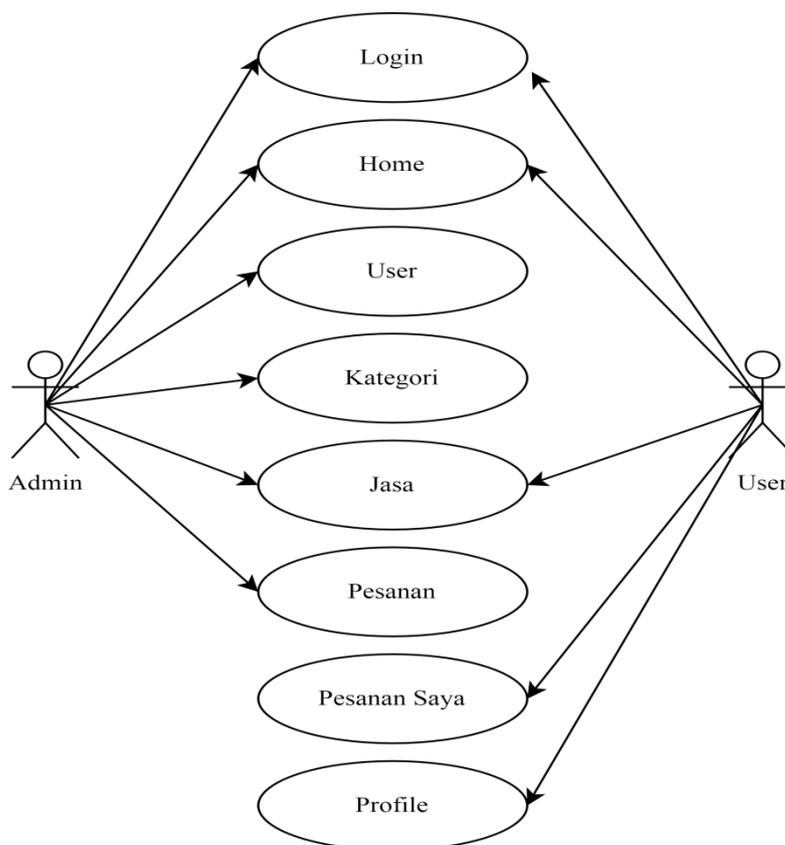
H. Desain Sistem

1. Sistem yang berjala



Gambar 3 1. Sistem yang Berjalan

2. Sistem yang diusulkan



Gambar 3 2. Sistem yang Diusulkan

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Aplikasi Pemesanan Jasa Fotografi dan Vidiografi berbasis Mobile, Menggunakan pemrograman PHP dan *Javascript*, sedangkan untuk pengelolaan basis data menggunakan MySQL. Metode yang digunakan dalam perancangan aplikasi ini adalah UML (*Unified Modelling Language*).

B. Pembahasan

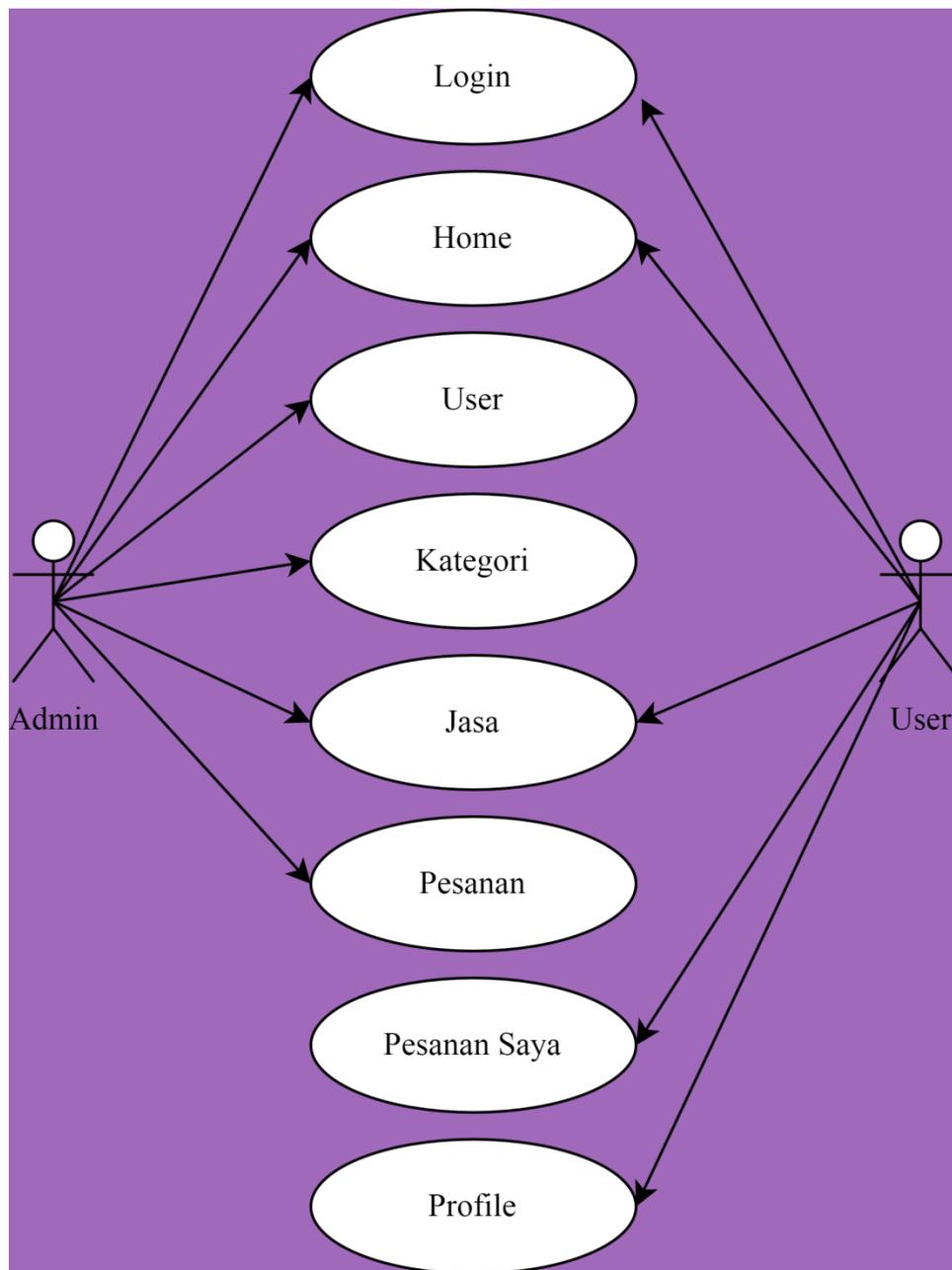
1. Analisis Aliran Data UML

1.1. 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* merupakan suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem.

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. *Use case Diagram* berfungsi untuk melakukan kelebihan sistem jika dilihat menurut sudut pandang individu yang berada di luar sistem (*User* dan *Admin*).



Gambar 4 1. Use Case Diagram.

Interaksi antara *User* dan *Admin* dengan aplikasi dijelaskan dalam skenario *use case* sebagai berikut:

1.2. 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	<i>User</i>	<i>User</i> merupakan aktor yang berperan sebagai pengguna sistem informasi akuntansi pembelian dan penjualan.
2	Admin	Admin merupakan aktor yang memiliki hak akses lebih tinggi dibanding <i>User</i> karena Admin dapat mengelola Jenis plafon dan halaman keranjang belanja..

Sedangkan untuk penjelasan definisi setiap *use case* sebagai berikut:

Tabel 4.2. Definisi *Use Case* Sistem

No	<i>Use Case</i>	Keterangan
1	<i>Login</i>	Melakukan Login akun.
2	<i>Home</i>	Menampilkan halaman utama aplikasi
3	<i>User</i>	Menampilkan halaman user.
4	Kategori	Menampilkan halaman kategori.
5	Jasa	Menampilkan halaman jasa
6	Pesanan	Manampilkan halaman pesanan.
7	Pesanan Saya	Menampilkan halaman pesanan saya.
8	Profile	Menampilkan halaman profile

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:

1) *Use Case Login*

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk Masuk ke aplikasi

Pre-condition : -

Post-condition : - Data baru sudah tersimpan.

Tabel 4.3. ..Skenario *Login*.

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan Username dan password.	
2. Menekan tombol Login.	3. Mengecek validitas isian data.
	4. Jika data yang dimasukkan valid, maka langsung masuk kehalaman utama aplikasi.
Jika data yang dimasukkan tidak valid, maka akan muncul pesan “Silahkan masukkan <i>user name</i> dan <i>password</i> dengan benar”.	

2) *Use Case Home*.

Aktor : Admin dan User

Deskripsi : Sebuah kegiatan untuk Menampilkan menu-menu

Pre-condition : - . Admin 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. Memilih salah satu menu.	
	4. Jika menu yang dipilih valid, maka menu tersebut akan ditampilkan di form baru.
Jika menu yang diakses tidak valid, maka menu tidak dapat diakses.	

3) Use Case User

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk menampilkan user

Pre-condition : .Admin berada di halaman user.

Post-condition : -.

Tabel 4.5. Skenario *Use Case User*

Aksi Aktor	Reaksi Sistem
1. Memilih menu user	
	2. Menampilkan form user

4) Skenario *Use Case Kategori*

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk melihat kategori

Pre-condition :- Admin berada di halaman kategori

Post-condition : -

Tabel 4.6. Skenario Use Case Kategori

Aksi Aktor	Reaksi Sistem
1. Menekan menu kategori	
	2. Menampilkan data yang dipilih (kategori) yang dimaksud dalam sistem.
	3. Menampilkan data yang dipilih sesuai data yang ada.

5) Skenario Use Case Jasa

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk melihat jasa

Pre-condition : Admin berada di halaman jasa

Post-condition : - Data dalam sistem akan diperbarui.

Tabel 4.7. Skenario *Use Case* jasa

Aksi Aktor	Reaksi Sistem
1. Memilih menu jasa	
	2. Menampilkan form jasa

6) Skenario Use Case Pesanan

Aktor : Admin

Deskripsi : Sebuah kegiatan untuk melihat pesanan

Pre-condition :- Admin berada di halaman pesanan

Post-condition : -

Tabel 4.8. Skenario *Use Case* pesanan

Aksi Aktor	Reaksi Sistem
1. Memilih menu pesanan	
	2. Menampilkan form pesanan

7) Skenario *Use Case* Pesanan Saya

Aktor : User

Deskripsi : Sebuah kegiatan untuk melihat pesanan saya

Pre-condition :- Admin berada di halaman pesanan saya

Post-condition : -

Tabel 4.9. Skenario *Use Case* pesanan saya

Aksi Aktor	Reaksi Sistem
1. Memilih menu pesanan saya	
	2. Menampilkan form pesanan saya

8) Skenario *Use Case* Profile

Aktor : User

Deskripsi : Sebuah kegiatan untuk melihat profile

Pre-condition : User berada di halaman profile

Post-condition : - Data dalam sistem akan diperbarui.

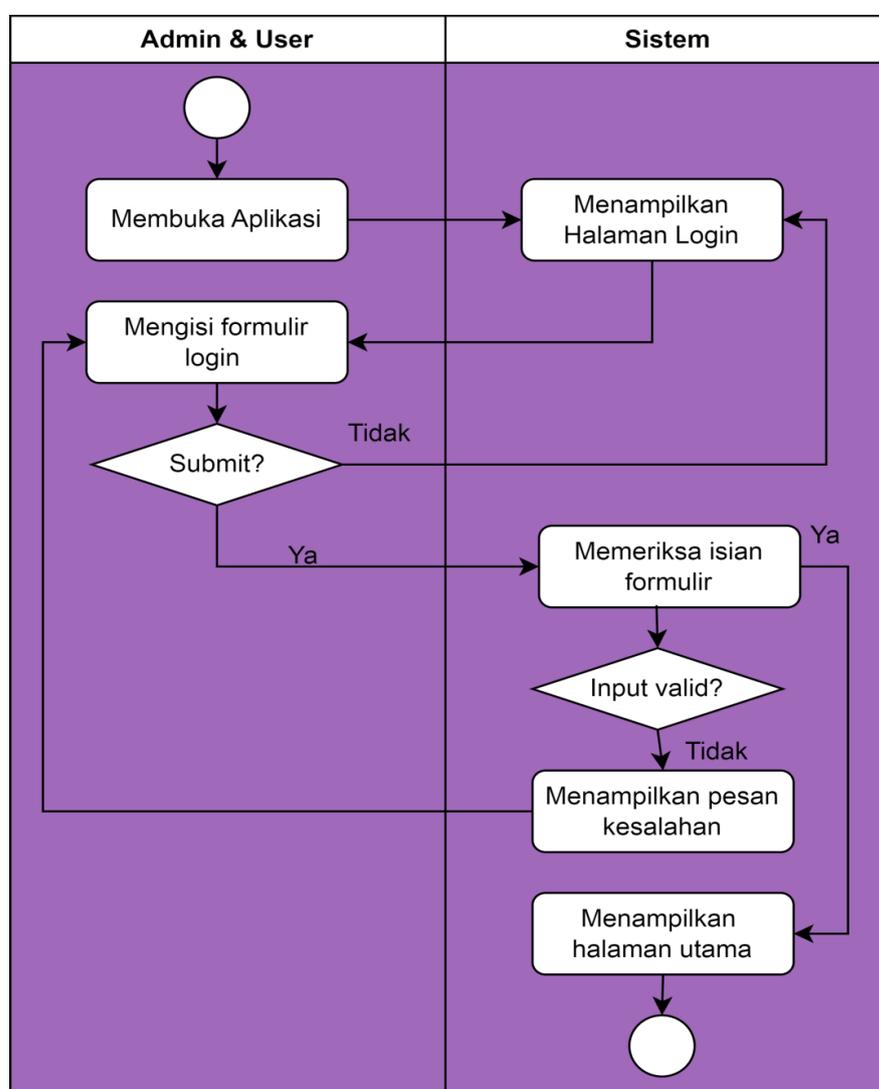
Tabel 4.10. Skenario *Use Case* profile

Aksi Aktor	Reaksi Sistem
1. Memilih menu profile	
	2. Menampilkan form profile

1.3. Activity Diagram

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. Activity diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sebuah sistem.

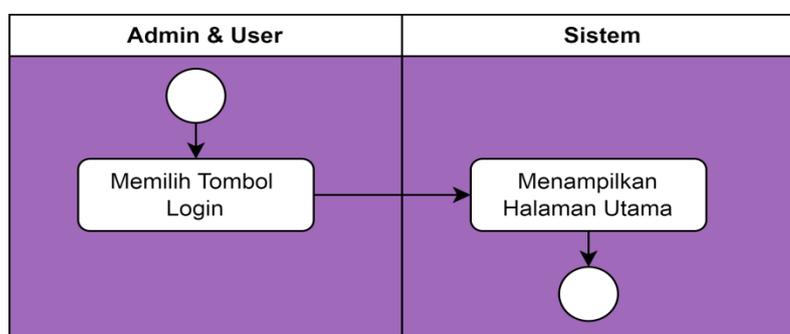
a) Activity Diagram Login



Gambar 4 2. Activity Diagram Login

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas login. Proses dimulai dengan admin 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.

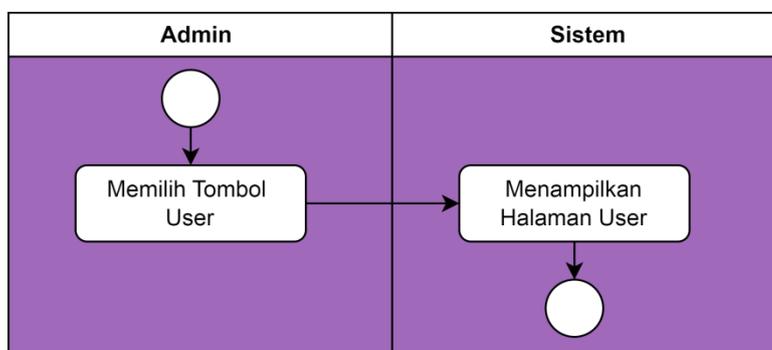
b) *Activity Diagram Home*



Gambar 4 3. Activity Diagram Home

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

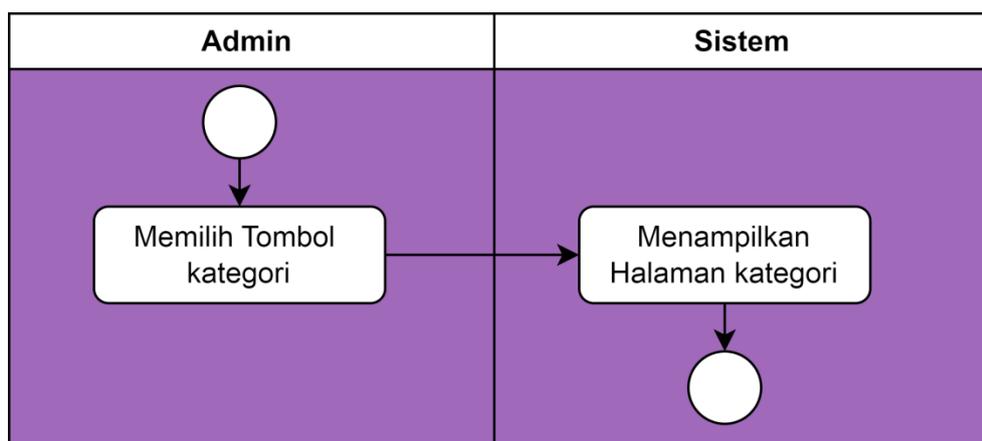
c) *Activity Diagram User*



Gambar 4 4. Activity Diagram User

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas user. Proses dimulai dengan admin memilih tombol user, kemudian sistem menampilkan halaman user.

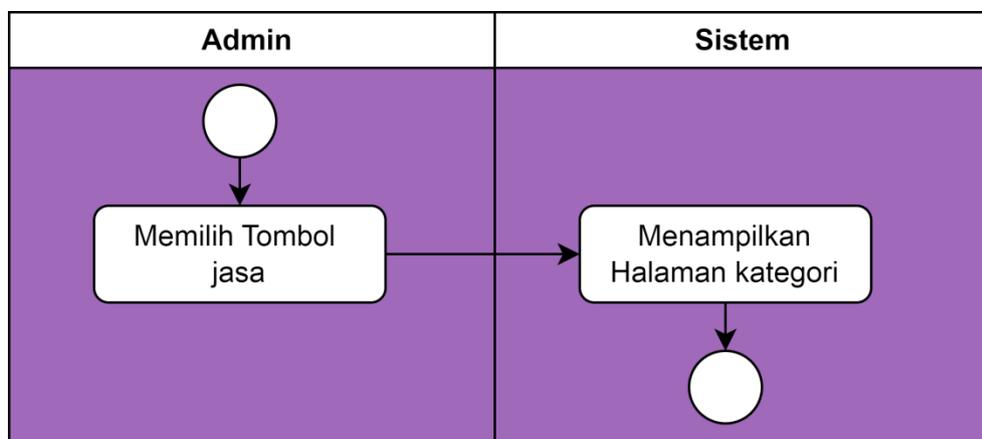
d) *Activity Diagram Kategori*



Gambar 4 5. Activity Diagram Kategori

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas kategori. Proses dimulai dengan admin memilih tombol kategori, kemudian sistem menampilkan halaman kategori.

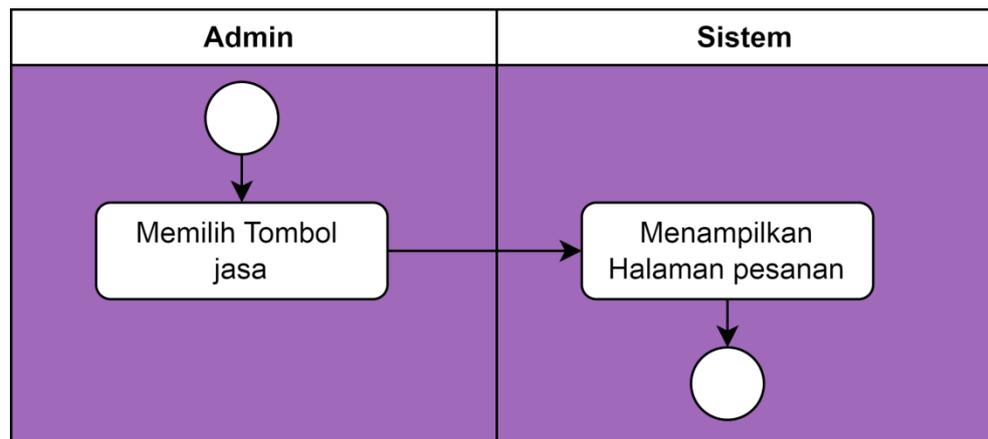
e) *Activity Diagram Jasa*



Gambar 4 6. Activity Diagram Jasa

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas jasa. Proses dimulai dengan admin memilih tombol jasa, kemudian sistem menampilkan halaman jasa.

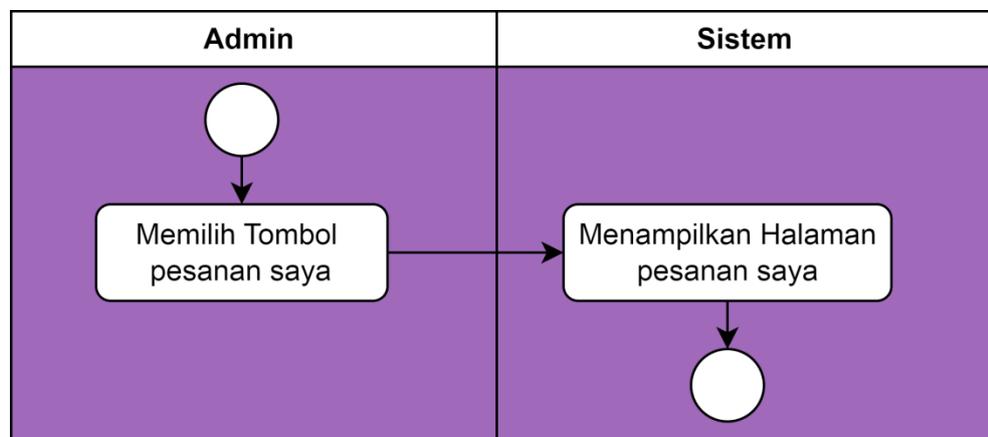
f) *Activity Diagram Pesanan*



Gambar 4 7. Activity Diagram pesanan

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas pesanan. Proses dimulai dengan user memilih tombol pesanan, kemudian sistem menampilkan halaman pesanan.

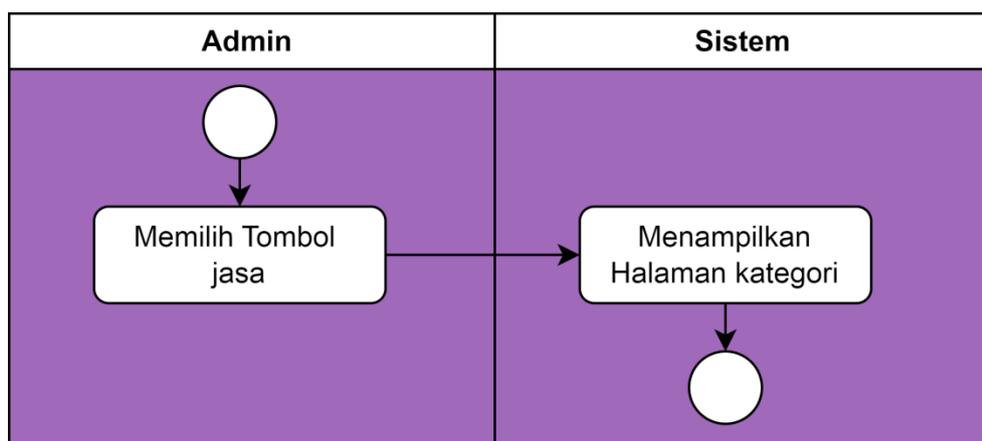
g) *Activity Diagram Pesanan Saya*



Gambar 4 8. Activity Diagram pesanan saya

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas pesanan saya. Proses dimulai dengan user memilih tombol pesanan saya, kemudian sistem menampilkan halaman pesanan saya.

h) Activity Diagram Profile



Gambar 4 9. Activity Diagram Profile

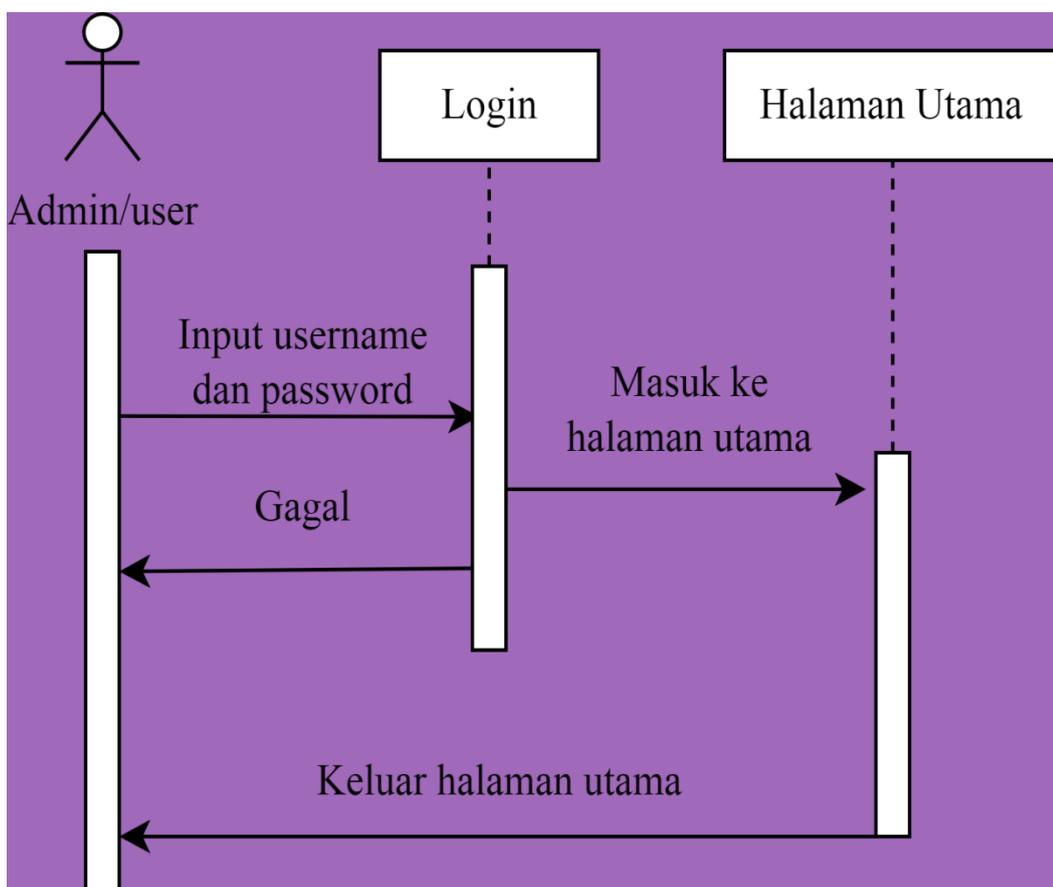
Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Profile. Proses dimulai dengan admin memilih tombol Profile, kemudian sistem menampilkan halaman Profile.

1.4. Sequence Diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan 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 pembahasan proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada sistem informasi yang dikembangkan:

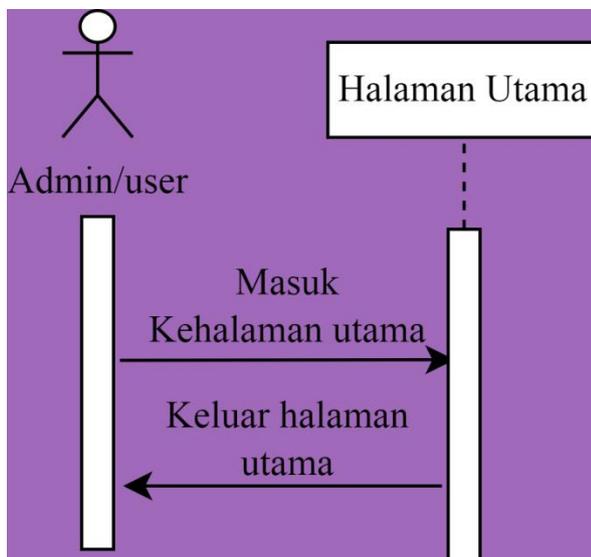
1) *Sequence Diagram: Login*



Gambar 4 10. Sequence Diagram login

Diagram Sequence diagram login Admin dan user merupakan penggambaran proses login yang dilakukan oleh admin dan *user*. Bila proses login benar admin dan user bisa masuk ke dalam halaman utama sebaliknya bila proses login gagal akan kembali ke menu *login*.

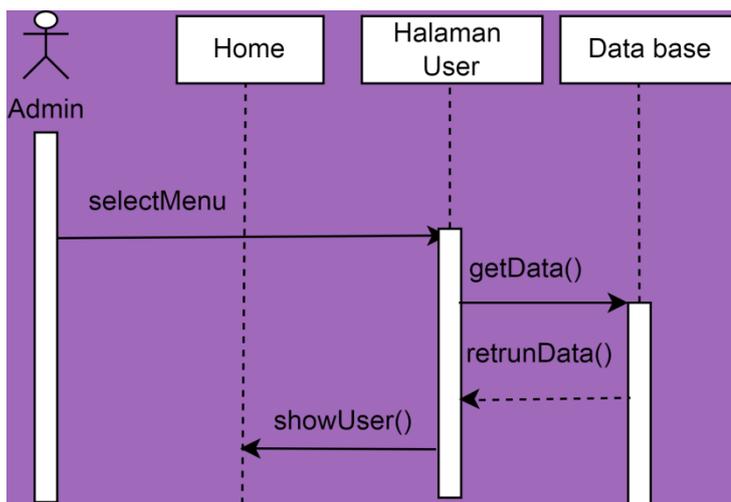
2) Sequence Diagram: Home



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.

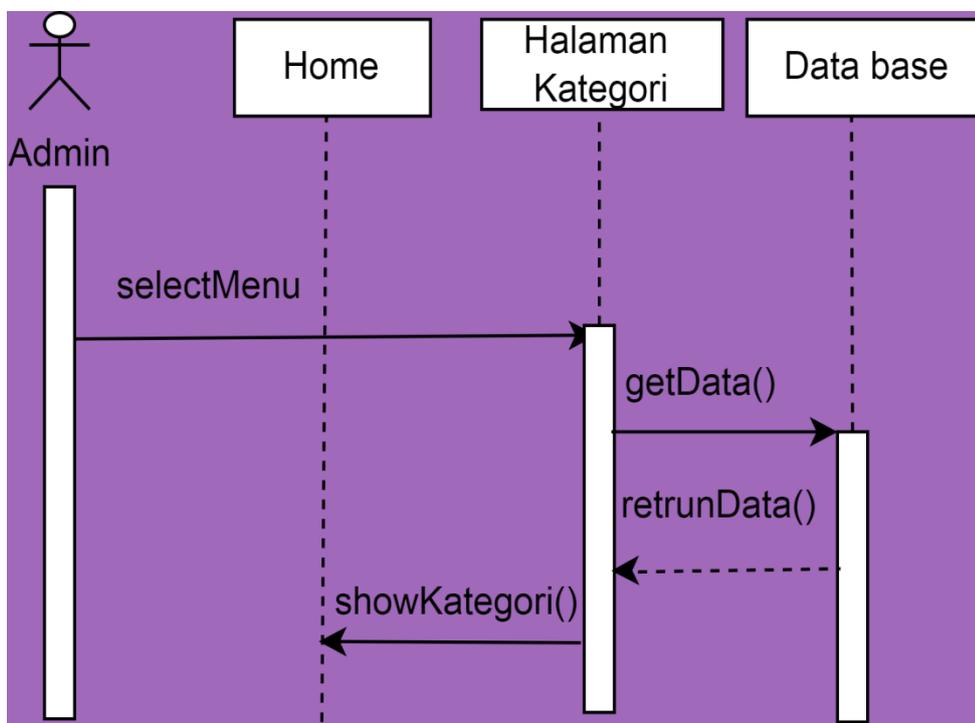
3) Sequence Diagram: User



Gambar 4 12. Sequence Diagram User

Diagram sekuen di atas menerangkan alur proses untuk aktivitas kelola user Admin memilih data yang akan ditampilkan pada halaman user, dengan method *selecMenu()*, kemudian method *getData()* digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowUser()* akan menampilkan User.

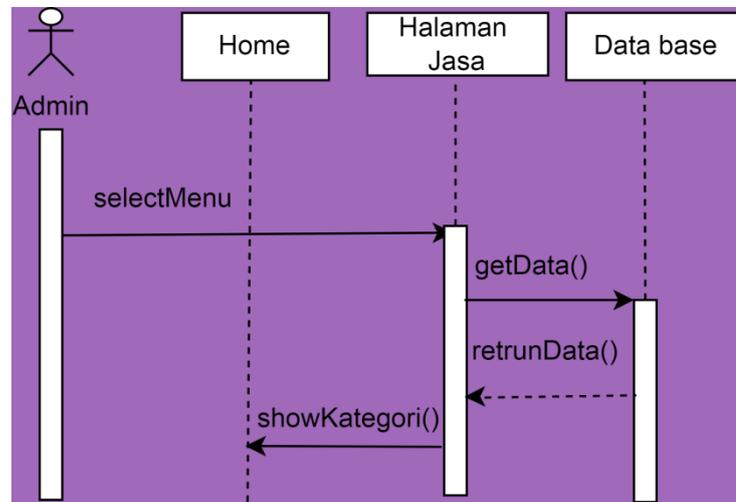
4) Sequence Diagram: Kategori



Gambar 4 13. Sequence Diagram Kategori

Diagram sekuen di atas menerangkan alur proses untuk aktivitas kelola kategori Admin memilih data yang akan ditampilkan pada halaman kategori, dengan method *selecMenu()*, kemudian method *getData()* digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowKategori()* akan menampilkan kategori.

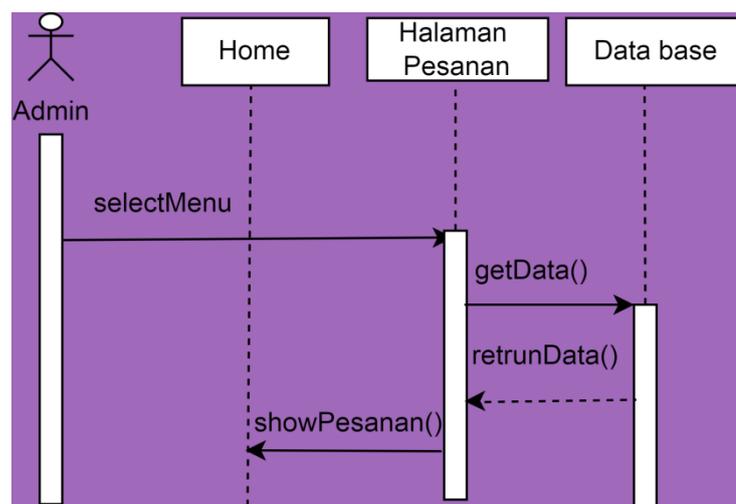
5) Sequence Diagram: Jasa



Gambar 4 14. Sequence Diagram Jasa

Diagram sekuen di atas menerangkan alur proses untuk aktivitas kelola jasa. Admin memilih data yang akan ditampilkan pada halaman jasa, dengan method `selectMenu()`, kemudian method `getData()` digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method `ShowJasa()` akan menampilkan jasa.

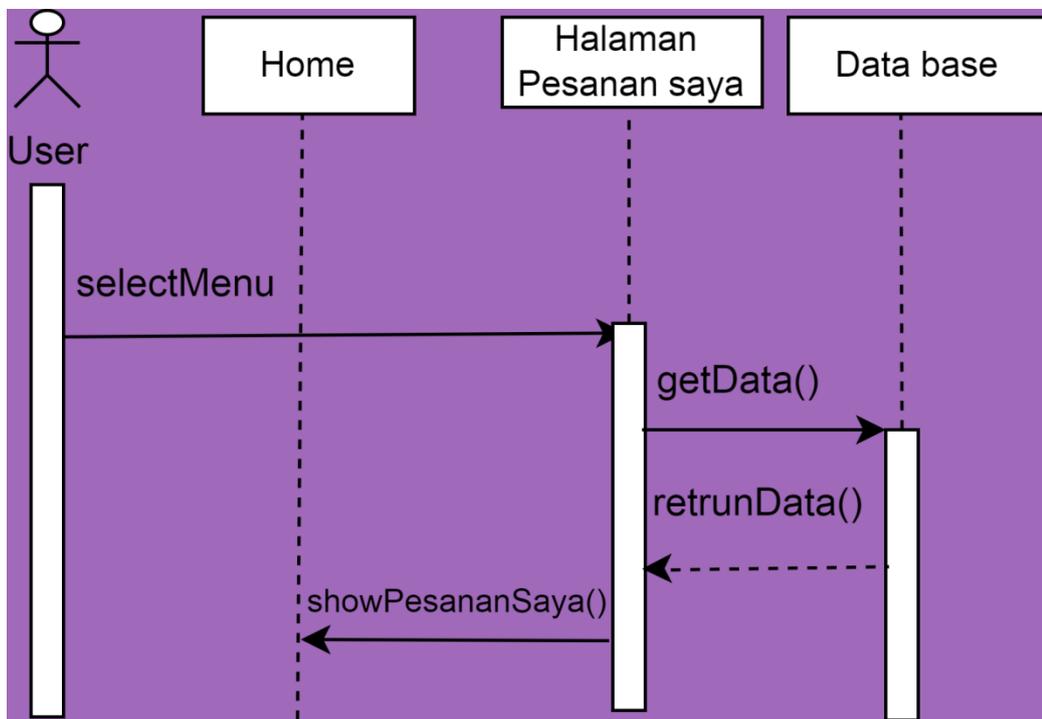
6) Sequence Diagram: Pesanan



Gambar 4 15. Sequence Diagram Pesanan

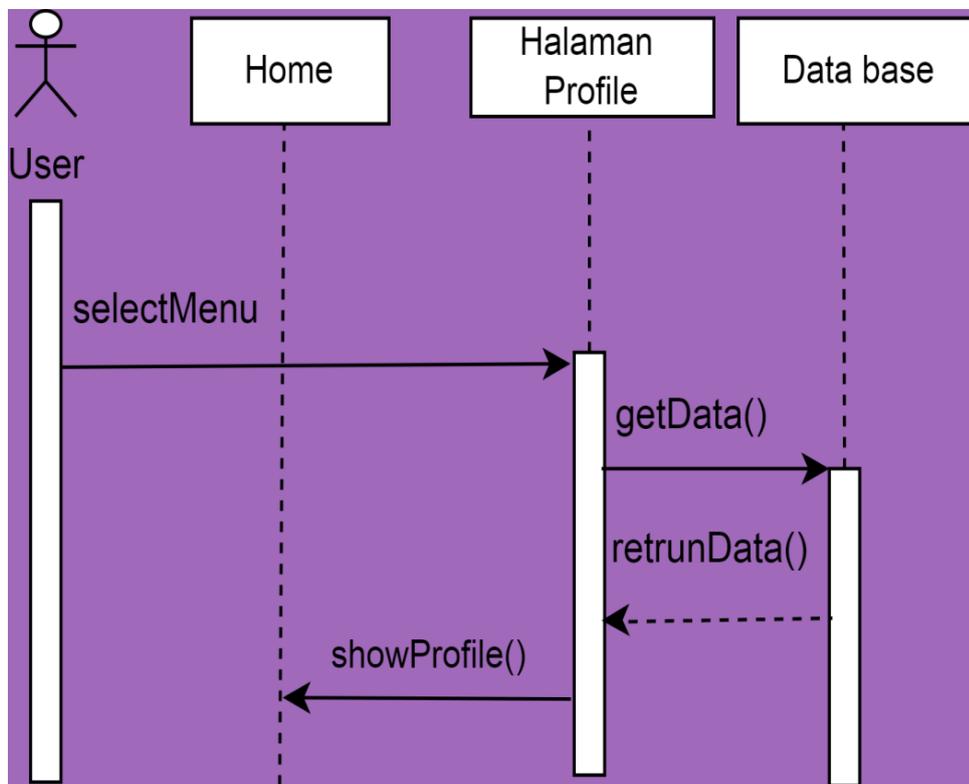
Diagram sekuen di atas menerangkan alur proses untuk aktivitas kelola pesanan Admin memilih data yang akan ditampilkan pada halaman pesanan, dengan method *selecMenu()*, kemudian method *getData()* digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowPesanan()* akan menampilkan pesanan.

7) *Sequence Diagram: Pesanan saya*



Gambar 4 16. Sequence Diagram Pesanan saya

Diagram sekuen di atas menerangkan alur proses untuk aktivitas pesanan saya user memilih data yang akan ditampilkan pada halaman pesanan saya, dengan method *selecMenu()*, kemudian method *getData()* digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowPesananSaya()* akan menampilkan pesanan saya.

8) *Sequence Diagram: Profile*

Gambar 4 17. Sequence Diagram Profile

Diagram sekuen di atas menerangkan alur proses untuk aktivitas profile user memilih data yang akan ditampilkan pada halaman profile, dengan method *selectMenu()*, kemudian method *getData()* digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method *ShowProfile()* akan menampilkan Profile.

1.5. Tabel Database

Adapun rancangan *database* dalam pembangunan aplikasi pemesanan jasa fotografi dan vidiografi berbasis mobile. sistem ini adalah terdiri dari beberapa tabel diantaranya sebagai berikut yaitu :

- a. Tabel User. Nama *table*: user, Adapun struktur *table* user sebagai berikut.

Tabel 4.11. Tabel User

No	Nama Field	Tipe Data	Ukuran
1	id_user	Varchar	12
2	Userid	Varchar	12
3	Passid	Varchar	8
4	nama	Varchar	20
5	alamat	Varchar	50

b. Tabel kategori. Nama *table*: kategori, Adapun struktur *table* kategori sebagai berikut.

Tabel 4.12. Tabel kategori

No	Nama Field	Tipe Data	Ukuran
1	id_kategori	Varchar	10
2	Id_vendor	Varchar	10
3	Id_jenis	Varchar	10
4	jumlah	Varchar	20

c. Tabel vendor Nama *table*: vendor , Adapun struktur *table* vendor sebagai berikut.

Tabel 4.13. Tabel vendor

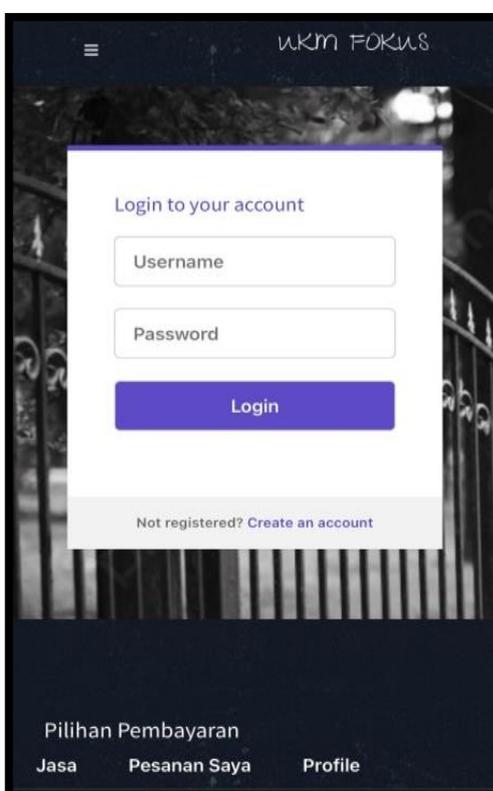
No	Nama Field	Tipe Data	Ukuran
1	id_vendor	Varchar	10
2	nama_vendor	Varchar	20
3	kode_vendor	Varchar	20

2. Perancangan Sistem

2.1. Halaman User

a. Halaman *Login* Akun

Halaman menu login merupakan halaman *login user*. *Login* merupakan tampilan yang ada pada sistem, yang digunakan user untuk masuk ke halaman selanjutnya.



Gambar 4 18. Login akun

b. Halaman Utama

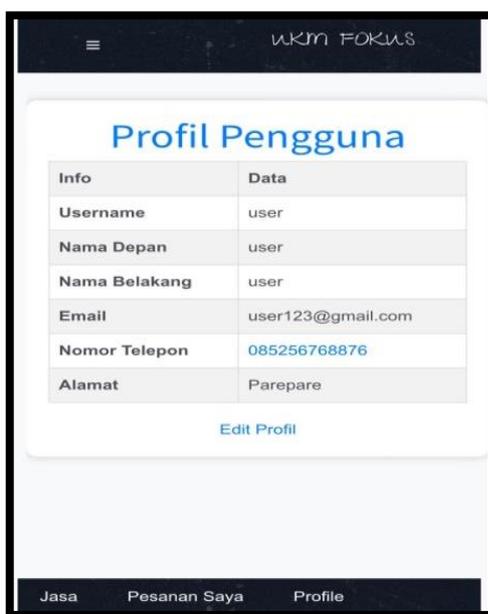
Berikut adalah tampilan halaman utama aplikasi ayo belajar bahasa arab. Halaman ini merupakan halaman awal pelanggan yang langsung menampilkan menu-menu yang ada pada halaman ini.



Gambar 4 19. Halaman Utama

c. Halaman Profil User

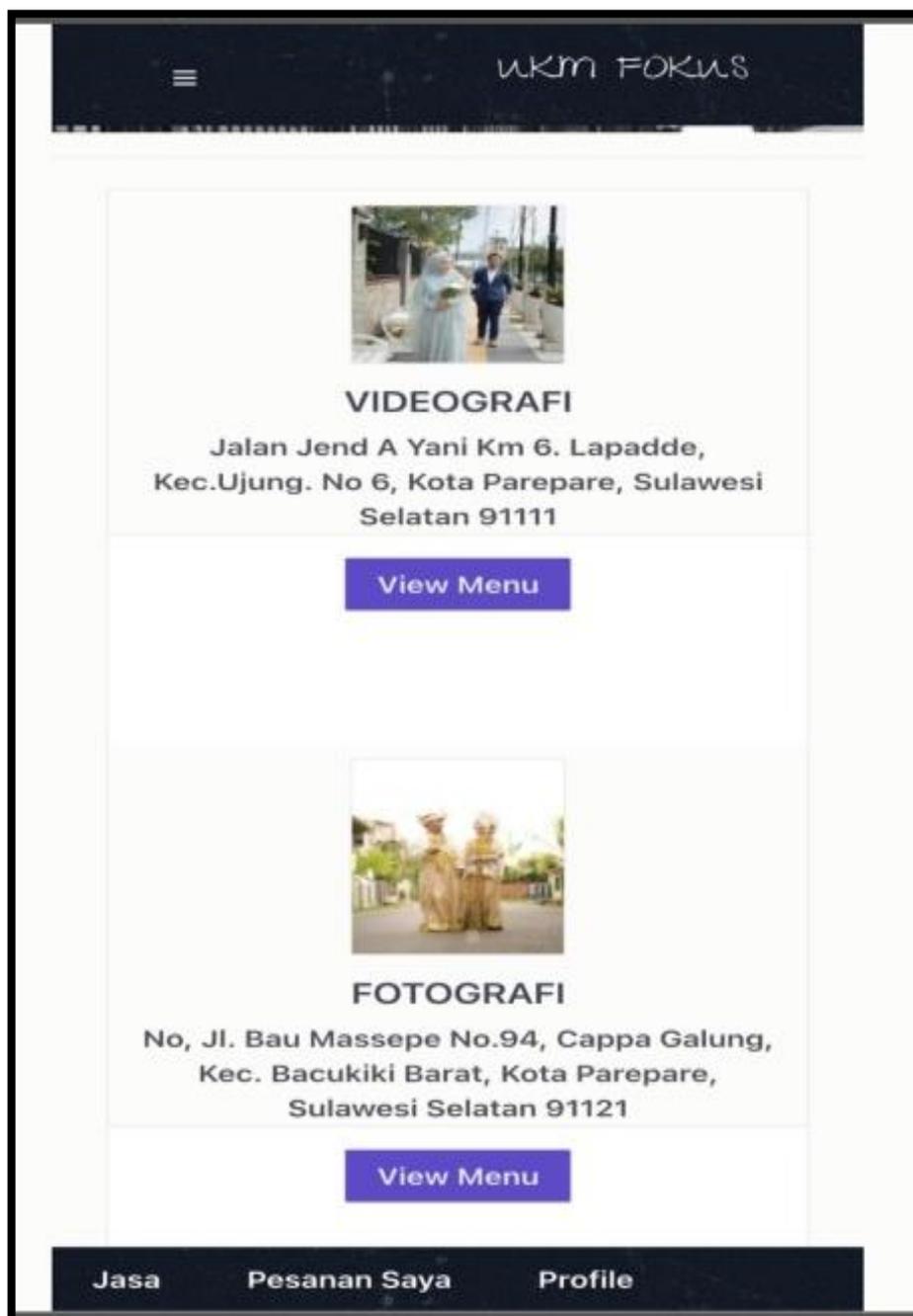
Berikut adalah tampilan profil pada aplikasi ayo belajar bahasa arab. Halaman profil ini menampilkan Edit Akun, Tentang, dan Logout.



Gambar 4 20. Profil User

d. Halaman Jasa

Berikut adalah tampilan halaman jasa. Halaman jasa menampilkan daftar jasa vidiografi dan fotografi lengkap dengan harga.



Gambar 4 21. Halaman Jasa

e. Halaman Pesanan Saya

Berikut adalah tampilan pesanan saya pada aplikasi pemesanan. Halaman pesanan saya menampilkan detail pesanan yang ada dalam aplikasi.

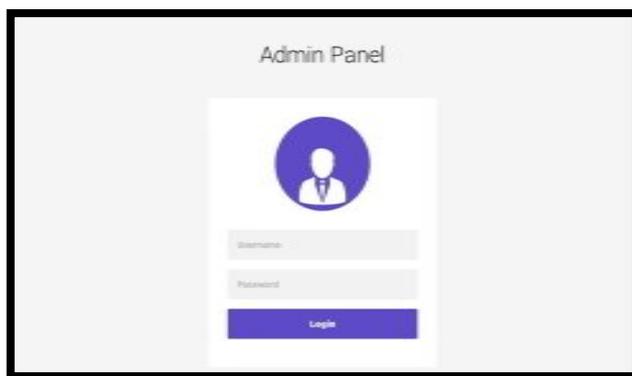
Jasa	Jumlah	Harga	Status	Pesanan	Mapus	Konfirmasi	Regulan	Regulan
Safoto Basic	1	Rp.3.200.000	⚠️ Dibatasi	2024-01-26 01:17:22	🚫	✅ Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 10:42:43	🚫	✅ Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Asyiah	1	Rp.1.500.000	⚠️ Dibatasi	2024-03-07 12:57:56	🚫	✅ Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Irs	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 12:37:58	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 11:59:47	🚫	✅ Konfirmasi Pembayaran	2024-03-12 11:00:00	ji pangaro
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 12:13:01	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 15:10:21	🚫	✅ Konfirmasi Pembayaran	2024-03-28 13:00:00	lasekita
Safoto Asyiah	1	Rp.1.500.000	⚠️ Dibatasi	2024-03-07 12:24:37	🚫	✅ Konfirmasi Pembayaran	2024-03-30 18:00:00	patem
Safoto Ulang Tahun	1	Rp.1.000.000	⚠️ Dibatasi	2024-03-07 12:36:48	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Pre Wedding	1	Rp.800.000	⚠️ Dibatasi	2024-03-07 12:38:10	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	u
Safoto Irs Wedding	1	Rp.800.000	⚠️ Dibatasi	2024-03-07 12:38:38	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	u
Safoto Ulang Tahun	1	Rp.1.000.000	⚠️ Dibatasi	2024-03-07 12:46:10	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	d
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 12:37:38	🚫	✅ Konfirmasi Pembayaran	2024-03-07 18:00:00	wirsa
Safoto Advance	1	Rp.8.000.000	⚠️ Dibatasi	2024-03-07 12:58:50	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	f
Safoto Asyiah	1	Rp.1.500.000	⚠️ Dibatasi	2024-03-07 13:01:02	🚫	✅ Konfirmasi Pembayaran	0000-00-00 00:00:00	w
Safoto Ulang Tahun	1	Rp.1.000.000	⚠️ Dibatasi	2024-03-07 13:03:42	🚫	✅ Konfirmasi Pembayaran	2024-03-30 22:03:32	ngaga
Safoto Pre Wedding	1	Rp.3.500.000	✅ Dibatasi	2024-03-07 13:10:30	🚫	✅ Konfirmasi Pembayaran	2024-03-28 13:06:00	Ren Pales
Safoto Advance	1	Rp.8.000.000	✅ Dibatasi	2024-03-07 13:19:09	🚫	✅ Konfirmasi Pembayaran	2024-04-08 22:18:09	Hutari Jempu

Gambar 4 22. Materi Terbaru

2.2. Admin

a. Halaman *Login Akun*

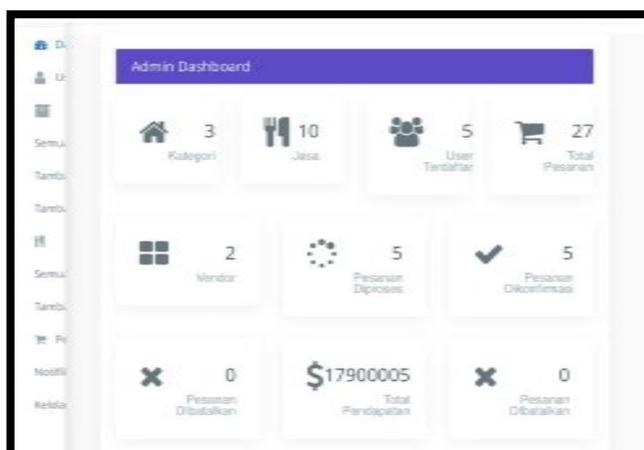
Halaman menu login merupakan halaman *login user*. *Login* merupakan tampilan yang ada pada sistem, yang digunakan user untuk masuk ke halaman selanjutnya.



Gambar 4 23.Login akun

b. Halaman Utama

Halaman ini merupakan halaman awal admin yang langsung menampilkan menu-menu yang ada pada halaman ini.



Gambar 4 24. Halaman Utama

c. Halaman User

Halaman ini merupakan halaman user pada menu admin yang langsung menampilkan halaman user.

Username	FirstName	LastName	Email	Phone
syarwansyakir	syarwan	syakir	syarwansyakir1302@gmail.com	08960

Gambar 4 25. Halaman User

d. Halaman Kategori

Halaman ini merupakan halaman kategori pada menu admin yang langsung menampilkan halaman kategori.

Vendor	Kategori	Email	Phone	Url
SAFOTO	FOTOGRAFI	fotografi@gmail.com	085234114277	fotografi.

Gambar 4 26. Halaman Kategori

e. Halaman Jasa

Halaman ini merupakan halaman jasa pada menu admin yang langsung menampilkan halaman jasa.

Kategori	Jasa	Deskripsi	Harga	Image	Hapus Atau Edit
FOTOGRAFI	ADAMSTORY	1. Album Magnetik Jumbo 2. 1 Album Kolase (30x40)	Rp.1,000,000		

Gambar 4 27. Halaman Jasa

f. Halaman Pesanan

Halaman ini merupakan halaman pesanan pada menu admin yang langsung menampilkan halaman pesanan.

User	Jasa	Jumlah	Harga	Lokasi	Status
User1	Wedding Cinematic	2	Rp.5	Parepare	Diterima!
User1	Wedding Cinematic	1	Rp.5	Parepare	Ditunggu!

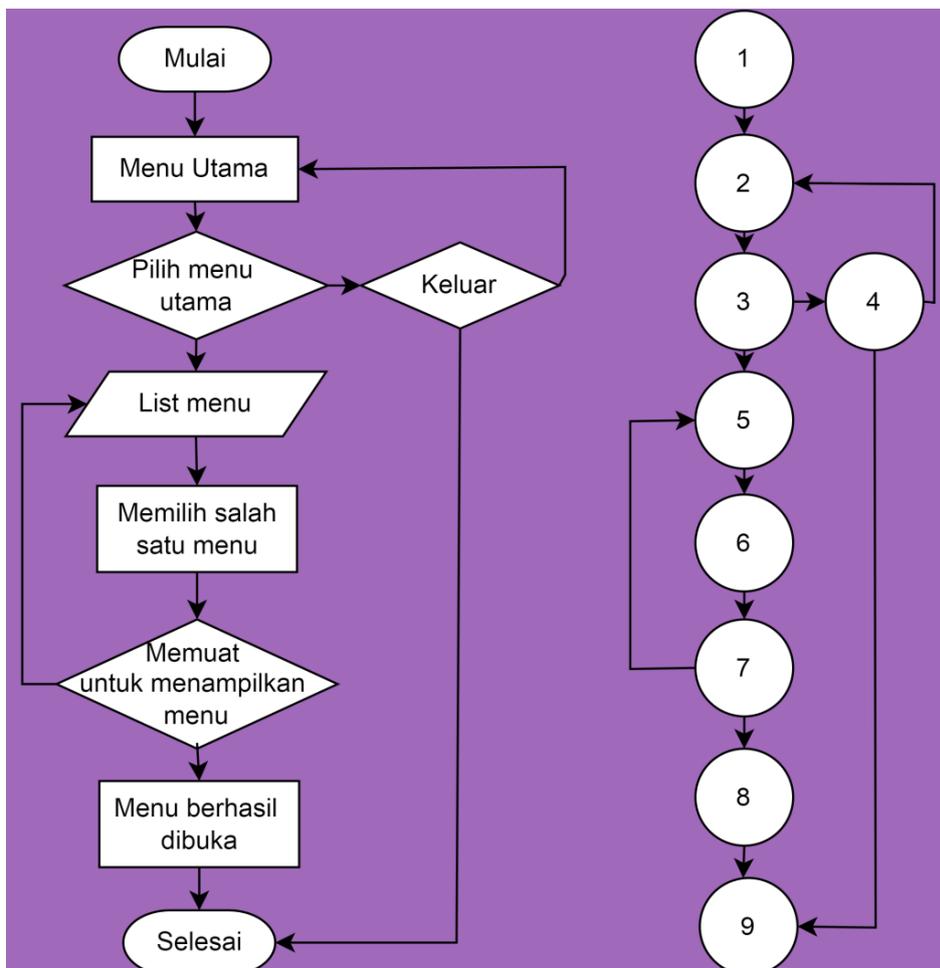
Gambar 4 28. Halaman Pesanan

3. Pengujian Sistem

3.1. White Box.User

White box testing atau pengujian kotak putih adalah salah satu metode pengujian perangkat lunak. Metode pengujian ini juga terkadang disebut juga *glass box testing* atau pengujian kotak kaca. Berbeda dengan *black box testing*, white box testing ini justru menguji perangkat lunak dari sisi internal tanpa memperhatikan fungsional seperti antarmuka perangkat lunak itu sendiri.

a) *White box* menu aplikasi.



Gambar 4 29. White box menu aplikasi.

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 9 \quad E = 11 \quad R = 2$$

Penyelesaian :

$$\begin{aligned} V(G) &= (E - N) + 2 \\ &= (11 - 9) + 2 \\ &= 4 \end{aligned}$$

Independent Path :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 2 - 3 - 5 - 6 - 7 - 8 - 9$$

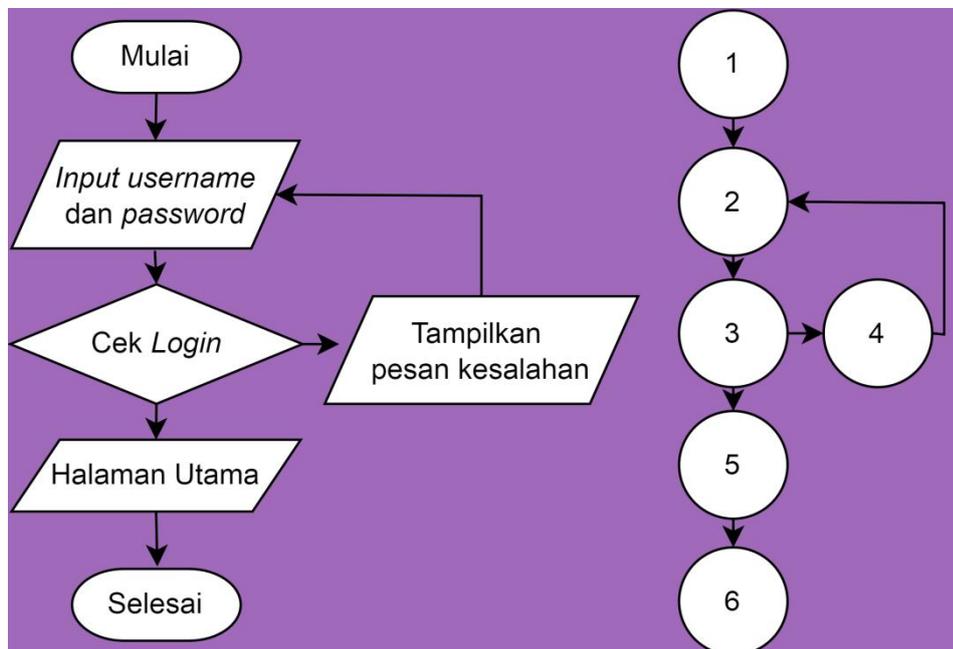
$$\text{Path 3} = 1 - 2 - 3 - 4 - 9$$

$$\text{Path 4} = 1 - 2 - 3 - 5 - 6 - 7 - 5 - 6 - 7 - 8 - 9$$

Tabel 4.15. *White Box* menu aplikasi

	1	2	3	4	5	6	7	8	9	E-1
1		1								1-1=0
2			1							1-1=0
3				1						1-1=0
4		1			1					3-1=2
5						1				1-1=0
6							1			1-1=0
7					1			1		2-1=1
8									1	1-1=0
9										1-1=0
SUM(E+1)										3+1=4

b) *White box* pada menu login



Gambar 4 30. White box pada menu login.

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6 \quad E = 6 \quad R = 2$$

Penyelesaian :

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

$$= (6 - 6) + 2$$

$$= 2$$

Independent Path :

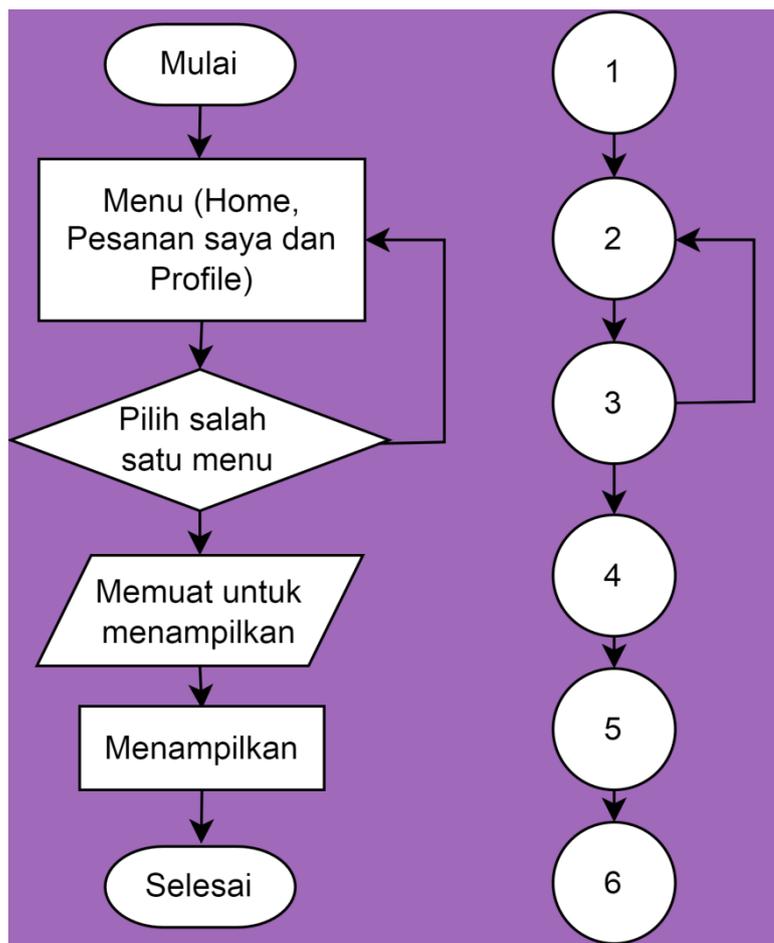
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

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

Tabel 4.16. White Box Perhitungan menu login

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

c) White box pada view menu user.



Gambar 4 31. White box pada View menu user

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6$$

$$E = 6$$

$$R = 2$$

Penyelesaian :

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

$$= (6 - 6) + 2$$

$$= 2$$

Independent Path :

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

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

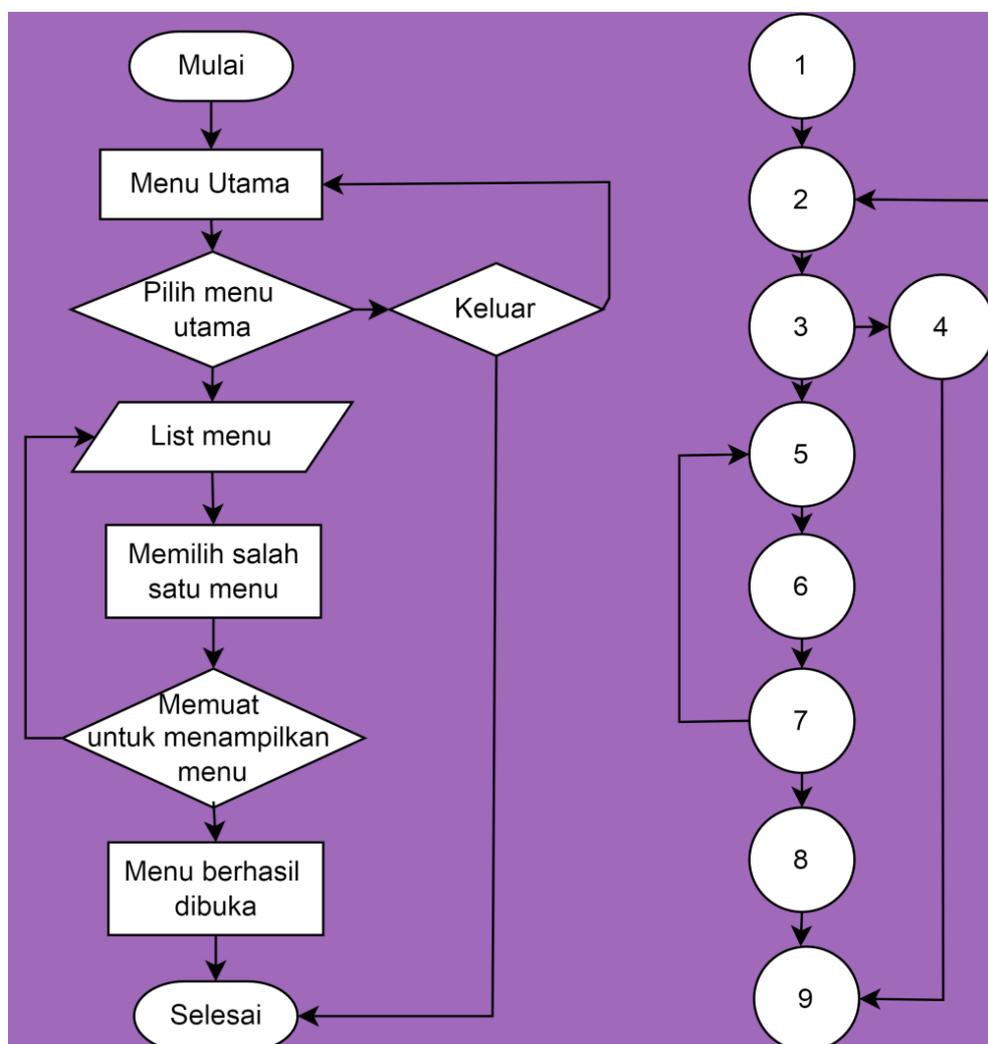
Tabel 4.17. *White Box* Perhitungan menu user.

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

3.2. White Box Admin

White box testing atau pengujian kotak putih adalah salah satu metode pengujian perangkat lunak. Metode pengujian ini juga terkadang disebut juga *glass box testing* atau pengujian kotak kaca. Berlainan dengan *black box testing*, white box testing ini justru menguji perangkat lunak dari sisi internal tanpa memperhatikan fungsional seperti antarmuka perangkat lunak itu sendiri.

a) *White box* menu aplikasi.



Gambar 4 32. White box menu aplikasi.

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 9 \quad E = 11 \quad R = 2$$

Penyelesaian :

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

$$= (11 - 9) + 2$$

$$= 4$$

Independent Path :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 2 - 3 - 5 - 6 - 7 - 8 - 9$$

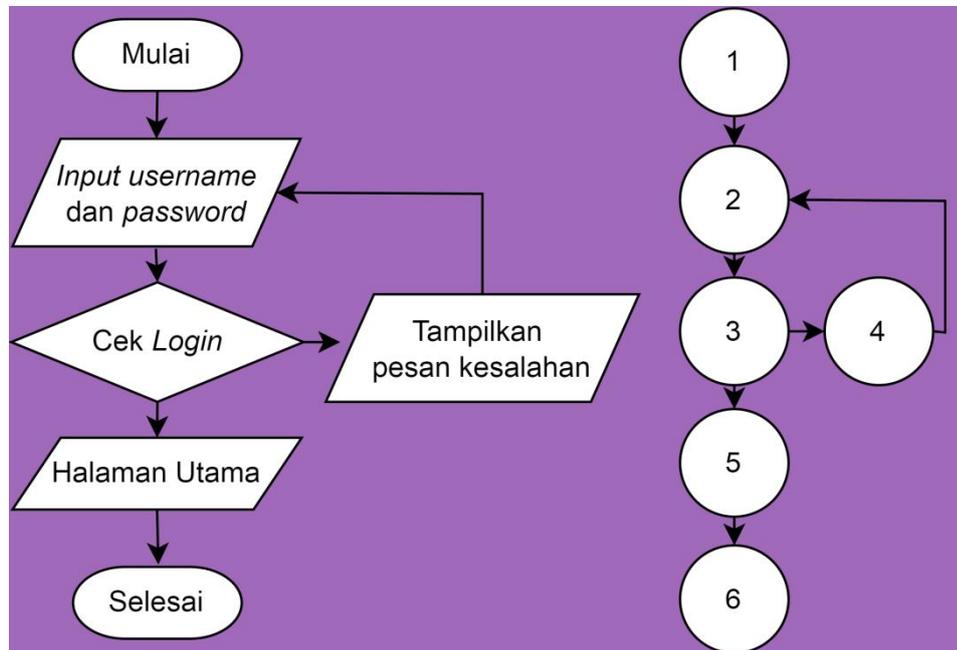
$$\text{Path 3} = 1 - 2 - 3 - 4 - 9$$

$$\text{Path 4} = 1 - 2 - 3 - 5 - 6 - 7 - 5 - 6 - 7 - 8 - 9$$

Tabel 4.18. *White Box* menu aplikasi

	1	2	3	4	5	6	7	8	9	E-1
1		1								1-1=0
2			1							1-1=0
3				1						1-1=0
4		1			1					3-1=2
5						1				1-1=0
6							1			1-1=0
7					1			1		2-1=1
8									1	1-1=0
9										1-1=0
SUM(E+1)										3+1=4

b) *White box* pada menu login



Gambar 4 33. .White box pada menu login..

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6 \quad E = 6 \quad R = 2$$

Penyelesaian :

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

$$= (6 - 6) + 2$$

$$= 2$$

Independent Path :

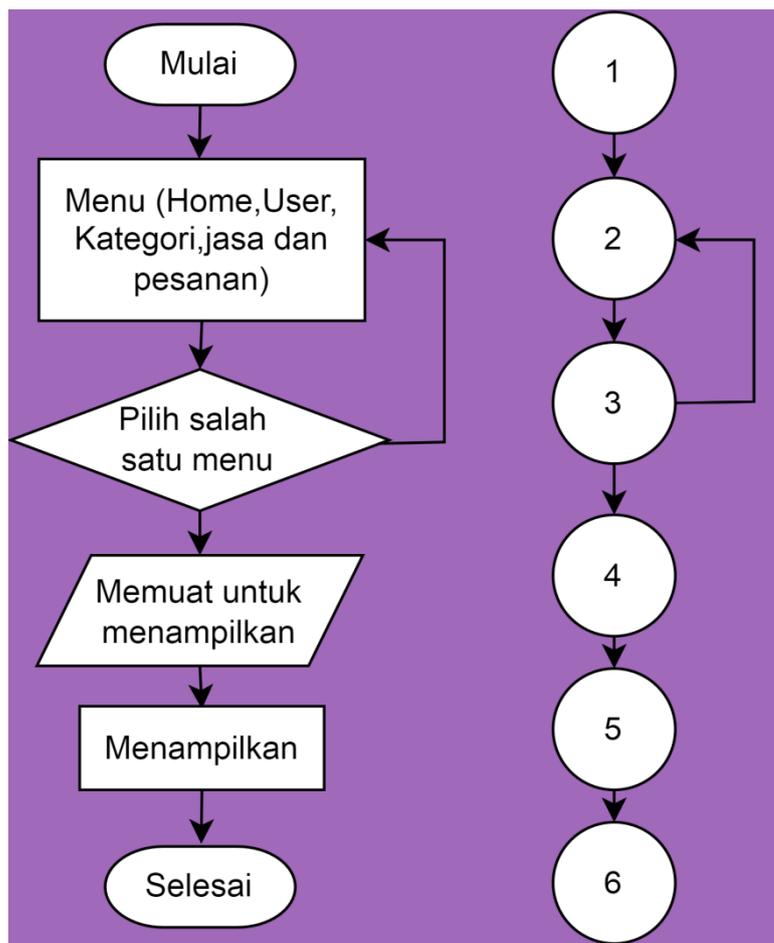
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

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

Tabel 4.19. White Box Perhitungan menu login

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

c) White box pada view menu Admin.



Gambar 4 34. White box pada View menu admin

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan $V(G)$ terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

$$N = 6$$

$$E = 6$$

$$R = 2$$

Penyelesaian :

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

$$= (6 - 6) + 2$$

$$= 2$$

Independent Path :

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

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

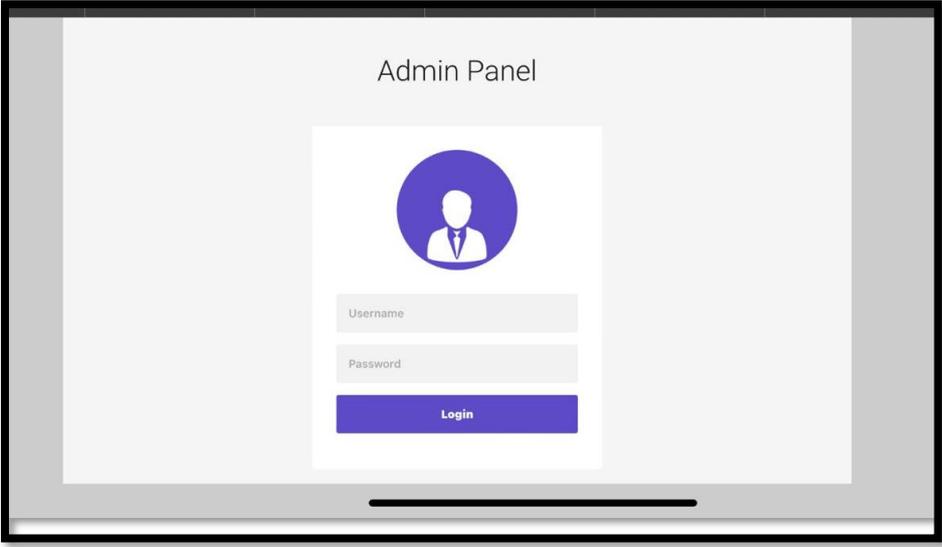
Tabel 4.20. *White Box* Perhitungan menu admin.

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

3.4. Black Box Admin

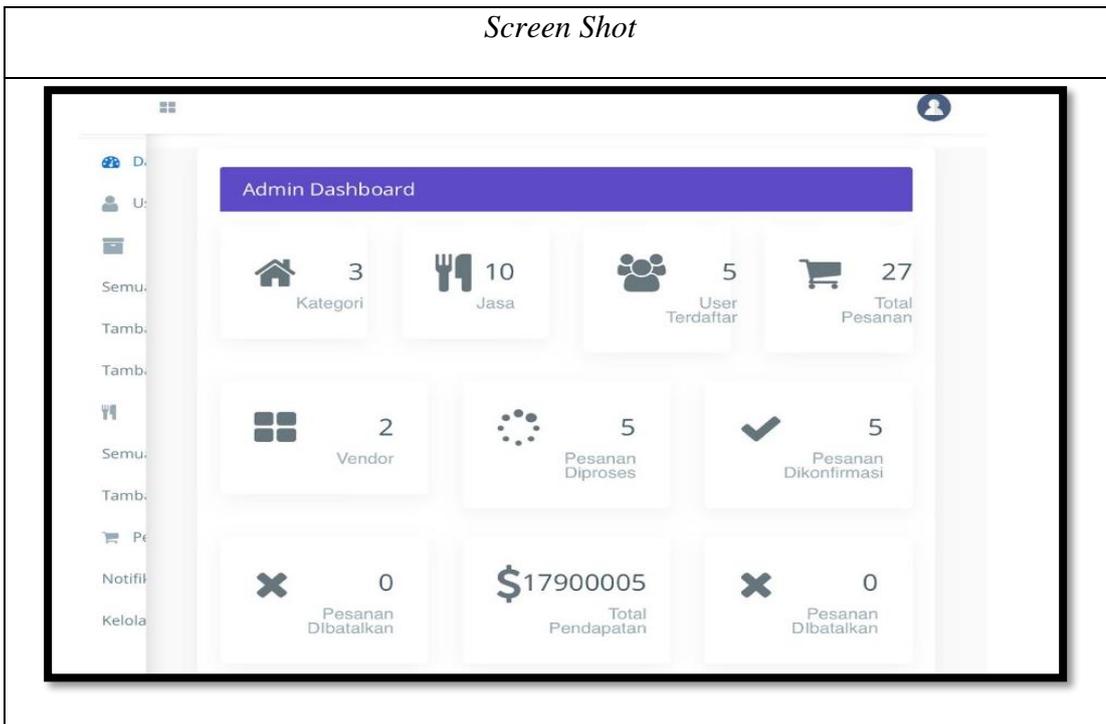
Pengujian *Black Box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau *user*, pengujian ini tidak melihat dan menguji Logika program.

Tabel 4.21. Tabel Pengujian Login admin

Test Faktor	Hasil	Kesimpulan
Login	✓	Sukses, karena ketika <i>admin</i> memasukkan <i>user</i> ID dan <i>Password</i> , akan masuk di halaman Utama
<i>Screen Shot</i>		
		

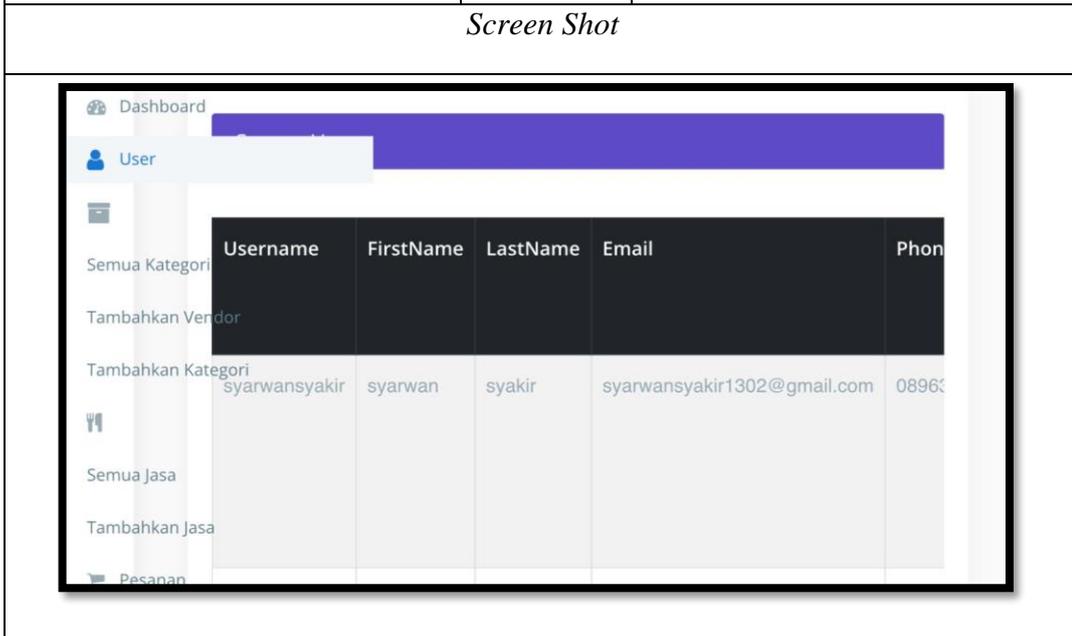
Tabel 4.22. Tabel Pengujian Halaman Utama Admin

Test Faktor	Hasil	Kesimpulan
Halaman Utama	✓	Sukses, karena ketika <i>admin</i> sudah <i>login</i> sebagai user maka langsung masuk ke halaman utama <i>admin</i>

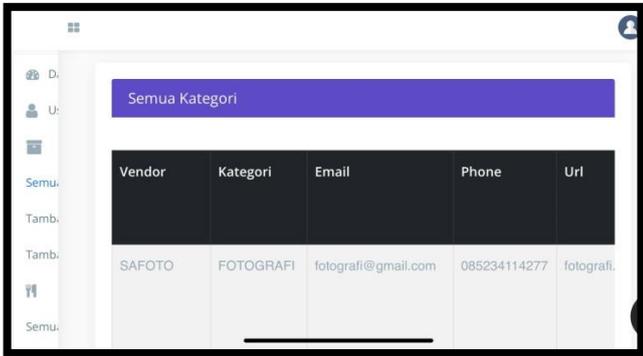


Tabel 4.23. Tabel Pengujian Halaman User

Test Faktor	Hasil	Kesimpulan
Halaman User	✓	Sukses, karena ketika <i>admin</i> sudah <i>login</i> sebagai user maka langsung masuk ke halaman Jasa.



Tabel 4.24. Tabel Pengujian Kategori

Test Faktor	Hasil	Kesimpulan
Profil Kategori	✓	Sukses, karena ketika admin menekan icon Kategori maka langsung menampilkan Kategori pada menu admin.
<i>Screen Shot</i>		
		

Tabel 4.25. Tabel Pengujian Jasa

Test Faktor	Hasil	Kesimpulan
Jasa	✓	Sukses, karena ketika <i>users</i> sudah <i>login</i> sebagai admin dan memilih menu jasa maka langsung masuk ke halaman tersebut.
<i>Screen Shot</i>		
		

Tabel 4.26. Tabel Pengujian Pesanan

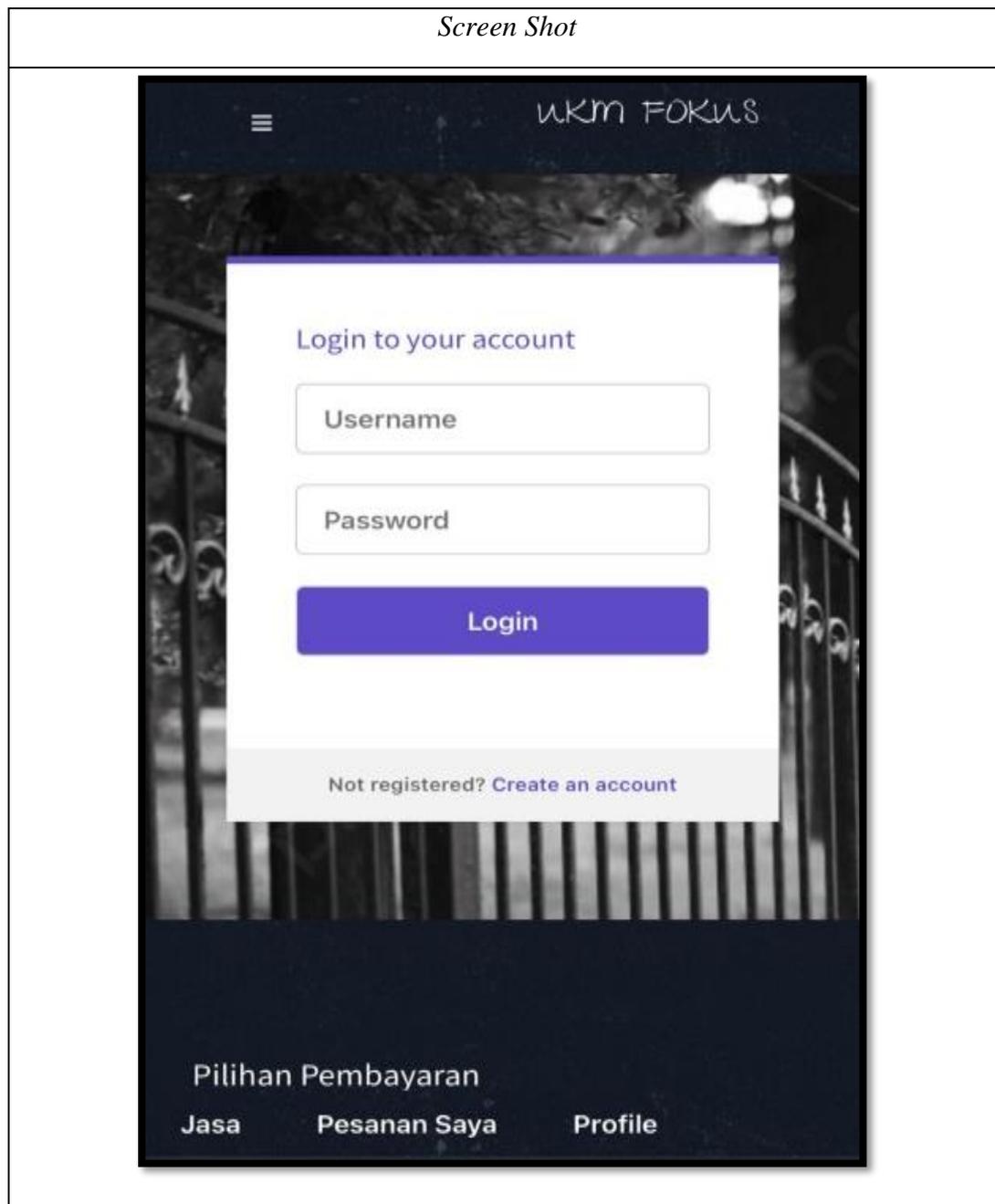
Test Faktor	Hasil	Kesimpulan
Pesanan	✓	Sukses, karena ketika <i>users</i> sudah <i>login</i> sebagai admin dan memilih menu pesanan maka langsung masuk ke halaman tersebut.
<i>Screen Shot</i>		
		

3.3. Black Box User

Pengujian *Black Box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau *user*, pengujian ini tidak melihat dan menguji Logika program.

Tabel 4.27. Tabel Pengujian Login

Test Faktor	Hasil	Kesimpulan
Login	✓	Sukses, karena ketika <i>users</i> memasukkan <i>user</i> ID dan <i>Password</i> , akan masuk di halaman Utama



Tabel 4.28. Tabel Pengujian Halaman Utama User

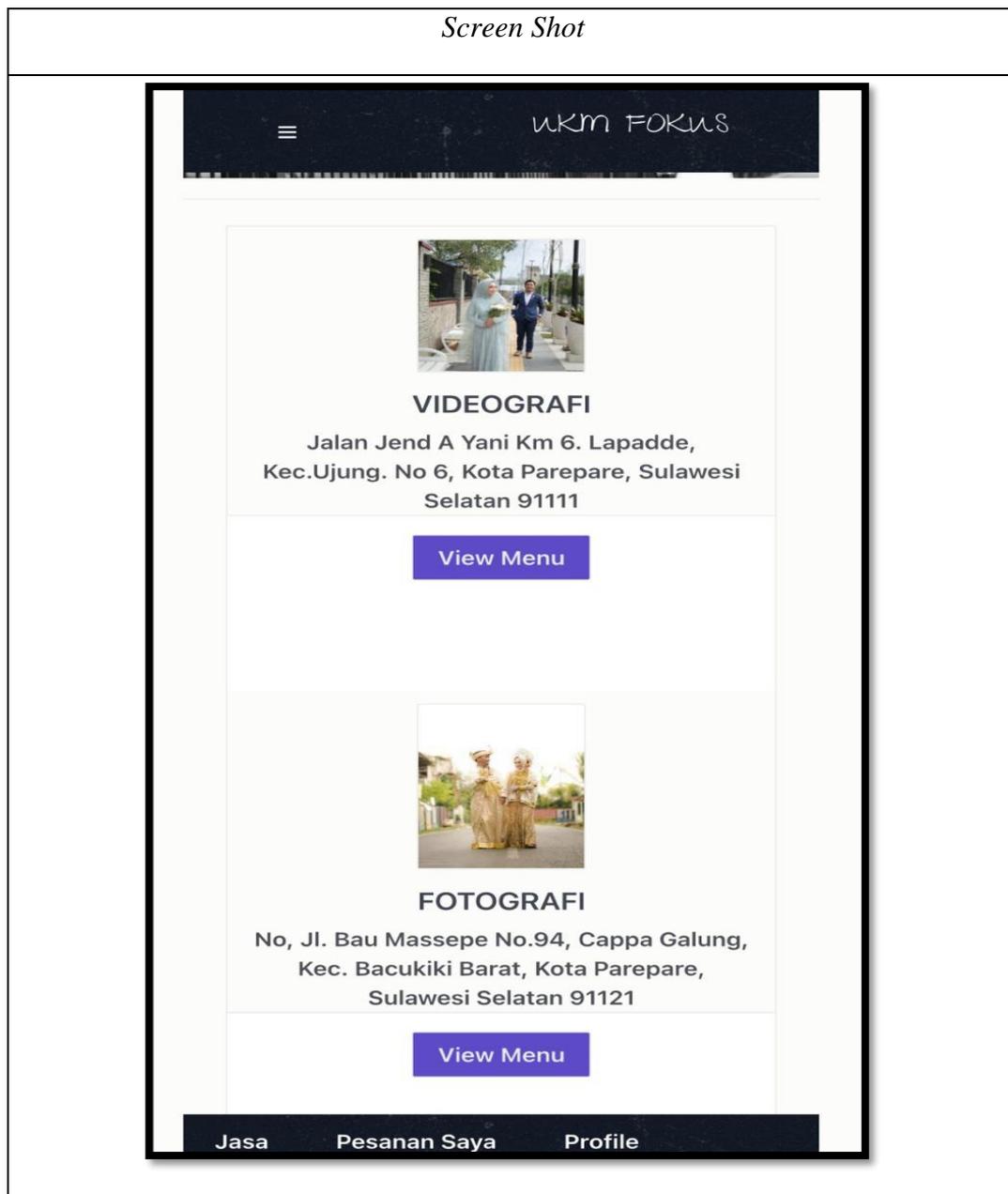
Test Faktor	Hasil	Kesimpulan
Halaman Utama User	✓	Sukses, karena ketika <i>users</i> sudah <i>login</i> sebagai user maka langsung masuk ke halaman utama user.

Screen Shot



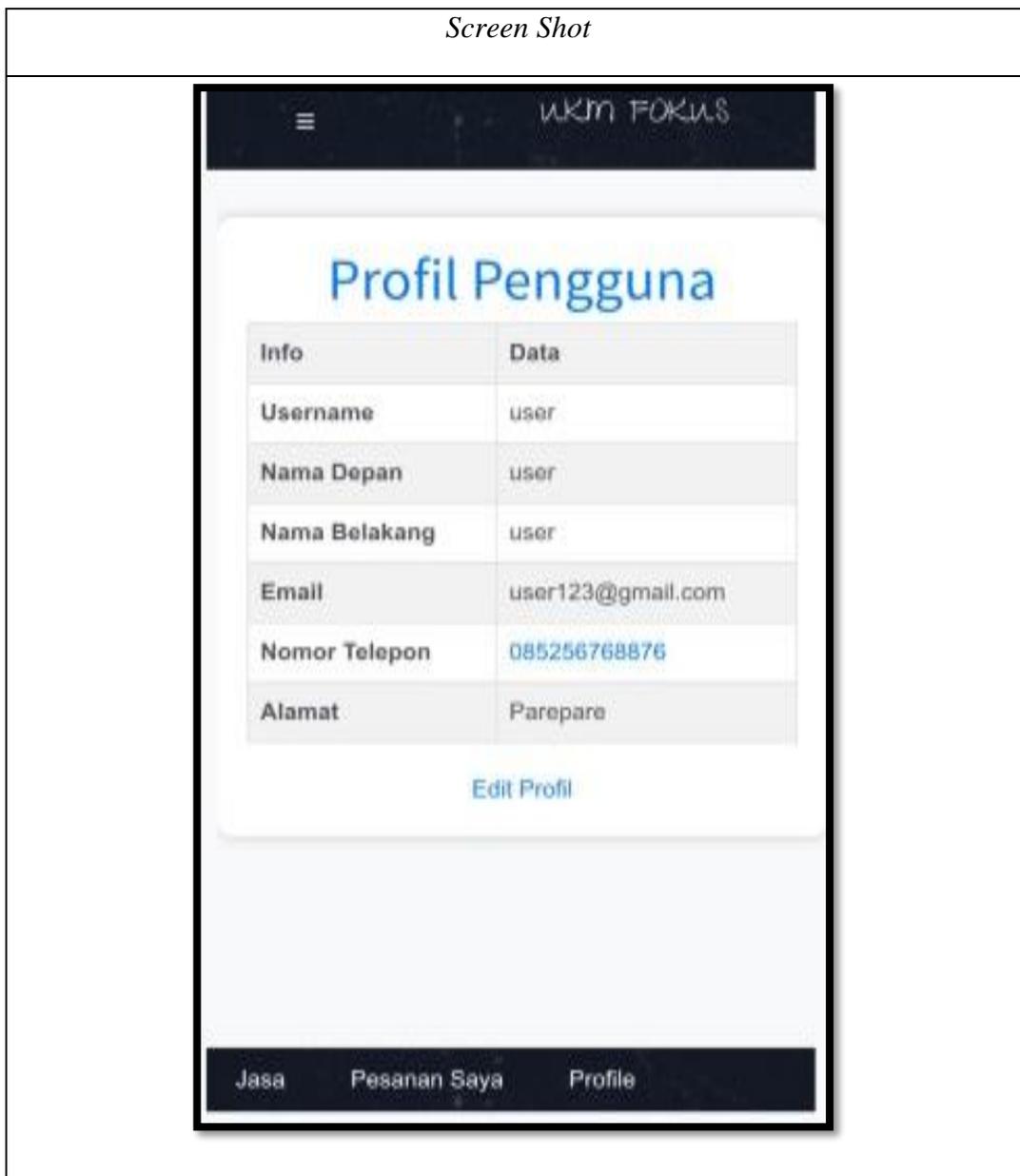
Tabel 4.29. Tabel Pengujian Halaman Jasa User

Test Faktor	Hasil	Kesimpulan
Halaman JasaUser	✓	Sukses, karena ketika <i>users</i> sudah <i>login</i> sebagai user maka langsung masuk ke halaman Jasa.



Tabel 4.30. Tabel Pengujian Profil User

Test Faktor	Hasil	Kesimpulan
Profil User	✓	Sukses, karena ketika users menekan icon profil maka langsung menampilkan profil pada menu user.



Tabel 4.31. Tabel Pengujian Pesanan saya

Test Faktor	Hasil	Kesimpulan
Pesanan saya	✓	Sukses, karena ketika <i>users</i> sudah <i>login</i> sebagai user dan memilih menu pesanan saya maka langsung masuk ke halaman tersebut.

Screen Shot

Jasa	Jumlah	Harga	Status	Tanggal Pesanan	Mapus	Konfirmasi	Tanggal Register	Lokasi Kegiatan
Safoto Basic	1	Rp.3,200,000	Ditunggu	2024-01-25 01:17:22	Batal	Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Advance	1	Rp.8,000,000	Ditunggu	2024-03-07 10:52:53	Batal	Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Asgah	1	Rp.1,500,000	Ditunggu	2024-03-07 12:57:55	Batal	Konfirmasi Pembayaran	2024-03-07 00:00:00	
Safoto Prio	1	Rp.6,500,000	Ditunggu	2024-03-07 12:57:58	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Advance	1	Rp.8,000,000	Ditunggu	2024-03-07 11:59:47	Batal	Konfirmasi Pembayaran	2024-03-12 11:00:00	parepare
Safoto Advance	1	Rp.8,000,000	Ditunggu	2024-03-07 12:13:01	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Advance	1	Rp.8,000,000	Ditunggu	2024-03-07 12:19:21	Batal	Konfirmasi Pembayaran	2024-03-23 12:00:00	lapadde
Safoto Asgah	1	Rp.1,800,000	Ditunggu	2024-03-07 12:24:19	Batal	Konfirmasi Pembayaran	2024-03-20 18:00:00	palen
Safoto Ulang Tahun	1	Rp.1,000,000	Ditunggu	2024-03-07 12:26:48	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Prio Wedding	1	Rp.500,000	Ditunggu	2024-03-07 12:30:10	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	
Safoto Prio Wedding	1	Rp.500,000	Ditunggu	2024-03-07 12:39:58	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	p
Safoto Ulang Tahun	1	Rp.1,000,000	Ditunggu	2024-03-07 12:46:19	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	d
Safoto Advance	1	Rp.9,000,000	Ditunggu	2024-03-07 12:57:38	Batal	Konfirmasi Pembayaran	2024-03-07 00:00:00	mane
Safoto Advance	1	Rp.8,000,000	Ditunggu	2024-03-07 12:58:50	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	i
Safoto Asgah	1	Rp.1,500,000	Ditunggu	2024-03-07 13:01:02	Batal	Konfirmasi Pembayaran	0000-00-00 00:00:00	w
Safoto Ulang Tahun	1	Rp.1,000,000	Ditunggu	2024-03-07 13:03:42	Batal	Konfirmasi Pembayaran	2024-03-23 22:03:32	epege
Safoto Prio Wedding	1	Rp.2,500,000	Diterima	2024-03-07 13:10:30	Batal	Konfirmasi Pembayaran	2024-03-28 12:00:00	Btn Palen
Safoto Advance	1	Rp.8,000,000	Diterima	2024-03-07 13:19:09	Batal	Konfirmasi Pembayaran	2024-04-08 22:18:00	Hutan Jompie

4. 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 implementasi dan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi ini berjalan baik sesuai dengan fungsinya. Aplikasi ini berjalan pada *smartphone* berbasis website yang merupakan aplikasi pemesanan fotografi dan vidiografi agar dapat memudahkan pihak vendor dan pengguna dalam melakukan proses transaksi pemesanan secara online.

B. Saran

Saran yang diberikan sebagai bentuk referensi dari pengembangan sistem yang telah dibangun, yaitu:

1. Untuk penelitian selanjutnya diharapkan dapat mengembangkan lagi sistem yang berbasis android dengan menambahkan fitur-fitur terbaru.
2. Aplikasi ini dapat dikembangkan kembali dengan desain interface yang lebih memudahkan pengguna,
3. Mengembangkan aplikasi untuk dapat memfokuskan pada mengukur tingkat kepuasan pelanggan yang menggunakan aplikasi ini.

DAFTAR PUSTAKA

- Adi Nugroho. 2019 *"Rekayasa perangkat lunak menggunakan UML dan java pustaka belajar"*, Yogyakarta.
- Andi Pratama Putra (2021) dengan judul penelitian "Aplikasi Pemesanan Personal Fotografer Berbasis Perangkat Bergerak"
- Abdul Kadir. 2005 *"Dasar Pemrograman Java 2, Edisi II"*, Penerbit buku Andi, Yogyakarta.
- Dimas Septa Eka Aditya (2019) dalam penelitiannya yang berjudul "Portal Jasa Fotografi Dan Videografi Online Berbasis Web Responsif"
- Nazruddin Safaat H. 2012 *"Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android"*, Penerbit buku Informatika Bandung, Bandung.
- Ir. Yuniar Supardi. 2011 *"Semua bisa menjadi programmer android"*, Penerbit buku Elex Media Komputindo, Jakarta.
- Eko Kurniawan K. 2019 *"Belajar Java Dasar"* .Penerbit buku Strip Bandung, Bandung,.
Tesseract Engine (Online) <https://www.petanikode.com/tutorial/java/>
- Budi Sutedjo Dharma Oetomo, "Perencanaan Dan Pembangunan Sistem InFormasi", Yogyakarta: Andi Offset, 2006.
- Ladjamudin. Al-Bahra, "Analisis dan Desain Sistem InFormasi", 1st ed, Yogyakarta : Graha Ilmu, 2019.
- Yakub, "Pengantar Sistem InFormasi", Yogyakarta : Graha Ilmu, 2012.
- Ozy Prayogi (2018) dalam penelitiannya yang berjudul " sistem inFormasi pemesanan jasa foto dokumentasi Dan video shooting pada ozy photography berbasis Web dan sms gateway"
- Wahana. 2019 *"Android Programming With Eclipse"*, Penerbit buku Andi, Semarang.
- Wahana Komputer. 2019 *"Step by Stp Menjadi Programmer Android"*, Penerbit buku Andi, Semarang.