BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi digital semakin hari semakin pesat. Hanya dengan menggunakan gadget, seseorang di Parepare bisa mengetahui apa yang sedang terjadi di Amerika Serikat pada saat itu juga. Hal ini tidak lepas dari inovasi di media sosial kita. Media sosial membuat kita saling terhubung satu sama lain, sesuatu yang sebenarnya berjarak sangat jauh di ruang fisik tetapi begitu dekat di dunia maya.

Akses informasi yang begitu cepat dapat menjadi peluang apabila dimanfaatkan untuk hal yang positif. Seperti menyebarkan informasi yang berisi aktivitas sosial suatu komunitas yang aktivitasnya ditujukan untuk pengembangan ilmu pengetahuan. Penyebaran informasi ini untuk memudahkan pengguna lain mengetahui berbagai kegiatan yang sedang, akan dan telah berlangsung. Serta memasarkan produk karya dari komunitas tersebut.

Di Parepare, ada satu lembaga yang menaruh perhatian pada aktivitas riset dan buku. Sampan Institute merupakan lembaga diseminasi ilmu pengetahuan yang bergerak dalam bidang penelitian, penerbitan, dan pengarsipan. Aktivitas utamanya terbagi dari 3 poin yaitu riset, penulisan dan arsip. Selama ini, Sampan telah aktif melakukan beberapa penelitian, serta membuat beberapa buku yang sebagian besar di tulis oleh penulis Parepare dan daerah di sekitarnya.

Dalam penginformasian di dunia digital, Sampan Institute telah menggunakan beberapa aplikasi sosial media, seperti Instagram dan facebook. Hal

ini cukup berdampak bagi penyebaran informasi aktivitas dan karya yang telah lembaga ini hasilkan. Untuk memudahkan pengguna gadget android mengetahui serta dapat memesan buku-buku terbitan mereka, perlu ada satu aplikasi yang mencakup keseluruhan informasi tentang Sampan beserta buku yang telah mereka terbitkan, Maka sebagai solusi permasalahan ini yang kemudian dituangkan dalam bentuk tugas akhir yang berjudul **"Rancang Bangun Sistem Informasi Sampan Institute Berbasis Android".**

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang ada, maka penulis mendapatkan rumusan masalah sebagai berikut;

1. Bagaimana rancangan dalam membangun sistem informasi yang terkait dengan Sampan Institute?
2. Bagaimana mengimplementasikan sistem informasi dan penjualan buku terbitan Sampan Institute?

C. Batasan Masalah

Agar pembahasan dan penyusunan Skripsi ini terarah dan mendapatkan gambaran yang jelas serta tidak melenceng dari pokok pembahasan, maka diperlukan batasan-batasan masalah, yaitu;

1. Sistem informasi ini melingkupi profile tentang Sampan Institute, katalog (buku yang diterbitkan lembaga ini), blog (jurnal dan artikel), serta tampilan program (Workshop kelas menulis, klub baca, diskusi buku dan penelitian sejarah kota, terkhususnya Kota Parepare).
2. Aplikasi ini terdapat fitur pemesanan buku terbitan Sampan Institute.

D. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Membangun aplikasi agar dapat mejangkau konsumen dengan mudah dan efisien.
2. Untuk memberikan informasi yang lebih detail terkait Sampan Institute beserta seluruh program kerjanya.

E. Manfaat Penelitian

Adapun Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi dunia akedemik

   Dapat memberikan suatu referensi yang berguna bagi dunia akademis khususnya dalam penelitian yang dilaksanakan oleh para peneliti mendatang dalam hal perkembangan teknologi.

2. Bagi pengguna

   Memberikan informasi mengenai produk terbitan Sampan Institute, serta menciptakan sebuah lingkungan sosial yang membantu masyarakat dalam menemukan, mengembangkan, dan mempromosikan bakat dan minat mereka.

3. Bagi penulis

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

F. Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan proposal ini adalah:

BAB I PENDAHULUAN

Dalam bab ini akan diuraikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan beberapa teori-teori yang mendukung dalam pembahasan penyusunan tugas akhir ini serta bahasa pemrograman yang digunakan sehingga memudahkan penulis dalam menyelesaikan masalah.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan mengenai tahapan - tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu tempat penelitian dan waktu penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang atau dibuat.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Dalam bab ini menjelaskan mengenai rancangan sistem yang telah dibuat, metode dan teknik pengujian.

BAB V KESIMPULAN DAN SARAN

Dalam bab ini menjelaskan mengenai kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Sampan Institute

Sampan Institute adalah lembaga diseminasi ilmu pengetahuan yang bergerak dalam bidang sosial, keagamaan, dan kemanusiaan. Aktivitas utamanya ialah melakukan penelitian, penulisan, penerbitan, pengarsipan, dan pemberdayaan.

Sampan berasal dari kata 'sampan', sebuah perahu tradisional yang digunakan untuk membawa barang atau orang dari satu tempat ke tempat lain dengan tenang dan stabil. Filosofi dari nama ini merujuk pada karakteristik perahu sampan yang selalu stabil, aman, dan dapat menjaga keseimbangan di dalamnya.

Visi dari sampan institute sendiri ialah menjadi ruang dan sarana yang dapat membantu dan mendorong masyarakat meningkatkan kesadaran akan potensi diri, alam, sosial dan budaya untuk diolah sebagai bahan kreativitas dalam menggali dan menumbuhkembangkan makna atau nilai hidup. Adapun misinya, yakni menyediakan berbagai program dan kegiatan untuk meningkatkan kompetensi, pengetahuan, dan kreativitas masyarakat melalui kebiasaan membaca, berdiskusi, menulis, dan berkesenian.

Tujuan utama dari pendirian lembaga ini adalah untuk menciptakan sebuah lingkungan sosial yang membantu masyarakat dalam menemukan, mengembangkan, dan mempromosikan bakat dan minat mereka. Dengan mengadopsi filosofi sampan yang dapat menjaga keseimbangan dan keselamatan di

dalamnya, Sampan yakin masyarakat dapat menghadapi tantangan dan masalah kehidupan dengan lebih tenang dan stabil.

Sampan memiliki prinsip-prinsip dasar yaitu kejujuran, keberanian, kecerdasan, kemandirian, keterbukaan, kesederhanaan, dan kerja sama. Dengan mempraktikkan prinsip-prinsip tersebut, Sampan percaya bahwa masyarakat akan dapat membangun karakter dan kreativitas yang kuat serta menghasilkan kontribusi positif bagi lingkungan sekitar.

Sampan memiliki unit usaha penerbitan bernama Sampan Institute, yang berdiri sejak 6 Januari tahun 2016. Sebelum menjadi yayasan, nama Sampan Institute inilah yang digunakan dalam melakukan segala aktivitas baik di bidang sosial, budaya, pendidikan, dan kesenian. Hingga saat ini Sampan Institute telah menerbitkan 27 judul buku yang terdiri dari karya sastra dan karya ilmiah. Selain usaha penerbitan buku, Akademi Imajinasi Sampan juga menyediakan jasa untuk program pelatihan menulis (Ilmiah, Jurnalistik, dan Sastra), desain komunikasi visual, dan jasa pendokumentasian berupa foto dan video.

Adapun macam-macam kategori buku yang Sampan Institute sendiri telah terbitkan yaitu:

a. Buku Fiksi

Buku fiksi adalah buku yang berisi cerita atau kisah rekaan atau imajinasi penulis, bukan berdasarkan fakta nyata. Ciri-ciri dari Buku fiksi yaitu:

1. Mengandung unsur imajinatif, tidak benar-benar terjadi.
2. Biasanya bertujuan menghibur.

3. Menggunakan gaya bahasa naratif (bercerita).

b. Buku Nonfiksi

Buku nonfiksi adalah buku yang ditulis berdasarkan fakta nyata, data, pengalaman, atau hasil penelitian. Isinya bersifat informatif dan edukatif. Ciri-ciri buku Nonfiksi yaitu:

1. Berdasarkan kenyataan (realitas).
2. Disusun secara sistematis dan logis.
3. Bertujuan memberi informasi atau pengetahuan.

b. Buku Referensi

Buku referensi adalah jenis buku nonfiksi yang digunakan untuk rujukan atau acuan dalam mencari informasi tertentu secara cepat dan ringkas. Buku ini tidak untuk dibaca secara berurutan dari awal hingga akhir, melainkan dibuka sesuai kebutuhan. Ciri-ciri dari buku Referensi yaitu:

1. Memuat informasi spesifik dan ringkas.
2. Tidak untuk dibaca dari awal sampai akhir.
3. Biasanya disusun berdasarkan abjad atau sistem tertentu.

2. Rancang Bangun

"Rancang Bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut". maka dapat disimpulkan bahwa rancang bangun merupakan sebuah proses penggambaran, Menurut Pressman (2012). "Rancang Bangun merupakan kegiatan menerjemahkan

hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada sebelumnya". Ada juga pendapat menurut perencanaan, dan pembuatan sistem baru atau pengembangan dari sistem yang sudah ada sebelumnya (Maulani1 et al., 2018).

3. Sistem Informasi

Sistem Informasi adalah suatu sistem yang menyediakan informasi untuk manajemen pengambilan keputusan/kebijakan dan menjalankan operasional dari kombinasi orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi, atau sistem informasi diartikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen.

Dari pengertian sistem informasi di atas, maka dapat disimpulkan bahwa sistem informasi memiliki hubungan dengan sistem data yang ada pada satu sisi serta sistem aktivitas pada sisi yang lainnya.

Pengertian Sistem Informasi Menurut Para Ahli sebagai berikut :

a. Menurut John F.Nash (1995:8) Sistem informasi adalah kombinasi dari manusia, fasilitas ataupun teknologi, media, prosedur serta pengendalian yang memiliki maksud untuk menata jaringan komunikasi yang penting, proses maupun transaksi tertentu secara rutin, membantu manajemen serta pemakai intern maupun ekstern dan menyediakan dasar dari pengambilan keputusan yang tepat.
b. Menurut Robert A. Leitch (1999:11) Sistem Informasi adalah suatu sistem yang ada di dalam suatu organisasi yang dapat mempertemukan kebutuhan

pengolahan transaksi harian, mendukung operasi yang bersifat manajerial serta kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

4. Pengertian *E-Commerce*

Menurut E. Turban, et al. dalam Rizki, Dkk (2019), *E-commerce* atau *Electronic Commerce* ialah perdagangan elektronik yang mencakup proses pembelian dan penjualan barang atau jasa, pertukaran produk, transfer dana, pelayanan serta informasi yang menggunakan jaringan komputer atau internet. *E-Commerce* juga bisa diartikan sebagai konsep penerapan *E-Business* dengan strategi jual beli barang atau jasa menggunakan jaringan elektronik yang mana melakukan transaksi data secara elektronik, sistem manajemen inventory yang dilakukan secara otomatis dan juga sistem pengumpulan data yang dapat dilakukan secara otomatis.

Salah satu hal yang sedang berkembang dikalangan masyarakat saat ini adalah *E-Commerce* (perdagangan elektronik). Menurut Guay dalam Nursani, et al. (2019), *E-Commerce* diartikan sebagai suatu transaksi ekononi yang dilakukan oleh penjual dan pembeli secara bersama-sama menggunakan media elektronik yang berasal dari internet dengan membuat kontrak perjanjian tentang pengiriman dan harga suatu barang serta menyelesaikan transaksi melalui pembayaran dan pengiriman suatu barang yang sesuai dengan kontrak yang sudah ditetapkan.

Menurut Laudon dan Laudon dalam Nursani, et al. (2019), *E-Commerce* merupakan suatu keadaan dimana proses jual beli produk dilakukan secara elektronik oleh konsumen serta dari perusahaan ke perusahaan dengan

menggunakan computer sebagai perantara dalam transaksi bisnis. *E-commerce* dikelompokkan menjadi tiga, yaitu sebagai berikut :

a. *Business to Consumer* (B2C), merupakan penjualan suatu produk serta layanan dan eceran kepada pembeli secara individu.
b. *Business to Business* (B2B), merupakan penjualan suatu produk dan layanan yang dilakukan antar perusahaan.
c. *Consumer to Consumer* (C2C), merpakan penjualan yang dilakukan secara langsung antara konsumen dengan konsumen.

Berdasarkan penjelasan diatas maka dapat disimpulkan bahwa *E-Commerce* atau perdagangan elektronik merupakan suatu media yang dapat digunakan untuk melakukan transaksi jual beli suatu produk atau jasa secara online atau daring diamanapun dan kapanpun dengan menggunakan jaringan komputer atau internet sehingga dapat memberikan kemudahan bagi para penggunanya dalam bertransaksi.

5. Pengertian Distribusi

Distribusi adalah kegiatan pemasaran yang berusaha memperlancar dan mempermudah penyampaian barang dari produsen sampai ke tangan konsumen, sehingga penggunaannya sesuai dengan jenis, jumlah, harga, tempat, dan waktu yang dibutuhkan.

Distribusi adalah salah satu bagian dari pemasaran. Distribusi juga dapat diartikan kegiatan pemasaran yang berusaha memperlancar dan mempermudah penyampaian barang dan jasa sehingga penggunaannya sesuai dengan yang diperlukan.

Distribusi adalah kegiatan yang sangat penting agar produk dari produsen bisa sampai ke tangan konsumen dengan efektif. Walaupun letak suatu pabrik/produsen sangat jauh dengan keberadaan masyarakat, dengan adanya kegiatan distribusi maka sangat mempermudah masyarakat mendapatkan produk/barang yang diinginkan (Putri, Rosmayani, & Rosmita, 2018).

Saluran Distribusi merupakan perantara yang turut serta dalam proses pemindahan barang dari produsen ke konsumen (Mursid, 2015).

Distribusi merupakan suatu organisasi yang membuat proses kegiatan penyaluran barang atau jasa untuk dipakai atau konsumsi oleh para konsumen. Istilah distribusi menurut Zylstra (2006) adalah suatu sistem yang menunjukkan segala sesuatu/sumber daya-sumber daya organisasi yang disimpan dalam antisipasinya disebut dengan istilah distribusi. Tetapi kita seharusnya tidak membatasi pengertian distribusi tidak hanya itu saja. Banyak organisasi perusahan menyimpan jenis-jenis distribusi lain seperti: uang, ruang fisik buka tutup, bangunan pabrik, peralatan dan tenaga kerja untuk memenuhi permintaan produk dan jasa (Karundeng, Mandey, & Sumarauw, 2018).

Menurut Sigit (dalam Danang, 2015: 180) saluran distribusi adalah perantaraperantara, para pembeli dan penjual yang dilalui oleh perpindahan barang baik fisik maupun perpindahan milik sejak dari produsen hingga ke tangan konsumen. Suatu perusahaan di dalam mendistribusikan barangnya dapat menggunakan salah satu atau lebih kegiatan penyaluran. Bagi perusahaan yang baru saja berdiri memilih saluran distribusi merupakan pekerjaan yang tidak mudah, karena baru masuk pasar (Putri, Rosmayani, & Rosmita, 2018). Sedangkan bagi

perusahaan yang sudah berjalan dengan saluran-saluran distribusi tertentu, masih harus tetap waspada dan mengawasi aparat-aparat distribusinya. Dalam kegiatan distribusi banyak aspek di dalamnya sehingga pendistribusian bias berjalan dengan efektif, salah satunya adalah distributor. Distributor merupakan subjek yang melakukan kegiatan distribusi. Tanpa adanya distributor tidak dapat ada yang menjembatani produsen dan pembeli.

6. Teori Program Aplikasi

a. Android

Pengertian Android adalah sistem operasi berbasis *Linux* yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, *smartphone* dan juga PC tablet. Secara umum Android adalah *platform* yang terbuka *(Open Source)* bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh berbagai piranti bergerak. Telepon pertama yang memakai sistem operasi Android adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Semenjak kehadirannya pada 9 Maret 2009, Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada aplikasinya, seperti jam alarm, *voice search*, pengiriman pesan dengan Gmail, dan pemberitahuan email. Pada perkembangannya, sistem operasi Android telah mengalami beberapa perubahan dan perbaikan. Dan yang paling menarik adalah versi keluaran Android yang diberi nama seperti nama-nama makanan. Menurut Teguh Arifianto (2011 : 1), android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang

berbasis *linux*. Menurut Hermawan (2011 : 1), Android merupakan *OS (Operating System) Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile, i-Phone OS, Symbian,* dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat 7 potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka. Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa android adalah sistem operasi berbasis *linux* yang sedang berkembang ditengah OS lainnya. Nazrudin safaat H (2012:1) menyatakan bahwa android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencangkup sistem opreasi, *middleware* dan aplikasi. Android menyiapkan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Nazrudin safaat H (2012:3) menyatakan bahwa Android dipuji sebagai "*platform mobile* pertama yang lengkap, terbuka, dan bebas".

b. Android Studio

Android studio adalah *IDE (Integrated Development Environment)* resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio mengantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools).

Android studio memiliki fitur :

a) Projek berbasis pada *Gradle Build*.

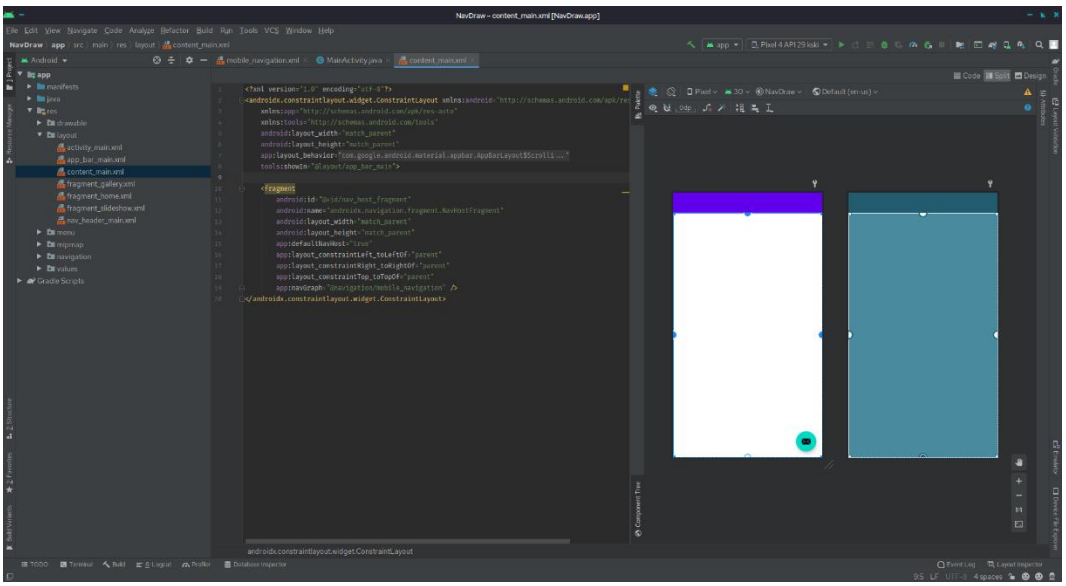
b) *Refactory* dan pembenahan bug yang cepat.

c) *Tools* baru yang bernama *"Lint"* dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.

d) Mendukung *Proguard And App-signing* untuk keamanan.

e) Memiliki GUI aplikasi android lebih mudah.

f) dukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.



Gambar 2. 1 *Android Studio*

c. *Java*

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama "Oak", namun pada tahun 1995 diganti namanya menjadi *"Java"*.



Gambar 2. 2 Logo Java

Adapun pengertian *Java* menurut para ahli yaitu :

Menurut (Rosa & Shalahuddin, 2010) *Java* adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan. *Java* berdiri di atas sebuah mesin penterjemah (interpreter) yang diberi nama *Java* Virtual Machine (JVM). JVM inilah yang akan membaca kode bit (bytecode) dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa *Java* disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa *Java* adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat

elektronik, sehingga *Java* harus bersifat tidak bergantung pada platform (platform independent). Itulah yang menyebabkan dalam dunia pemrograman *Java* dikenal adanya istilah *"write once, run everywhere"*, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (platform) manapun, tanpa harus melakukan perubahan kode program.

d. *PHP*

PHP adalah singkatan dari "*PHP*: *Hypertext Preprocessor*" dan merupakan bahasa skrip yang diintegrasikan ke dalam *HTML*. Beberapa sintaksnya mirip dengan *C*, *Java*, *Asc*, dan *Perl*, di samping beberapa fitur khusus *PHP*. Tujuan utama dari bahasa ini adalah untuk memungkinkan para perancang web menulis halaman web dinamis dengan cepat.

e. *MySQL*

Beberapa pengertian *MySQL* menurut para ahli:

1. Menurut Raharjo (2011:21), "*MySQL* merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user".
2. Menurut Kadir (2008:2), "*MySQL* adalah sebuah *software open source* yang digunakan untuk membuat sebuah database." Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu software atau program yang digunakan untuk membuat sebuah database yang bersifat *open source*.
3. Menurut Didik Dwi Prasetyo (2004) : *MySQL* mempunyai beberapa keunggulan diantaranya :

a) Cepat, handal dan mudah dalam penggunaannya, *MySQL* lebih cepat tiga sampai empat kali daripada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL.*

b) Didukung oleh berbagai Bahasa *database server, MySQL* dapat memberikan pesan eror dalam berbagai Bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.

c) Mampu membuat table berukuran sangat besar.

d) Ukuran maksimal dari setiap table yang dapat dibuat dengan *MySQL* adalah 4GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.

e) Lebih murah, *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX platform, OS/2* dan *Windows Platform.*

f) Melekatnya integrasi *PHP* dengan *MySQL.*

g) Keterikatan antara *PHP* dengan *MySQL* yang sama-sama *software open-source* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya.

f. UML (Unified Modelling Language)

Pemodelan adalah proses merancang piranti lunak sebelum melakukan pengkodean. Model piranti dapat dianalogikan seperti membuat cetak biru pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan

teknik pemodelan yang baik.

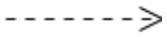
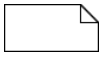
Pada teknik perancangan berdasarkan objek oriented, hasil pemodelan konseptual digunakan untuk menggambarkan fungsionalitas sesuai persyaratan yang ditentukan pelanggan. Pemodelan konseptual itu diwujudkan dalam bentuk sebuah standar untuk merancang model sebuah sistem yang bernama *Unified Modelling Language (UML)*. Menurut Yuni Sugiarti (2013:34), *UML* adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. Dalam *UML* sendiri terdapat beberapa diagram yaitu :

a. Use Case Diagram

Use case atau *diagram use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa & Salahuddin, 2011:130).

Tabel 2. 1 *Use Case Diagram*

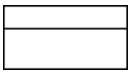
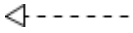
NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yangsatu adalah fungsi yang lebih umum.

NO	GAMBAR	NAMA	KETERANGAN
4		*Include*	Relasi use case tambahan ke sebuah use case dimanause case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya.
5		*Extend*	Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri.
6		*Association*	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
7		*System*	Menspesifikasikan paket yang menampilkan sistemsecara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkansistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaborati on*	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan danmencerminkan suatu sumber daya komputasi.

b. Class Diagram

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada *UML* yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi.

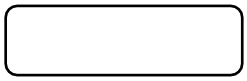
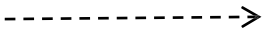
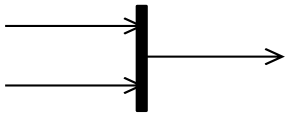
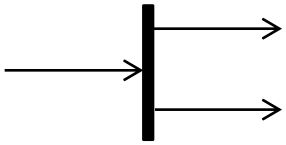
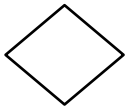
Tabel 2. 2 Simbol *Class Diagram*

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-banar dilakukan oleh suatu *object.*
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.

c. Activity Diagram

Activity Diagram Adalah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir.

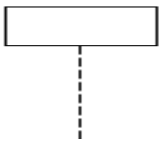
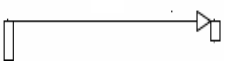
Tabel 2. 3 Simbol *Activity Diagram*

NO	BENTUK SIMBOL	NAMA SIMBOL	FUNGSI SIMBOL
1.		*Activity*	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		*Control Flow*	Menunjukan Urutan Eksekusi.
3.		*Object Flow*	Menunjukan aliran objek dari sebuah *action* atau *activity* ke *action*.
4.		*Start Point*	Menyatakan bahwa sebuah objek dibentuk atau diawali.
5.		*End Point*	Menyatakan bahwa sebuah objek dibentuk atau diakhiri.
6.		*Join*/Penggabungan	Menyatakan untuk menggabungkan kembali *activity* atau *action* yang *parallel*.
7.		*Fork*	Menyatakan untuk memecah *behavio*r menjadi *activity* atau *action*. yang *parallel*
8.		*Decision*	Menunjukan penggambaran suatu keputusan/tindakan yang harus di ambil pada kondisi tertentu.

d. Sequence Diagram

Sequence Diagram adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

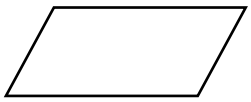
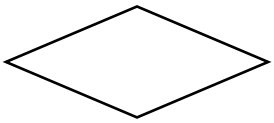
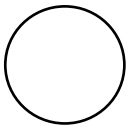
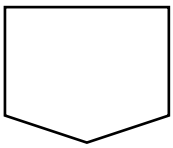
Tabel 2. 4 Simbol *Sequence Diagram*

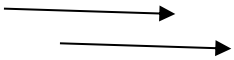
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi .
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi

g. FlowChart

Menurut (Arifianto) "*Flowchart* adalah suatu bagan dengann simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program".

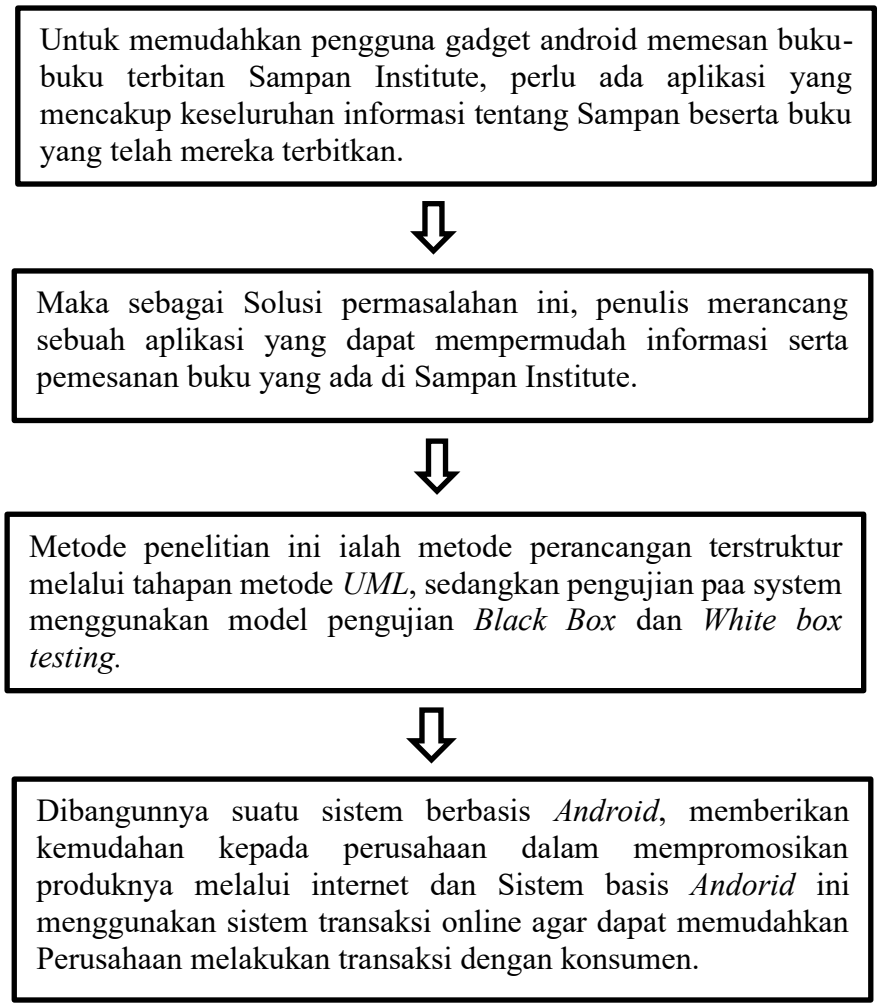
Tabel 2. 5 Simbol *Flowchart*

NO	Simbol	Nama	Fungsi
1		Terminal	Menyatakan permulaan atau akhir suatu program
2		*Input/Output*	Menyatakan proses input atau output tanpa tergantung jenis peralatanya
3		*Process*	Menyatakan suatu Tindakan (Proses) yang dilakukan oleh komputer
4		*Decision*	Menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemunkinan jawaban:ya / tidak
5		*Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang sama
6		*Offline Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang berbeda
7		*Prefefned Process*	Menyatakan penyediaan tempat penyimpanan suatu pengolaan untuk memberi harga awal
8		*Punched card*	Menyatakan input berasal dari kartu atau output ditulis ke kartu

NO	Simbol	Nama	Fungsi
9		*Document*	Mencetak keluaran dalam bentuk dokumen(Melalui printer)
10		*Flow*	Menyatakan jalannya arus suatu proses

B. Kerangka Berfikir

Untuk lebih memahami alur penelitian di atas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram berikut ini :



Gambar 2. 3 Kerangka Bergikir

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Metode penelitian yang penulis gunakan dalam penelitian ini adalah menggunakan *kualitatif* dengan teknik pengumpulan data secara observasi, wawancara, dan studi pustaka. Penulis melakukan observasi untuk mendapatkan data yang cukup dengan mengumpulkan data yang terkait dengan judul penelitian ini, sehingga tidak keluar dari pokok permasalahan.

B. Waktu dan Tempat Penelitian

1. Lokasi penelitian berada kantor Sampan Institute, tepatnya di Kota Parepare, Sulawesi Selatan, Jl. Rajawali, No. 23 Perumnas Wekke'E Blok F, Kel. Lompoe, Kec. Bacukiki.
2. Waktu yang dipergunakan untuk pelaksanaan penelitian adalah (±) 2 bulan.

C. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan dalam pembuatan aplikasi ini.

1. Alat
   a. *Hardware* (Perangkat keras)

      Perangkat keras yang digunakan untuk membuat aplikasi dapat dilihat pada sebagai berikut :

      Laptop *Zyrex* dengan spesifkasi:

1) *Processor* : *Intel(R) Celeron(R) CPU N3350 @ 1.10GHz 1.10 GHz*
2) *RAM* : *4GB*
3) *SSD* : *M.2 256 GB*
4) *Model* : *Notebook*

b. *Software* (Perangkat lunak)

Perangkat lunak yang digunakan untuk membuat aplikasi dapat dilihat pada sebagai berikut :

1. *Windows 10 Home Single Language*
2. *Android Studio*

2. Bahan Penelitian

Bahan yang digunakan dalam penelitian ini 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. Tahapan Penelitian

1. Persiapan Penelitian

Tahap persiapan adalah tahap yang dilakukan sebelum melakukan penelitian. Pada tahapan ini dimulai dengan mengkaji permasalahan yang telah ada kemudian melakukan studi literatur mengenai permasalahan yang sedang diteliti.

2. Studi Literatur

Pada tahapan ini ada empat langkah yang harus dipenuhi guna untuk mencapai hasil maksimal dalam penelitian yaitu tahapan pengumpulan data dengan menggunakan beberapa metode pengumpulan data yang telah dijelaskan pada poin

di atas, pengolahan data, analisis data dan selanjutnya penafsiran hasil analisis.

Setelah kegiatan tersebut, proses selanjutnya yaitu dengan melakukan tugas lapangan dalam rangka mengumpulkan data, untuk kemudian akan diproses. Proses yang dimaksud meliputi penyuntingan, penerapan masalah dalam aplikasi program, serta analisis sebagai penarikan kesimpulan hasil akhir.

1. Pengumpulan Data

Pada tahap ini peneliti melakukan pencarian data-data di berbagai sumber untuk dikumpulkan lalu dikaji lebih lanjut.

2. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap permasalahan yang diteliti kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

3. Perancangan

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

4. Pengujian

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

5. Implementasi

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

6. Tahap Penyelesaian

Tahap penyelesaian merupakan tahap akhir yang dilakukan pada penelitian yang dilakukan. Pada tahap ini penyusunan laporan penelitian disusun.

E. Teknik Pengumpulan Data

Pada kegiatan penelitian ini, penulis menggunakan metode yang dijadikan sebagai cara pengumpulan data yang dibutuhkan yaitu :

1. Observasi

Metode Observasi adalah suatu metode pengumpulan data yang dilakukan dengan mengamati langsung, melihat, dan mengambil suatu data yang dibutuhkan dengan teliti dan *sistematis* guna mendapatkan informasi yang akurat.

2. Wawancara (*interview*)

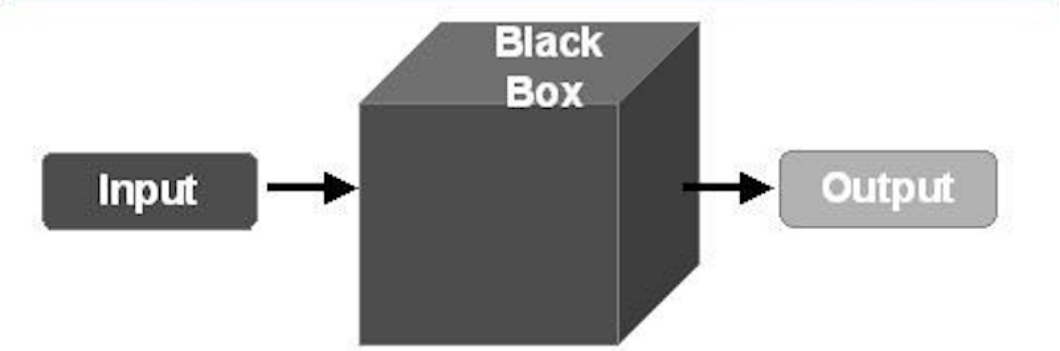
Wawancara adalah percakapan dengan maksud tertentu. Percakapan itu dilakukan oleh dua pihak, yaitu pewawancara (*interview*) yang mengajukan pertanyaan dan terwawancara (*interview*) yang memberikan jawaban atas pertanyaan itu. Ciri utama wawancara adalah kontak langsung dengan tatap muka antara pencari informasi dan sumber informasi. Dalam wawancara sudah disiapkan berbagai macam pertanyaanpertanyaan tetapi muncul berbagai pertanyaan lain saat meneliti.

F. Metode Pengujian

a. Black Box Testing

Pengujian *Black box* adalah jenis pengujian perangkat lunak di mana fungsionalitas aplikasi diuji daripada operasi atau struktur internalnya. Biasanya tidak perlu memiliki pemahaman khusus tentang kode aplikasi atau struktur internal atau pengetahuan pemrograman. Untuk menghasilkan kasus tes, ia menggunakan

deskripsi eksternal dari perangkat lunak yang mencakup persyaratan, desain, dan spesifikasi.



Gambar 3. 1 *Black Box Testing*

Meskipun mereka biasanya fungsional, tes ini juga bisa non-fungsional. Perancang tes memilih input yang dapat diterima dan tidak sesuai dan memastikan hasil yang tepat. Semua tahap pengujian perangkat lunak unit, penerimaan, fungsional, sistem, dan integrasi dapat menggunakan pendekatan tes. Ini juga dapat mendominasi tes unit, tetapi biasanya terdiri dari sebagian besar, jika tidak semua, tes di tingkat atas. Adapun tahapan-tahapan yang dilakukan penulis adalah sebagai berikut:

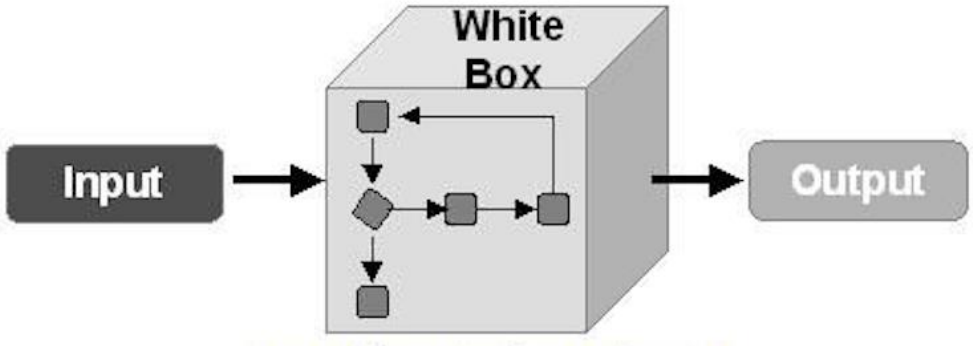
1) Analisis Sistem dilakukan terhadap kebutuhan dari Toko Jaya ABC. Analisa proses bisnis yang terjadi dilakukan dengan cara pengamatan secara langsung mulai dari proses pemesanan, pencatatan stok, pencatatan pesanan hingga proses penjualan dan pengiriman kepada pelanggan. Selain itu juga melakukan wawancara owner dan kasir mengenai kebutuhan apa saja yang akan ditampilkan didalam aplikasi.

2) *Design* Pada tahap ini dilakukan perancangan *user interface* yang dibutuhkan penguna guna memastikan bahwa aplikasi yang akan dibuat dapat diterima dan nyaman digunakan oleh pengguna.
3) Pengodean Perangkat Lunak Pada tahap ini akan dilakukan pembuatan kode program sesuai dengan design dan *planning* yang sebelumnya telah dilaksanakan, serta proses *refactoring*, yaitu proses perubahan *sourcecode* tanpa mengubah hasil yang dikeluarkan untuk meningkatkan *efficiency, readability,* atau *performance* dari *source code* tersebut.
4) Pengujian dan Implementasi Program 76 Pada tahap ini dilakukan testing terhadap *code* yang dihasilkan pada tahap *coding* yang bertujuan untuk memastikan semua fitur yang telah direncanakan pada tahap *planning* telah berjalan sesuai harapan. Pengujian aplikasi *Point of Sale* dilakukan dengan dengan metode *Black-box*. Metode pengujian *Black-box* memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Pengujian dilakukan dengan memilih sejumlah modul dengan berbagai tipe data untuk memastikan bahawa programini hanya menerima input dengan tipe data yang benar. Serta tampilan interface aplikasi itu sendiri.

b. White Box Testing

White box testing adalah pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. (Cholifah, Yulianingsih, & Sagita, 2018), dan pengujian yang

didasarkan pada pengujian design program secara prosedural, secara strctural, pengujian berbasis logika atau pengujian berbasis kode (Irawan, 2017).



Gambar 3. 2 *White Box Testing*

Metode jalur dasar adalah salah satu metode *white box testing*, di mana dalam proses pengujian diperlukan untuk membuat *flowgraph* dari program skrip dan juga menentukan nilai kompleksitas siklomatik. Tes ini bertujuan untuk menganalisis kebenaran struktur program yang dibuat dan kinerja program (Rahayuda & Santiari, 2017).

Berikut ini adalah teknik yang dapat digunakan untuk melakukan white box testing pada perangkat lunak:

1. *Basis path testing*

   Teknik pertama adalah *Basis Path Testing*. Teknik bertujuan untuk mengukur kompleksitas kode program dan mendefinisikan alur yang dieksekusi.

2. *Branch coverage*

   Kemudian ada *Branch Coverage*. Pengujian ini dirancang agar setiap branch code diuji setidaknya satu kali.

3. *Condition coverage*

Selanjutnya adalah teknik condition coverage, tujuannya untuk menguji seluruh kode agar menghasilkan nilai TRUE atau FALSE. Dengan begitu, tester dapat memastikan perangkat lunak dapat bekerja dan mengeluarkan output sesuai dengan input dari pengguna.

4. *Loop testing*

Teknik *Loop Testing* yaitu Pengujian ini yang wajib dilakukan untuk menguji berbagai perulangan/looping yang ada dalam program. seperti do-while, for, dan while. Dalam pengujian ini kamu juga dapat memeriksa kondisi dari perulangan, apakah sudah berjalan dengan benar atau tidak.

5. *Multiple condition coverage*

Berikutnya adalah *Multiple Condition Coverage*. Teknik ini dilakukan untuk menguji seluruh kombinasi dari kode yang mungkin digunakan dalam berbagai kondisi. Seluruh kombinasi harus diuji minimal satu kali, tujuannya untuk memastikan perangkat lunak agar berjalan dengan baik.

6. *Statement coverage*

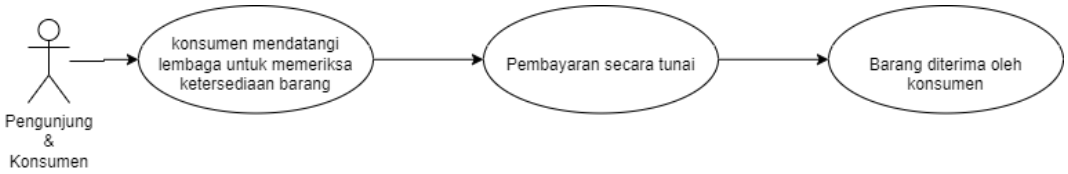
Teknik terakhir adalah *Statement Coverage*. Teknik ini dilakukan minimal satu kali untuk menguji setiap statement dalam perangkat lunak. Dengan pengujian ini, kamu dapat mengetahui kode-kode yang error sehingga dapat segera memperbaikinya.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Perancangan Sistem

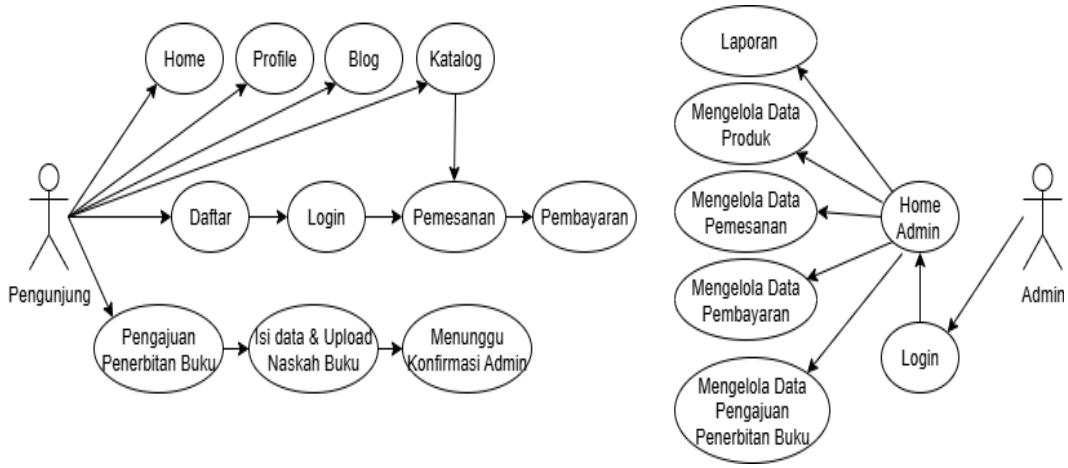
1. Desain Sistem Berjalan



Gambar 4. 1 Desain Sistem Berjalan

Pada gambar 4.1 menjelaskan mengenai alur kegiatan pemesanan produk Pada Sampan Institute, yaitu masih menggunakan cara manual, dimana konsumen secara langsung datang ke Kantor untuk memesan dan memeriksa ketersediaan produk.

2. Desain Sistem yang di usulkan



Gambar 4. 2 Desain Sistem di Usulkan

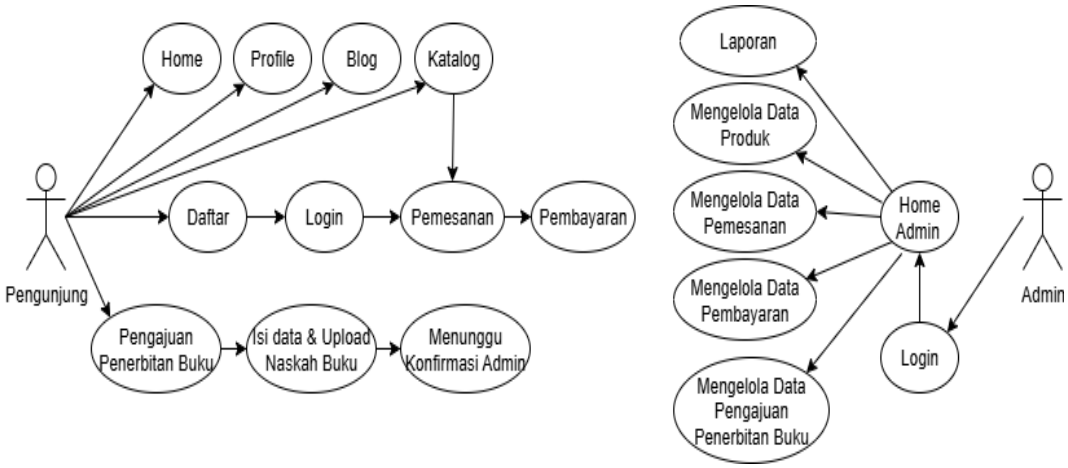
Prosedur Sistem di Usulkan

a. Admin melakukan akses login ke halaman *Home administrator*, dengan memasukkan *Username* dan *password* di form login admin.
b. Pengunjung mengakses semua yang ada pada halaman aplikasi. (Login, *home,* profil tentang Sampan Institute, produk/buku terbitan Sampan, blog yang berisi kegiatan-kegiatan Sampan, pemesanan, dan Pengejuan Penerbitan Buku).
c. Login mengakses aplikasi Sistem Informasi Sampan Institute.
d. Laporan, admin dapat melihat laporan diagram penjualan dan pendapatan perusahaan.

3. Use Case Diagram

Dalam analisis sistem ini, penulis menggunakan *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram.*

Interaksi yang terjadi antara aktor dan proses sistem yang dikembangkan juga dapat diwakili dengan menggunakan kasus penggunaan. Sebuah skenario, atau fase yang menggambarkan aliran peristiwa antara pengguna dan sistem, juga dimasukkan dalam desain kasus penggunaan. Berikut ini adalah deskripsi dari diagram kasus penggunaan untuk Sistem Informasi Sampan Institute dalam penelitian ini:



Gambar 4. 3 *Use Case Diagram*

4. 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	Pengunjung	Pengunjung merupakan aktor yang berperan sebagai pengguna yang dapat melakukan pembelian di sistem informasi sampan institute
2	Admin	Admin dapat mengelola data pengguna, yaitu pelanggan, dan melakukan tugas manajemen data termasuk menghapus, memperbarui, dan menambahkan data pelanggan, selain men-download konten laporan barang yang tiba dan laporan lainnya, Admin adalah aktor dengan lebih banyak hak akses daripada pelanggan.

Sedangkan untuk penjelasan definisi setiap *Use Case* sebagai berikut:

Tabel 4. 2 Definisi *Use Case System*

No	*Use Case*	Keterangan
1	Register	Menu register untuk mendaftar sebagai pelanggan Sampan Institute.
2	Beli Produk	Setelah memilih barang-barang yang ingin mereka beli, pengunjung dapat menambahkan mereka ke keranjang belanja mereka dan melanjutkan ke *checkout*.
3	Pembayaran	Pengunjung dapat membayar barang yang telah mereka pesan dan mengisi formulir konfirmasi pembayaran kemudian melakukan pembayaran. Setelah itu, mereka akan dibawa ke riwayat pembelian jika pembayaran berhasil.
4	Riwayat Pembelian	Menampilkan detail pembelian.
5	*Login*	Pada menu *Login*, Admin dan Pengunjung diminta untuk memasukkan *Username* dan *Password* untuk dapat mengakses halaman tertentu.
6	Mengelola Produk	Admin dapat mengelola produk yang ingin di jual pada *Web*.
7	Mengelola Pesanan Pelanggan	Admin mengelola data pemesanan pengunjung yang sudah membayar dan melakukan pengiriman produk.
8	Mengelola Laporan Penjualan	Admin dapat melihat laporan penjualan berdasarkan waktu pembelian, dan setelahnya laporan dapat diolah kembali, serta laporan dapat dicetak.
9	Daftar Pengunjung	Manampilkan daftar Pengunjung.

Dalam use case yang telah didefinisikan di atas, terdapat beberapa use case yang memiliki alur proses yang 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 dan Pengunjung

Deskripsi : Sebuah kegiatan untuk masuk ke halaman Admin dan Pelanggan

Pre-condition : -

Post-condition : - Masuk kehalaman Dashboard masing-masing.

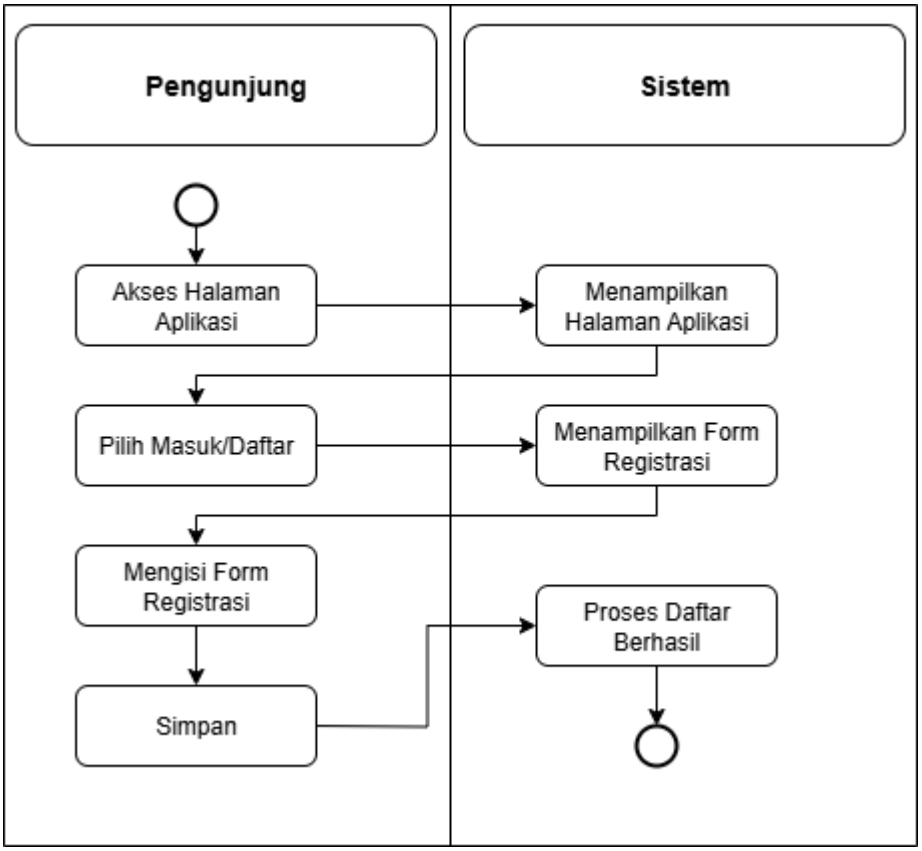
Tabel 4. 3 *Skenario Use Case Login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengisi form Username dan Password.	
2. Menekan tombol Login.	3. Mengecek validitas.
	4. Jika data yang dimasukkan valid, maka akan masuk ke halaman awal".
Jika data yang di masukkan tidak valid, maka akan muncul pesan "Gagal Login".	

5. *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 sistem.

a) Activity Diagram Login

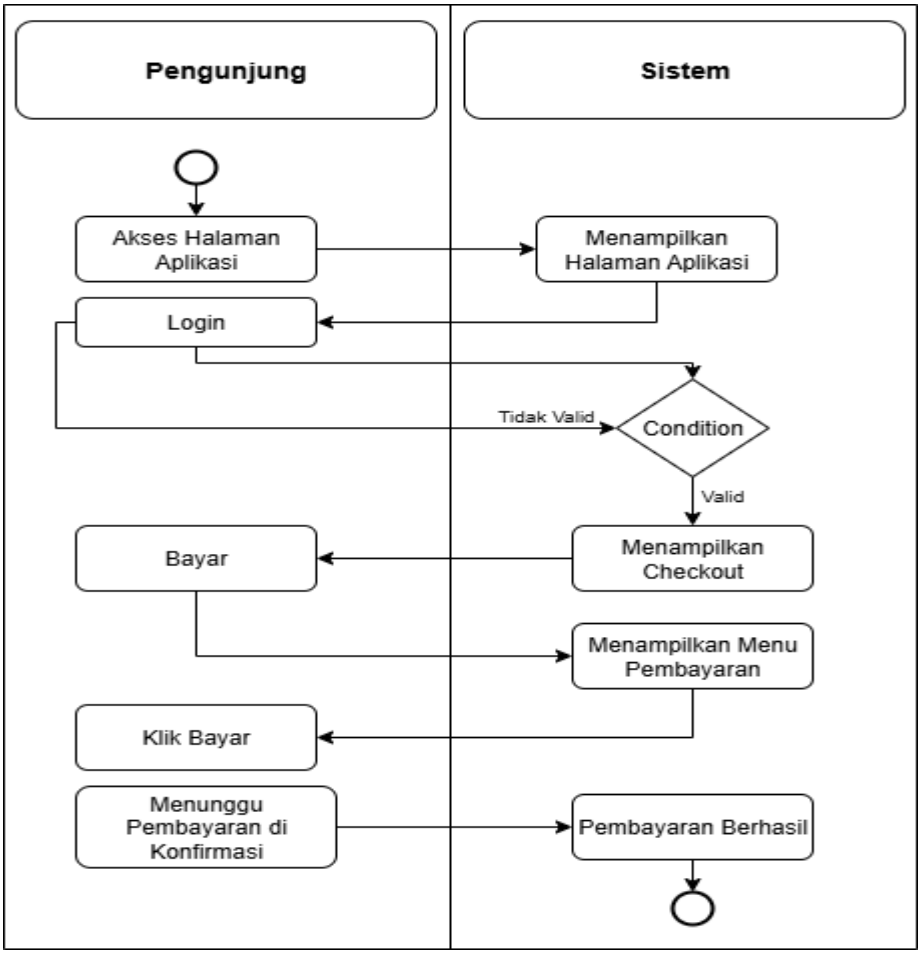


Gambar 4. 4 *Activity Diagram* Login

Alur proses untuk *Activity* Diagram Login dapat dipahami dari diagram aktivitas di atas. Pengunjung mengakses halaman aplikasi, setelah itu sistem menampilkan halaman aplikasi. Pengunjung memilih daftar, dan sistem kemudian menampilkan formulir untuk Pengunjung untuk di isi. Setelah disimpan, sistem akan memperbarui data dalam database untuk

memverifikasi data yang dikirim; jika ada masalah, pemberitahuan kesalahan akan ditampilkan; jika tidak, prosedur pendaftaran berhasil.

b) Activity Diagram Pembayaran

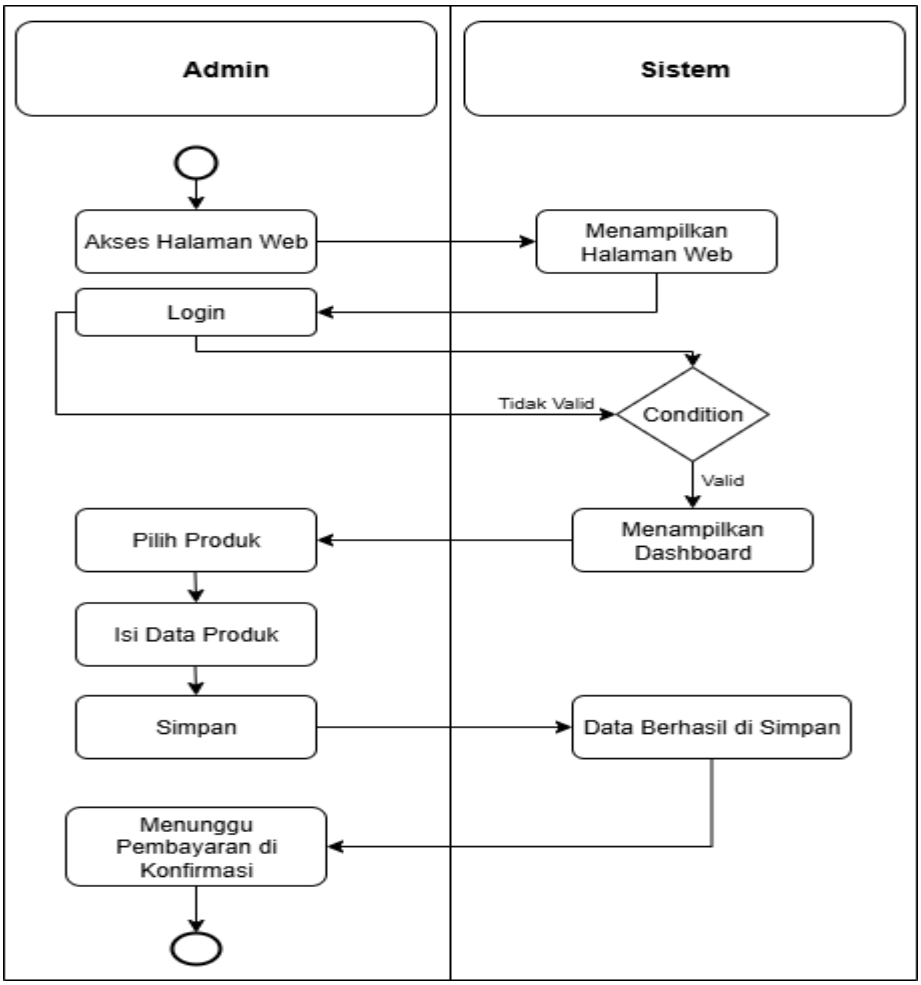


Gambar 4. 5 *Activity Diagram* Pembayaran

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas Pembayaran. Proses dimulai dengan Pengunjung mengakses halaman aplikasi, kemudian sistem menampilkan halaman aplikasi dan Pengunjung melakukan *Login* ketika tidak valid maka akan kembali ke halaman *Login* jika valid maka sistem akan menampilkan halaman *Checkout* dan memilih Bayar kemudian sistem akan menampilkan Menu

Pembayaran, setelah data sesuai maka Pengunjung memilih Klik Bayar dan menunggu Pembayaran di Konfirmasi, jika terdapat kesalahan akan menampilkan pesan kesalahan, dan jika tidak, maka Pembayaran Berhasil.

c) *Activity* Diagram Tambah Produk.



Gambar 4. 6 *Activity Diagram* Tambah Produk.

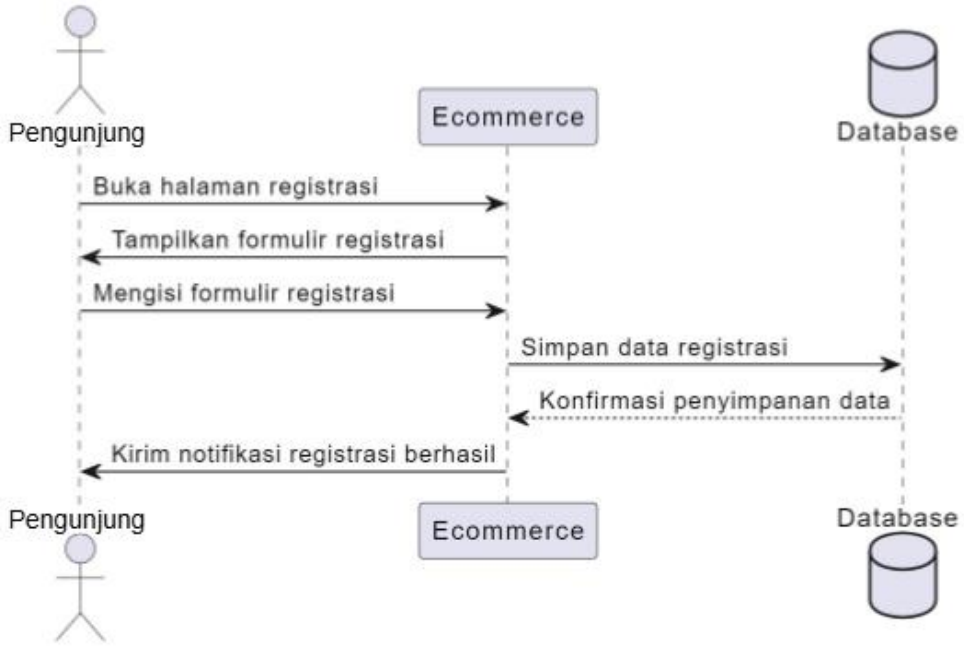
Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Tambah Produk. Proses dimulai dengan Admin mengakses halaman *Login* Admin, kemudian sistem menampilkan halaman *Login* kemudian Admin *Login* Jika tidak valid, maka sistem akan kembali ke halaman *Login* tapi Jika valid, maka sistem menampilkan halaman *Dashboard*, kemudian Admin memilih Tambah Produk dan

mengisi Data Produk Baru setelah itu Klik Simpan dan sistem akan menyimpan data.

6. *Sequence Diagram*

Sequence diagram menggambarkan perilaku objek dalam sebuah use case dengan mendeskripsikan siklus hidup objek dan pesan yang dikirim dan di terima antar objek. Proses menggambarkan diagram sekuens membutuhkan pengetahuan tentang objek-objek yang terlibat dalam kasus penggunaan dan metode-metode milik kelas yang di inisialisasi pada objek-objek yang digambar dalam diagram kelas. Berikut ini adalah ringkasan sequence diagram dari sistem informasi yang dikembangkan:

1) *Sequence Diagram*: Daftar Pengunjung Baru



***Gambar 4. 7** Sequence Diagram* Daftar Pengunjung Baru

Berdasarkan **Gambar 4. 7** Diatas *Sequence Diagram* Daftar Pengunjung

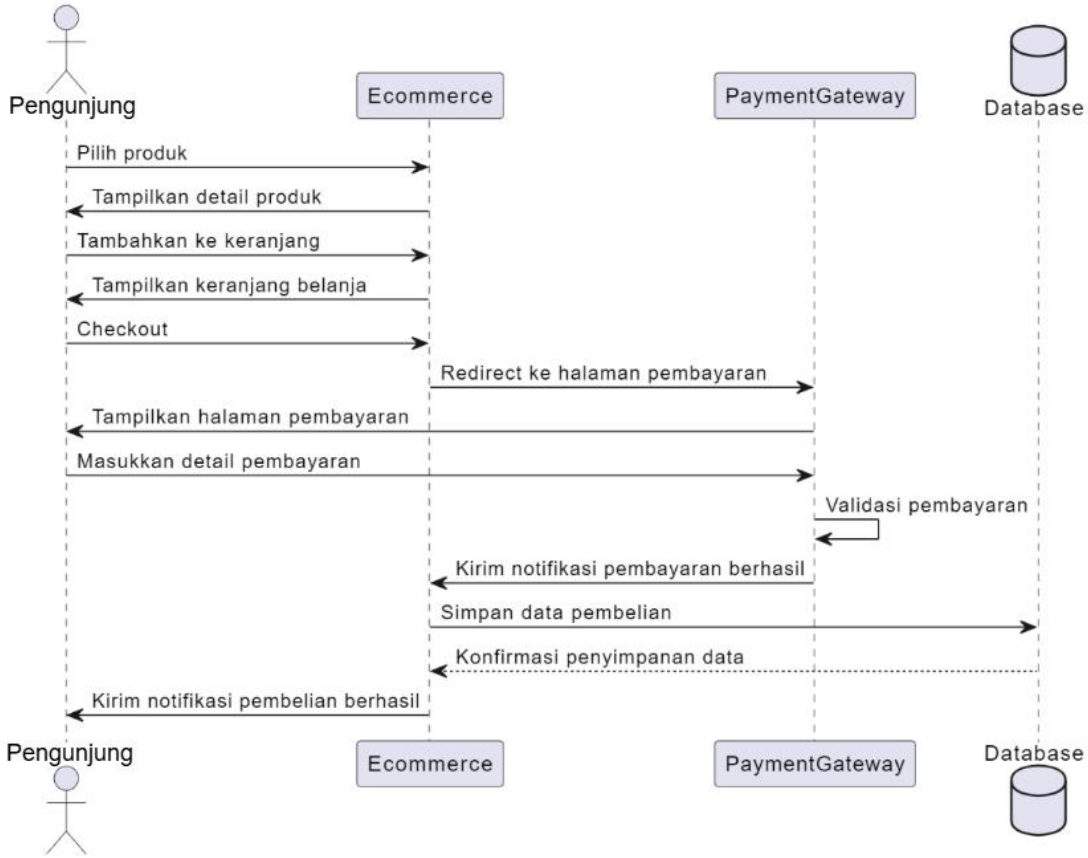
Baru, terdapat:

a. Terdapat 1 (satu) actor, sebagai sebagai pelaku kegiatan.

b. Terdapat 2 (dua) life line antar muka yang saling berinteraksi.

c. Terdapat 6 (enam) message, spesifikasi dari komunikasi antar obyek yang membuat informasi-informasi aktifitas terjadi.

2) *Sequence Diagram*: Pembayaran



Gambar 4. 8 *Sequence Diagram* Pembayaran

Berdasarkan **Gambar 4.8** *Sequence Diagram* Pembayaran terdapat:

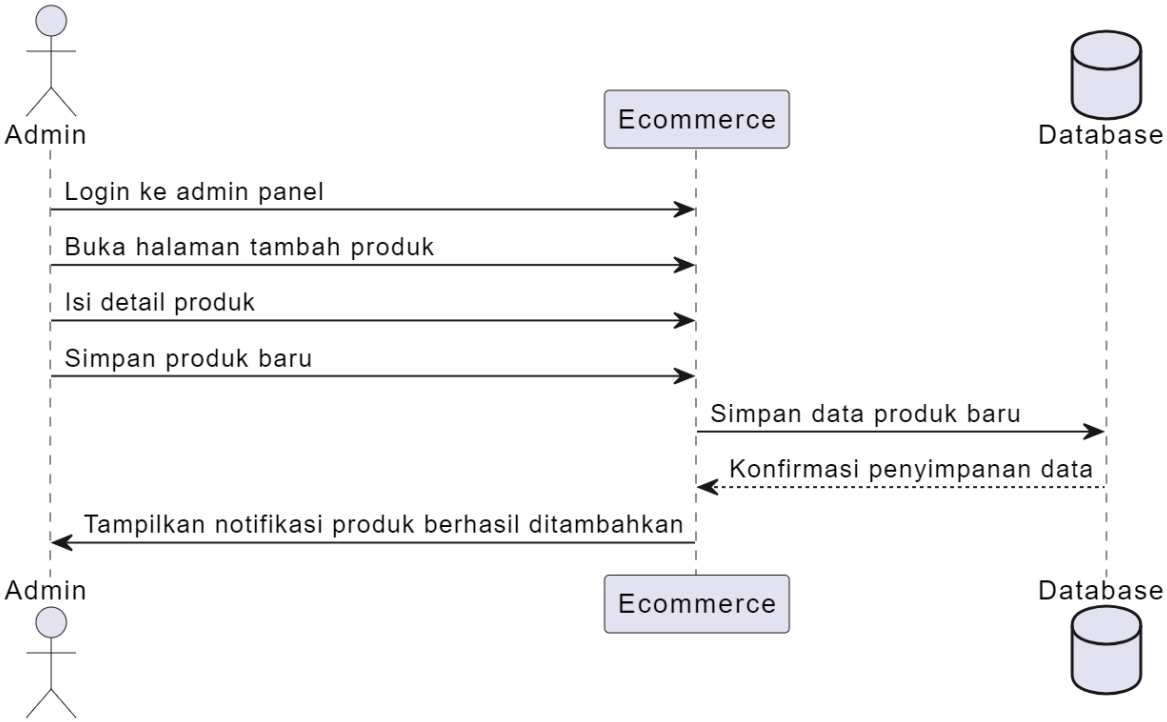
a. Terdapat 1 (satu) actor, sebagai sebagai pelaku kegiatan.

b. Terdapat 3 (tiga) lifeline antar muka yang saling berinteraksi.

c. Terdapat 13 (tiga belas) message, spesifikasi dari komunikasi antar obyek

yang membuat informasi-informasi aktifitas terjadi.

3) *Sequence Diagram*: Tambah Produk Baru



Gambar 4. 9 *Sequence Diagram* Tambah Produk Baru

Berdasarkan **Gambar 4.9.** *Sequence Diagram* Tambah Produk Baru terdapat:

1. Terdapat 1 (satu) *actor*, sebagai aktor dalam suatu kegiatan.

2. Terdapat 2 (dua) *life line* antar muka yang saling berinteraksi.

3. Terdapat 7 (tujuh) message, spesifikasi dari komunikasi antar obyek yang membuat informasi-informasi aktivitas terjadi.

7. Tabel *Database*

Adapun rancangan *database* dalam pembangunan sistem ini adalah terdiri dari beberapa tabel diantaranya yaitu :

Tabel 4. 4 *wpud_actionscheduler_actions*

No	Nama Field	Tipe Data	Ukuran/Detail
1	action_id	bigint	20
2	hook	varchar	191
3	status	varchar	20
4	scheduled_date_gmt	datetime	-
5	scheduled_date_local	datetime	-
6	priority	tinyint	3
7	args	varchar	191
8	schedule	longtext	-
9	group_id	bigint	20
10	attempts	int	11
11	last_attempt_gmt	datetime	-
12	last_attempt_local	datetime	-
13	claim_id	bigint	20
14	extended_args	varchar	8000

Tabel 4. 5 *wpud_actionscheduler_claims*

No	Nama Field	Tipe Data	Ukuran/Detail
1	claim_id	bigint	20
2	date_created_gmt	datetime	-

Tabel 4. 6 wpud_actionscheduler_groups

No	Nama Field	Tipe Data	Ukuran/Detail
1	group_id	bigint	20
2	slug	varchar	255

Tabel 4. 7 *wpud_actionscheduler_logs*

No	Nama Field	Tipe Data	Ukuran/Detail
1	log_id	bigint	20
2	action_id	bigint	20
3	message	text	-
4	log_date_gmt	datetime	-
5	log_date_local	datetime	-

Tabel 4. 8 *wpud_commentmeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	meta_id	bigint	20
2	comment_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 9 *wpud_comments*

No	Nama Field	Tipe Data	Ukuran/Detail
1	comment_ID	bigint	20
2	comment_post_ID	bigint	20
3	comment_author	tinytext	-
4	comment_author_email	varchar	100
5	comment_author_url	varchar	200
6	comment_author_IP	varchar	100
7	comment_date	datetime	-
8	comment_date_gmt	datetime	-
9	comment_content	text	-
10	comment_karma	int	11
11	comment_approved	varchar	20
12	comment_agent	varchar	255
13	comment_type	varchar	20
14	comment_parent	bigint	20
15	user_id	bigint	20

Tabel 4. 10 *wpud_links*

No	Nama Field	Tipe Data	Ukuran/Detail
1	link_id	bigint	20
2	link_url	varchar	255
3	link_name	varchar	255
4	link_image	varchar	255
5	link_target	varchar	25
6	link_description	varchar	255
7	link_visible	varchar	20
8	link_owner	bigint	20
9	link_rating	int	11
10	link_updated	datetime	-
11	link_rel	varchar	255
12	link_notes	mediumtext	-
13	link_rss	varchar	255

Tabel 4. 11 *wpud_options*

No	Nama Field	Tipe Data	Ukuran/Detail
1	option_id	bigint	20
2	option_name	varchar	191
3	option_value	longtext	-
4	autoload	varchar	20

Tabel 4. 12 wpud_pengajuan_buku

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	mediumint	9
2	time	datetime	-
3	nama_pengirim	varchar	100
4	judul_buku	varchar	255
5	nama_penulis	varchar	100
6	sinopsis	text	-
7	file_buku	varchar	255
8	jumlah_halaman	int	11
9	has_cover	tinyint	1
10	cover_image	varchar	255
11	nomor_whatsapp	varchar	20
12	status	varchar	20

Tabel 4. 13 *wpud_postmeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	meta_id	bigint	20
2	post_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 14 *wpud_posts*

No	Nama Field	Tipe Data	Ukuran/Detail
1	ID	bigint	20
2	post_author	bigint	20
3	post_date	datetime	-
4	post_date_gmt	datetime	-
5	post_content	longtext	-
6	post_title	text	-
7	post_excerpt	text	-
8	post_status	varchar	20
9	comment_status	varchar	20
10	ping_status	varchar	20
11	post_password	varchar	255

No	Nama Field	Tipe Data	Ukuran/Detail
12	post_name	varchar	200
13	to_ping	text	-
14	pinged	text	-
15	post_modified	datetime	-
16	post_modified_gmt	datetime	-
17	post_content_filtered	longtext	-
18	post_parent	bigint	20
19	guid	varchar	255
20	menu_order	int	11
21	post_type	varchar	20
22	post_mime_type	varchar	100
23	comment_count	bigint	20

Tabel 4. 15 *wpud_termmeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	meta_id	bigint	20
2	term_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 16 *wpud_terms*

No	Nama Field	Tipe Data	Ukuran/Detail
1	term_id	bigint	20
2	name	varchar	200
3	slug	varchar	200
4	term_group	bigint	10

Tabel 4. 17 *wpud_term_relationships*

No	Nama Field	Tipe Data	Ukuran/Detail
1	object_id	bigint	20
2	term_taxonomy_id	bigint	20
3	term_order	int	11

Tabel 4. 18 *wpud_term_taxonomy*

No	Nama Field	Tipe Data	Ukuran/Detail
1	term_taxonomy_id	bigint	20
2	term_id	bigint	20
3	taxonomy	varchar	32
4	description	longtext	-

No	Nama Field	Tipe Data	Ukuran/Detail
5	parent	bigint	20
6	count	bigint	20

Tabel 4. 19 *wpud_usermeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	umeta_id	bigint	20
2	user_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 20 *wpud_wc_admin_notes*

No	Nama Field	Tipe Data	Ukuran/Detail
1	ID	bigint	20
2	user_login	varchar	60
3	user_pass	varchar	255
4	user_nicename	varchar	50
5	user_email	varchar	100
6	user_url	varchar	100
7	user_registered	datetime	-
8	user_activation_key	varchar	255
9	user_status	int	11
10	display_name	varchar	250

Tabel 4. 21 *wpud_wc_admin_notes*

No	Nama Field	Tipe Data	Ukuran/Detail
1	note_id	bigint	20
2	name	varchar	255
3	type	varchar	20
4	locale	varchar	20
5	title	longtext	-
6	content	longtext	-
7	content_data	longtext	-
8	status	varchar	200
9	source	varchar	200
10	date_created	datetime	-
11	date_reminder	datetime	-
12	is_snoozable	tinyint	1
13	layout	varchar	20
14	image	varchar	200

No	Nama Field	Tipe Data	Ukuran/Detail
15	is_deleted	tinyint	1
16	is_read	tinyint	1
17	icon	varchar	200

Tabel 4. 22 *wpud_wc_admin_note_actions*

No	Nama Field	Tipe Data	Ukuran/Detail
1	action_id	bigint	20
2	note_id	bigint	20
3	name	varchar	255
4	label	varchar	255
5	query	longtext	-
6	status	varchar	255
7	actioned_text	varchar	255
8	nonce_action	varchar	255
9	nonce_name	varchar	255

Tabel 4. 23 *wpud_wc_category_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	category_tree_id	bigint	20
2	category_id	bigint	20

Tabel 4. 24 *wpud_wc_customer_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	customer_id	bigint	20
2	user_id	bigint	20
3	username	varchar	60
4	first_name	varchar	255
5	last_name	varchar	255
6	email	varchar	100
7	date_last_active	timestamp	-
8	date_registered	timestamp	-
9	country	char	2
10	postcode	varchar	20
11	city	varchar	100
12	state	varchar	100

Tabel 4. 25 *wpud_wc_download_log*

No	Nama Field	Tipe Data	Ukuran/Detail
1	download_log_id	bigint	20
2	timestamp	datetime	-
3	permission_id	bigint	20
4	user_id	bigint	20
5	user_ip_address	varchar	100

Tabel 4. 26 *wpud_wc_orders*

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	bigint	20
2	status	varchar	20
3	currency	varchar	10
4	type	varchar	20
5	tax_amount	decimal	26,8
6	total_amount	decimal	26,8
7	customer_id	bigint	20
8	billing_email	varchar	320
9	date_created_gmt	datetime	-
10	date_updated_gmt	datetime	-
11	parent_order_id	bigint	20
12	payment_method	varchar	100
13	payment_method_title	text	-
14	transaction_id	varchar	100
15	ip_address	varchar	100
16	user_agent	text	-
17	customer_note	text	-

Tabel 4. 27 *wpud_wc_orders_meta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	bigint	20
2	order_id	bigint	20
3	meta_key	varchar	255
4	meta_value	text	-

Tabel 4. 28 *wpud_wc_order_addresses*

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	bigint	20
2	order_id	bigint	20

No	Nama Field	Tipe Data	Ukuran/Detail
3	address_type	varchar	20
4	first_name	text	-
5	last_name	text	-
6	company	text	-
7	address_1	text	-
8	address_2	text	-
9	city	text	-
10	state	text	-
11	postcode	text	-
12	country	text	-
13	email	varchar	320
14	phone	varchar	100

Tabel 4. 29 *wpud_wc_order_coupon_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_id	bigint	20
2	coupon_id	bigint	20
3	date_created	datetime	-
4	discount_amount	double	-

Tabel 4. 30 *wpud_wc_order_operational_data*

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	bigint	20
2	order_id	bigint	20
3	created_via	varchar	100
4	woocommerce_version	varchar	20
5	prices_include_tax	tinyint	1
6	coupon_usages_are_counted	tinyint	1
7	download_permission_granted	tinyint	1
8	cart_hash	varchar	100
9	new_order_email_sent	tinyint	1
10	order_key	varchar	100
11	order_stock_reduced	tinyint	1
12	date_paid_gmt	datetime	-
13	date_completed_gmt	datetime	-
14	shipping_tax_amount	decimal	26,8
15	shipping_total_amount	decimal	26,8
16	discount_tax_amount	decimal	26,8
17	discount_total_amount	decimal	26,8
18	recorded_sales	tinyint	1

Tabel 4. 31 *wpud_wc_order_product_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_item_id	bigint	20
2	order_id	bigint	20
3	product_id	bigint	20
4	variation_id	bigint	20
5	customer_id	bigint	20
6	date_created	datetime	-
7	product_qty	int	11
8	product_net_revenue	double	-
9	product_gross_revenue	double	-
10	coupon_amount	double	-
11	tax_amount	double	-
12	shipping_amount	double	-
13	shipping_tax_amount	double	-

Tabel 4. 32 *wpud_wc_order_stats*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_id	bigint	20
2	parent_id	bigint	20
3	date_created	datetime	-
4	date_created_gmt	datetime	-
5	date_paid	datetime	-
6	date_completed	datetime	-
7	num_items_sold	int	11
8	total_sales	double	-
9	tax_total	double	-
10	shipping_total	double	-
11	net_total	double	-
12	returning_customer	tinyint	1
13	status	varchar	200
14	customer_id	bigint	20

Tabel 4. 33 *wpud_wc_order_tax_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_id	bigint	20
2	tax_rate_id	bigint	20
3	date_created	datetime	-
4	shipping_tax	double	-
5	order_tax	double	-

No	Nama Field	Tipe Data	Ukuran/Detail
6	total_tax	double	-

Tabel 4. 34 *wpud_wc_product_attributes_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	product_id	bigint	20
2	product_or_parent_id	bigint	20
3	taxonomy	varchar	32
4	term_id	bigint	20
5	is_variation_attribute	tinyint	1
6	in_stock	tinyint	1

Tabel 4. 35 *wpud_wc_product_download_directories*

No	Nama Field	Tipe Data	Ukuran/Detail
1	url_id	bigint	20
2	url	varchar	256
3	enabled	tinyint	1

Tabel 4. 36 *wpud_wc_product_meta_lookup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	product_id	bigint	20
2	sku	varchar	100
3	global_unique_id	varchar	100
4	virtual	tinyint	1
5	downloadable	tinyint	1
6	min_price	decimal	19,4
7	max_price	decimal	19,4
8	onsale	tinyint	1
9	stock_quantity	double	-
10	stock_status	varchar	100
11	rating_count	bigint	20
12	average_rating	decimal	3,2
13	total_sales	bigint	20
14	tax_status	varchar	100
15	tax_class	varchar	100

Tabel 4. 37 *wpud_wc_rate_limits*

No	Nama Field	Tipe Data	Ukuran/Detail
1	rate_limit_id	bigint	20

No	Nama Field	Tipe Data	Ukuran/Detail
2	rate_limit_key	varchar	200
3	rate_limit_expiry	bigint	20
4	rate_limit_remaining	smallint	10

Tabel 4. 38 *wpud_wc_reserved_stock*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_id	bigint	20
2	product_id	bigint	20
3	stock_quantity	double	-
4	timestamp	datetime	-
5	expires	datetime	-

Tabel 4. 39 *wpud_wc_tax_rate_classes*

No	Nama Field	Tipe Data	Ukuran/Detail
1	tax_rate_class_id	bigint	20
2	name	varchar	200
3	slug	varchar	200

Tabel 4. 40 *wpud_wc_webhooks*

No	Nama Field	Tipe Data	Ukuran/Detail
1	webhook_id	bigint	20
2	status	varchar	200
3	name	text	-
4	user_id	bigint	20
5	delivery_url	text	-
6	secret	text	-
7	topic	varchar	200
8	date_created	datetime	-
9	date_created_gmt	datetime	-
10	date_modified	datetime	-
11	date_modified_gmt	datetime	-
12	api_version	smallint	4
13	failure_count	smallint	10
14	pending_delivery	tinyint	1

Tabel 4. 41 *wpud_woocommerce_api_keys*

No	Nama Field	Tipe Data	Ukuran/Detail
1	key_id	bigint	20

No	Nama Field	Tipe Data	Ukuran/Detail
2	user_id	bigint	20
3	description	varchar	200
4	permissions	varchar	10
5	consumer_key	char	64
6	consumer_secret	char	43
7	nonces	longtext	-
8	truncated_key	char	7
9	last_access	datetime	-

Tabel 4. 42 *wpud_woocommerce_attribute_taxonomies*

No	Nama Field	Tipe Data	Ukuran/Detail
1	attribute_id	bigint	20
2	attribute_name	varchar	200
3	attribute_label	varchar	200
4	attribute_type	varchar	20
5	attribute_orderby	varchar	20
6	attribute_public	int	1

Tabel 4. 43 *wpud_woocommerce_downloadable_product_permissions*

No	Nama Field	Tipe Data	Ukuran/Detail
1	permission_id	bigint	20
2	download_id	varchar	36
3	product_id	bigint	20
4	order_id	bigint	20
5	order_key	varchar	200
6	user_email	varchar	200
7	user_id	bigint	20
8	downloads_remaining	varchar	9
9	access_granted	datetime	-
10	access_expires	datetime	-
11	download_count	bigint	20

Tabel 4. 44 *wpud_woocommerce_log*

No	Nama Field	Tipe Data	Ukuran/Detail
1	log_id	bigint	20
2	timestamp	datetime	-
3	level	smallint	4
4	source	varchar	200
5	message	longtext	-

No	Nama Field	Tipe Data	Ukuran/Detail
6	context	longtext	-

Tabel 4. 45 *wpud_woocommerce_order_itemmeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	meta_id	bigint	20
2	order_item_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 46 *wpud_woocommerce_order_items*

No	Nama Field	Tipe Data	Ukuran/Detail
1	order_item_id	bigint	20
2	order_item_name	text	-
3	order_item_type	varchar	200
4	order_id	bigint	20

Tabel 4. 47 *wpud_woocommerce_payment_tokenmeta*

No	Nama Field	Tipe Data	Ukuran/Detail
1	meta_id	bigint	20
2	payment_token_id	bigint	20
3	meta_key	varchar	255
4	meta_value	longtext	-

Tabel 4. 48 *wpud_woocommerce_payment_tokens*

No	Nama Field	Tipe Data	Ukuran/Detail
1	token_id	bigint	20
2	gateway_id	varchar	200
3	token	text	-
4	user_id	bigint	20
5	type	varchar	200
6	is_default	tinyint	1

Tabel 4. 49 *wpud_woocommerce_sessions*

No	Nama Field	Tipe Data	Ukuran/Detail
1	session_id	bigint	20
2	session_key	char	32

No	Nama Field	Tipe Data	Ukuran/Detail
3	session_value	longtext	-
4	session_expiry	bigint	20

Tabel 4. 50 *wpud_woocommerce_shipping_zones*

No	Nama Field	Tipe Data	Ukuran/Detail
1	zone_id	bigint	20
2	zone_name	varchar	200
3	zone_order	bigint	20

Tabel 4. 51 *wpud_woocommerce_shipping_zone_locations*

No	Nama Field	Tipe Data	Ukuran/Detail
1	location_id	bigint	20
2	zone_id	bigint	20
3	location_code	varchar	200
4	location_type	varchar	40

Tabel 4. 52 *wpud_woocommerce_shipping_zone_methods*

No	Nama Field	Tipe Data	Ukuran/Detail
1	zone_id	bigint	20
2	instance_id	bigint	20
3	method_id	varchar	200
4	method_order	bigint	20
5	is_enabled	tinyint	1

Tabel 4. 53 *wpud_woocommerce_tax_rates*

No	Nama Field	Tipe Data	Ukuran/Detail
1	tax_rate_id	bigint	20
2	tax_rate_country	varchar	2
3	tax_rate_state	varchar	200
4	tax_rate	varchar	8
5	tax_rate_name	varchar	200
6	tax_rate_priority	bigint	20
7	tax_rate_compound	int	1
8	tax_rate_shipping	int	1
9	tax_rate_order	bigint	20
10	tax_rate_class	varchar	200

Tabel 4. 54 *wpud_woocommerce_tax_rate_locations*

No	Nama Field	Tipe Data	Ukuran/Detail
1	location_id	bigint	20
2	location_code	varchar	200
3	tax_rate_id	bigint	20
4	location_type	varchar	40

Tabel 4. 55 *wpud_wpfm_backup*

No	Nama Field	Tipe Data	Ukuran/Detail
1	id	int	11
2	backup_name	text	-
3	backup_date	text	-

B. Detail Sistem

1. Tampilan Halaman Utama

   Berikut adalah tampilan Halaman Utama pada Sistem Informasi Sampan Institute.



Gambar 4. 10 Tampilan Halaman Utama

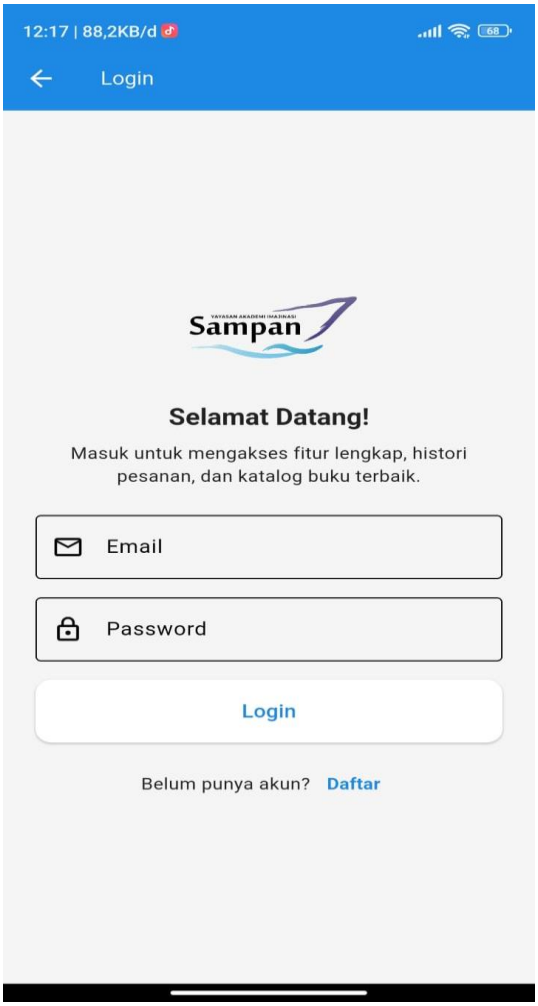
MainActivity.java

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:appsampanins/core/theme/app_theme.dart';
import 'package:appsampanins/di/injection.dart';
import 'package:appsampanins/providers/book_provider.dart';
import 'package:appsampanins/providers/cart_provider.dart';
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  // Konfigurasi Logger
  Logger.root.level = Level.ALL; // Mengaktifkan semua level
log
  Logger.root.onRecord.listen((record) {
    debugPrint('${record.level.name}: ${record.time}:
${record.message}');
  });
@override
  void didChangeAppLifecycleState(AppLifecycleState state) {
    debugPrint('App lifecycle state changed to: $state');
    if (state == AppLifecycleState.paused || state ==
AppLifecycleState.detached) {
      // Aplikasi di-pause atau ditutup, simpan keranjang
      debugPrint('App is paused or closed, syncing cart to
server');
      cartProvider.syncCartToServer();
    }
  }
}
'/register': (context) => const RegisterScreen(),
      '/cart': (context) => const CartScreen(),
      '/checkout': (context) => const CheckoutScreen(),
      '/order-confirmation': (context) => const
OrderConfirmationScreen(),
      '/book-detail': (context) => const BookDetailScreen(),
      '/new-books': (context) => const NewBooksScreen(),
      '/login': (context) => const LoginScreen(),
      '/blog-detail': (context) => const BlogDetailScreen(),
      '/blog-list': (context) => const BlogListScreen(),
      '/catalog': (context) => const CatalogScreen(),
      '/publish-book': (context) => const
BookSubmissionHomeScreen(),
      '/track-shipment': (context) => const
TrackShipmentScreen()
```

2. Halaman *Login* Akun

Berikut adalah tampilan login akun pada Sistem Informasi Sampan Institute. Halaman *Login* merupakan halaman *Login user. Login* merupakan tampilan yang ada pada sistem, yang digunakan *User* untuk masuk ke Akun Pelanggan.



Gambar 4. 11 Tampilan Halaman Login

LoginActivity.java

```
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.URL;
import java.util.Arrays;
import java.util.concurrent.Executor;
import java.util.concurrent.Executors;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

ChangeStatusBarColor(Color.parseColor(getPreferences().getStrin
g(Constant.HEADER_COLOR, Constant.HEAD_COLOR)));
    binding =
ActivityLogInBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());
    mAuth = FirebaseAuth.getInstance();
    setCLickEvent();
    setScreenLayoutDirection();
    loginWithFB();
    setColor();
    getIntentData();

    if (Constant.APPLOGO != null &&
!Constant.APPLOGO.equals("")) {

Glide.with(this).load(Constant.APPLOGO).error(R.drawable.logo).
into(binding.ivLogo);
    }

    GoogleSignInOptions gso = new GoogleSignInOptions
            .Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)

.requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build();

    mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
    binding.ivBlackBackButton.setOnClickListener(v ->
finish());
}
public void getIntentData() {
    bundle = getIntent().getExtras();
    if (bundle != null && bundle.containsKey("is_splash")) {
        isSplash = bundle.getBoolean("is_splash");
    }
}
```

3. Halaman Registrasi

Berikut adalah tampilan Registrasi pada Sistem Informasi Sampan Institute. Halaman ini digunakan pelanggan untuk membuat akun baru.



Gambar 4. 12 Tampilan Halaman Registrasi

LoginActivity.java

```
import com.bumptech.glide.Glide;
import com.ciyashop.library.apicall.PostApi;
import com.ciyashop.library.apicall.URLS;
import
com.ciyashop.library.apicall.interfaces.OnResponseListner;
import com.example.sampan.R;
import com.example.sampan.customview.bounceview.BounceView;
import com.example.sampan.customview.edittext.EditTextMedium;
import com.example.sampan.customview.edittext.EditTextRegular;
import com.example.sampan.databinding.ActivityLogInBinding;

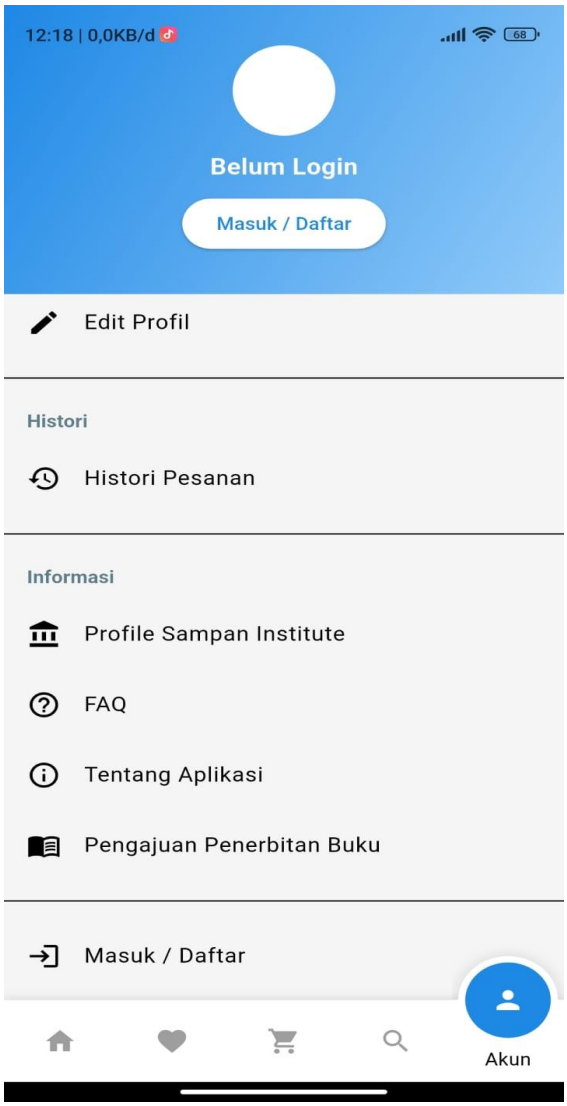
public void loginWithFB() {
    callbackManager = CallbackManager.Factory.create();

    // If you are using in a fragment, call
loginButton.setFragment(this);
LoginManager.getInstance().registerCallback(callbackManager,
            new FacebookCallback<LoginResult>() {
                @Override
                public void onSuccess(LoginResult loginResult)
{
                    // App code
                    String accessToken =
loginResult.getAccessToken()
                            .getToken();
                    Log.e("accessToken", accessToken);

                    GraphRequest request =
GraphRequest.newMeRequest(loginResult.getAccessToken(),
                            (object, response) -> {
                                Log.e("LoginActivity",
response.toString());
                                try {
                                    String id =
object.getString("id");
                                    faceBookImageUrl =
"https://graph.facebook.com/" + id + "/picture?type=large";
                                    fbJsonObject = object;
                                    getBitmapInAsync();
                                } catch (JSONException e) {
```

4. Halaman Utama Profil Pelanggan

Berikut adalah tampilan halaman profil pelanggan pada Sistem Informasi Sampan Institute. Halaman ini merupakan halaman awal pelanggan yang langsung menampilkan profil pelanggan.



Gambar 4. 13 Tampilan Halaman Utama Profil Pelanggan

AccountActivity.java

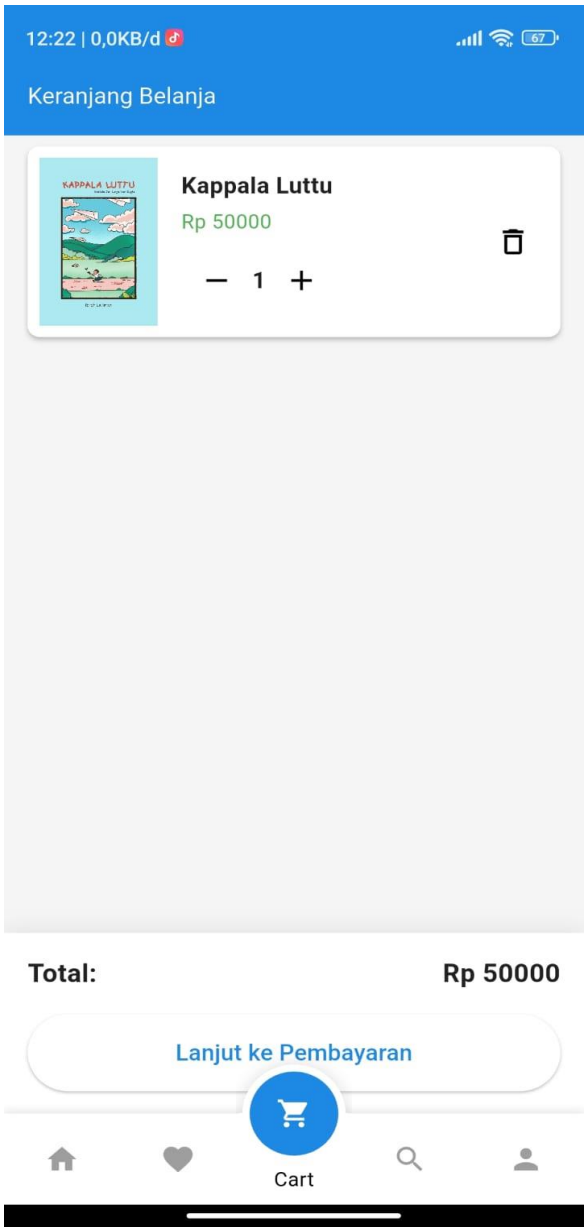
```
import android.Manifest;
import android.content.ActivityNotFoundException;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.content.res.ColorStateList;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.PorterDuffColorFilter;
import android.graphics.Typeface;
import android.graphics.drawable.ColorDrawable;
import android.graphics.drawable.Drawable;
ActivityResultLauncher<Intent> cameraActivityResultLauncher;
ActivityResultLauncher<Intent> galleryActivityResultLauncher;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    binding =
ActivityAccountBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    prepareCameraLauncher();
    prepareGalleryLauncher();
    setClickEvent();
    setToolbarTheme();
    showDownload();
    setScreenLayoutDirection();
    settvTitle(getResources().getString(R.string.account));
    hideSearchNotification();
    showBackButton();
    setBottomBar("account", binding.svHome);
    customerId =
getPreferences().getString(RequestParamUtils.ID, "");
    isWallet =
getPreferences().getBoolean(RequestParamUtils.IS_WALLET,
false);
    infoPages();
    if (Config.IS_CATALOG_MODE_OPTION) {
        binding.tvMyOrder.setVisibility(View.GONE);
    } else {
        binding.tvMyOrder.setVisibility(View.VISIBLE);
    }
```

5. Halaman *Cart*

Berikut adalah tampilan *Cart* atau Keranjang pada Sistem Informasi Sampan Institute Halaman ini menampilkan produk yang di pesan oleh Pelanggan, apabila memilih proses *checkout* maka Pelanggan akan beralih ke halaman *checkout*.



Gambar 4. 14 Tampilan Halaman Cart

CartActivity.java

```
import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
private ActivityCartBinding binding;
private LayoutEmptyBinding emptyBinding;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

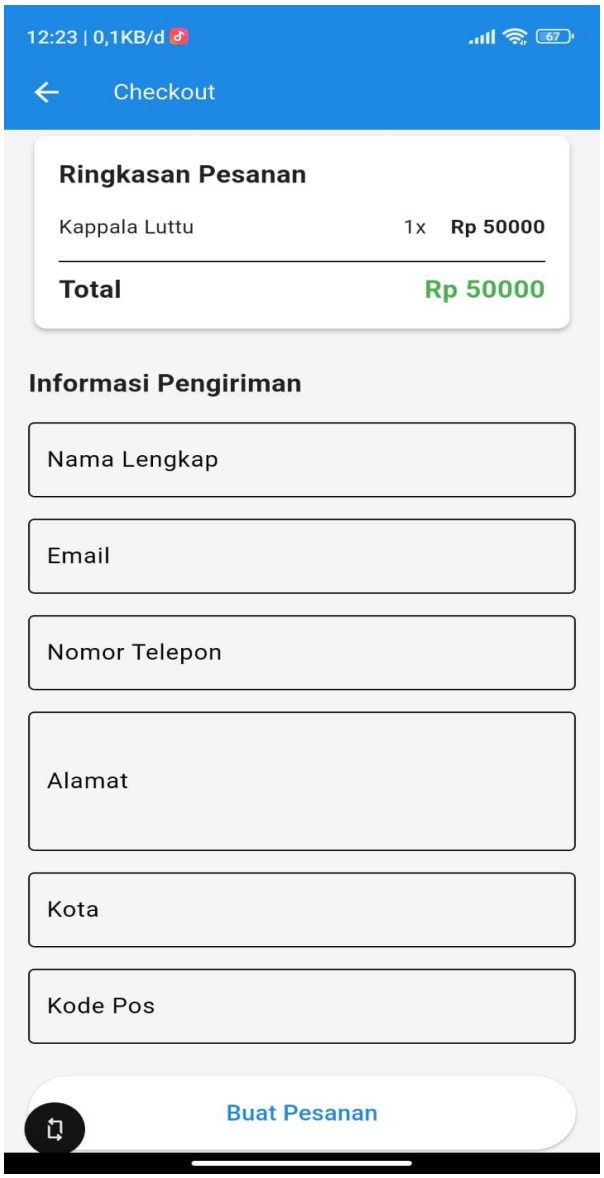
    binding = ActivityCartBinding.inflate(getLayoutInflater());
    emptyBinding = LayoutEmptyBinding.bind(binding.getRoot());
    setContentView(binding.getRoot());

    databaseHelper = new DatabaseHelper(this);
    emptyBinding.llEmpty.setVisibility(View.GONE);

    setClickEvent();
    setToolbarTheme();
    setThemeColor();
    setScreenLayoutDirection();
    getIntentData();
    setCartAdapter();
    settvTitle(getString(R.string.cart));
    showBackButton();
    hideSearchNotification();
    getCartData();
    setBottomBar("cart", binding.svHome);
    customerId =
getPreferences().getString(RequestParamUtils.ID, "");
}
public void getIntentData() {
    bundle = getIntent().getExtras();
    if (bundle != null) {
        id = bundle.getString(RequestParamUtils.ID);
        buyNow = bundle.getInt(RequestParamUtils.buynow);
    }
}
```

6. Halaman *Checkout*

Berikut adalah tampilan *Checkout* pada Sistem Informasi Sampan Institute. Pada halaman ini pelanggan akan mengisi data alamat penerima pesanan.



Gambar 4. 15 Tampilan Halaman Checkout

MyOrderActivity.java

```
import android.app.ActivityManager;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Toast;
private boolean loading = true;
private boolean Splashscreen = false;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    binding =
ActivityMyOrderBinding.inflate(getLayoutInflater());
    emptyBinding = LayoutEmptyBinding.bind(binding.getRoot());
    orderPlaceholderBinding =
LayoutOrderPlaceholderBinding.bind(binding.getRoot());
    setContentView(binding.getRoot());
    emptyBinding.tvEmptyTitle.setText(R.string.no_order_found);
    setToolbarTheme();
    setScreenLayoutDirection();
    settvTitle(getResources().getString(R.string.my_orders));

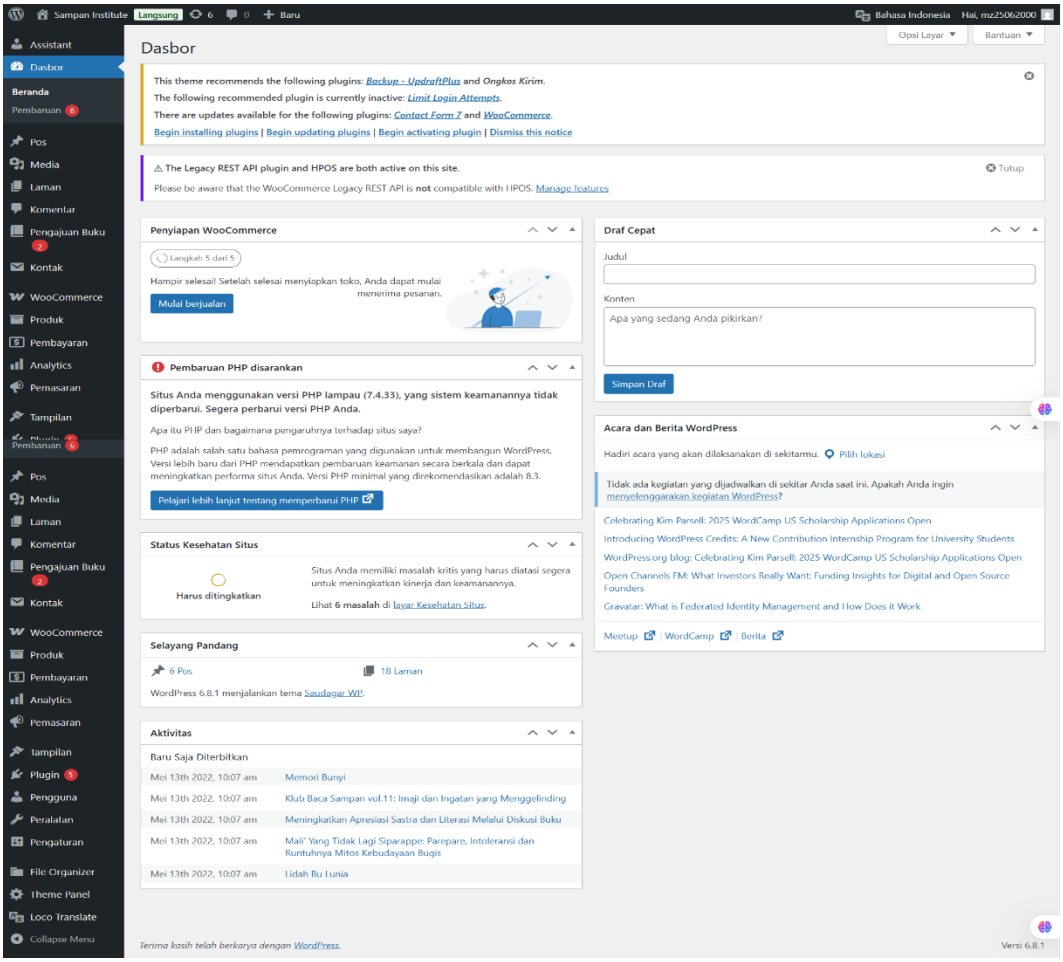
    if (getPreferences().getString(RequestParamUtils.ID,
"").equals("")) {
        Intent myOrderIntent = new Intent(this,
LogInActivity.class);
        startActivity(myOrderIntent);
    }

    showBackButton();
    setEmptyColor();
    hideSearchNotification();
    setMyOrderAdapter();
    myOrder(true);
    Intent intent = getIntent();
    if (intent.hasExtra(RequestParamUtils.Splashscreen)) {
        Splashscreen =
intent.getBooleanExtra(RequestParamUtils.Splashscreen, true);
    } else {
        Splashscreen = false;
    }
    ivBack.setOnClickListener(view -> backPressed());
}
```

7. Admin

a. Halaman *Dashboard*

Berikut adalah tampilan *Dashboard* pada Sistem Informasi Sampan Institute. Halaman *Dashboard* ini menampilkan Laporan Penjualan, Pesanan Terbaru, Daftar Pelanggan, Produk Yang Baru Ditambahkan, serta beberapa menu seperti Kategori, Sub Kategori dan lain-lain.



Gambar 4. 16 Tampilan Halaman *Dashboard* Admin

Index.php *wp-admin*

```
<?php
/**
 * Dashboard Administration Screen
 *
 * @package WordPress
 * @subpackage Administration
 */

/** Load WordPress Bootstrap */
require_once __DIR__ . '/admin.php';

/** Load WordPress dashboard API */
require_once ABSPATH . 'wp-admin/includes/dashboard.php';

wp_dashboard_setup();

wp_enqueue_script( 'dashboard' );

if ( current_user_can( 'install_plugins' ) ) {
    wp_enqueue_script( 'plugin-install' );
    wp_enqueue_script( 'updates' );
}
if ( current_user_can( 'upload_files' ) ) {
    wp_enqueue_script( 'media-upload' );
}
add_thickbox();

if ( wp_is_mobile() ) {
    wp_enqueue_script( 'jquery-touch-punch' );
}
// Used in the HTML title tag.
$title       = __( 'Dashboard' );
$parent_file = 'index.php';

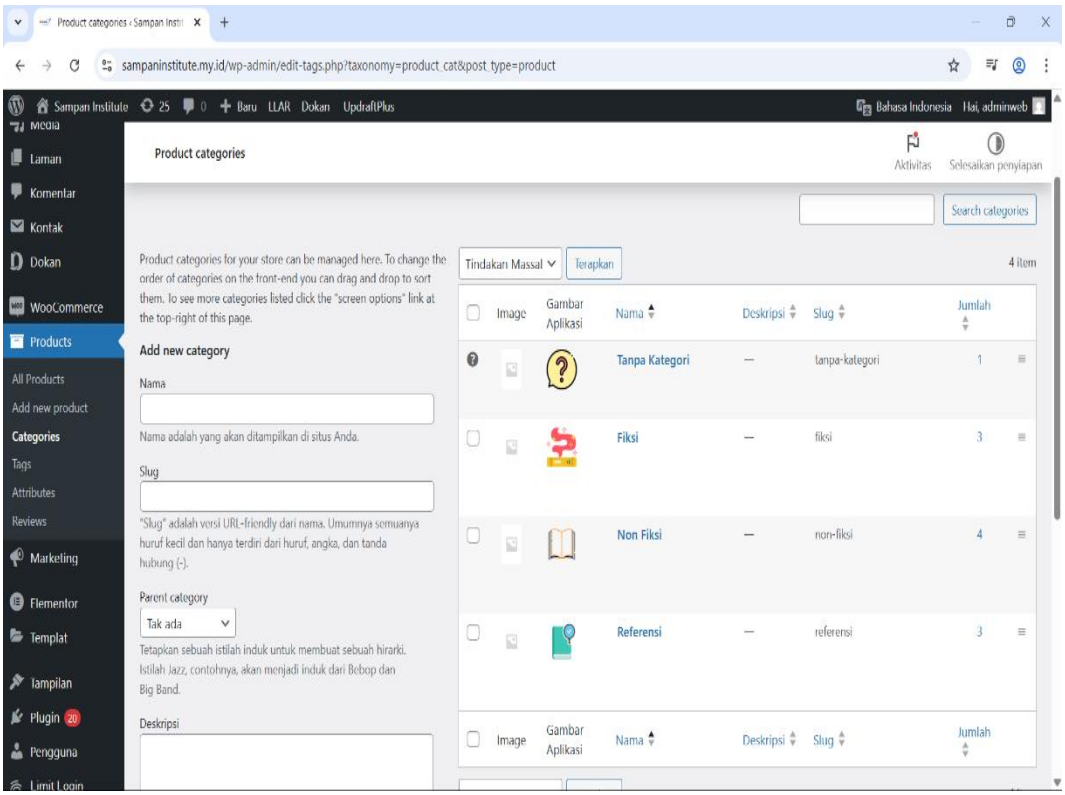
$help  = '<p>' . __( 'Welcome to your WordPress Dashboard!' ) .
'</p>';
$help .= '<p>' . __( 'The Dashboard is the first place you will
come to every time you log into your site. It is where you will
find all your WordPress tools. If you need help, just click the
&#8220;Help&#8221; tab above the screen title.' ) . '</p>';

$screen = get_current_screen();

$screen->add_help_tab(
    array(
        'id'      => 'overview',
        'title'   => __( 'Overview' ),
        'content' => $help,
    )
);
```

b. Halaman Tambah Kategori

Berikut adalah tampilan Tambah Kategori pada Sistem Informasi Sampan Institute. Halaman Tambah Kategori merupakan halaman yang digunakan Admin untuk menambah Kategori Produk.



Gambar 4. 17 Tampilan Halaman Menambahkan Kategori

ProductCategories.php

```
<?php
/**
 * Comment Management Screen
 *
 * @package WordPress
 * @subpackage Administration
 */

/** Load WordPress Bootstrap */
require_once __DIR__ . '/admin.php';

$parent_file  = 'edit-comments.php';
$submenu_file = 'edit-comments.php';

/**
 * @global string $action
 */
global $action;

$action = ! empty( $_REQUEST['action'] ) ? sanitize_text_field(
$_REQUEST['action'] ) : '';

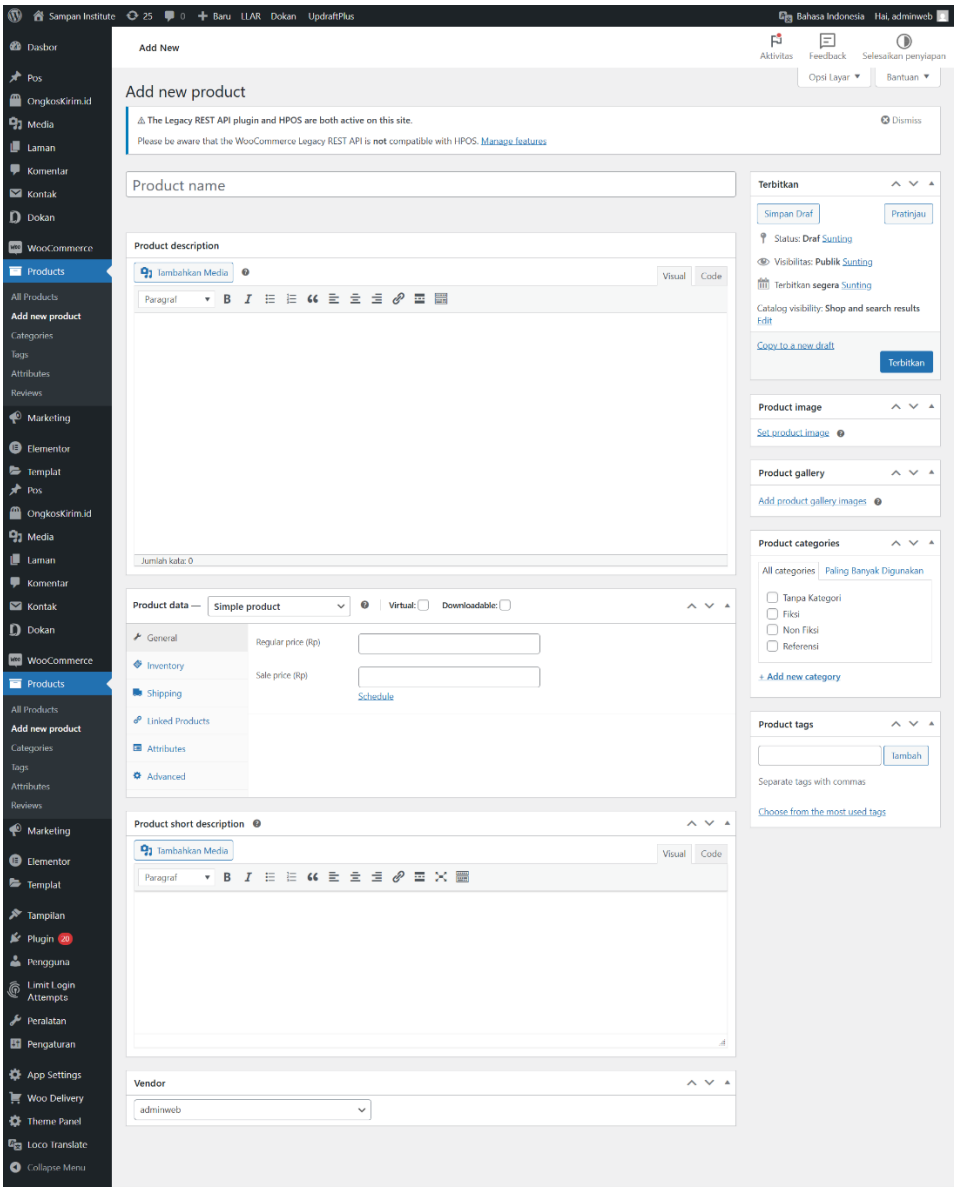
if ( isset( $_POST['deletecomment'] ) ) {
    $action = 'deletecomment';
}

if ( 'cdc' === $action ) {
    $action = 'delete';
} elseif ( 'mac' === $action ) {
    $action = 'approve';
}

if ( isset( $_GET['dt'] ) ) {
    if ( 'spam' === $_GET['dt'] ) {
        $action = 'spam';
    } elseif ( 'trash' === $_GET['dt'] ) {
        $action = 'trash';
    }
}
wp_admin_notice(
            '<strong>' . __( 'Caution:' ) . '</strong> ' .
$caution_msg,
            array(
                'type' => 'warning',
                'id'   => 'message',
            )
        );
```

c. Halaman Tambah Produk

Berikut adalah tampilan halaman tambah Produk pada Sistem Informasi Sampan Institute. Di halaman ini Admin akan mengisi beberapa formulir jika ingin menambahkan produk baru pada *Web* dan setelah di isi kemudian Admin menekan tombol tambah untuk menambahkan produk.



Gambar 4. 18 Tampilan Halaman Menambahkan Produk

.

AddNewProduct.php

```
<?php
/**
 * WordPress Administration Bootstrap
 *
 * @package WordPress
 * @subpackage Administration
 */

/**
 * In WordPress Administration Screens
 *
 * @since 2.3.2
 */
if ( ! defined( 'WP_ADMIN' ) ) {
    define( 'WP_ADMIN', true );
}

if ( ! defined( 'WP_NETWORK_ADMIN' ) ) {
    define( 'WP_NETWORK_ADMIN', false );
}

if ( ! defined( 'WP_USER_ADMIN' ) ) {
    define( 'WP_USER_ADMIN', false );
}

if ( ! WP_NETWORK_ADMIN && ! WP_USER_ADMIN ) {
    define( 'WP_BLOG_ADMIN', true );
}

if ( isset( $_GET['import'] ) && ! defined( 'WP_LOAD_IMPORTERS'
) ) {
    define( 'WP_LOAD_IMPORTERS', true );
}
if ( get_option( 'db_upgraded' ) ) {

    flush_rewrite_rules();
    update_option( 'db_upgraded', false, true );

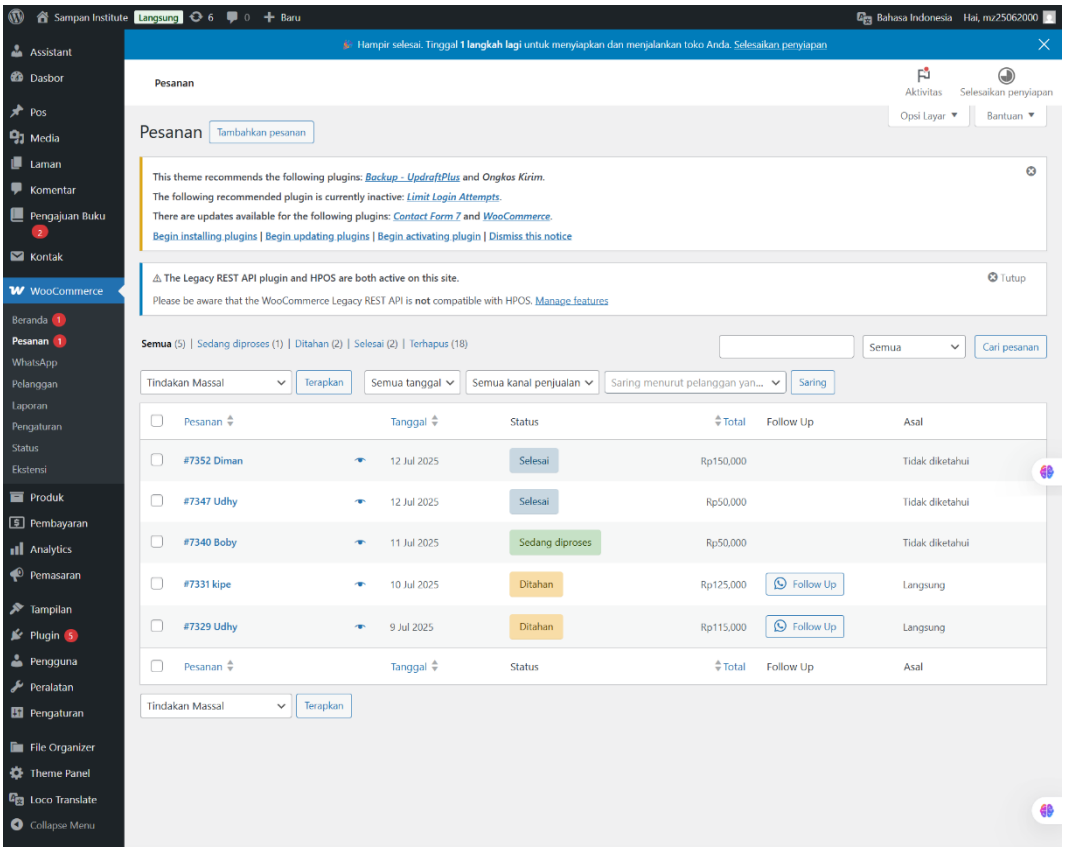
    /**
     * Fires on the next page load after a successful DB
upgrade.
     *
     * @since 2.8.0
     */
    do_action( 'after_db_upgrade' );

} elseif ( ! wp_doing_ajax() && empty( $_POST )
    && (int) get_option( 'db_version' ) !== $wp_db_version
) {

    if ( ! is_multisite() ) {
        wp_redirect( admin_url( 'upgrade.php?_wp_http_referer='
. urlencode( wp_unslash( $_SERVER['REQUEST_URI'] ) ) ) );
        exit;
    }
```

d. Halaman Laporan Order

Berikut adalah tampilan laporan Order pada Sistem Informasi Sampan Institute. Halaman Laporan Order ini merupakan halaman yang menampilkan nama-nama Pelanggan lengkap dengan produk yang dipesan.



Gambar 4. 19 Tampilan Halaman Laporan Order.

Order.php

```
<?php
/**
 * Build Administration Menu.
 *
 * @package WordPress
 * @subpackage Administration
 */

// Don't load directly.
if ( ! defined( 'ABSPATH' ) ) {
    die( '-1' );
}

/**
 * Constructs the admin menu.
 *
 * The elements in the array are:
 *     0: Menu item name.
 *     1: Minimum level or capability required.
 *     2: The URL of the item's file.
 *     3: Page title.
 *     4: Classes.
 *     5: ID.
 *     6: Icon for top level menu.
 *
 * @global array $menu
 */

$menu[2] = array( __( 'Dashboard' ), 'read', 'index.php', '',
'menu-top menu-top-first menu-icon-dashboard', 'menu-
dashboard', 'dashicons-dashboard' );

$submenu['index.php'][0] = array( __( 'Home' ), 'read',
'index.php' );

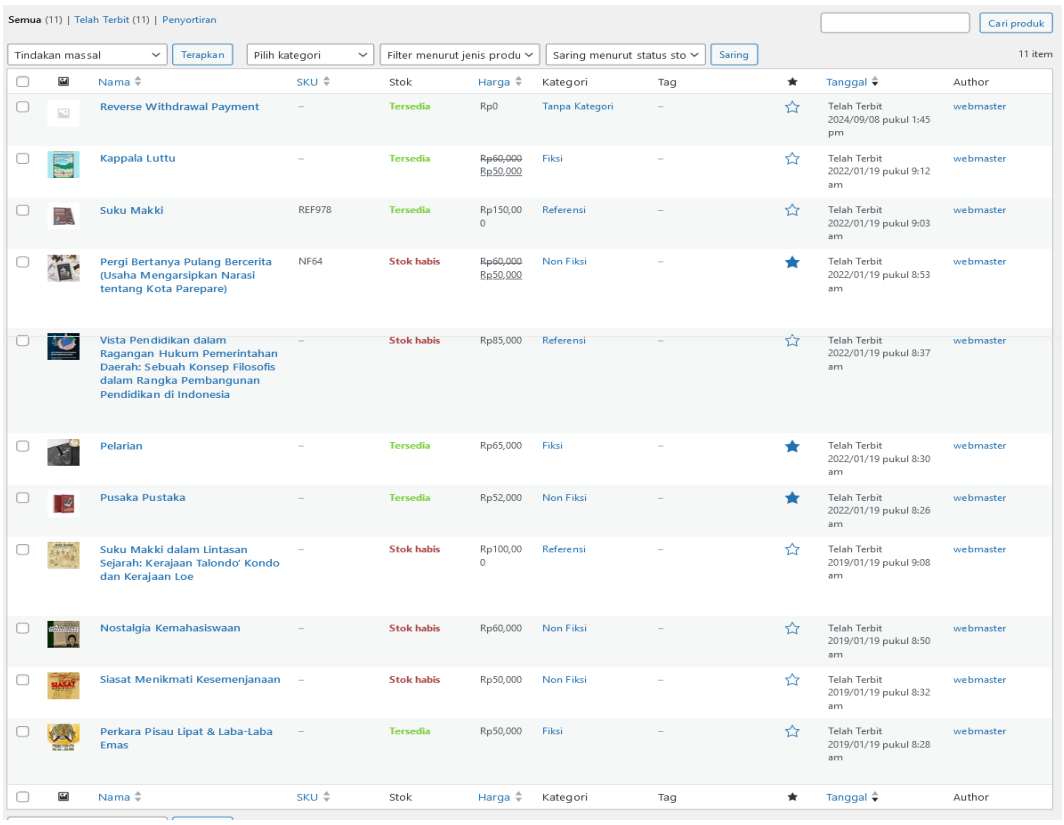
if ( is_multisite() ) {
    $submenu['index.php'][5] = array( __( 'My Sites' ), 'read',
'my-sites.php' );
}

if ( ! is_multisite() || current_user_can( 'update_core' ) ) {
    $update_data = wp_get_update_data();
}

    }
                number_format_i18n(
$update_data['counts']['total'] )
            )
        ),
```

e. Data User

Berikut ini adalah Tampilan Data User pada sistem informasi Sampan Institute yang dimana admin dapat mengelola dan menambahkan data user. Admin dalat mengelola password atau kata sandi akun user.



Gambar 4. 20 Daftar Produk

Data diatas digunakan untuk diinput kedalam Aplikasi sistem informasi Sampan Institute.

Product.php

```
<?php
/**
 * Edit user administration panel.
 *
 * @package WordPress
 * @subpackage Administration
 */

/** WordPress Administration Bootstrap */
require_once __DIR__ . '/admin.php';

/** WordPress Translation Installation API */
require_once ABSPATH . 'wp-admin/includes/translation-
install.php';

$action          = ! empty( $_REQUEST['action'] ) ?
sanitize_text_field( $_REQUEST['action'] ) : '';
$user_id         = ! empty( $_REQUEST['user_id'] ) ? absint(
$_REQUEST['user_id'] ) : 0;
$wp_http_referer = ! empty( $_REQUEST['wp_http_referer'] ) ?
sanitize_url( $_REQUEST['wp_http_referer'] ) : '';

$current_user = wp_get_current_user();

if ( ! defined( 'IS_PROFILE_PAGE' ) ) {
    define( 'IS_PROFILE_PAGE', ( $user_id === $current_user->ID
) );
}

if ( ! $user_id && IS_PROFILE_PAGE ) {
    $user_id = $current_user->ID;
} elseif ( ! $user_id && ! IS_PROFILE_PAGE ) {
    wp_die( __( 'Invalid user ID.' ) );
} elseif ( ! get_userdata( $user_id ) ) {
    wp_die( __( 'Invalid user ID.' ) );
}

wp_enqueue_script( 'user-profile' );

if ( wp_is_application_passwords_available_for_user( $user_id )
) {
    wp_enqueue_script( 'application-passwords' );
}

if ( IS_PROFILE_PAGE ) {
    // Used in the HTML title tag.
    $title = __( 'Profile' );
} else {
    $submenu_file = 'profile.php';
}

if ( current_user_can( 'edit_users' ) && ! is_user_admin() ) {
    $parent_file = 'users.php';
}
```

C. Pengujian Sistem

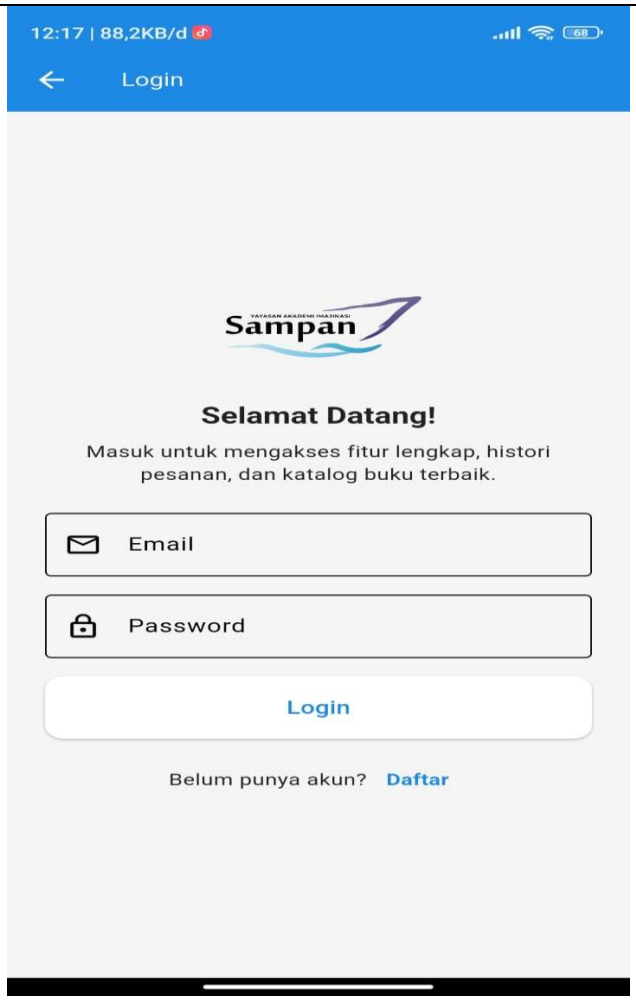
Pengujian sistem pada penelitian ini menggunakan 2 metode yaitu Black Box Testing dan White Box Testing. Berikut ini hasil pengujian menggunakan sistem tersebut.

1. *Black Box*

   Pengujian *Black Box* tidak memeriksa atau menguji logika program sebaliknya, pengujian ini berfokus pada spesifikasi aplikasi, fungsi-fungsinya, dan seberapa baik aliran fungsi sesuai dengan proses yang ingin dilakukan pengguna.

Tabel 4. 56 Tabel Pengujian *Login*

Test Faktor	Hasil	Kesimpulan
Login	✓	Berhasil, ketika Pelanggan memasukkan *Username* dan *Password*, maka akan masuk ke halaman Utama
Screen Shot		



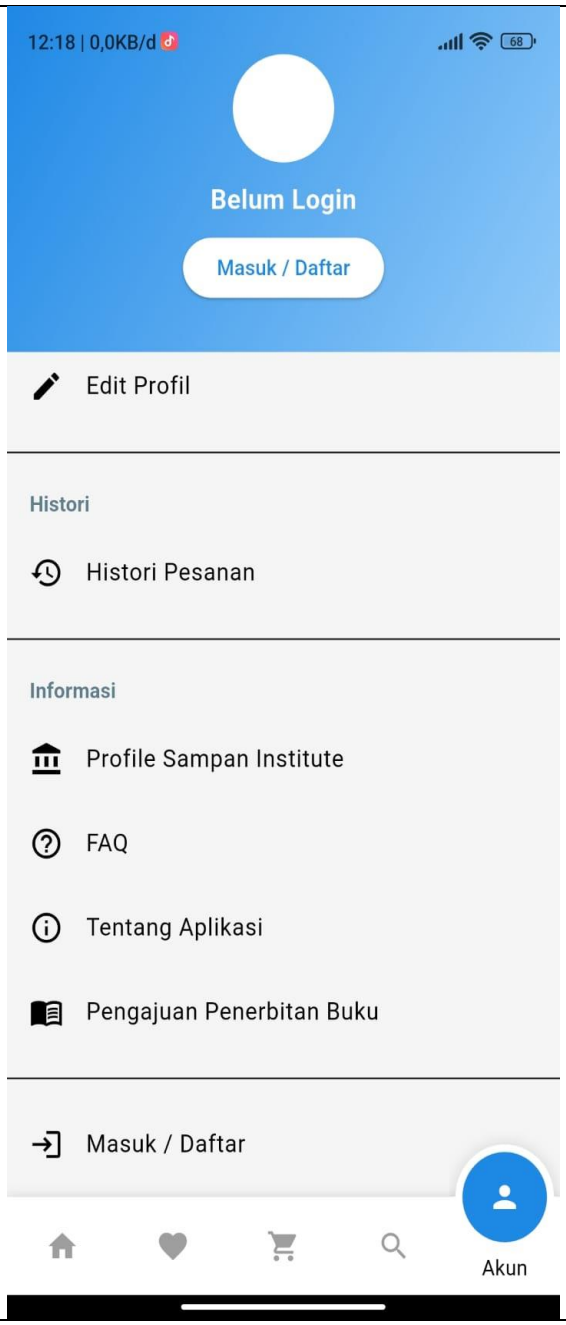
Tabel 4. 57 Tabel Pengujian Registrasi

Test Faktor	Hasil	Kesimpulan
Registrasi	✓	Berhasil, ketika Pelanggan memasukkan Nama, *Email*, Telpon, dan *Password*, maka akan masuk di halaman Login.
Screen Shot		



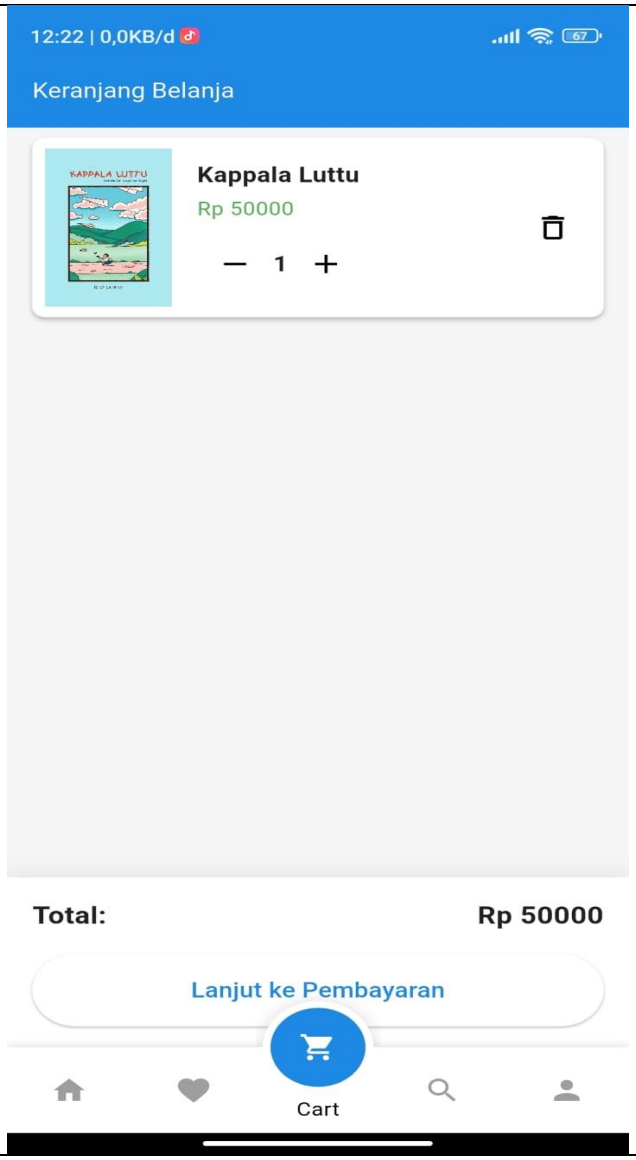
Tabel 4. 58 Tabel Pengujian Halaman Profil Pelanggan

Test Faktor	Hasil	Kesimpulan
Halaman Profil Pelanggan	✓	Berhasil, ketika Pelanggan *Login* maka akan masuk ke halaman profil pelanggan.
Screen Shot		



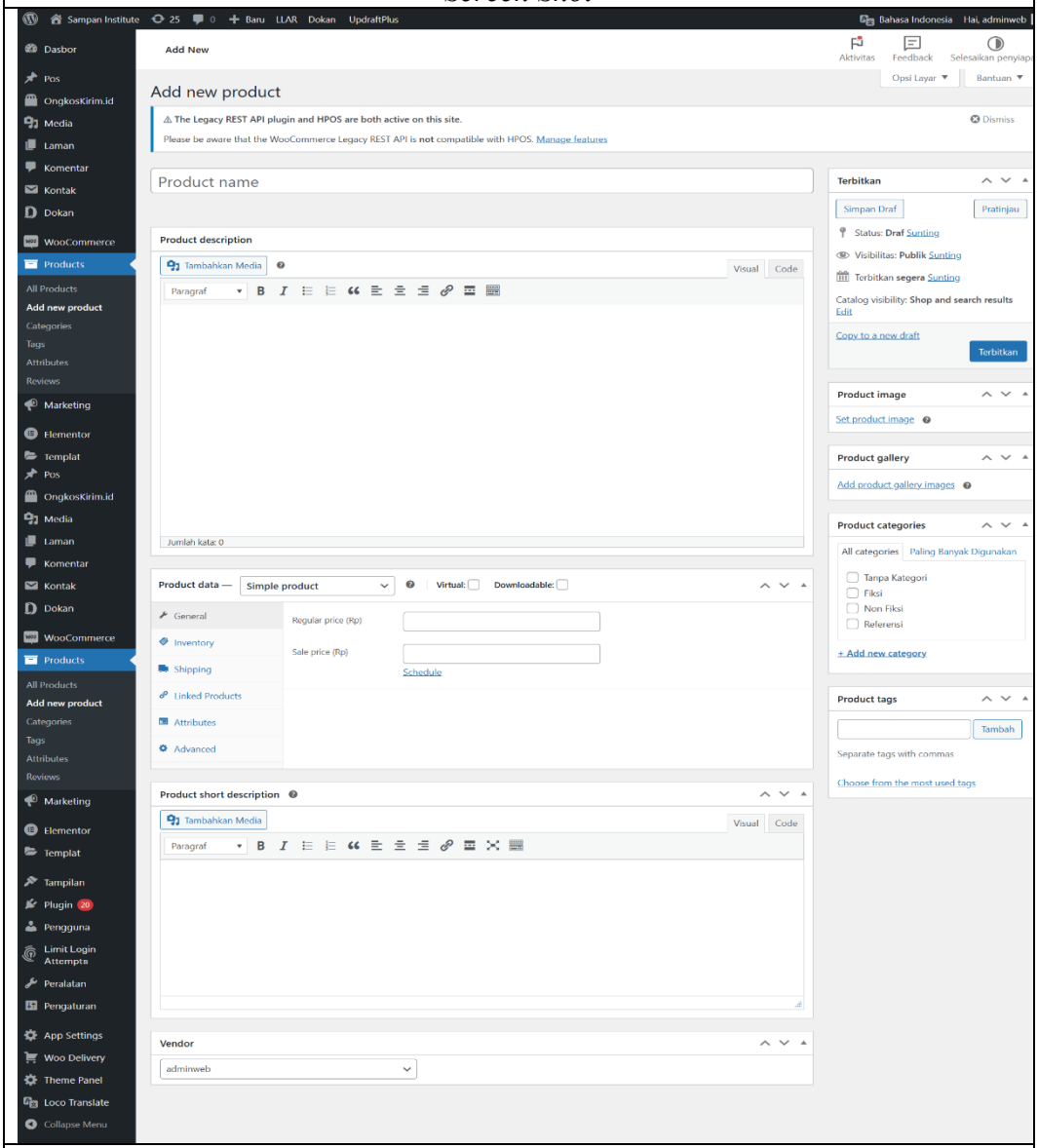
Tabel 4. 59 Tabel Pengujian Tambah ke *Cart* (Pelanggan)

Test Faktor	Hasil	Kesimpulan
Tambah ke *Cart*	✓	Berhasil, ketika Pelanggan menekan Tambah ke *Cart* maka di tampilkan sebuah pesan Berhasil di Tambahkan.
Screen Shot		



Tabel 4. 60 Tabel Pengujian Tambah Produk

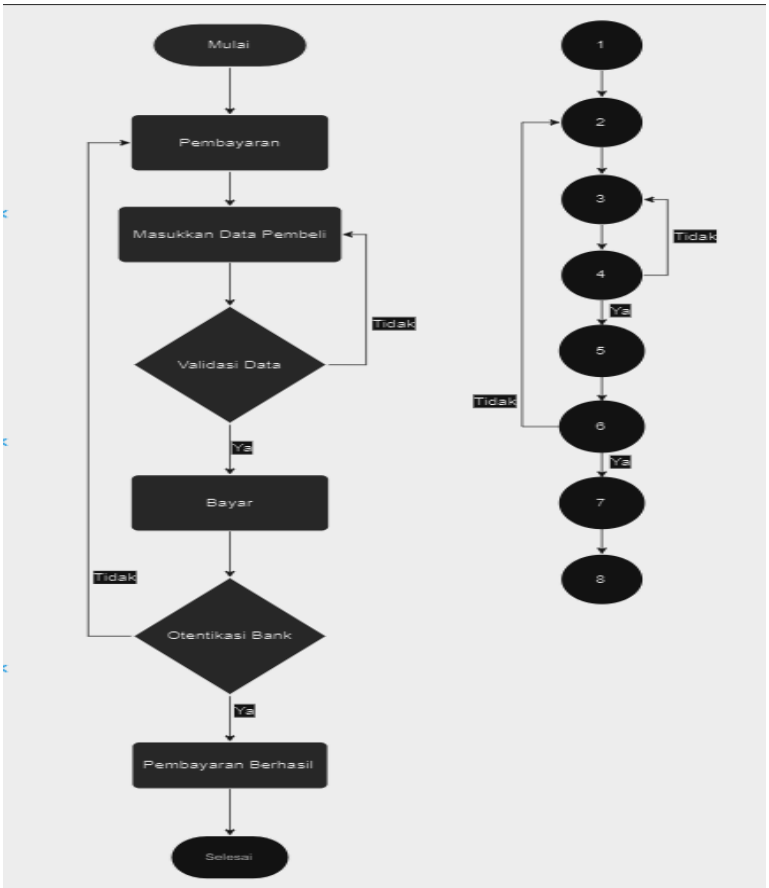
Test Faktor	Hasil	Kesimpulan
Tambah Produk	✓	Berhasil, ketika Pelanggan menekan *icon* Tambah maka Produk akan tampil di daftar Produk dan produk berhasil di tambahkan.
Screen Shot		



2. *White Box.*

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 pengujian kotak hitam, pengujian kotak putih sebenarnya menguji perangkat lunak dari sisi internal tanpa memperhatikan fungsionalitas yang serupa dengan antarmuka perangkat lunak itu sendiri.

a) *White box* menu Pembayaran Sistem Informasi Sampan Institute



Gambar 4. 21 *White box* menu Pembayaran.

Pengujian *white box* membagi pengujian menjadi beberapa kasus

pengujian dengan membaginya secara prosedural berdasarkan struktur kontrol desain perangkat lunak dan memverifikasi spesifikasi desain. Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan V(G) terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

N = 8 E = 9 R = 2

Penyelasaian :

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

= (9 – 8) + 2

= 3

Independent Path :

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

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

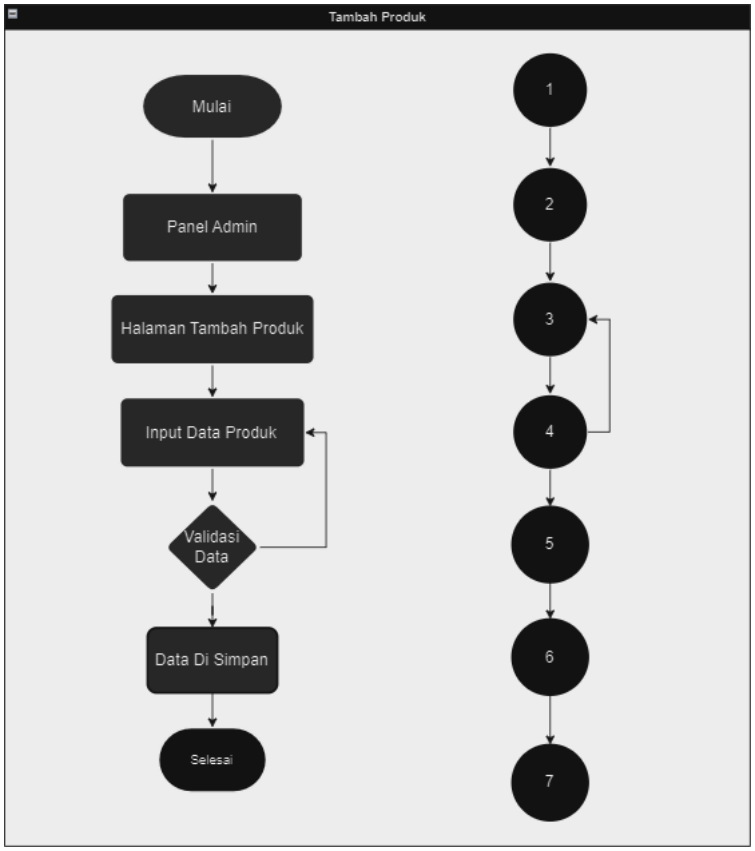
Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 2 – 3 – 4 – 5 – 6 – 7 – 8

Tabel 4. 61 *White Box* Perhitungan menu Pembayaran.

	1	2	3	4	5	6	7	8	E-1
1		1							1-1=0
2			1						1-1=0
3				1					1-1=0
4			1		1				2-1=1
5						1			1-1=0

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

b) *White box* pada menu Tambah Produk



Gambar 4. 22 *White box* pada menu Tambah Produk.

Pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan V(G) terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

N = 7 E = 7 R = 2

Penyelasaian :

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

= (7 – 7) + 2

= 2

Independent Path :

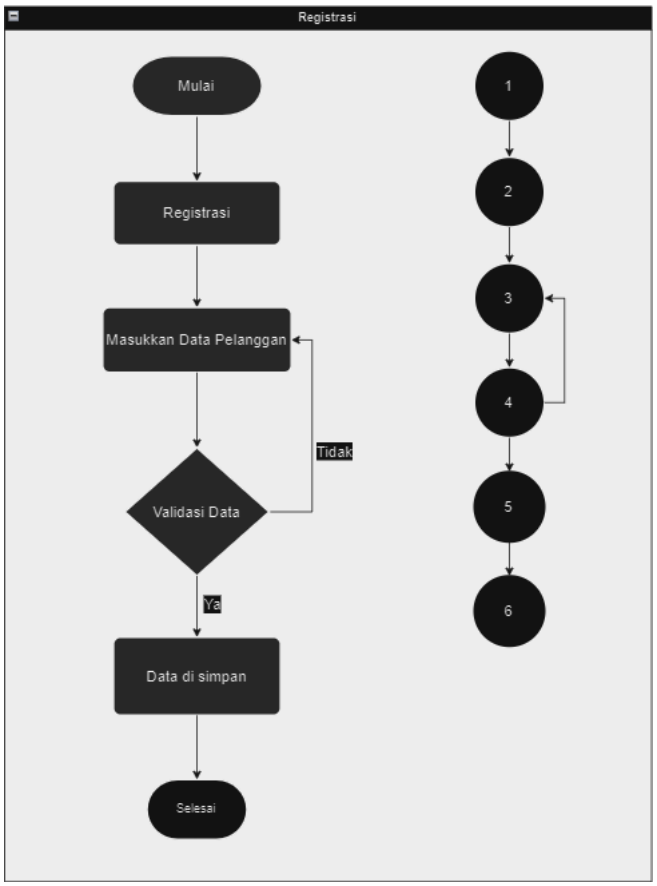
Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7

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

Tabel 4. 62 *White Box* Perhituangan menu Tambah Produk

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

c) *White box* pada menu *Registrasi*



Gambar 4. 23 *White box* pada menu Registrasi.

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

Penyelasaian :

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

= (6 – 6) + 2

= 2

Independent Path :

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 4 – 3 – 4 –5 – 6

Tabel 4. 63 *White Box* Perhituangan menu Registrasi.

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

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dan pada pembahasan bab sebelumnya, maka dapat diperoleh kesimpulan yaitu:

1. Dibangunnya suatu sistem berbasis Android, memberikan kemudahan kepada perusahaan dalam mempromosikan produknya melalui internet.
2. Sistem basis Android ini menggunakan sistem transaksi online agar dapat memudahkan Perusahaan melakukan transaksi dengan konsumen.

B. Saran

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

1. Diyakini bahwa sistem berbasis Android ini dapat dikembangkan lebih lanjut untuk penelitian di masa depan dengan memasukkan fitur-fitur terbaru tergantung pada permintaan pengguna.
2. Peneliti di masa depan harus membangun mekanisme yang memungkinkan laporan dikirim langsung ke manajer atau pemimpin melalui email.
3. Diharapkan kepada peneliti selanjutnya agar meningkatkan keamanan sistem ini untuk menghindari pencurian informasi.

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.*

Christy, F. E. (2020). Pengguna e commerce. Tempo.Co

I, Sofiani. Al, Nurhidayat. 2019. Rancang Bangun Aplikasi E-Marketplace Hasil Pertanian Berbasis Website Dengan Menggunakan Framework Codeigniter. SMS Gateway. Fakultas Teknik, Universitas Negeri Surabaya

Jogiyanto, 2015. *Analisis & Desain Sistem Informasi : Pendekatan terstruktur teori dan praktik aplikasi bisnis.Andi Offset*. Yogyakarta

KBBI, 2015. *Kamus Besar Bahasa Indonesia (KBBI). [Online] Available at: http://kbbi.web.id/di [Diakses 07 Oktober 2022]*.

Hartiwi, E. N. (2022). Aplikasi Inventori Barang Menggunakan Java dengan PHP Myadmin. Cross-borderVol. 5No. 1 Fakultas Ilmu Komputer, *Universitas Gunadarma, Indonesia.*

Maimunah. (2017). Pengertian dan Penjelasan PHP (Hypertext Preprocessor). *Pengertian dan Penjelasan PHP (Hypertext Preprocessor)*.

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

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

Triandini, E., & Suardika, I. (2012). Step by Step Desain Proyek Menggunakan UML. *Step by Step Desain Proyek Menggunakan UML*.

Wahyuddin, W., Alam, S., & Said, I. R. (2021). E-COMMERCE BUMBU MASAKAN KELOMPOK TANI KWT (KELOMPOK WANITA TANI) SETIA DESA PAKKODI KAB. ENREKANG. *Jurnal Sintaks Logika*, *1*(3), 209-214.

Wahyuddin, W., & Wafiah, A. (2022). Aplikasi Pemesanan Menu Pada Warkop Shearlock Berbasis Abdroid. *Jurnal Sintaks Logika*, *2*(3), 11-16