

BAB I

PENDAHULUAN

A. Latar Belakang

Desain grafis adalah suatu bentuk komunikasi visual yang menggunakan gambar untuk menyampaikan informasi atau pesan seefektif mungkin. Dalam desain grafis, text juga dianggap gambar karena merupakan hasil abstraksi simbol-simbol yang bisa dibunyikan. Desain grafis di terapkan dalam desain komunikasi dan fine art. Seperti jenis desain lainnya, desain grafis dapat merujuk kepada proses pembuatan, metoda merancang, produk yang dihasilkan (rancangan), ataupun disiplin ilmu yang digunakan (desain). Seni desain grafis mencakup kemampuan kognitif dan keterampilan visual, termasuk di dalamnya tipografi, ilustrasi, fotografi, pengolahan gambar dan tata letak.

Di Indonesia desain grafis sangat diminati di kalangan masyarakat yang membuat dan menikmati seni jenis ini, seperti para desainer grafis, orang-orang yang bergerak di dalam dunia Editor, Layouter, Art Director, Fotografer, Animator, maupun orang-orang yang menikmati adanya desain tersebut, seperti para kostumer yang tertarik untuk membeli suatu jasa, karena mereka tertarik dengan desain grafis yang ditampilkan oleh desainer tersebut. Namun untuk memperoleh jasa seperti Drafter, Editor, Layouter, atau semacamnya terkadang menjadi masalah tersendiri, terutama jika tidak ada wadah untuk menampung kostumer dan supplier untuk saling berinteraksi jual beli satu sama lain.

Dengan adanya E-marketplace Desain Grafis ini di harapkan bisa mengatasi permasalahan yang ada dalam bisnis desain grafis, serta menumbuhkan niat para creator-creator Indonesia khususnya daerah parepare untuk kembali berkarya dan mendapatkan hasil dari jasa yang ditawarkan tersebut.

Berdasarkan latar belakang tersebut diatas, penulis menganggap penelitian ini perlu untuk dilakukan dengan membuat penelitian yang berjudul “Perancangan E-marketplace Desain Grafis Berbasis Web” yang nantinya akan menjadi wadah bagi para konsumen untuk mencari jasa, layanan atau keahlian yang diinginkan dan juga sebaliknya produsen dapat menawarkan jasa, layanan atau keahlian yang konsumen butuhkan.

B. Rumusan Masalah

Dari latar belakang masalah yang ditemukan diatas maka pokok permasalahan yang terdapat dalam penelitian ini adalah Sebagai Berikut:

Bagaimana membuat suatu Aplikasi Perancangan *E-marketplace* Desain Grafis Berbasis Web?

C. Batasan Masalah

Batasan masalah dalam penelitian tugas akhir ini adalah sebagai berikut :

1. penelitian ini meliputi info desai grafis, pesan jasa desain
2. Aplikasi ini hanya menggunakan Bahasa pemrograman PHP
3. Pengambilan data hanya pada kota parepare.

D. Tujuan Penelitian

Tujuan penelitian ini adalah Adapun tujuan dari penelitian ini adalah merancang e-marketplace desain grafis berbasis web sehingga memudahkan setiap produsen dan konsumen dalam melakukan transaksi jual beli secara *online*.

E. Manfaat Penelitian

Adapun manfaat penelitian yaitu sebagai berikut :

1. Manfaat bagi mahasiswa

Sebagai penerapan teori yang telah di dapatkan dari bangku kuliah dan diharapkan dapat menambah pengetahuan dan wawasan tentang bagaimana cara mengetahui jasa pesan desain grafis sesuai yang diinginkan dengan berbasis *website*.

2. Manfaat bagi penulis

Menambah pengetahuan, pengalaman, serta wawasan tentang metode perkembangan dan melatih diri dalam menyelesaikan masalah untuk maju dan berkembang.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan teori

1. Pengertian Perancangan

Perancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari perancangan ialah untuk memberi gambaran yang jelas lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan harus berguna dan mudah dipahami sehingga mudah digunakan. Perancangan adalah Sebuah Proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.(Syukroni et al., 2017)

2. Aplikasi

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam Bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu sub kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2019) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut (Alda et al., 2024), Aplikasi adalah program komputer yang dibuat untuk

melakukan tugas tertentu kepada pengguna. Aplikasi adalah serangkaian aktivitas atau perintah yang dilakukan oleh komputer. Berdasarkan pengertian tersebut, dapat disimpulkan bahwa perancangan aplikasi adalah suatu model program komputer yang dibuat dengan menggunakan teknik dan prinsip tertentu agar dapat memproses dan melakukan tugas-tugas khusus. (Alda et al., 2024).

3. E-marketplace

Astuti, Wati Aris, (2023), Menjelaskan *E-marketplace* merupakan pasar online yang menyediakan kebijakan jual beli dari berbagai toko.

4. Pengertian Desain Grafis

Desain grafis adalah suatu bentuk komunikasi yang dilakukan secara visual. Desain grafis akan memanfaatkan suatu gambar sebagai media dalam penyampaian pesan. Informasi yang disampaikan di dalamnya akan dilakukan seefektif mungkin.

Desain grafis berasal dari dua kata, yaitu kata desain dan kata grafis. Desain adalah suatu metode perancangan estetika. Metode perancangan ini didasarkan pada kreativitas. Sedangkan grafis memiliki arti ilmu dari sebuah perancangan titik, maupun garis. Maka hal itu akan membentuk sebuah gambar, yang bisa memberikan informasi serta berhubungan dengan proses percetakan. Maka bisa ditarik kesimpulan bahwa desain grafis adalah ilmu yang mengandalkan sebuah kreatifitas dalam menciptakan suatu rancangan yang berbentuk gambar. Tujuan akhirnya adalah sebagai kepentingan dari percetakan.

Menurut (Putri et al., 2024), Desain grafis atau rancangan grafis merupakan proses komunikasi yang menggunakan elemen visual seperti tulisan, bentuk,

dan gambar untuk menyampaikan pesan. Hal ini ditujukan untuk sebuah produksi. Selain itu, akan dikomunikasikan juga sebagai sebuah pesan. Bentuk dari seni desain grafis nantinya akan dibuat oleh *graphic designer*.

Graphic designer adalah orang yang memiliki spesialisasi dalam menginterpretasikan pesan dalam bentuk gambar. Dengan kata lain, seorang *graphic designer* adalah seorang komunikator visual. Tugasnya untuk menciptakan konsep visual secara manual, atau dapat menggunakan *software* komputer. Mereka akan mengkomunikasikan sebuah ide untuk membuat sebuah inspirasi. Kemudian pesan tersebut akan diinformasikan melalui seni visual. Seperti tulisan, gambar dan grafik.

5. Jenis-jenis Desain Grafis

a. Drafterr

Drafter atau bisa juga dikatakan sebagai perancang memiliki fungsi untuk membuat suatu rancangan bangunan. Tugas seorang *drafter* adalah berkaitan dengan suatu industri atau arsitektur. Untuk menjadi seorang perancang, perlu ketelitian yang tinggi. Oleh sebab itu, seorang perancang harus selalu hati-hati pada pekerjaannya. Beberapa aplikasi yang bisa digunakan pada jenis desain grafis ini adalah *Archicad*, *Autocad* dan *3D Revit Architecture*.

b. Editor

Jenis desain grafis yang kedua adalah editor. Editor memiliki beberapa fungsi. Seperti membuat *banner*, membuat *cover* atau sampul, membuat logo, stiker, poster, kartu nama dan lain sebagainya. *Editor* sering dibutuhkan di dunia publikasi. Umumnya, dunia periklanan juga sering membutuhkan seorang

editor. Beberapa aplikasi yang bisa digunakan seorang *editor* antara lain seperti *corel draw*, *adobe photoshop* dan *illustrator*.

c. Layouter

Layouter adalah jenis desain grafis yang fungsinya membuat tatanan letak di majalah atau koran. Tujuannya supaya lebih enak dipandang. *Layouter* juga sering dibutuhkan di dalam dunia percetakan. Beberapa aplikasi yang bisa digunakan untuk *layouter* adalah *Adobe Page Maker*, *Ms. Publisher* dan *Adobe InDesign*.

d. Art Director

Art director memiliki fungsi untuk membuat karya seni seperti hiasan atau visual *effect*. Dalam proses pembuatannya, diperlukan kreativitas yang tinggi. Beberapa aplikasi yang bisa digunakan untuk art *director* seperti *Photo paint*, *Photoshop*, *Art creator* dan *Corel draw*.

e. Fotografer

Fotografer adalah jenis desain grafis yang memiliki fungsi untuk pengeditan foto. Dalam hal ini, perlu adanya keahlian yang tinggi di dalam dunia *fotografi*. Seperti ahli dalam menyunting foto sesuai *event* yang *perfect*. Sama seperti jenis desain grafis lainnya, fotografer juga membutuhkan kreativitas yang tinggi. Selain itu, dituntut untuk menguasai *software* aplikasi *editing*. *Fotografer* sering dibutuhkan di dalam dunia edit foto, wartawan dan *fotografi*. Beberapa aplikasi yang dapat digunakan seperti *Ieworks* dan *Photo studio*.

f. Animator

Animator adalah jenis desain grafis memiliki fungsi untuk menunjang pekerjaan di dalam bidang motion grafis, film fantasi dan sebuah iklan. Dalam proses pengerjaannya, seorang animator membutuhkan daya tahan yang tinggi. Tidak hanya itu, dibutuhkan juga kesabaran, kreativitas dan pengetahuan yang tinggi pula. Seorang animator akan sering dijumpai di dunia pertelevisian, *advertising* dan perfilman. Aplikasi yang dibutuhkan seperti 3D maya, *Adobe Flash*, *Gif Animator* dan *After Effect*.

g. Visualisator

Fungsi *visualisator* dalam desain grafis adalah memberikan sebuah gambaran di suatu karya produk. Produk yang dibuat umumnya berbentuk nyata atau 3 dimensi. Setiap orang yang ingin menggunakan jenis desain grafis ini harus memiliki kemampuan otak bagian kanan yang tinggi. Dalam menguasai aplikasi desain grafis, visualisator akan sering dibutuhkan dalam dunia visualisasi produk serta presentasi produk. Beberapa aplikasi yang dapat digunakan yaitu *Digital Clay*, *Autocad* dan *3D Max*.

h. Video Editor

Fungsi dari seorang video editor adalah untuk mengedit video atau film sekaligus merangkap sebagai seorang video *shooter*. Dalam melakukannya, diperlukan imajinasi yang tinggi. Video editor sering dibutuhkan di dalam dunia industri musik dan dunia perfilman. Aplikasi yang bisa digunakan seperti *After Effect*, *Sony Vegas* dan *Video Studio*.

i. Integrater Designer

Fungsinya adalah untuk mengedit video dan film, yang sekaligus akan memerankan sebagai seorang video *shooter*. Aplikasi yang dapat digunakan seperti *Premiere Pro*.

B. Sekilas Tentang Aplikasi Yang Digunakan

1. Web

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses dimana saja dengan biaya relatif murah. Web merupakan bentuk implementasi dari bahasa pemrograman web (*web programming*). Sejarah perkembangan bahasa pemrograman web diawali dengan munculnya HTML (*Hypertext Markup Language*), yang kemudian dikembangkan dengan munculnya CSS (*Cascading Style Sheet*) yang bertujuan untuk memperindah tampilan *website*. (Granitzer et al., 2024)

Bahasa pemrograman saat ini sudah sangat berkembang dengan berbagai kemudahan dalam penyajian dan *interface* yang lebih *friendly used*. Penyajian yang baik dari suatu Bahasa pemrograman akan menghasilkan sebuah web yang dinamis sehingga pengunjung akan lebih mudah mendapatkan informasi yang dibutuhkan. (Deng et al., 2023). Beberapa bahasa pemrograman web berbasis server (*Server Side*) mulai muncul dan dikembangkan oleh beberapa perusahaan perangkat lunak, seperti : ASP (*Active Server Pages*) oleh *Microsoft*, JPS (*Java Server Pages*) oleh *Sun Microsystem*, dan CGI (*Common gateway Interface*) oleh perusahaan UNIX yang sekarang diadopsi oleh NCSA (*National Central For Supercomputing Application*) web server. (Vallabhaneni et al., 2024).

2. Internet (Interconneted Network)

Internet adalah sekumpulan jaringan komputer yang saling terhubung secara fisik dan memiliki kemampuan untuk membaca dan menguraikan protokol komunikasi tertentu yang disebut *Internet Protocol (IP)* dan *Transmission Control Protocol (TCP)*. Protocol adalah spesifikasi sederhana mengenai bagaimana komputer saling bertukar informasi. (WIDIARTO, 2023).

3. WWW (World Wide Web)

World Wide Web atau WWW atau juga dikenal dengan WEB adalah suatu ruang informasi yang dipakai oleh pengenal global yang disebut pengidentifikasi sumber seragam untuk mengenal pasti sumber daya berguna. WWW sering dianggap sama dengan internet. (Birkle et al., 2020).

WWW merupakan kumpulan web dari seluruh dunia yang mempunyai kegunaan untuk menyediakan data dan informasi untuk dapat digunakan bersama. WWW adalah bagian yang paling menarik dari internet. Melalui web, para pengguna dapat mengakses informasi- informasi yang tidak hanya berupa teks tetapi bisa juga berupa gambar, suara, video dan animasi. WWW adalah suatu program yang ditemukan oleh Tim burners-Lee pada tahun 1991. Awalnya Berners Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya. Dengan program itu menciptakan jaringan yang manautkan berbagai arsip sehingga memudahkan pencairan informasi yang dibutuhkan. Inilah yang kelak menjadi dasar sebuah perkembangan pesat yang dikenal sebagai WWW. (Ramadhan Arif. 2006).

4. HTTP (Hypertext Tranfer Protocol)

HTTP (*Hypertext Transfer Protocol*) adalah protocol yang dipergunakan untuk mentransfer dokumen dalam *World Wide Web* (WWW). *protocol* ini adalah *protocol* ringan tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen. Pengembangan HTTP dikoordinasikan oleh Konsorsium *World Wide Web* (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC yang menjelaskan HTTP/1,1, versi HTTP yang digunakan umum sekarang ini. HTTP adalah sebuah *protocol* meminta / menjawab antara *client* dan server.

Sebuah client HTTP seperti web *browser*, biasanya melalui permintaan dengan membuat hubungan TCP/IP ke *port* tertentu. Sebuah server HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan di ikuti dengan pesan MIME yang memiliki beberapa informasi data tertentu. (Deka M Wildan, 2014).

5. URL (Uniform Resource Locator)

URL (*Uniform Resource Locator*) merupakan suatu *pathname* untuk mengidentifikasi sebuah dokumen di web. Dibandingkan dengan *logical pathname* yang membuat nama direktori dan subdirektori yang menunjuk ke satu file di sebuah komputer, cakupan URL lebih luas lagi. Didalam URL dimuat lokasi atau nama mesin komputer yang diminta, nama *file* dari dokumen itu (termasuk *logical pathnamanya*), serta protoko untuk mengaksesnya. (Santoso, Harip. 2010)

6. HTML (Hypertext Markup Language)

HTML (*HyperText Markup Language*) Adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan penformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak yang digunakan di dunia penerbitan dan pencetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Concorium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-Lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). (Santoso,Harip. 2010)

HTML juga bisa berupa dasar atau tulang punggung dalam pembuatan web. Dalam setiap file HTML akan selalu di dahului dengan tag (HTML). Tag ini akan membuat web browser mencoba untuk membaca file yang ada didalam tag tersebut untuk menampilkan dalam web *browser*. (Santoso,Harip. 2010)

7. PHP (Personal Home Page)

PHP (*Personal Home Page*) merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya semua sintaks dan perintah program yang ditulis akan sepenuhnya dijalankan oleh server, tetapi dapat disertakan pada halaman HTML biasa. Pada umumnya, semua aplikasi yang dibangun menggunakan PHP akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan pada server. Pada prinsipnya, server akan bekerja apabila ada permintaan dari client. Dalam hal ini, client menggunakan kode-kode PHP akan mengirim perintah ke server ketika menggunakan PHP sebagai *serverside embedded script language*. (Alan Nur Aditya, 2011)

8. MySQL

Sutarman, (2007) MySQL merupakan salah satu program yang dapat digunakan sebagai database, dan merupakan salah satu *software* untuk database server yang banyak digunakan. MySQL bersifat *Open Source* dan menggunakan SQL. MySQL bisa dijalankan diberbagai platform misalnya *Windows, linux* dan lain sebagainya.

MySQL merupakan salah satu perangkat lunak sistem manajemen basis data (*database management system*) atau DBMS yang menggunakan perintah standar SQL (*Structured Query Language*). Dimana MySQL mampu untuk melakukan banyak eksekusi perintah *query* dalam suatu permintaan (*multithread*), baik itu mengirim dan menerima data. MySQL juga *multi-User* dalam arti dapat

dipergunakan oleh banyak pengguna dalam waktu bersamaan. Dengan sekitar enam juta Instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia dalam perangkat lunak gratis dibawah lisensi *GNU General Public Lisence* (GPL) dan juga menjual dalam lisensi komersial untuk keperluan jika penggunanya tidak cocok menggunakan lisensi GPL. (Nugroho, Bunafit. 2014)

9. *Adobe Dreamweaver CS6*

Adobe Dreamweaver adalah sebuah HTML (*Hyper Text Markup Language*) editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web. Saat ini terdapat *software* dari kelompok adobe yang belakangan banyak digunakan untuk mendesain suatu situs web. Versi terbaru dari *adobe dreamweaver* saat ini adalah *Dreamweaver CS6*. Pada *Dreamweaver CS6*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk mendesain web saja tetapi juga untuk menyunting kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman web, antara lain HTML, PHP, CSS dan sebagainya.

Dreamweaver merupakan *software* utama yang digunakan oleh web desainer maupun web pemrograman dalam mengembangkan suatu situs web. Hal ini disebabkan ruang kerja, fasilitas dan kemampuan *dreamweaver* yang mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu situs web. (Madcoms Madiun, 2012)

10. **PhpMyAdmin**

PhpMyAdmin adalah perangkat lunak yang ditulis dalam bahasa pemrograman *PHP* digunakan untuk menangani *Administrasi MySQL* dengan menggunakan protokol *WWW* secara langsung dari *browser*. *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, bidang *fields*, relasi *relations*, indeks, pengguna *Users*, perijinan *permissions*, dan lain-lain. Dengan menggunakan *phpMyAdmin* diharapkan akan memudahkan dalam memajemen database *MySQL*. (Madcoms Madiun, 2012)

11. **XAMPP (Cross-Platform (x), Apache, MySQL dan Perl)**

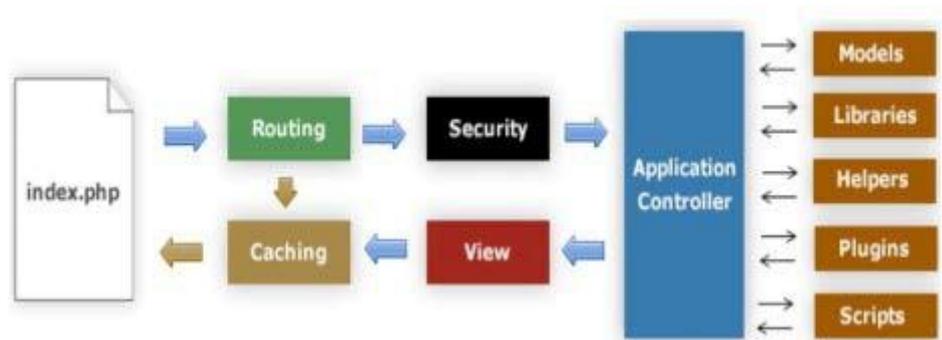
XAMPP adalah perangkat lunak besar, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP server, *MySQL* database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama XAMPP merupakan singkatan dari X (tempat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam GNU *general Public license* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. (Madcoms Madiun, 2012).

12. **Framework Codeigniter**

Menurut (Madcoms Madiun, 2012) “*CodeIgniter* adalah sebuah *framework* *PHP* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis *PHP* dibanding jika menulis semua kode program dari awal”. Didalam *CI* ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrograman dalam mengembangkan aplikasi. File

dokumentasi ini disertakan secara langsung pada saat Pengunduhan paket *framework CI*.

CodeIgniter menggunakan konsep MVC (*Model View Controller*). Menurut Akhmad Sofwan (2003), konsep MVC adalah konsep pemisahan antara logika dengan tampilan dan database. Manfaat dari konsep ini adalah membuat pengodean logika lebih *simple*, karena sudah dipisahkan dengan kode untuk tampilan dan membuat *programmer* dapat bekerja secara terpisah dengan desainer. *Programmer* mengerjakan logika, sedangkan desainer berkeutatan dengan desain dan tampilan. Model (M) berisi kode penghubung *database*, View (V) berisi kode desain tampilan, dan *Controller* (C) berisi kode logika.



Gambar 2. 1 *CodeIgniter*

C. Perancangan basis data

Menurut (Jogiyanto Hartono, 2005:700). Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasi. Sedangkan menurut penadapat saya Basis Data adalah data data yang tersimpan dalam komputer dan masih harus di proses agar saling

berhubungan satu sama lain. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakai. Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

1. Entity Relationship Diagram (ERD)

Menurut Andi, ERD berguna untuk mengorganisasi data-data (*hanya data*) dalam hubungannya dengan data-data yang lain. Sedangkan menurut saya ERD berfungsi untuk mengelompokkan data- data agar terjadi keselarasan hubungan antar data lainnya. ERD mempunyai koleksi objek-objek dasar yang dinamakan entitas serta hubungan (*relationship*) antara entitas –entitas itu.

Entitas adalah sesuatu atau objek pada dunia nyata yang dapat dibedakan satu terhadap yang lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan. Sedangkan *Relationship* adalah hubungan antara beberapa entitas.

2. Normalisasi

Menurut Andi, Normalisasi sebagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tak memiliki masalah tersebut. Sedangkan Normalisasi menurut pendapat saya adalah proses pengolahan data yang masih salah dan memiliki data ganda untuk kemudian di perbaiki sehingga menghasilkan relasi-relasi yang lebih baik dari sebelumnya. Normalisasi dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal. Bentuk normal adalah keadaan relasi

yang dihasilkan dengan menerapkan aturan sederhana yang berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan sebagai berikut.

a. Bentuk normal pertama.

Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*Multivalued attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (*mungkin saja nilai null*) pada perpotongan setiap baris dan kolom.

b. Bentuk normal kedua.

Suatu relasi adalah dalam bentuk normal kedua jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh pada kunci primer.

c. Bentuk normal ketiga.

Relasi adalah dalam bentuk normal ke tiga adalah jika berada dalam bentuk normal kedua dan tidak dijumpai kebergantungan *transitif*. Kebergantungan *transitif* dalam suatu relasi adalah kebergantungan fungsional antara 2 atau lebih atribut bukan kunci. Ketika relasi memiliki lebih dari 1 kunci kandidat, anomali-anomali seperti kita bahas di atas masih mungkin dijumpai meskipun relasi yang bersangkutan sudah dalam bentuk normal ke tiga.

3. Tabel Relasi

Tabel relasi adalah tabel hubungan antara suatu himpunan entitas dengan entitas yang lainnya.

4. Kunci

Kunci (*key*) merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan entitas yang lainnya dalam suatu himpunan entitas.

D. Tahap-tahap Perancangan Sistem

1. Use Case

(Pranoto et al., 2024), *Use case diagram* adalah abstraksi dari interaksi antara sistem dan *actor*. Use case bekerja mendeskripsikan tipe interaksi antara *User* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

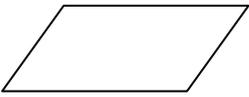
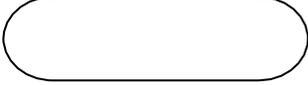
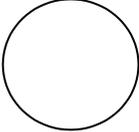
2. Kamus Data

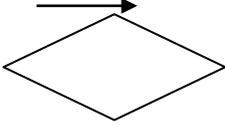
Kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan informasi yang mengalir pada sistem dengan lengkap dari suatu sistem informasi. Kamus data di buat berdasarkan arus data yang mengalir pada *Data Flow Diagram* (DFD). Kamus data merupakan tempat penyimpanan dari aliran-aliran data, *file-file* dan proses-proses dalam sebuah sistem. Bagian ini menjelaskan secara detail proses-proses yang terjadi disetiap proses, *file* dan struktur data mengenai model sistem yang digambarkan. Kamus data digunakan untuk menjelaskan semua data yang mengalir atau digunakan dalam sistem, yaitu mengenai arus data yang masuk ke dalam sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

3. Bagan Alir Program

Bagan alir program (*program Flowchart*) merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir sistem (Bagan alir program dibuat dengan menggunakan symbol-simbol sebagai berikut :

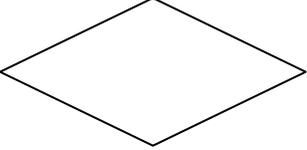
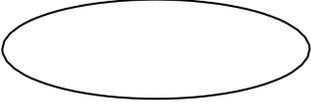
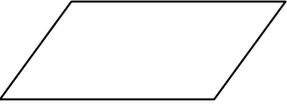
Tabel 2. 1 Simbol *Flowchart*

Simbol	Keterangan
	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i>
	Simbol proses, digunakan untuk mewakili suatu proses
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus dihalaman yang masih sama
	Simbol garis alir, menunjukkan arus dari proses

Simbol	Keterangan
	Simbol keputusan, digunakan untuk suatu penyelesaian kondisi didalam program

4. Entity Relationship Diagram

Tabel 2. 2 Simbol *Entity Relationship Diagram*

NOTASI	KETERANGAN
	Entitas, yaitu kumpulan dari objek yang diidentifikasi secara unik
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain : satu ke satu, satu ke banyak dan banyak ke banyak
	Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi.
	<i>Input/Output</i> data, yaitu proses <i>input/output</i> data, parameter, informasi.

Entity Relationship Diagram (ERD) adalah adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan antar data. ERD merupakan suatu model untuk

menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antardata, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

a. Entity

Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*.

c. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2) Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3) Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

5. UML (*Unified Modelling Language*)

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

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

a. Sesuatu (*things*)

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

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

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

b. Relasi (*relationship*)

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

1. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengaruhi elemen yang bergantung padanya.
2. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
3. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
4. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

c. Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

d. Diagram *use case*

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

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

Tabel 2. 3 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

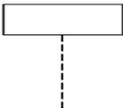
NO	GAMBAR	NAMA	KETERANGAN
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2. 4 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

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

Tabel 2. 5 Simbol *Sequence Diagram*

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

Tabel 2. 6 Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan

NO	GAMBAR	NAMA	KETERANGAN
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 7 Simbol *Activity Diagram*

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

E. Penelitian Terdahulu

Berbagai penelitian sebelumnya merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Dalam hal ini,

fokus penelitian terdahulu yang dijadikan acuan adalah terkait dengan masalah teknologi informasi. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui internet.

1. Ahmad Sidqi (2020) program studi teknik informatika – program sarjana fakultas teknologi industry Universitas islam indonesia, Melakukan Penelitian Yang Berjudul **“pengaruh penggunaan media desain grafis berbasis adobe photoshop dalam meningkatkan keterampilan editing foto siswa kelas x di smk”** Proses pembelajaran tahfiz di Madrasah Tsanawiyah Sunan Pandanaran (MTs SPA), pembukuan hasil perkembangan hafalan santri dilakukan secara manual oleh guru. Belum adanya sistem pencatatan yang baik menyebabkan guru terhambat dalam memperoleh informasi santri. Guru hendaknya dapat mengelola informasi seperti presensi santri dan berapa jumlah hafalan santri. Selain itu, tidak adanya laporan terhadap perkembangan santri membuat kepala sekolah kesulitan dalam mengetahui hasil pembelajaran santri secara menyeluruh. Pembuatan sistem ini menggunakan beberapa tahapan. Tahapan awal yaitu analisis proses bisnis yang meliputi pengumpulan data, kemudian tahapan kedua yaitu analisis kebutuhan sistem yang meliputi analisis kebutuhan *input*, analisis kebutuhan proses, dan analisis kebutuhan *otput*. Tahapan ketiga pengimplementasian sistem menggunakan framework CI, tahapan keempat yaitu pengujian yang telah dilakukan menggunakan pengujian kuesioner dan *User Acceptance Test (UAT)*.
2. Ridwan & Mutia, (2022) Fakultas Ilmu Komputer, Universitas Mercu Buana, dalam penelitiannya yang berjudul **“Pengembangan Aplikasi Web dan**

Mobile Pada dinas tenaga kerja ” Kabupaten Pidie merupakan salah satu lembaga pendidikan yang memiliki program tahfidz. Saat ini pencatatan hafalan Al-Qur’an setiap siswa dilakukan pada satu lembar kertas. Adanya teknologi informasi menuntut suatu lembaga pendidikan untuk mengimplementasikan aplikasi TI dengan baik. Penelitian ini bertujuan untuk membangun sistem monitoring dan evaluasi tahfidz qur’an yang baik dalam hal penyajian data kepada MUQ Kabupaten Pidie dan orang tua. Data informasi pada sistem aplikasi web dan mobile yang diusulkan ini berupa data informasi berupa tabel, teks, grafik dan laporan hasil hafalan. Metode penelitian menggunakan pendekatan penelitian dan pengembangan dengan model pengembangan aplikasi (waterfall). Aplikasi yang digunakan menggunakan konsep web service yang menghubungkan antara web dan aplikasi mobile. Hasil penelitian menunjukkan bahwa pengembangan sistem monitoring dan evaluasi berupa aplikasi web dan mobile untuk pengelolaan data pengembangan tahfidz dengan menyajikan data dalam bentuk teks, grafik dan tabel. Hal ini secara efektif memberikan fungsi pemantauan dan evaluasi yang lebih mudah, cepat dan efisien. Hasil uji efektivitas penerapan menunjukkan nilai rata-rata persentase 19 soal adalah 96%. Berdasarkan hasil kuesioner, nilai rata-ratanya diatas 70%. Secara umum, aplikasi ini sangat membantu para orang tua dalam memantau perkembangan hafalan anaknya.”

3. Syukroni et al., (2017), penelitian yang berjudul **“Rancang Bangun Knowledge Management Sistem Berbasis Web Pada Madrasah Muallimin Al-Islamiyah Uteran Geger Madiun”** Sistem Manajemen Pengetahuan

(*Knowledge Management System*) sangat diperlukan untuk memudahkan dalam pengelolaan pengetahuan, penyimpanan pengetahuan, dan penyebaran yang efektif dan efisien agar terjadi pertukaran pengetahuan antar guru, sehingga dapat mengembangkan kompetensi guru dan meningkatkan pelayanan pendidikan

4. Fulka Juhdan Hanief (2020), Fakultas Teknik dan Ilmu Komputer unikom dalam penelitiannya yang berjudul “**Rancang Bangun Sistem Informasi Monitoring desain grafis**” Ponpes Uswatun Hasanah saat ini melakukan monitoring kegiatan harian para santri dengan melakukan rapat evaluasi bulanan rutin yang dihadiri oleh seluruh QST, IST, para guru(pembimbing) dan kepala sekolah. Yang dibahas pada rapat evaluasi bulanan ini salah satunya adalah perkembangan akademik santri, baik itu perkembangan santri yang berprestasi maupun yang perkembangan akademiknya bermasalah. Informasi perkembangan akademik santri ini meliputi kehadiran santri disetiap kegiatan, kemampuan bacaan Al-Quran, hapalan Qur'an, kemampuan bahasa arab/inggris dan nilai raport santri. Informasi tersebut dituliskan secara deskriptif pada kartu evaluasi bulanan yang dikumpulkan oleh semua guru kepada kepala sekolah. Kemudian kepala sekolah melaporkan perkembangan akademik santri yang diperoleh tersebut kepada ketua yayasan. Dalam Proses monitoring yang berjalan saat ini ditemukan beberapa kendala dan keterbatasan, diantaranya yaitu belum adanya proses perhitungan yang mengeloladata nilai-nilai disetiap aspek kegiatan pondok dan belum adanya system yang terintegrasi yang memudahkan para guru dalam mengelola data-

data santri sehingga guru kesulitan dalam monitoring kemampuan setiap santrinya. Informasi monitoring perkembangan santri yang didapatkan kepala sekolah melalui kartu evaluasi bulanan hanya berisi tentang informasi umum perkembangan santri sehingga menyulitkan kepala sekolah untuk membuat laporan monitoring perkembangan santri. Hal ini menyebabkan ketua yayasan pun tidak bisa mengetahui bagaimana informasi monitoring perkembangan santri karena belum ada sistem pelaporan yang jelas

BAB III

METODOLOGI PENELITIAN

A. Waktu Dan Lokasi Penelitian

Pelaksanaan penelitian dilakukan pada bulan Mei 2024 dan lokasi Kota Parepare.

B. Jenis Penelitian

Penulis dalam hal ini menggunakan metode penelitian Deskriptif yaitu kegiatan yang dilakukan dengan cara mengumpulkan data secara langsung dari objek penelitian melalui wawancara, pengamatan langsung dan pengumpulan dokumen.

C. Metode Pengumpulan Data

Di penelitian ini, penulis memperoleh data dengan metode sebagai berikut:

1. Observasi

Dilakukan dengan mengadakan penelitian langsung dengan kantor yang terkait untuk mengumpulkan informasi dan data yang dibutuhkan dalam menunjang permasalahan.

2. Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dengan responden. Komunikasi ini berlangsung dalam bentuk tanya jawab dalam hubungan tata muka guna memperoleh data langsung

3. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data dengan menggunakan buku-buku, laporan-laporan atau bacaan lain sebagai bahan referensi dalam penulisan laporan dan pembuatan sistem.

D. Alat Dan Bahan Penelitian

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan tersebut. Alat dan bahan yang diperlukan antara lain:

1. Perangkat keras

Perangkat keras yang digunakan dalam Aplikasi Perancangan E-Marketplace Desain Grafis Berbasis Web adalah sebuah laptop dengan spesifikasi minimal yang dapat dilihat pada tabel 4 berikut:

Tabel 3. 1 Spesifikasi Minimal Perangkat Keras

Jenis	Spesifikasi
Laptop	Asus
Processor	Core-i3-3217U 1,8 GHz
Memory	4 GB
Hardisk	500 GB

2. Perangkat Lunak

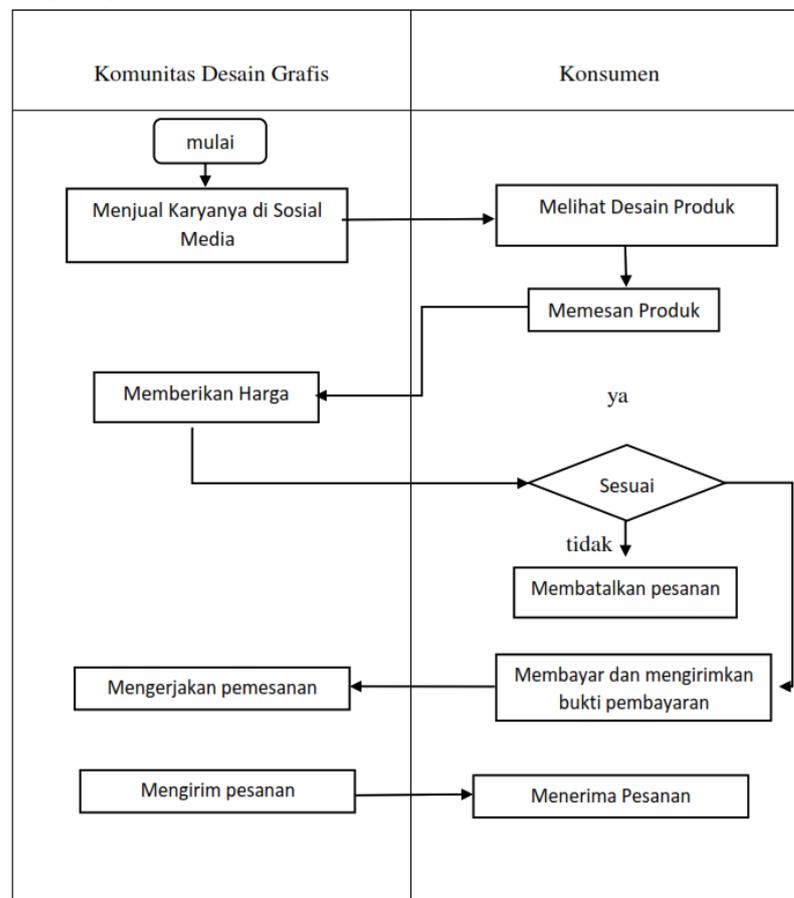
Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada Tabel 5 berikut :

Tabel 3. 2 Spesifikasi Minimal Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	Windows 11
Bahasa Pemrograman	PHP
Database	MySql
Tools	Notepad++, Netbeans, XAMPP

E. Desain Sistem

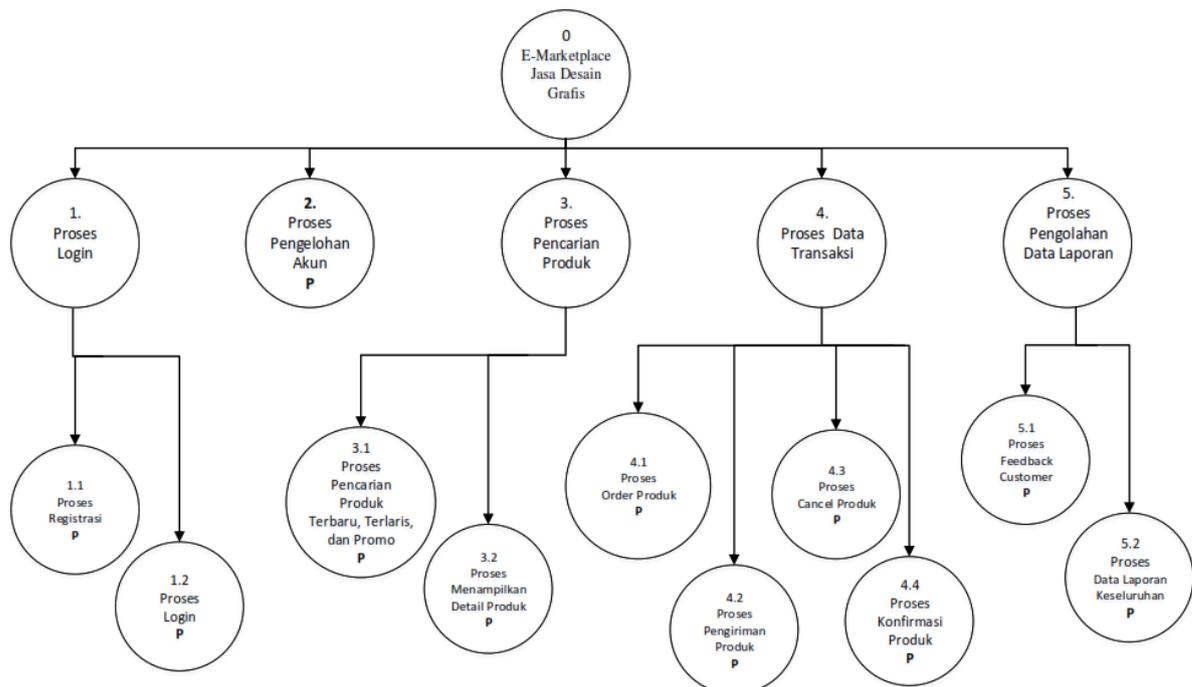
1. Sistem yang berjalan

**Gambar 3. 1** Activity plan diagram

Adapun sistem yang sedang berjalan yaitu konsumen mencari informasi tentang penyedia jasa desain grafis dengan surfing melalui web sosial media, atau bertatap

muka langsung dengan produsen membahas tentang cara pemesanan dan layanan seperti apa yang diinginkan, adapun cara kedua yaitu surfing melalui website sosial media mencari produk-produk desain yang diinginkan kemudian berkomunikasi menggunakan fitur chat mengenai harga yang ditawarkan apabila disetujui maka pesanan akan dilanjutkan apabila tidak akan ditolak, namun cara tersebut di atas masih kurang maksimal dari segi pembayaran yang memungkinkan terjadinya penipuan salah satu pihak contoh penipuan yang banyak terjadi adalah banyaknya produsen yang tidak mengirim jasa desain yang telah dipesan setelah menerima sejumlah uang yang di kirim oleh konsumen, atau sebaliknya.

2. Sistem yang Diusulkan



Adapun penjelasan dari proses sistem yang diusulkan yaitu kostumer mencari jasa yang diinginkan lalu di proses oleh *Admin* dan disampaikan kepada seller. Kemudian seller meninjau kembali apakah permintaan kostumer sesuai dengan

yang ditawarkan oleh seller. Jika sesuai pelayanan akan diterima dan jika tidak pemesanan akan ditolak karena tidak memenuhi persyaratan yang diajukan oleh pembeli. Contoh penolakan yaitu, tidak sesuainya permintaan kostumer dengan jasa yang ditawarkan oleh produsen. Adapun tugas dari *Admin* yaitu meninjau kembali pesanan yang telah dipesan oleh kostumer dan melakukan pengiriman pemberitahuan kepada produsen apabila ingin membatalkan pesanan yang telah dilakukan.

BAB IV

PEMBAHASAN DAN HASIL

A. Analisis Aliran Data UML

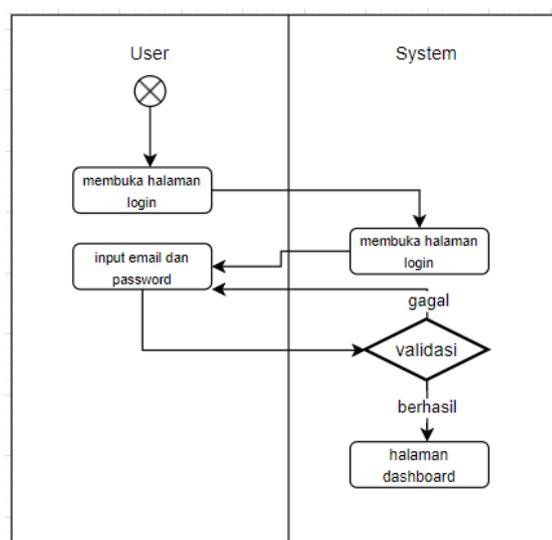
Dalam bagian ini, akan dibahas dua diagram UML yang menggambarkan aliran data dan interaksi dalam aplikasi APLIKASI PERANCANGAN E-MARKETPLACE DESAIN GRAFIS BERBASIS WEB, yaitu *Activity diagram* dan *Sequence diagram*.

1. Activity Diagram

Activity diagram digunakan untuk memodelkan aliran kerja atau aliran kontrol dari sebuah proses bisnis atau *use case*. Diagram ini menunjukkan aktivitas-aktivitas yang terjadi dalam sistem dan urutan eksekusinya.

a. Activity diagram login

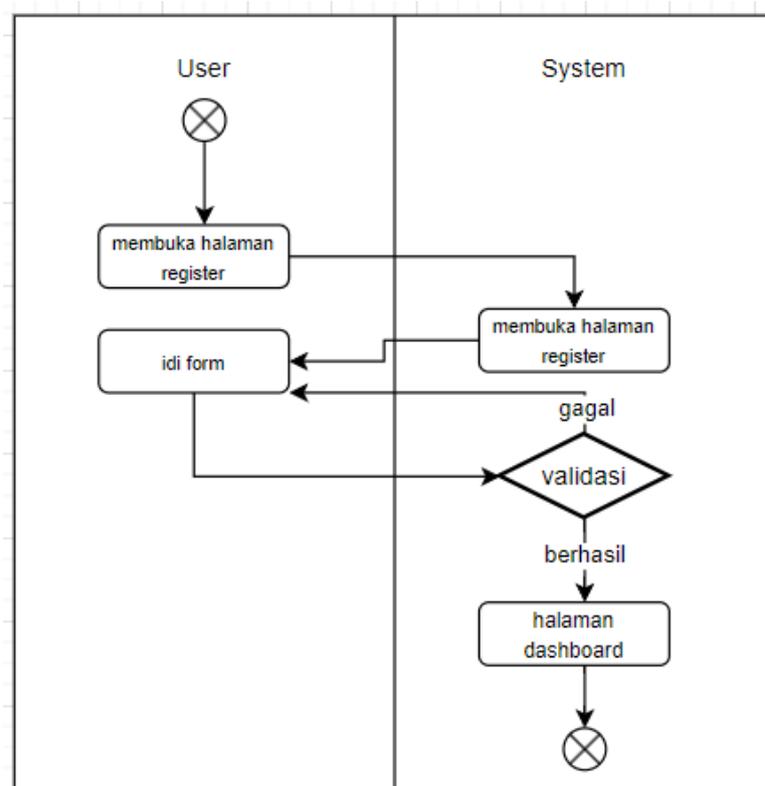
1) Activity diagram login



Gambar 4. 1 Activity diagram login

Pada gambar 4. 1 menjelaskan cara masuk login. *pengguna* harus mengunjungi website terlebih dahulu. Setelah itu, formulir login akan muncul di layar, mengharuskan *pengguna* memasukkan alamat email dan kata sandi pengguna. Setelah itu, sistem akan memvalidasi data dan masuk ke halaman *dashboard* jika semuanya sudah benar, kembali ke halaman login jika tidak.

2) Activity diagram Register

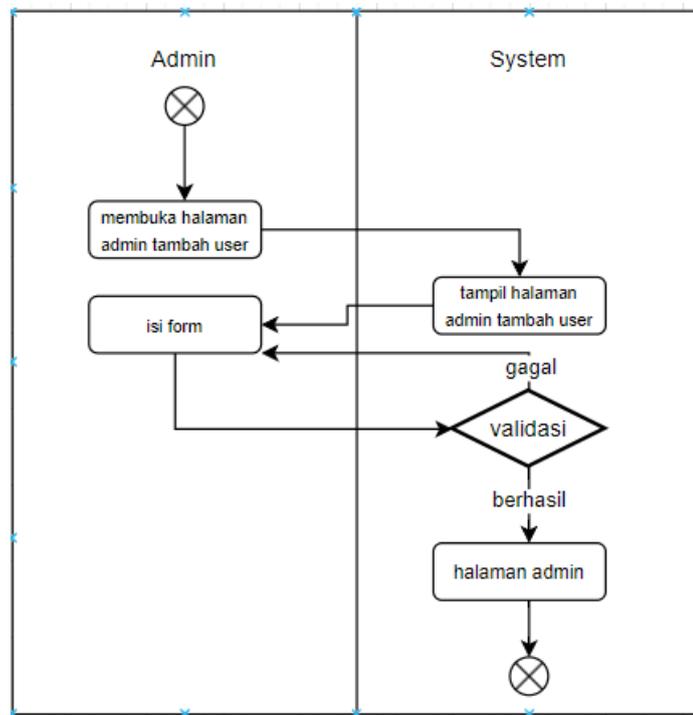


Gambar 4. 2 Activity diagram Register

Pada gambar 4. 2 menjelaskan cara Register. *pengguna* harus mengunjungi website terlebih dahulu. Setelah itu, formulir Register akan muncul di layar, mengharuskan *pengguna* mengisi formlulir. Setelah itu, sistem akan memvalidasi dan masuk ke halaman *dashboard* jika semuanya sudah benar, kembali ke halaman login jika tidak.

b. Activity Diagram *Admin*

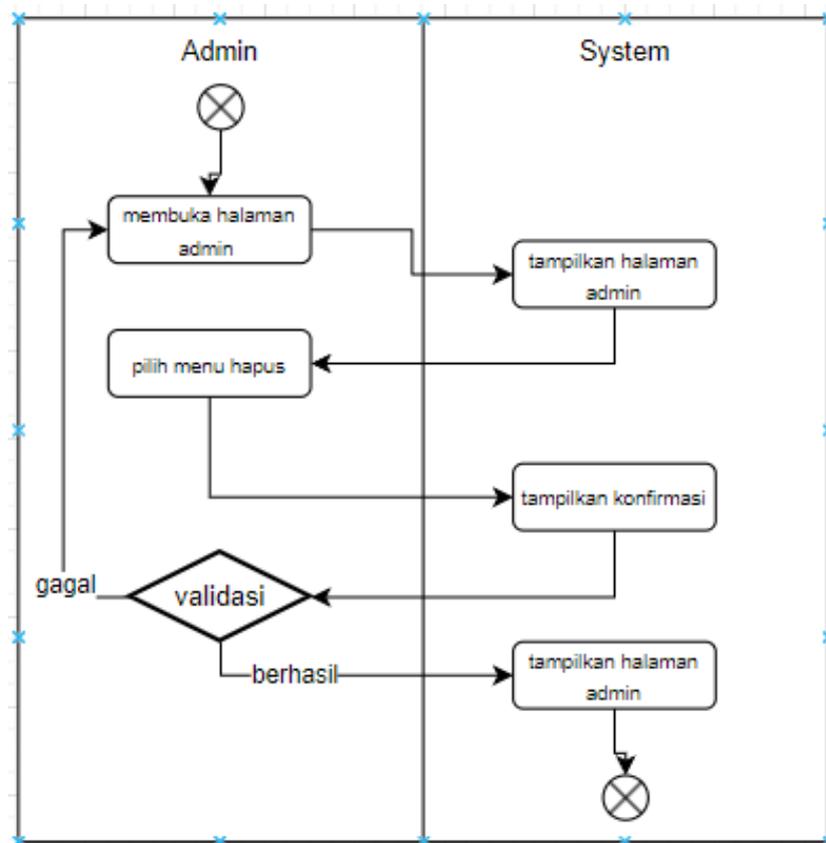
1) Activity diagram tambah pengguna



Gambar 4. 3 Activity diagram tambah pengguna

Pada gambar 4. 3 menjelaskan langkah-langkah yang digunakan oleh *Admin* untuk menambahkan *User*. Setelah *Admin* membukanya, sistem menampilkan halaman *Admin*. Sistem kemudian menampilkan halaman menu yang dipilih *Admin*, dan *Admin* mengklik tombol tambah pengguna setelah memilih item menu tambah pengguna. Sistem kemudian akan menampilkan form tambah pengguna. Setelah itu *Admin* diminta untuk mengisi form tambah pengguna. Setelah selesai, sistem akan memverifikasi info pengguna dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk pengguna yang baru ditambahkan. *Admin* perlu mengisi kembali formulir penambahan pengguna dengan benar jika tidak berfungsi.

2) Activity diagram hapus pengguna

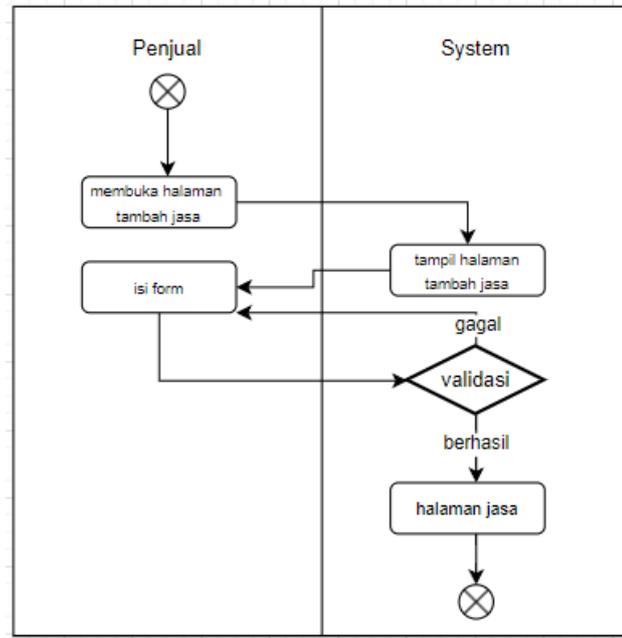


Gambar 4. 4 Activity diagram hapus pengguna

Pada gambar 4. 4 menjelaskan prosedur yang digunakan *Admin* untuk menghapus *User*. Setelah *Admin* membukanya, sistem menampilkan halaman *Admin*. Selanjutnya *Admin* memilih pengguna mana yang akan dihapus, sistem akan menampilkan pengguna yang dipilih *Admin*, dan *Admin* mengklik opsi hapus pengguna. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. *Admin* kemudian memverifikasi. Pengguna akan dihapus jika *Admin* menyetujui; jika tidak, pengguna akan dibatalkan.

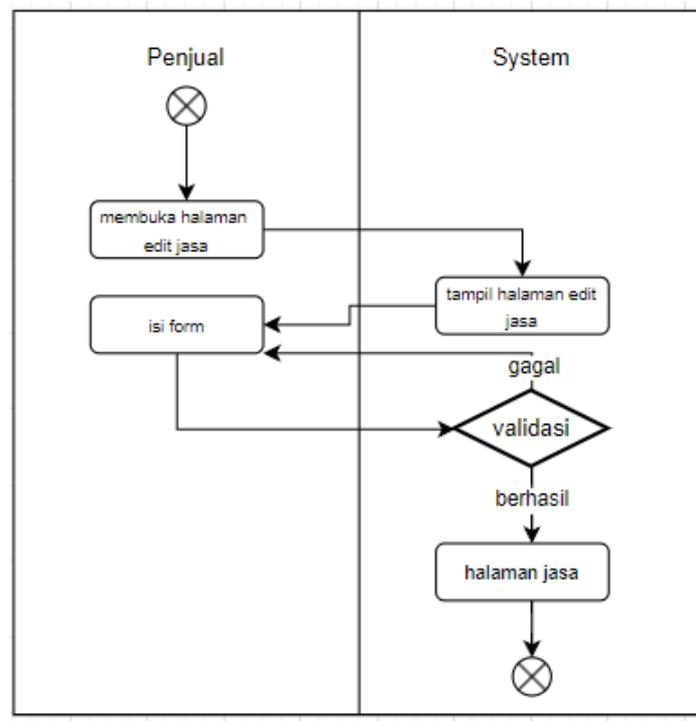
c. Activity diagram penjual

1) Activity diagram tambah jasa



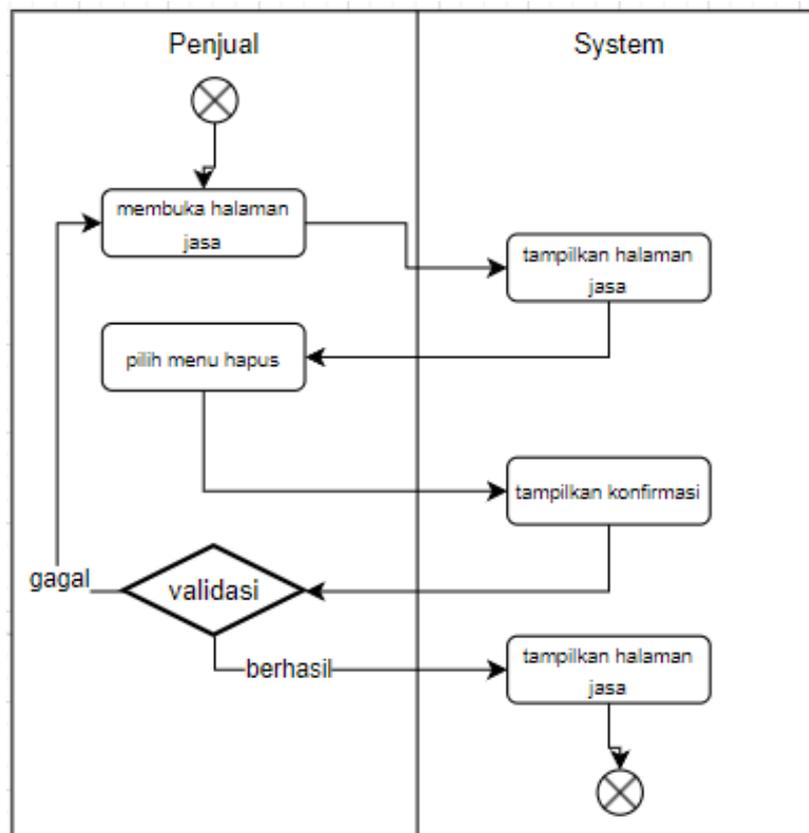
Gambar 4. 5 Activity diagram tambah jasa

Pada gambar 4.5 menjelaskan langkah-langkah yang digunakan oleh *jasa* untuk menambahkan *jasa*. Setelah *jasa* membukanya, sistem menampilkan halaman *jasa*. Sistem kemudian menampilkan halaman menu yang dipilih *jasa*, dan *jasa* mengklik tombol tambah *jasa* setelah memilih item menu tambah *jasa*. Sistem kemudian akan menampilkan form tambah *jasa*. Setelah itu *jasa* diminta untuk mengisi form tambah *jasa*. Setelah selesai, sistem akan memverifikasi info *jasa* dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk *jasa* yang baru ditambahkan. *Jasa* perlu mengisi kembali formulir penambahan *jasa* dengan benar jika tidak berfungsi.

2) *Activity* diagram edit jasa

Gambar 4. 6 *Activity* diagram edit jasa

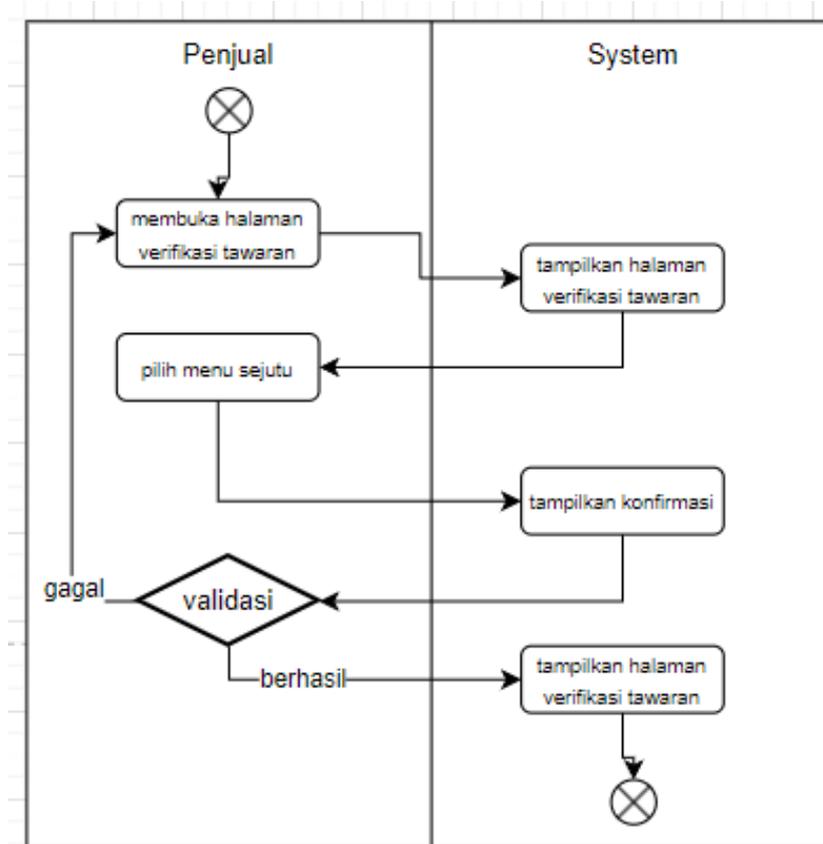
Pada gambar 4. 6 menjelaskan untuk mengubah data *jasa*, *jasa* harus menuju ke halaman *jasa* dan *jasa* harus menekan tombol edit untuk menuju ke form pengisian data *jasa*. Tindakan ini akan meminta sistem untuk menampilkan form pengisian data pengguna. Kemudian, setelah *jasa* mengisi form, sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka sistem akan menuju kembali ke halaman *jasa* dan jika belum terisi maka sistem akan kembali menampilkan form pengisiannya serta menampilkan error .

3) *Activity* diagram hapus jasa

Gambar 4.7 *Activity* diagram hapus jasa

Pada gambar 4.7 menjelaskan prosedur yang digunakan *penjual* untuk menghapus *jasa*. Setelah *penjual* membukanya, sistem menampilkan halaman *penjual*. Selanjutnya *penjual* memilih *jasa* mana yang akan dihapus, sistem akan menampilkan *jasa* yang dipilih *penjual*, dan *penjual* mengklik opsi hapus *jasa*. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. *Penjual* kemudian memverifikasi. *Jasa* akan dihapus jika *penjual* menyetujui; jika tidak, *jasa* akan dibatalkan.

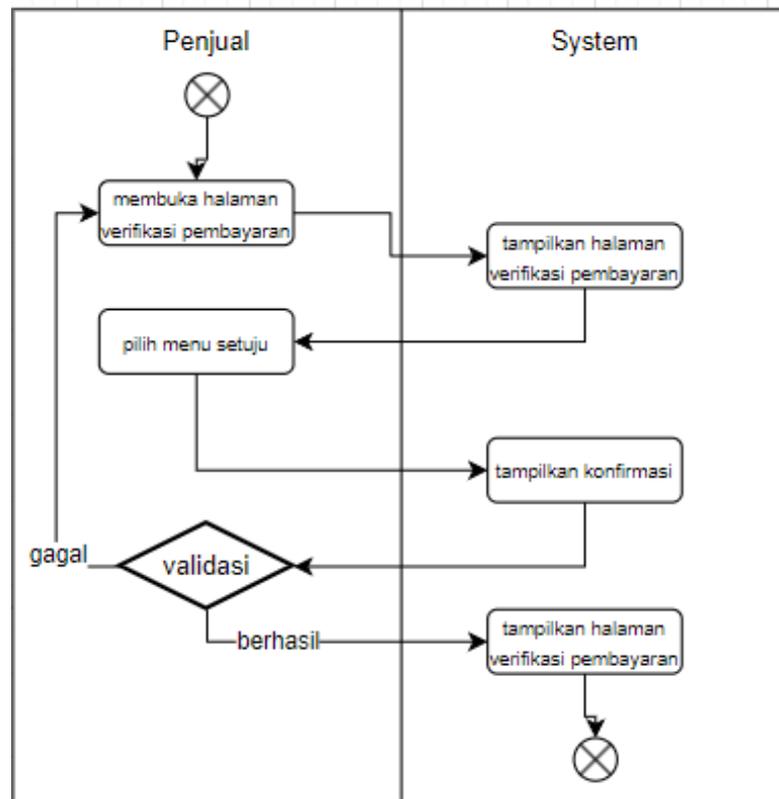
4) Activity diagram Verifikasi tawaran



Gambar 4. 8 Activity diagram verifikasi tawaran

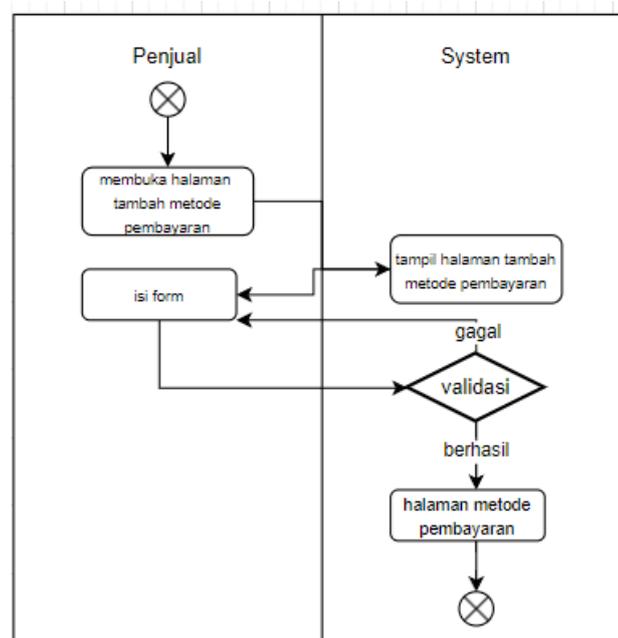
Pada gambar 4. 8 menjelaskan prosedur yang digunakan *penjual* untuk *menerima tawaran jasa*. Setelah *penjual* membukanya, sistem menampilkan halaman *penjual*. Selanjutnya *penjual* memilih *jasa* mana yang akan diterima, sistem akan menampilkan *jasa* yang dipilih *penjual*, dan *penjual* mengklik opsi terima *jasa*. Setelah itu, sistem akan menampilkan konfirmasi penerimaan. *Penjual* kemudian memverifikasi. *Jasa* akan diterima jika *penjual* menyetujui; jika tidak, *jasa* akan dibatalkan.

5) Activity diagram Verifikasi pembayaran



Gambar 4. 9 Activity diagram verifikasi pembayaran

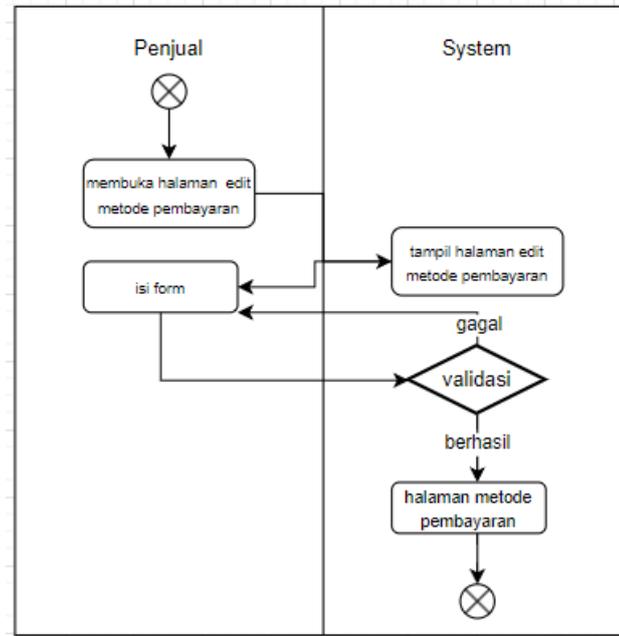
Pada gambar 4. 9 menjelaskan prosedur yang digunakan *penjual* untuk menerima pembayaran. Setelah *penjual* membukanya, sistem menampilkan halaman *penjual*. Selanjutnya *penjual* memilih membayar jasa mana yang akan diterima, sistem akan menampilkan membayar jasa yang dipilih *penjual*, dan *penjual* mengklik opsi terima membayar jasa. Setelah itu, sistem akan menampilkan konfirmasi pengterimaan. Penjual kemudian memverifikasi. Membayar jasa akan diterima jika *penjual* menyetujui; jika tidak, membayar jasa akan dibatalkan.

6) *Activity* diagram tambah metode pembayaran

Gambar 4. 10 *Activity* diagram tambah metode pembayaran

Pada gambar 4.10 menjelaskan langkah-langkah yang digunakan oleh metode pembayaran untuk menambahkan metode pembayaran. Setelah metode pembayaran membukanya, sistem menampilkan halaman metode pembayaran. Sistem kemudian menampilkan halaman menu yang dipilih metode pembayaran, dan metode pembayaran mengklik tombol tambah metode pembayaran setelah memilih item menu tambah metode pembayaran. Sistem kemudian akan menampilkan form tambah metode pembayaran. Setelah itu metode pembayaran diminta untuk mengisi form tambah metode pembayaran. Setelah selesai, sistem akan memverifikasi info metode pembayaran dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk metode pembayaran yang baru ditambahkan. Metode pembayaran perlu mengisi kembali formulir penambahan metode pembayaran dengan benar jika tidak berfungsi.

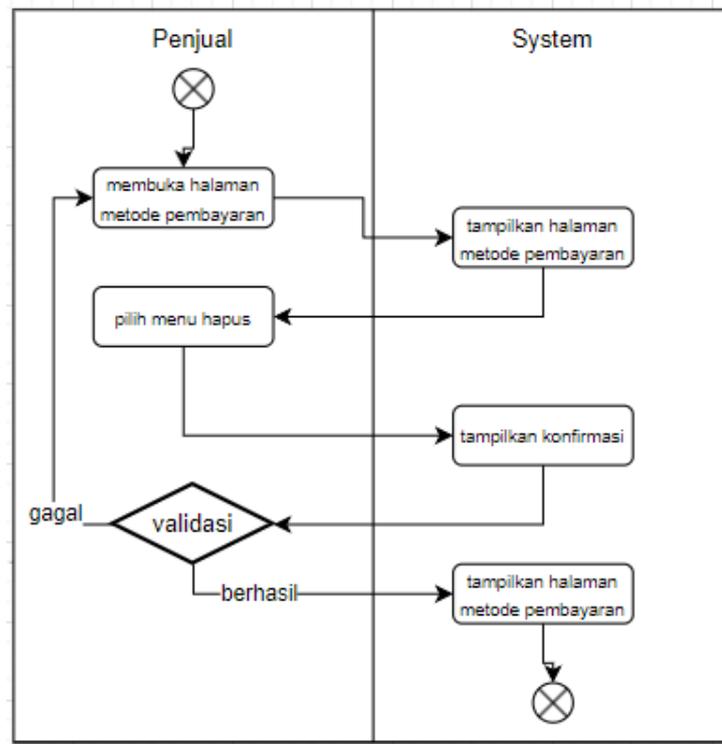
7) Activity diagram edit metode pembayaran



Gambar 4. 11 Activity diagram edit metode pembayaran

Pada gambar 4. 11 menjelaskan untuk mengubah data metode pembayaran, metode pembayaran harus menuju ke halaman metode pembayaran dan metode pembayaran harus menekan tombol edit untuk menuju ke form pengisian data metode pembayaran. Tindakan ini akan meminta sistem untuk menampilkan form pengisian data pengguna. Kemudian, setelah metode pembayaran mengisi form, sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka sistem akan menuju kembali ke halaman metode pembayaran dan jika belum terisi maka sistem akan kembali menampilkan form pengisiannya serta menampilkan error

8) Activity diagram hapus metode pembayaran

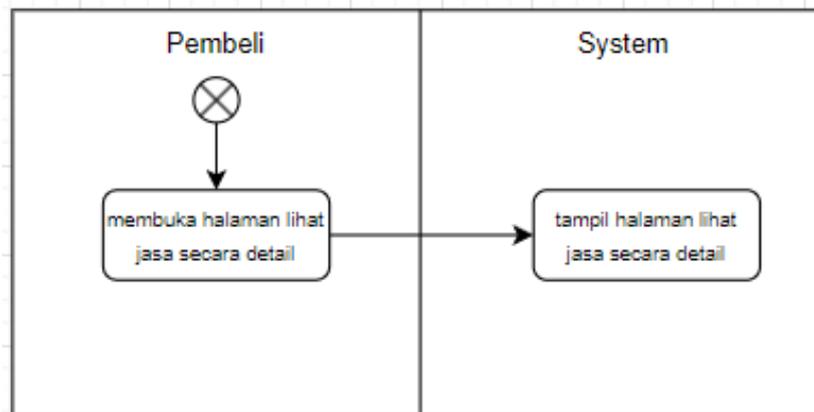


Gambar 4. 12 Activity diagram hapus metode pembayaran

Pada gambar 4. 12 menjelaskan prosedur yang digunakan *penjual* untuk menghapus metode pembayaran. Setelah *penjual* membukanya, sistem menampilkan halaman *penjual*. Selanjutnya *penjual* memilih metode pembayaran mana yang akan dihapus, sistem akan menampilkan metode pembayaran yang dipilih *penjual*, dan *penjual* mengklik opsi hapus metode pembayaran. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. *penjual* kemudian memverifikasi. Metode pembayaran akan dihapus jika *penjual* menyetujui; jika tidak, metode pembayaran akan dibatalkan.

d. Activity diagram pembeli

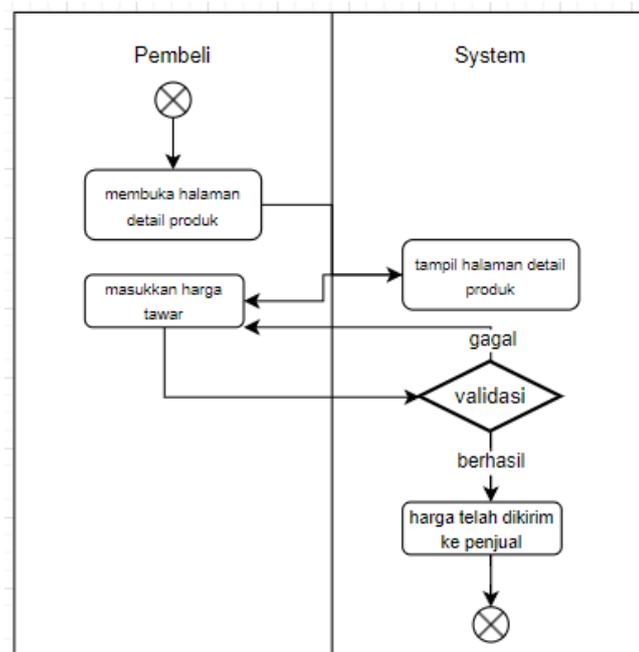
1) Activity diagram lihat jasa



Gambar 4. 13 Activity diagram lihat jasa secara detail

Pada gambar 4. 13 untuk melihat informasi detail jasa, pembeli harus mengklik tombol lihat untuk memasuki halaman detail jasa yg ingin di lihat.

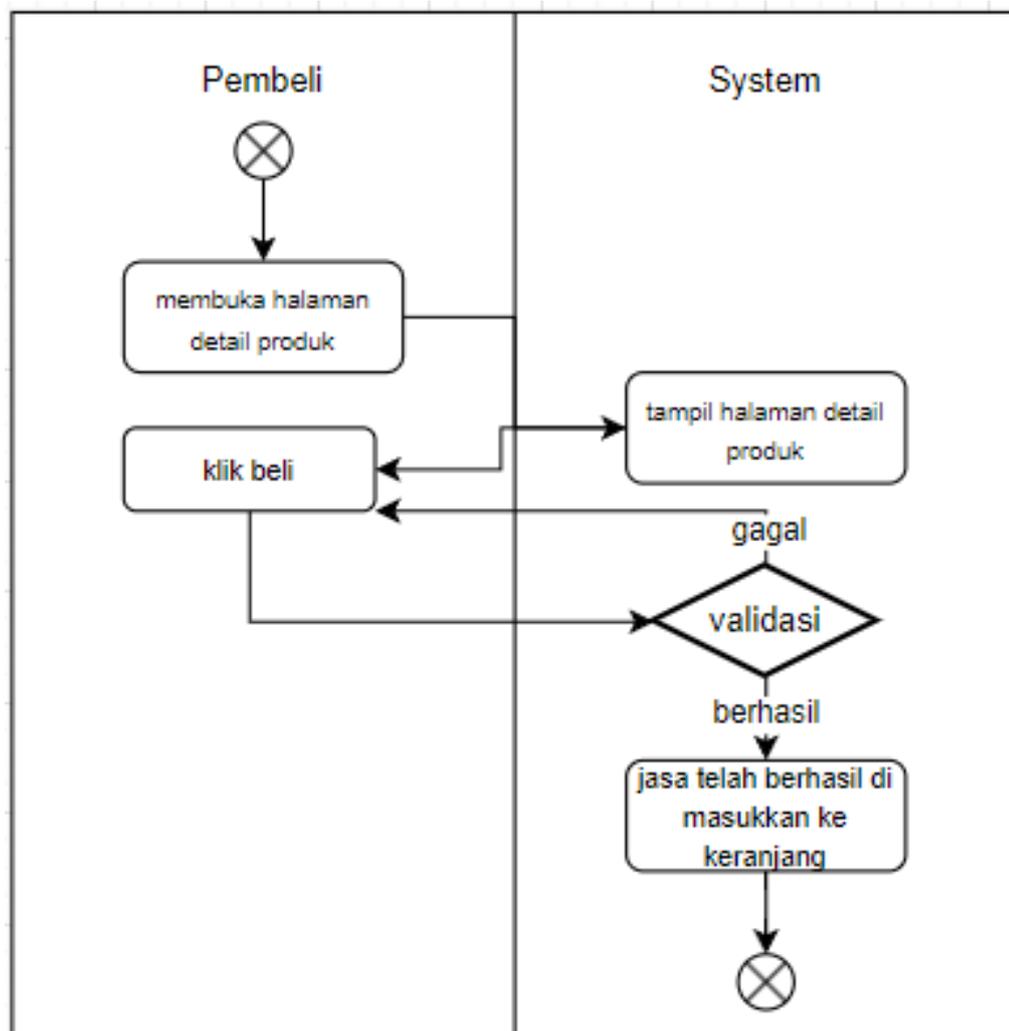
2) Activity diagram tawar harga



Gambar 4. 14 Activity diagram tawar harga

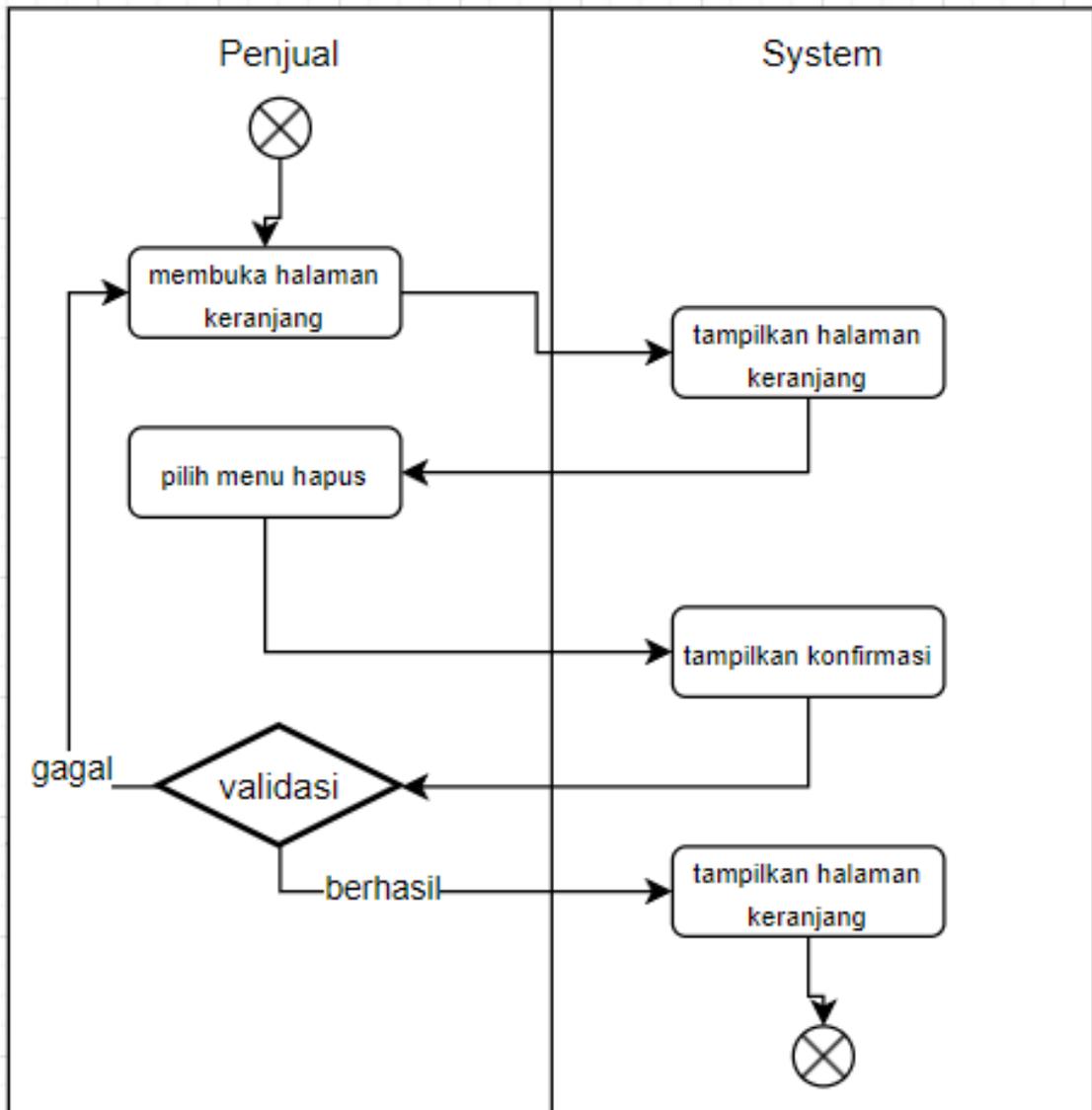
Pada gambar 4. 14 pembeli bisa menawar harga jasa dengan memasukkan harga yg diinginkan lalu tunggu sampai jawaban penjualnya.

3) *Activity* diagram tambah produk ke keranjang



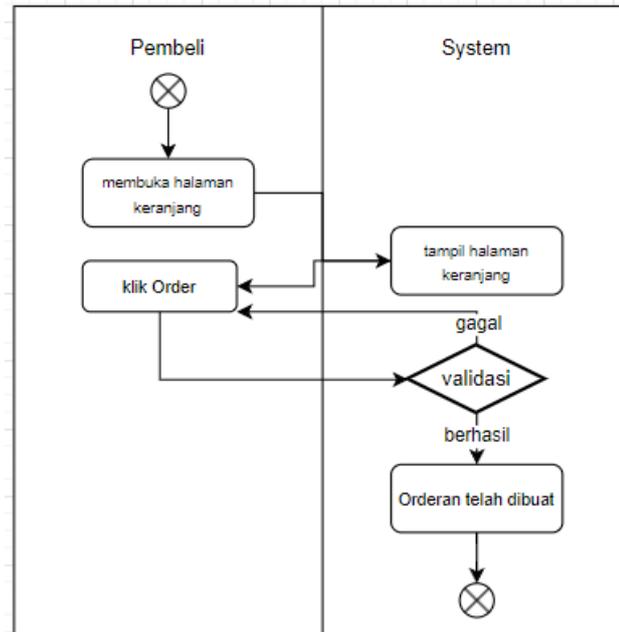
Gambar 4. 15 *Activity* diagram tambahkan ke keranjang

Pada gambar 4. 15 pembeli bisa memasukkan jasa yg ingin di pesan dengan cara mengklik tombol order dan jasa akan masuk kedalam keranjang.

4) *Activity* diagram hapus keranjang

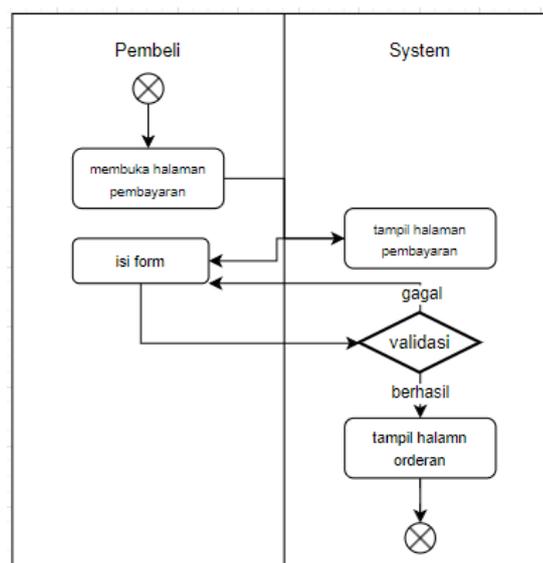
Gambar 4. 16 *Activity* diagram hapus keranjang

Pada gambar 4.16 jika pembeli ingin menghapus jasa dari keranjang maka pembeli bisa mengklik tombol hapus

5) *Activity diagram buat orderan*

Gambar 4. 17 *Activity diagram order*

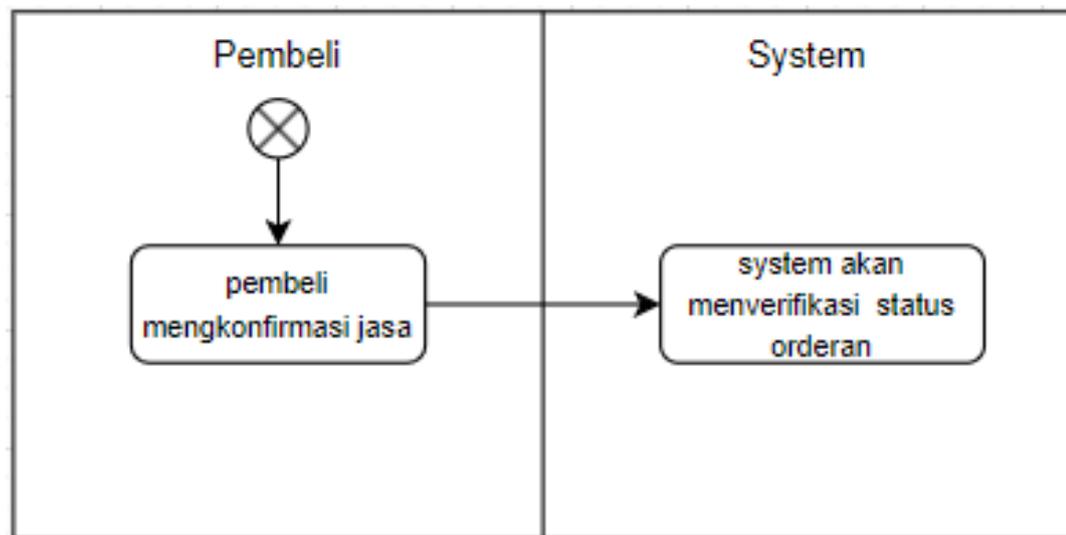
Pada gambar 4.17 menjelaskan tentang cara membuat orderan. pembeli tinggal menekan tombol order maka semua jasa yg ada dikeranjang akan masuk ke orderan.

6) *Activity diagram pembayaran*

Gambar 4. 18 *Activity diagram pembayaran*

Pada gambar 4.18 menjelaskan tentang cara melakukan pembayaran.pembeli harus mengisi semua form pembayaran dan mengirim bukti pembayaran , setelah itu penjual akan memeriksa pembayarannya .setelah itu status orderan akan berubah menjadi pengerjaan .

7) *Activity diagram* Konfirmasi pesanan



Gambar 4. 19 Konfirmasi pesanan

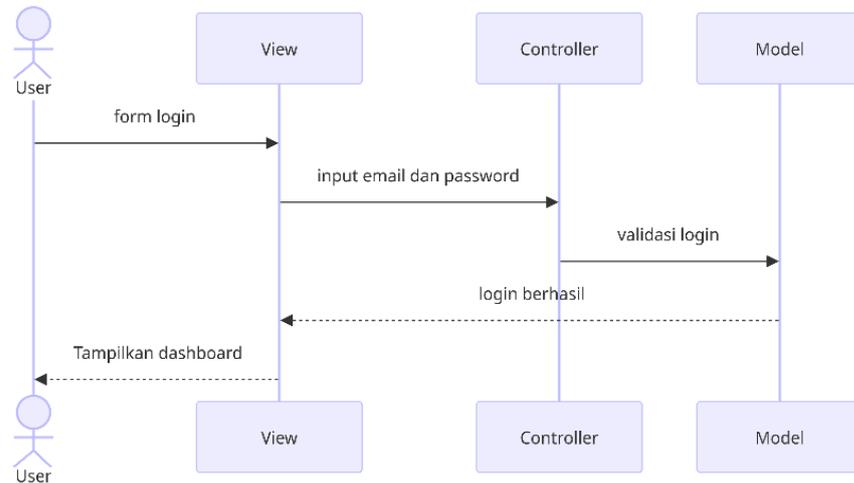
Pada gambar 4.19 menjelaskan tentang konfirmasi pesanan.apabila pesanan telah dikirimkan, maka pembeli harus mengkonfirmasi pesanannya agar status orderannya menjadi selesai.

B. *Sequence Diagram*

Sequence diagram merupakan diagram yang menggambarkan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu. Diagram ini menunjukkan bagaimana objek-objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case*.

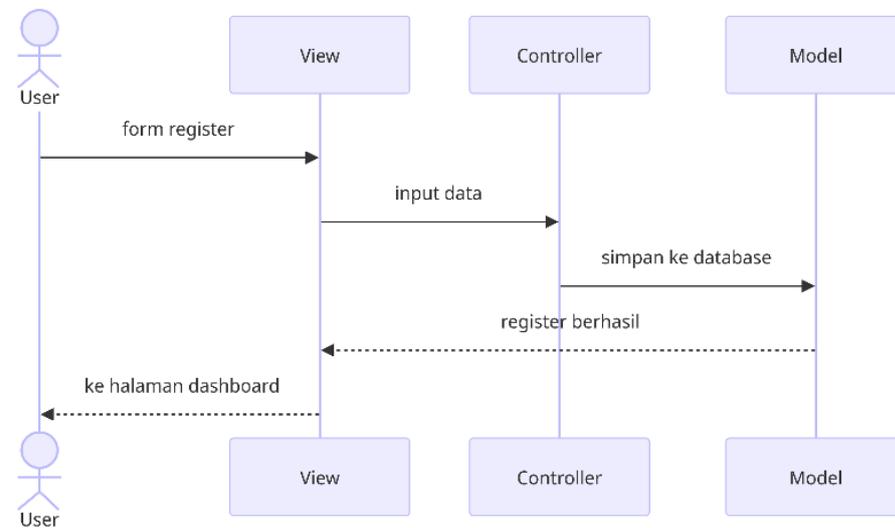
1. Sequence diagram login

1) Sequence diagram login



Gambar 4. 20 Sequence diagram login

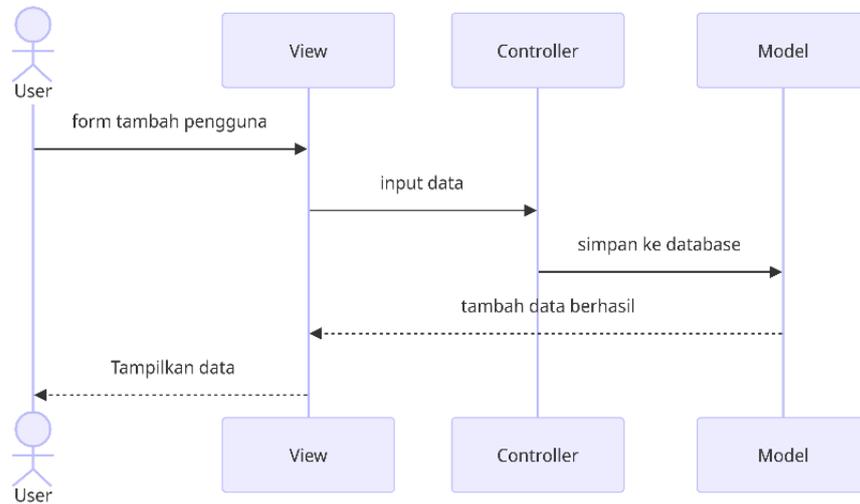
2) Sequence diagram Register



Gambar 4. 21 Sequence diagram Register

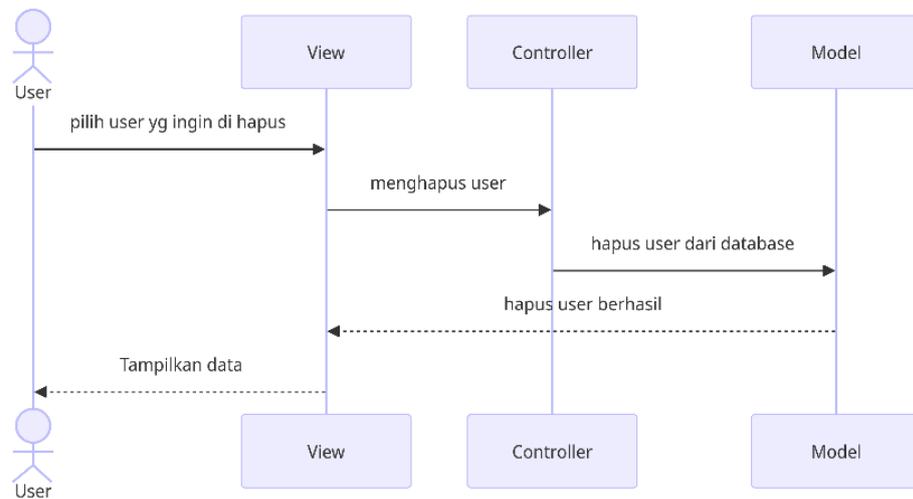
2. Sequence diagram Admin

1) Sequence diagram tambah User



Gambar 4. 22 Sequence diagram tambah User

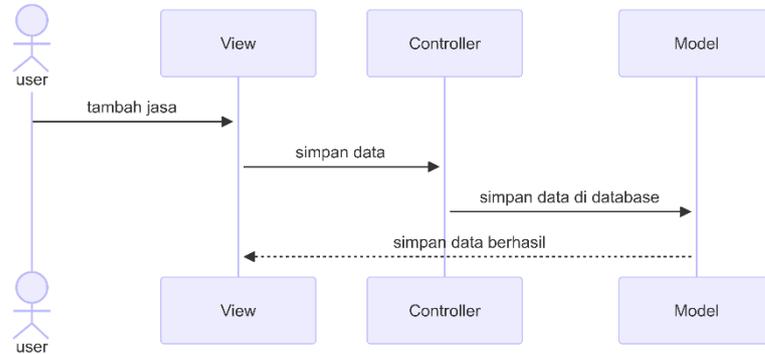
2) Sequence diagram hapus User



Gambar 4. 23 Sequence diagram hapus User

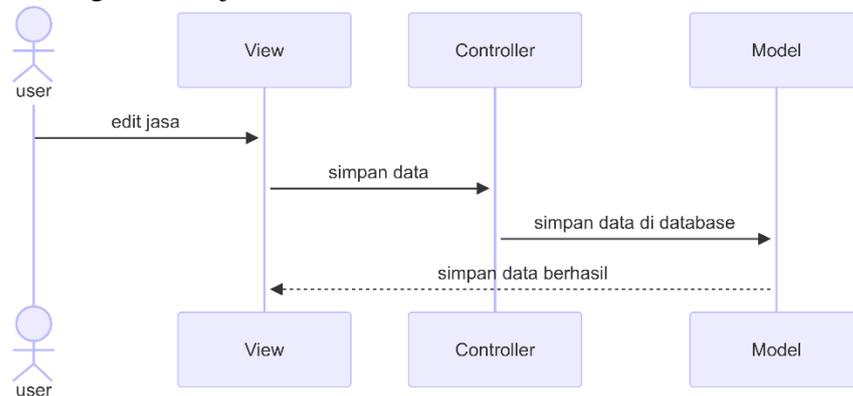
3. Sequence diagram penjual

1) Sequence diagram tambah jasa



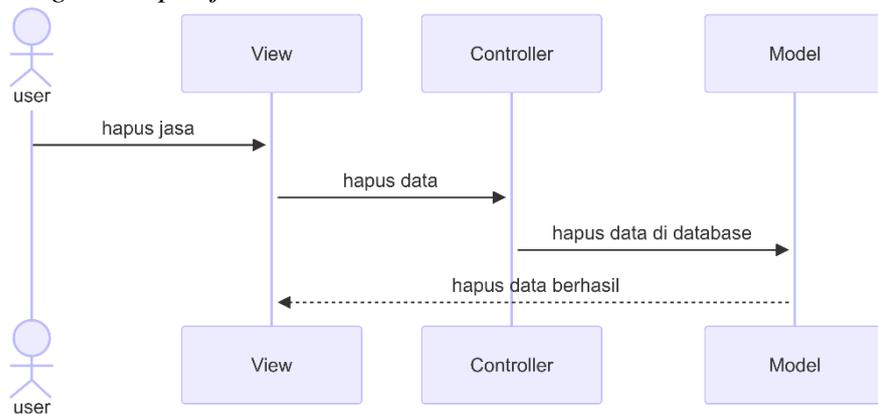
Gambar 4. 24 Sequence diagram tambah jasa

2) Sequence diagram edit jasa

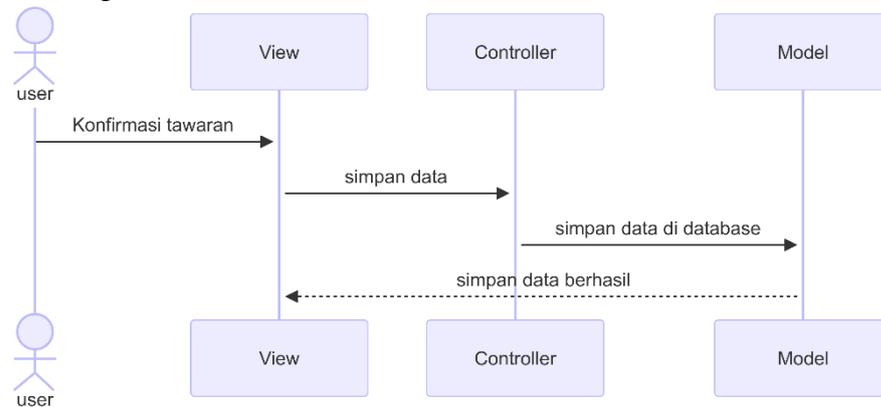
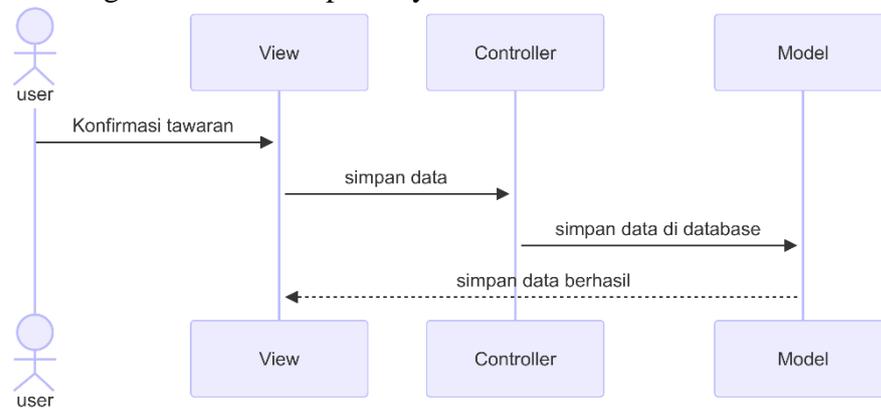
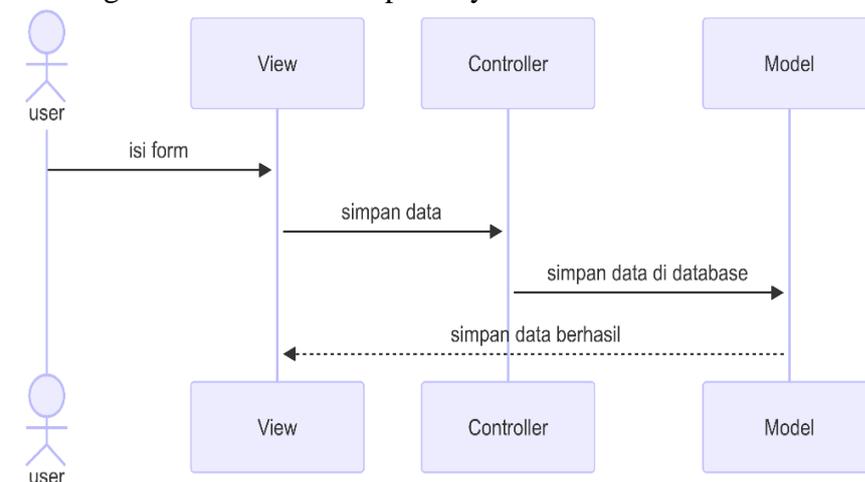


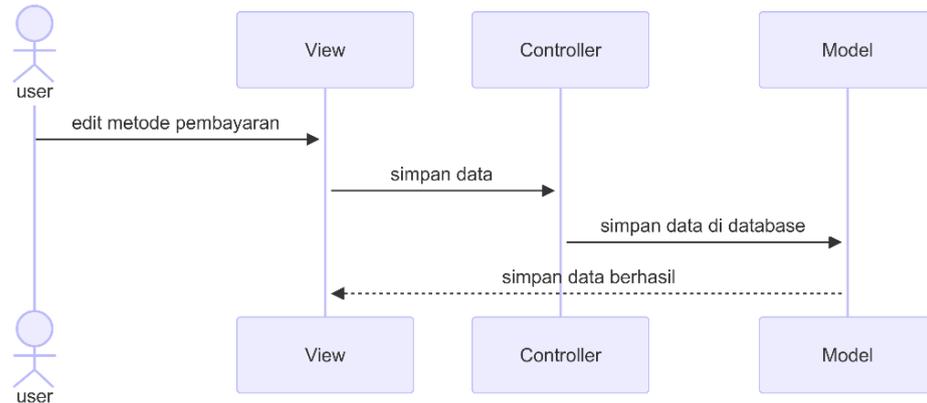
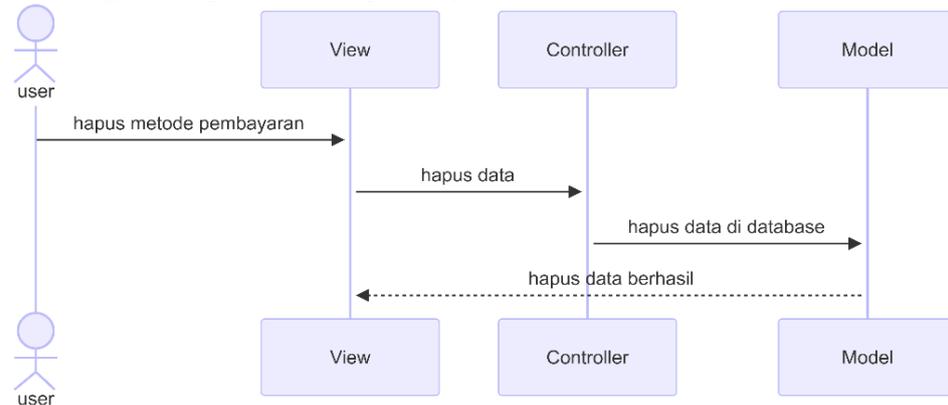
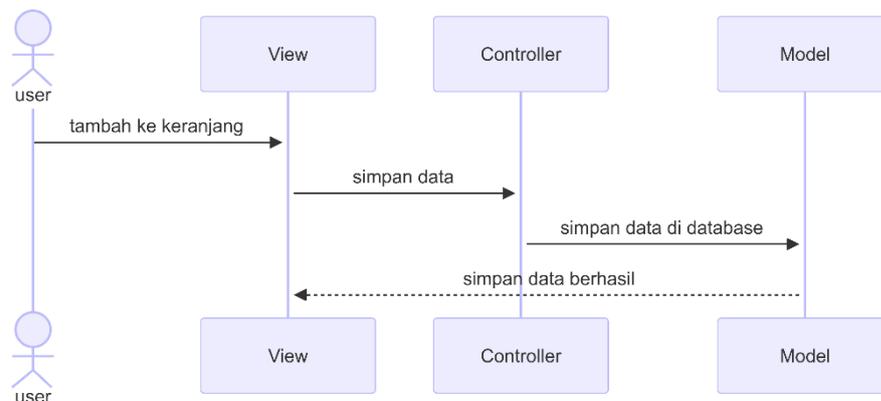
Gambar 4. 25 Sequence diagram edit jasa

3) Sequence diagram hapus jasa

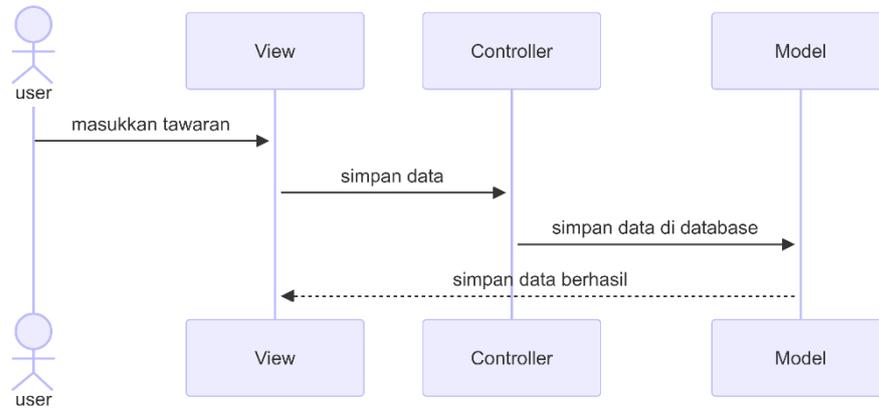


Gambar 4. 26 Sequence diagram hapus jasa

4) *Sequence* diagram konfirmasi tawaranGambar 4. 27 *Sequence* diagram konfirmasi tawaran5) *Sequence* diagram konfirmasi pembayaranGambar 4. 28 *Sequence* diagram konfirmasi pembayaran6) *Sequence* diagram tambah metode pembayaranGambar 4. 29 *Sequence* diagram tambah metode pembayaran

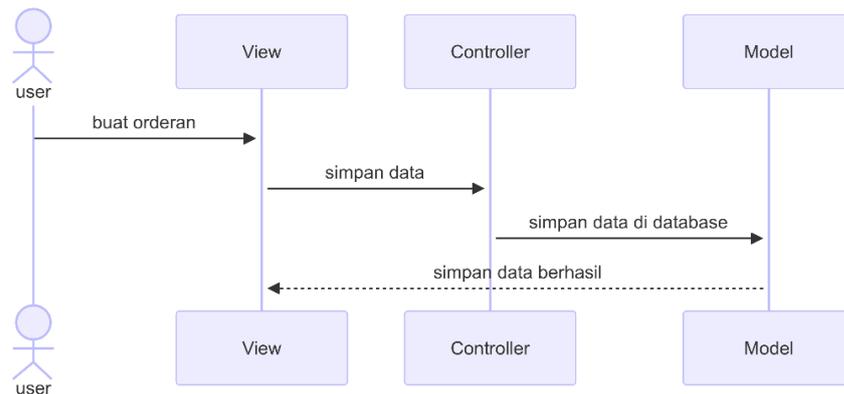
7) *Sequence diagram edit metode pembayaran*Gambar 4. 30 *Sequence diagram edit metode pembayaran*8) *Sequence diagram hapus metode pembayaran*Gambar 4. 31 *Sequence hapus metode pembayaran*4. *Sequence diagram pembeli*1) *Sequence diagram tambah jasa ke keranjang*Gambar 4. 32 *Sequence diagram tambah jasa ke keranjang*

2) *Sequence* diagram tawar harga jasa



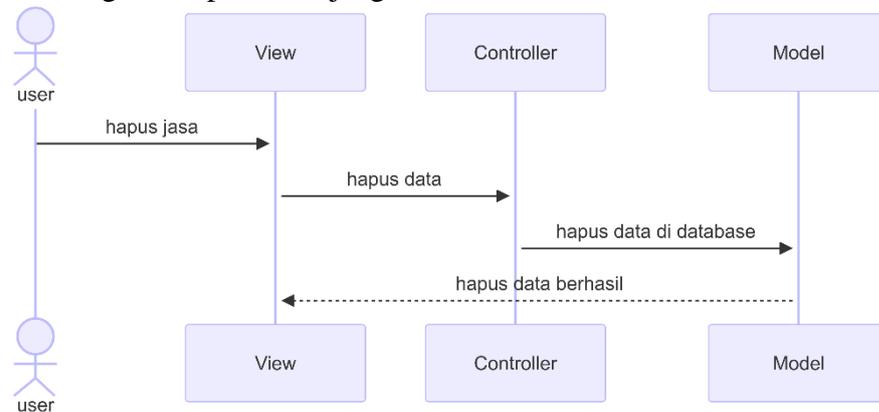
Gambar 4. 33 *Sequence* diagram tawar harga jasa

3) *Sequence* diagram buat orderan



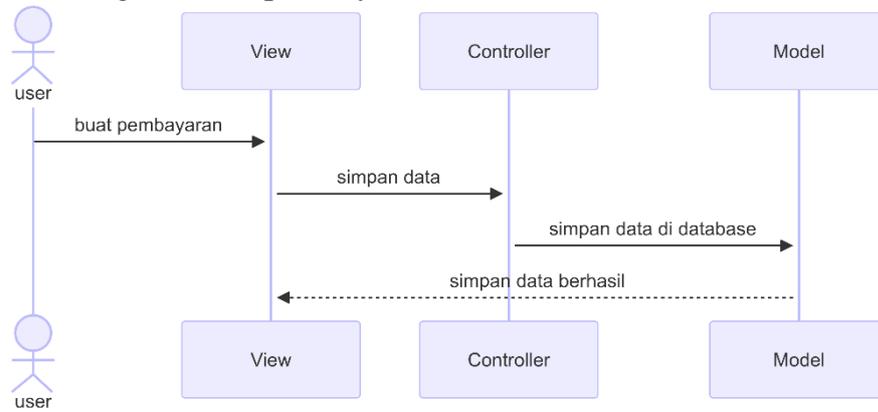
Gambar 4. 34 *Sequence* diagram buat orderan

4) *Sequence* diagram hapus keranjang



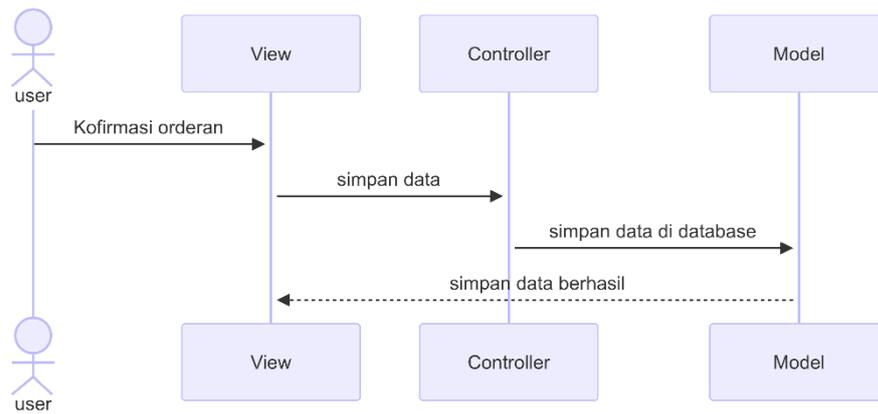
Gambar 4. 35 *Sequence* diagram hapus keranjang

Sequence diagram buat pembayaran



Gambar 4. 36 *Sequence diagram buat pembayaran*

6) *Sequence diagram konfirmasi pesanan*



Gambar 4. 37 *Sequence diagram konfirmasi pesanan*

C. Detail Sistem

1. Halaman *Welcome*

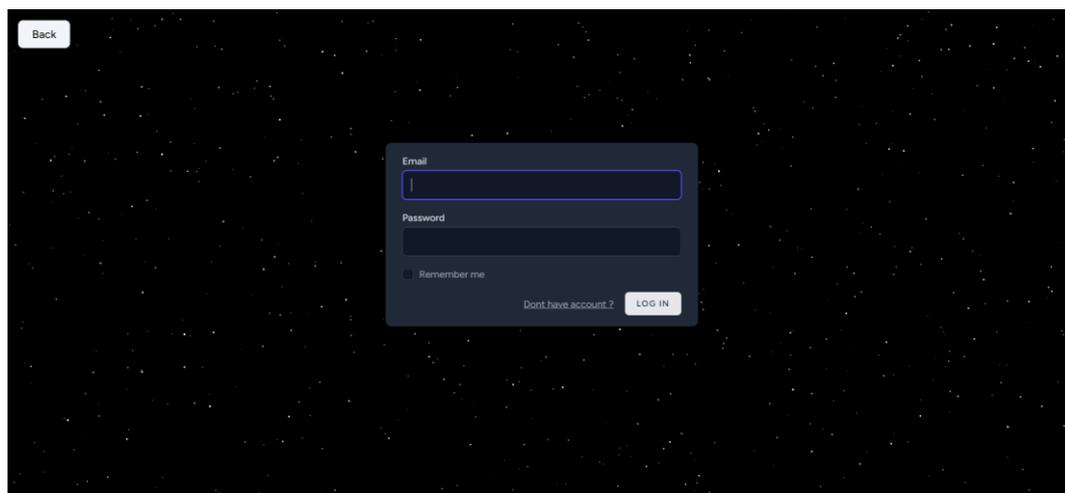
Halaman *welcome* merupakan halaman awal yg muncul Ketika pertama kali mengakses aplikasi.



Gambar 4. 38 Halaman *welcome*

2. Halaman *Login*

Halaman *login* merupakan halaman yg di gunakan *User* untuk login.



Gambar 4. 39 Halaman *login*

Halaman Register

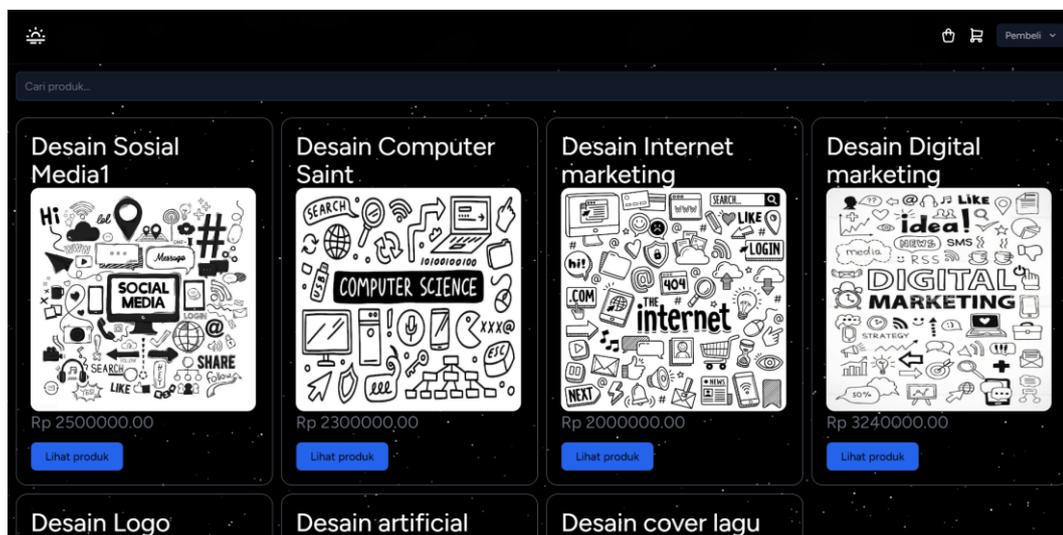
Halaman *Register* merupakan halaman yg di gunakan *User* untuk mendaftar .

The screenshot shows a registration form on a dark background. The form has a 'Back' button in the top left corner. The fields are: Name, Email, Nomor Hp, Password, and Confirm Password. At the bottom of the form, there is a link 'Already registered?' and a 'REGISTER' button.

Gambar 4. 40 Halaman Register

4. Halaman dashboard utama

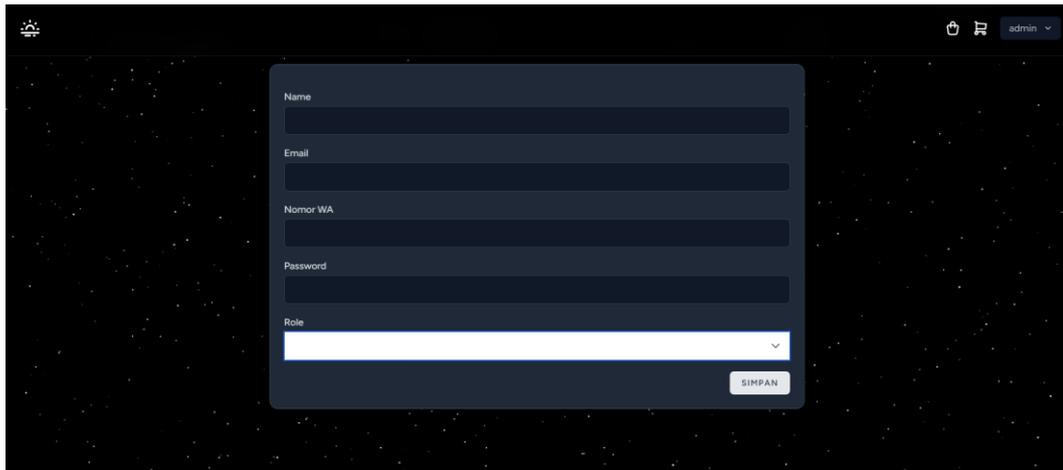
Halaman *dashboard utama* merupakan halaman yg bisa diakses oleh semua *User*.



Gambar 4. 41 Halaman dashboard utama

5. Halaman tambah *User*

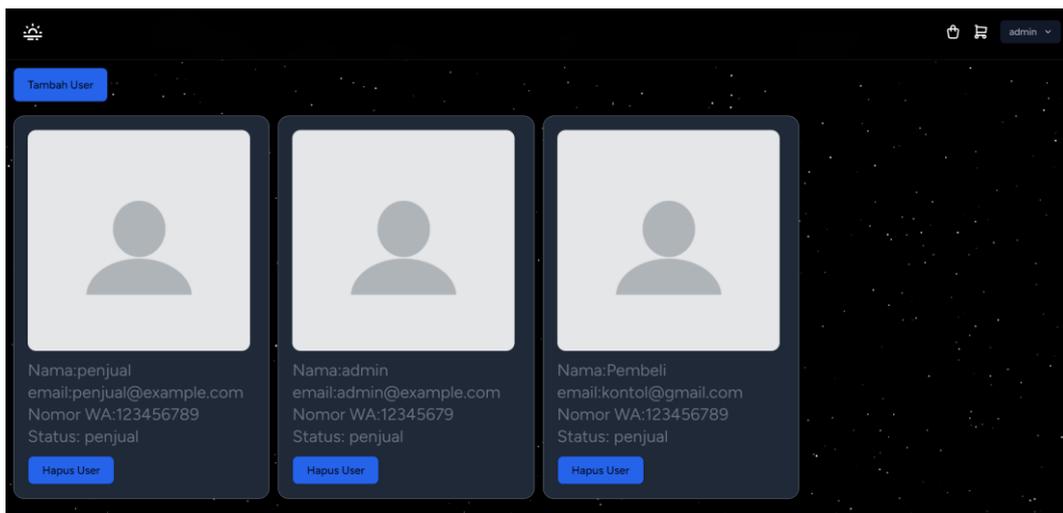
Halaman tambah *User* merupakan halaman untuk menambah *User*.berbeda dengan *Register*, halaman ini hanya khusus untuk *Admin* untuk menambahkan *User*.



Gambar 4. 42 Halaman tambah *User*

6. Halaman *Users*

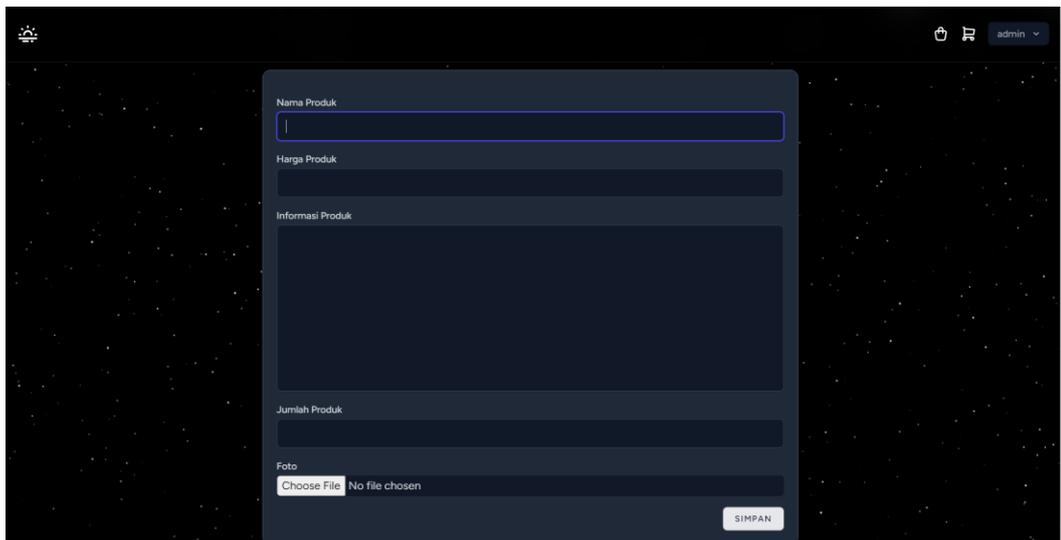
Halaman *User* merupakan halaman untuk melihat semua *User*.



Gambar 4. 43 Halaman *Users*

7. Halaman Tambah Jasa

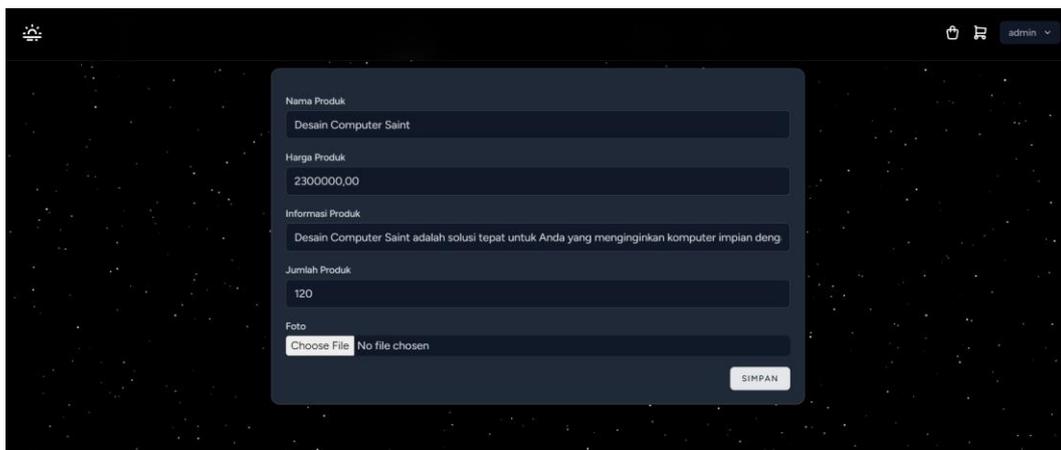
Halaman tambah jasa merupakan halaman untuk menambah jasa.berbeda dengan *Register*, halaman ini hanya khusus untuk *Admin* untuk menambahkan jasa.



Gambar 4. 44 Halaman tambah jasa

8. Halaman edit jasa

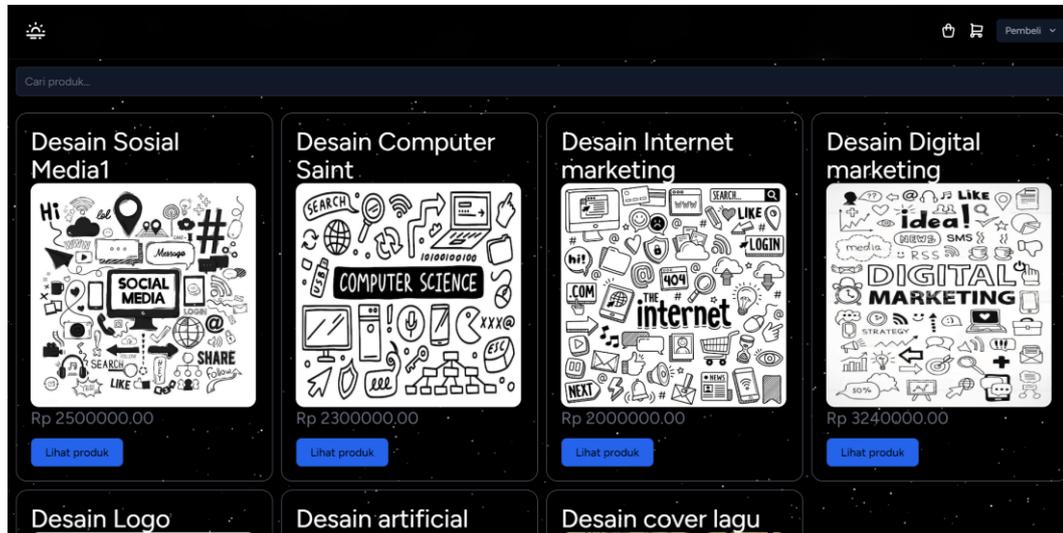
Halaman edit jasa merupakan halaman untuk mengubah data jasa.halaman ini berguna jika dalam penambahan jasa terdapat kesalahan ketik.



Gambar 4. 45 Halaman edit jasa

9. Halaman Jasa

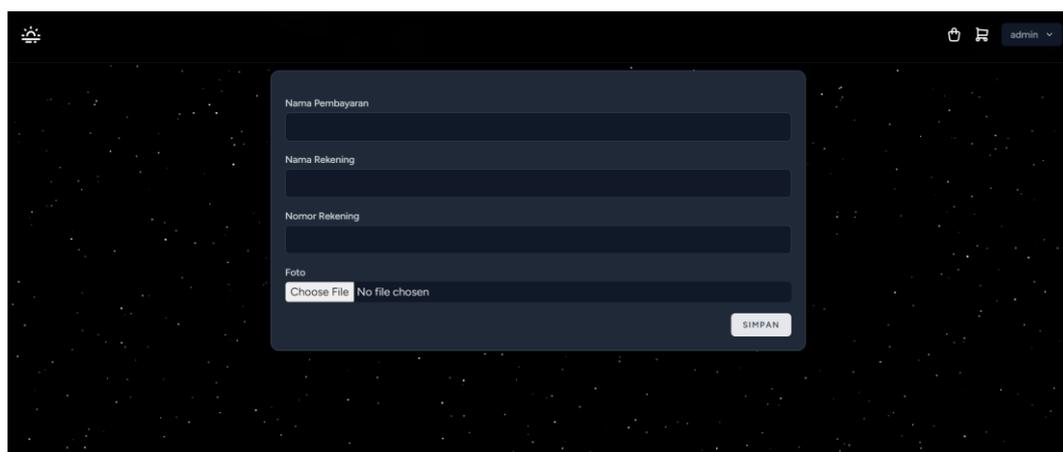
Halaman jasa merupakan halaman untuk melihat semua jasa.



Gambar 4. 46 Halaman jasa

10. Halaman Tambah metode pembayaran

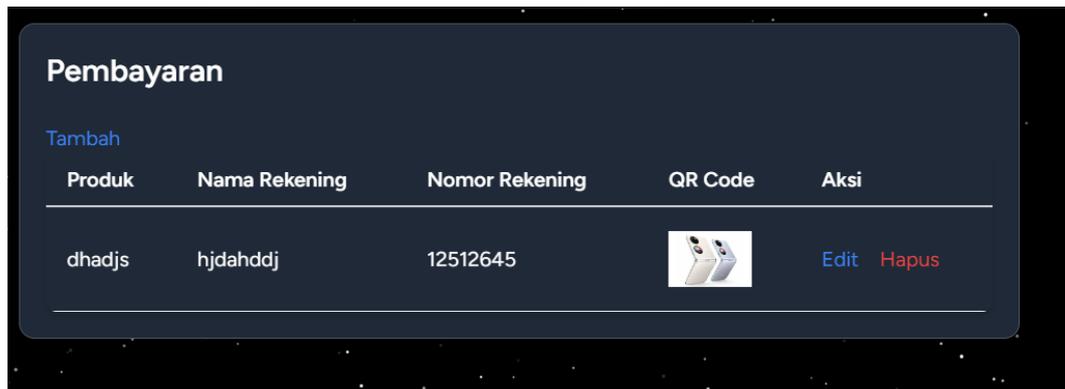
Halaman tambah metode pembayaran merupakan halaman untuk menambah metode pembayaran.berbeda dengan *Register*, halaman ini hanya khusus untuk *Admin* untuk menambahkan metode pembayaran.



Gambar 4. 47 Halaman tambah metode pembayaran

11. Halaman Metode Pembayaran

Halaman metode pembayaran merupakan halaman untuk melihat semua metode pembayaran.



Gambar 4. 48 Halaman metode pembayaran

12. Halaman edit metode pembayaran

Halaman edit metode pembayaran merupakan halaman untuk mengubah data metode pembayaran. halaman ini berguna jika dalam penambahan metode pembayaran terdapat kesalahan ketik.

Nama pembayaran
dhadjs

Harga rekening
hjdahddj

nomor_rekening
12512645

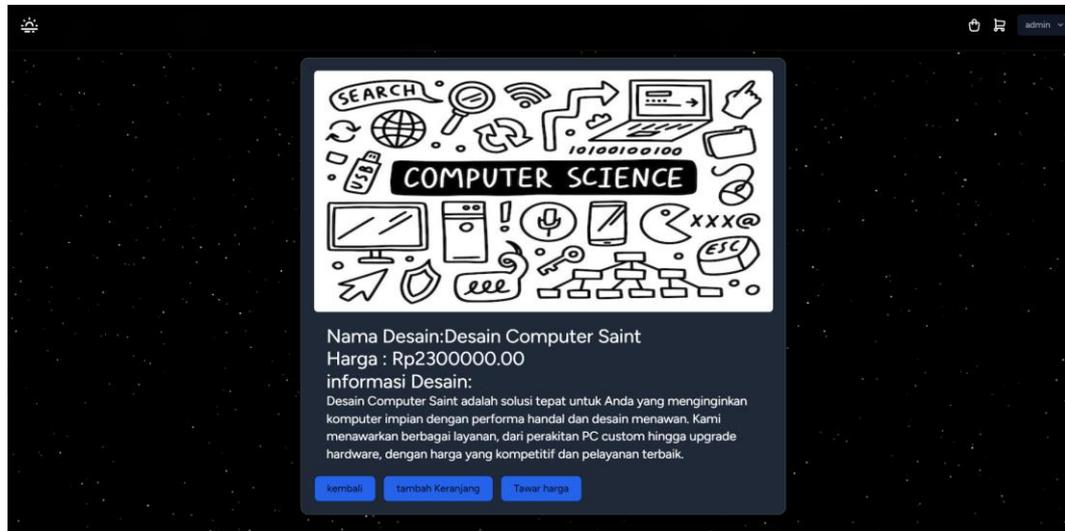
Foto
Choose File No file chosen

SIMPAN

Gambar 4. 49 Halaman edit metode pembayaran

13. Halaman Lihat Informasi Jasa

Halaman ini merupakan halaman untuk melihat informasi jasa secara detail.



Gambar 4. 50 Halaman informasi jasa

14. Halaman list tawaran harga

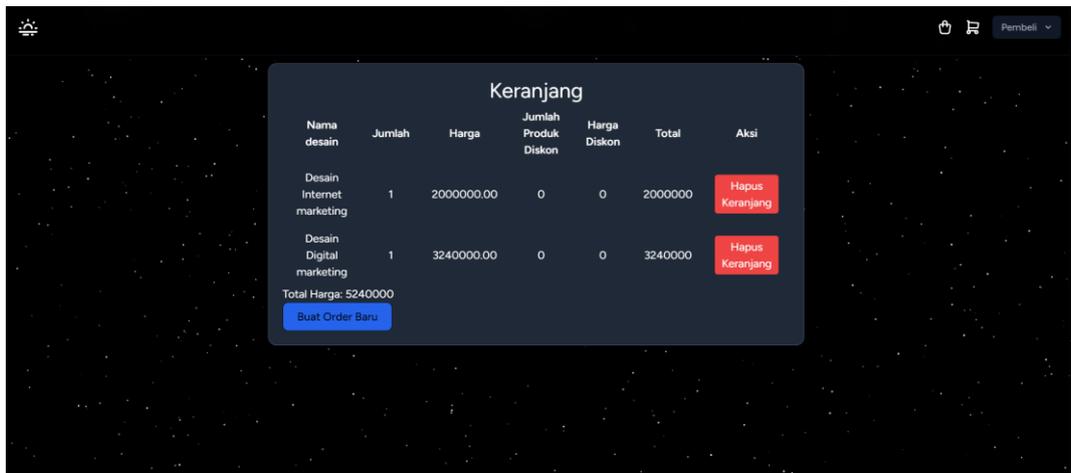
Halaman ini adalah halaman untuk melihat semua tawaran harga .



Gambar 4. 51 Halaman daftar tawaran harga

15. Halaman keranjang

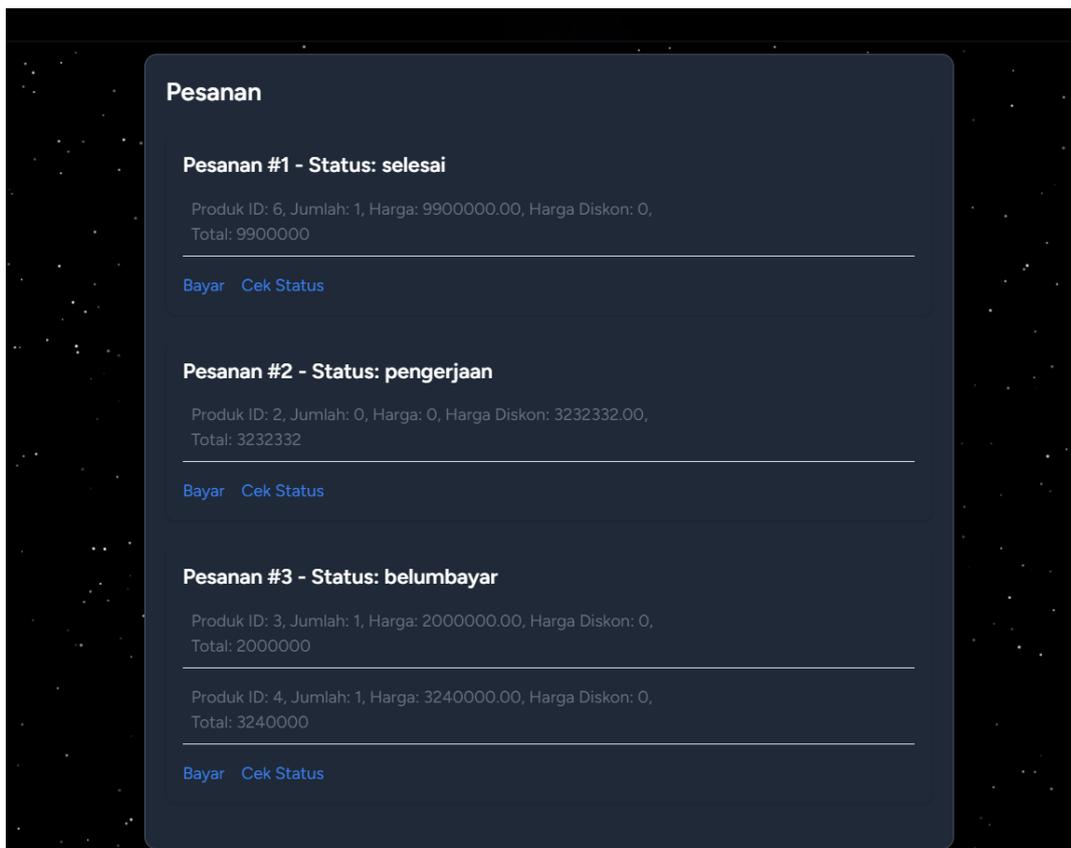
Halaman ini adalah halaman untuk melihat keranjang .



Gambar 4. 52 Halaman keranjang

16. Halaman *Orders*

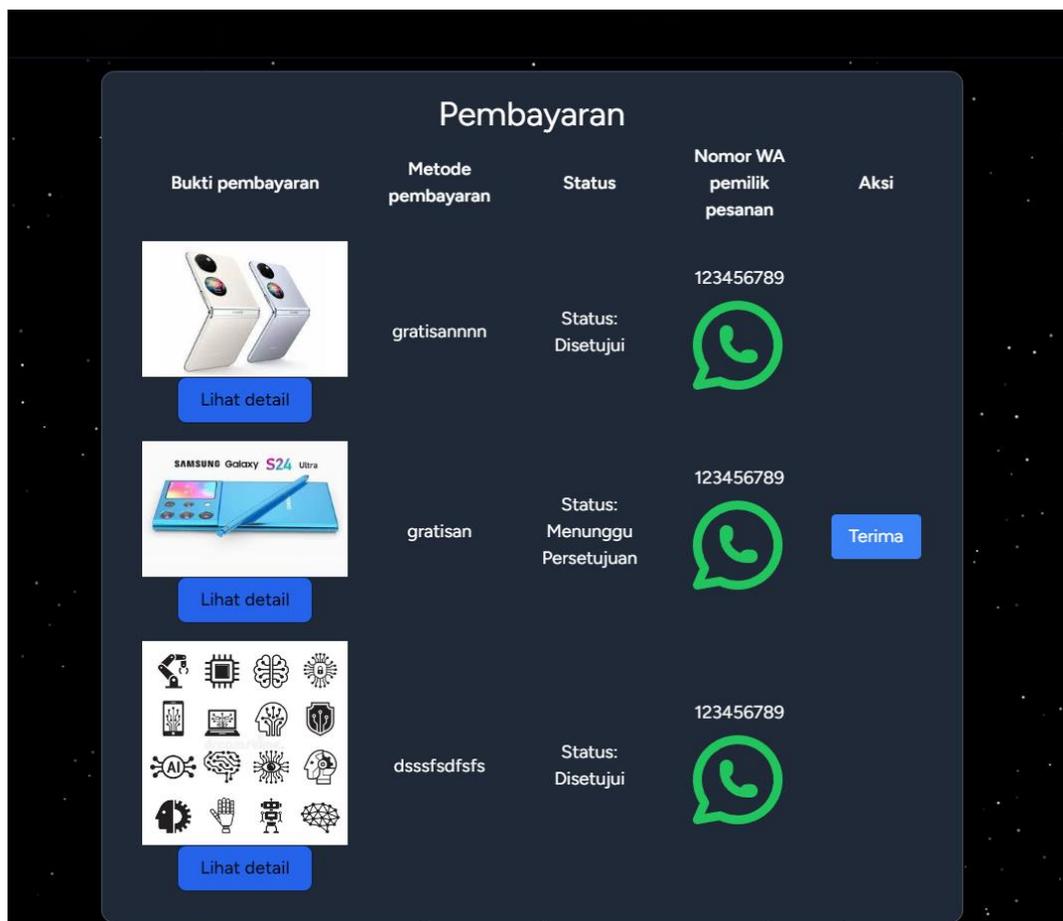
Halaman ini merupakan halaman untuk melihat semua orderan.



Gambar 4. 53 Halaman order

17. Halaman terima pembayaran

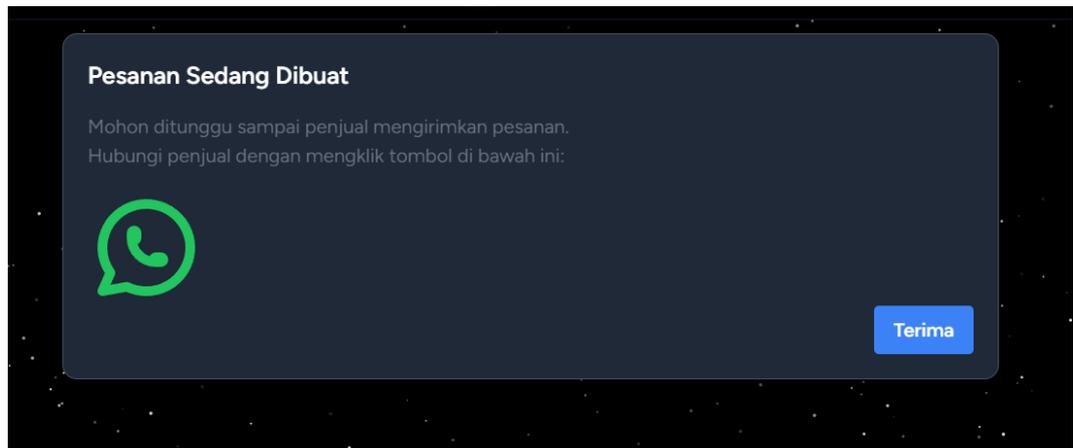
Halaman terima pembayaran merupakan halaman untuk menerima semua pembayaran yg masuk.



Gambar 4. 54 Halaman terima pembayaran

18. Halaman Konfirmasi Pesanan

Dihalaman ini, pembeli akan menunggu pengerjaan produknya dari penjual. Dan setelah penjual mengirimkan produknya maka pembeli akan mengkonfirmasi pesannya.



Gambar 4. 55 Halaman konfirmasi pesanan

D. Pengujian Sistem

Pengujian sistem bertujuan untuk mengidentifikasi kesalahan, kesenjangan, atau kekurangan dalam sistem sebelum digunakan. Ada dua pendekatan utama dalam pengujian sistem ini, yaitu pengujian *Black box* dan pengujian *White box*.

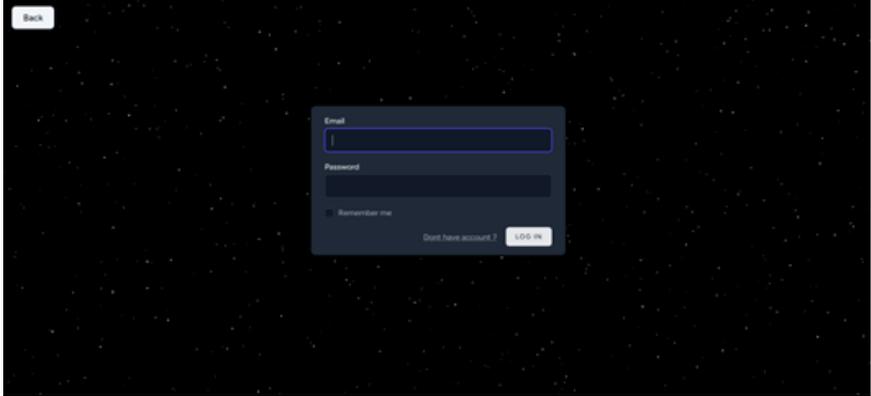
1. *Black box*

a. *Black box testing* kesalahan *Username* dan *password*

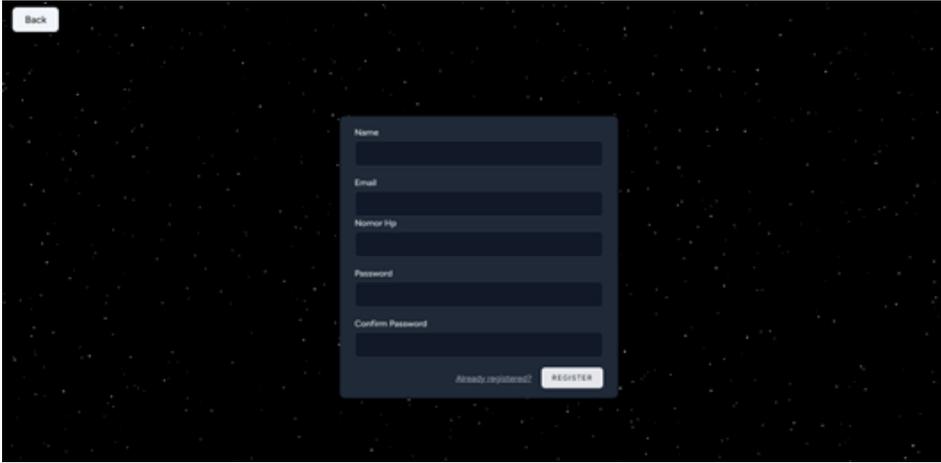
Tabel 4. 1 *Black box testing* kesalahan *Username* dan *password*

Tes faktor	Hasil	Keterangan
Memasukkan <i>Username</i> atau <i>password</i> yang tidak sesuai.	✓	Berhasil, ketika <i>Username</i> atau <i>password</i> tidak sesuai tampil <i>login gagal</i> .
<i>Screenshot</i>		

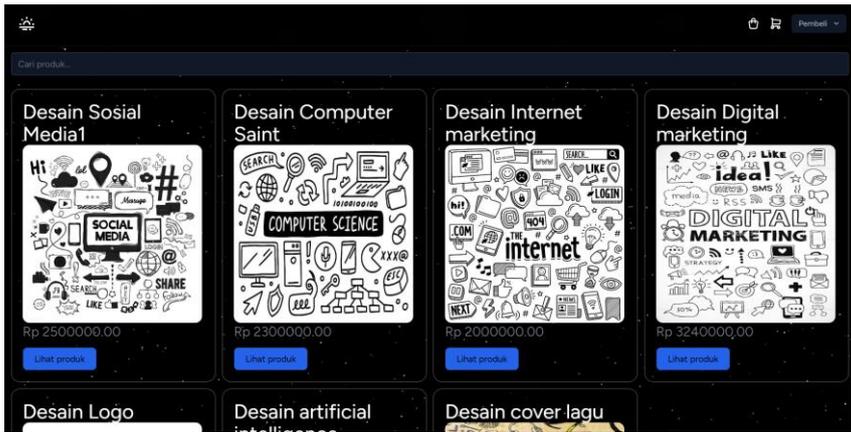
b. *Black box testing* login berhasil**Tabel 4. 2** *Black box testing* login berhasil

Tes faktor	Hasil	Keterangan
Memasukkan <i>Username</i> atau <i>password</i> yang benar	✓	Sistem berhasil menampilkan halaman dashboard.
<i>Screenshot</i>		
		

c. *Black box testing Register* berhasil**Tabel 4. 3** *Black box testing Register* berhasil

Tes faktor	Hasil	Keterangan
Mengisi form dengan benar	✓	Sistem berhasil menampilkan halaman dashboard.
<i>Screenshot</i>		
		

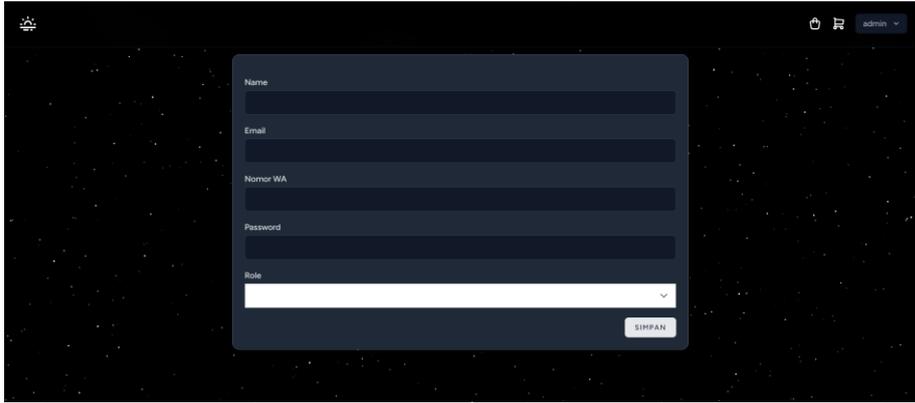
d. *Black box testing dashboard utama***Tabel 4. 4** *Black box testing dashboard utama*

Tes faktor	Hasil	Keterangan
User melakukan login mengakses halaman dashboard	✓	Halaman dashboard akan tampil
Screenshot		
		

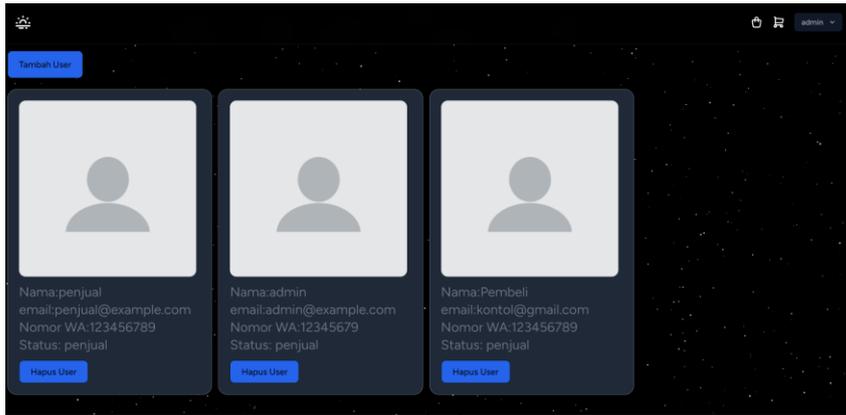
e. *Black box testing halaman awal***Tabel 4. 5** *Blackbox testing halaman awal*

Tes faktor	Hasil	Keterangan
User mengakses halaman awal	✓	Halaman awal akan tampil
Screenshot		
		

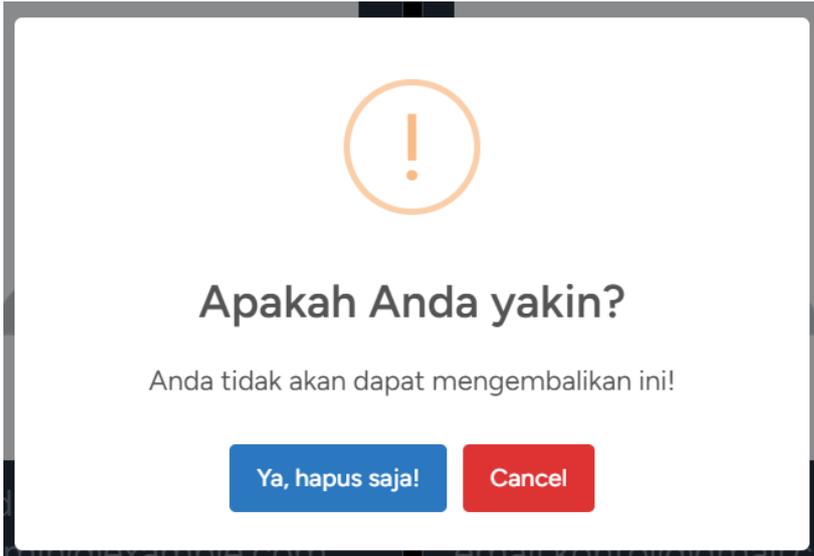
f. *Black box testing tambah User***Tabel 4. 6** *Black box testing tambah*

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol simpan.	✓	<i>User</i> akan bertambah .
Screenshot		
		

g. *Black box testing Users***Tabel 4. 7** *Black box testing Users*

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengakses halaman <i>Users</i> dashboard.	✓	Halaman <i>Users</i> dashboard akan tampil
Screenshot		
		

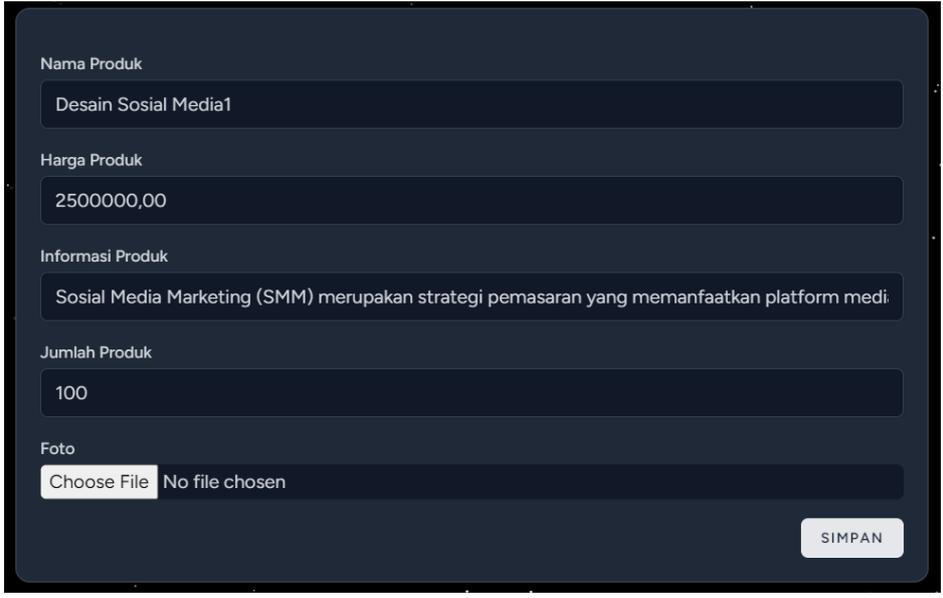
h. *Black box testing hapus User***Tabel 4. 8** *Black box testing hapus User*

Tes faktor	Hasil	Keterangan
<i>Admin</i> menghapus <i>User</i> yg dipilih.	✓	<i>User</i> yg dipilih <i>Admin</i> akan dihapus.
<i>Screenshot</i>		
		

i. *Black box testing tambah jasa***Tabel 4. 9** *Black box testing tambah jasa*

Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengisi form tambah dan menekan tombol simpan.	✓	jasa akan bertambah.
<i>Screenshot</i>		
		

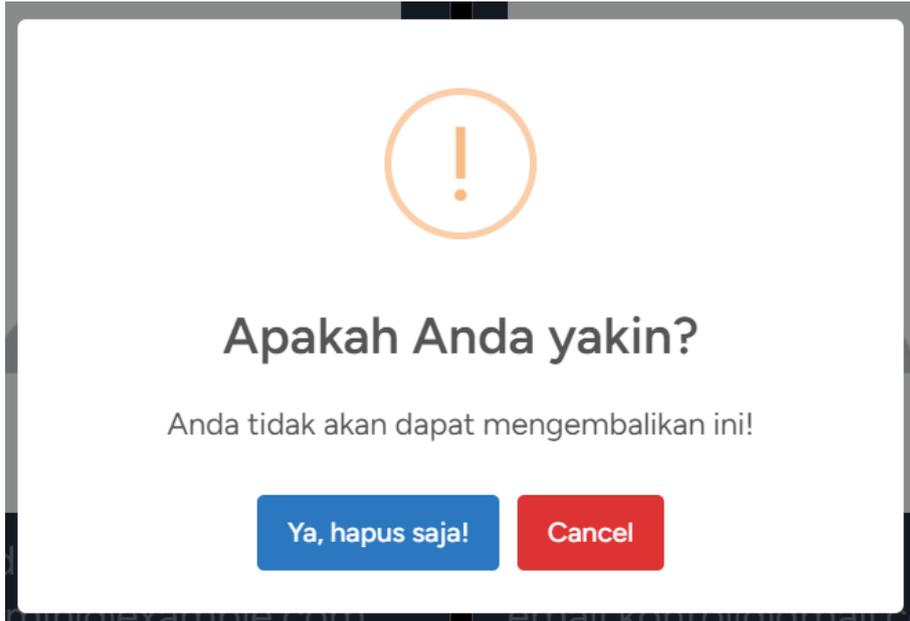
j. *Black box testing* edit jasa**Tabel 4. 10** *Black box testing* edit jasa

Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengisi form edit dan menekan tombol simpan.	✓	jasa akan berubah.
<i>Screenshot</i>		
		

k. *Black box testing* informasi jasa**Tabel 4. 11** *Black box testing* jasa

Tes faktor	Hasil	Keterangan
<i>User</i> mengakses salah satu jasa	✓	Halaman informasi jasa akan tampil.
<i>Screenshot</i>		
		

1. *Black box testing* hapus jasa**Tabel 4. 12** *Black box testing* hapus jasa

Tes factor	Hasil	Keterangan
<i>Penjual</i> menghapus jasa yg dipilih.	✓	jasa yg dipilih <i>Admin</i> akan dihapus.
<i>Screenshot</i>		
		

m. *Black box testing* tawar harga**Tabel 4. 13** *Black box testing* tawar harga

Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengisi form tawaran dan menekan tombol simpan.	✓	Tawaran harga akan terkirim ke penjual.
<i>Screenshot</i>		
		

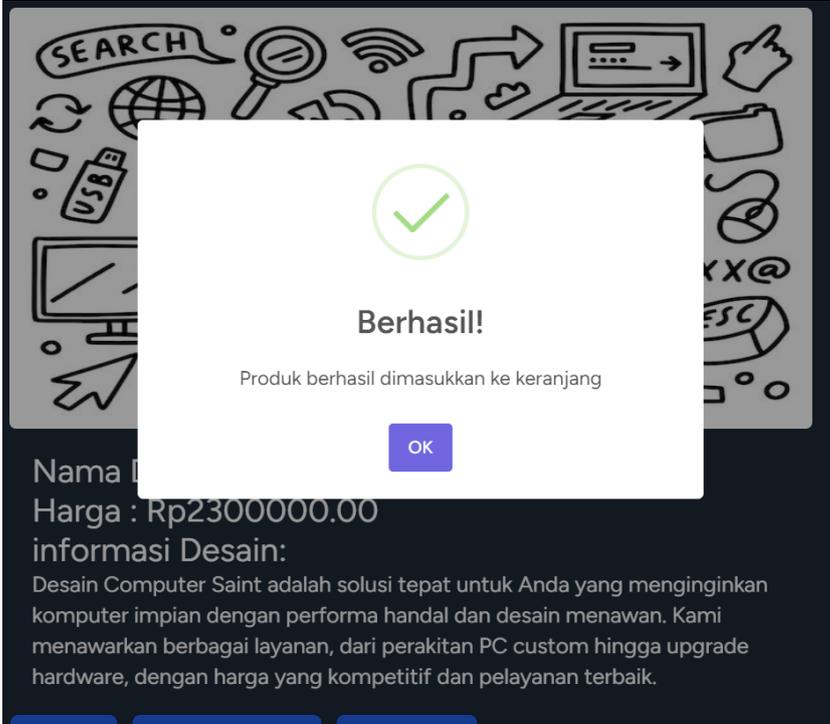
- n. *Black box* terima tawaran harga

Tabel 4. 14 *Black box testing* terima tawaran harga

Tes faktor	Hasil	Keterangan
Penjual mengklik tombol terima	✓	Harga tawar akan disejutui.
<i>Screenshot</i>		
		

- o. *Black box* tambah jasa ke keranjang

Tabel 4. 15 *Black box testing* tambah jasa ke keranjang

Tes faktor	Hasil	Keterangan
Pembeli mengklik tombol tambah ke keranjang.	✓	Jasa yg dipilih akan masuk keranjang.
<i>Screenshot</i>		
		

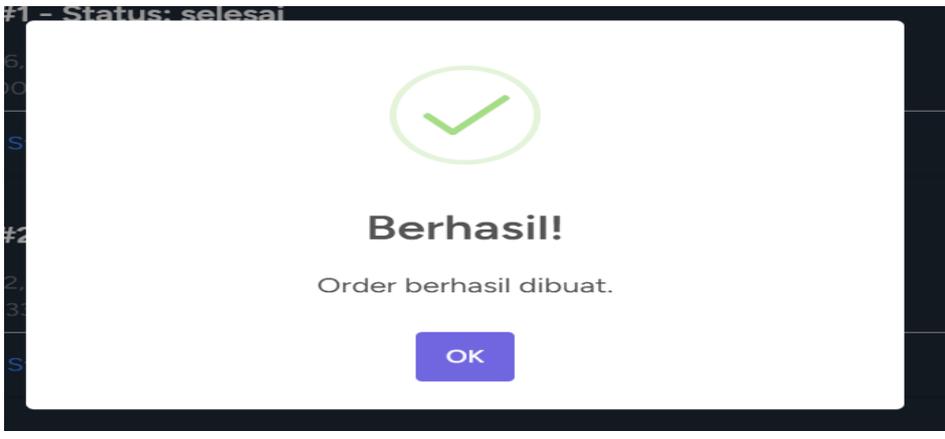
p. *Black box* keranjang

Tabel 4. 16 *Blackbox testing* keranjang

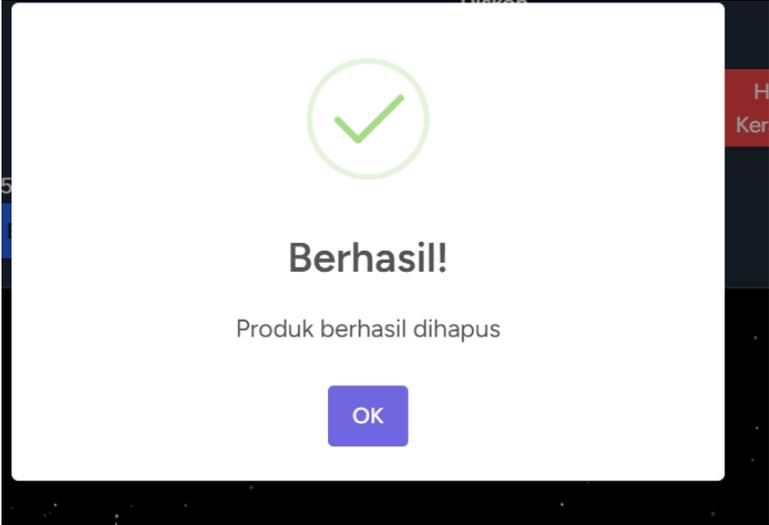
Tes faktor	Hasil	Keterangan
<i>pembeli</i> mengakses halaman keranjang.	✓	Halaman keranjang akan tampil.
<i>Screenshot</i>		
		

q. *Black box* buat orderan

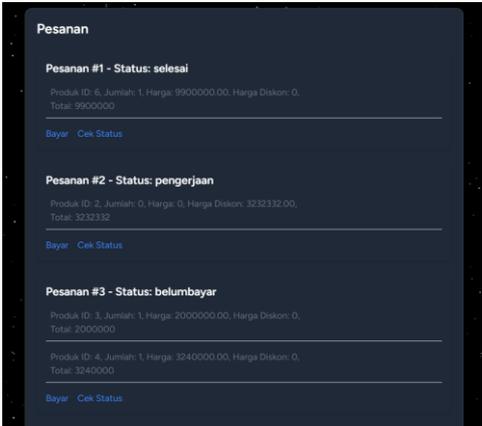
Tabel 4. 17 *Blackbox testing* buat orderan

Tes faktor	Hasil	Keterangan
<i>Pembeli</i> mengklik tombol buat orderan	✓	Semua isi keranjang akan ditambahkan ke orderan.
<i>Screenshot</i>		
		

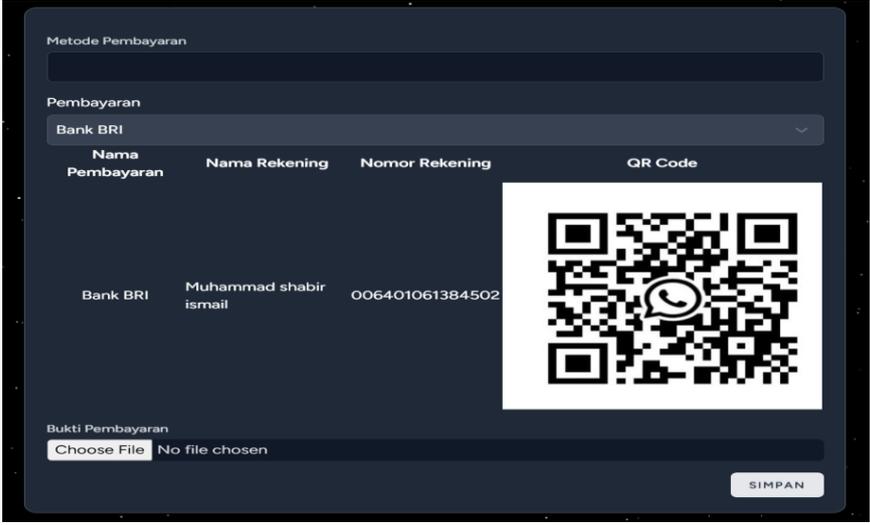
r. *Blackbox* hapus keranjang**Tabel 4. 18** *Blackbox testing* hapus keranjang

Tes faktor	Hasil	Keterangan
<i>Pembeli</i> menghapus jasa yg dipilih.	✓	jasa yg dipilih <i>pembeli</i> akan dihapus.
Screenshot		
		

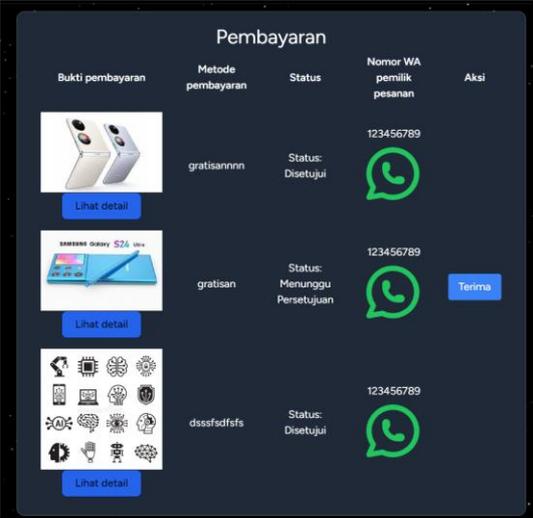
s. *Blackbox* orderan**Tabel 4. 19** *Blackbox testing* orderan

Tes faktor	Hasil	Keterangan
<i>Pembeli</i> mengakses halaman orderan.	✓	Sistem akan menampilkan semua orderan.
Screenshot		
		

t. *Blackbox* Buat pembayaran**Tabel 4. 20** *Blackbox testing* buat pembayaran

Tes faktor	Hasil	Keterangan
<i>Pembeli</i> mengisi form pembayaran.	✓	Pembayaran akan terkirim ke penjual.
<i>Screenshot</i>		
		

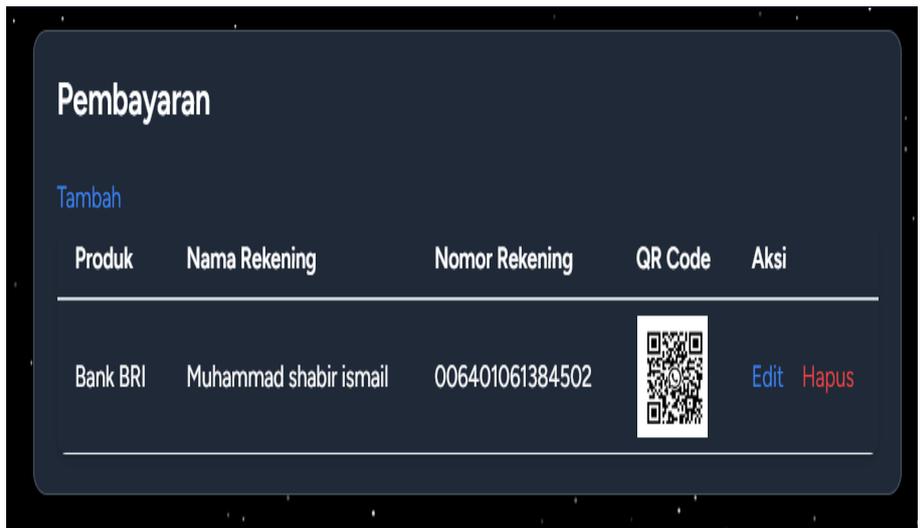
u. *Blackbox* terima pembayaran**Tabel 4. 21** *Blackbox testing* terima pembayaran

Tes faktor	Hasil	Keterangan
<i>Penjual</i> memeriksa dan menerima pembayaran.	✓	Pembayaran akan diterima.
<i>Screenshot</i>		
		

v. *Blackbox* tambah metode pembayaran**Tabel 4. 22** *Blackbox testing* tambah metode pembayaran

Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengisi form metode membayar.	✓	Metode pembayaran bertambah.
<i>Screenshot</i>		
		

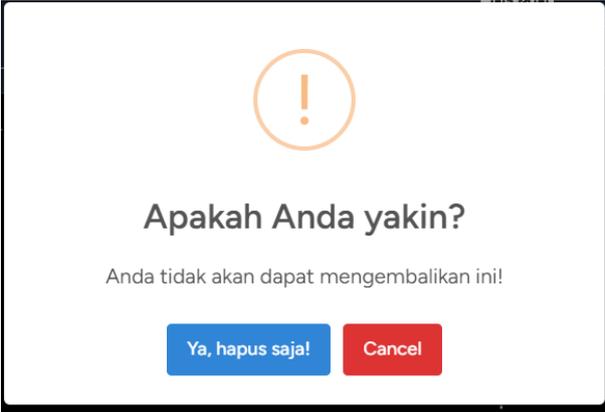
w. *Blackbox* testing metode pembayaran**Tabel 4. 23** *Blackbox testing* metode pembayaran

Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengakses metode pembayaran.	✓	Metode pembayaran akan tampil.
<i>Screenshot</i>		
		

x. *Blackbox testing* edit metode pembayaran**Tabel 4. 24** *Blackbox testing* edit metode pembayaran

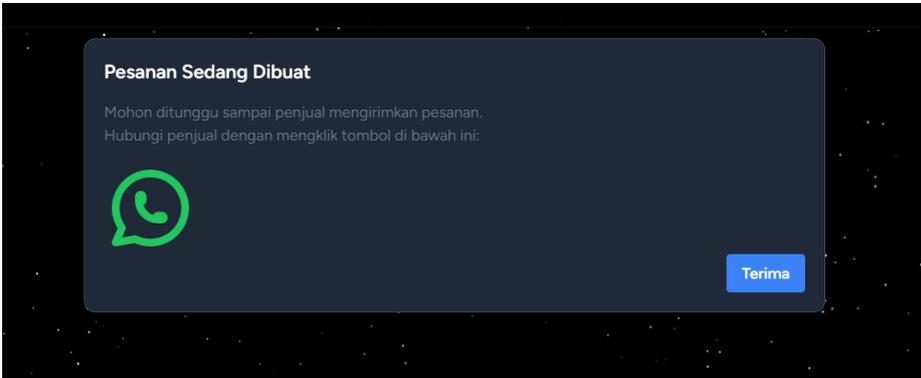
Tes faktor	Hasil	Keterangan
<i>Penjual</i> mengisi form metode pembayaran.	✓	Metode pembayaran berubah.
<i>Screenshot</i>		
		

y. *Blackbox testing* hapus**Tabel 4. 235** *Blackbox testing* hapus

Tes faktor	Hasil	Keterangan
<i>Penjual</i> pilih metode pembayaran yg ingin di hapus.	✓	Metode pembayaran yg dipilih akan dihapus.
<i>Screenshot</i>		
		

z. *Blackbox testing* konfirmasi pesanan

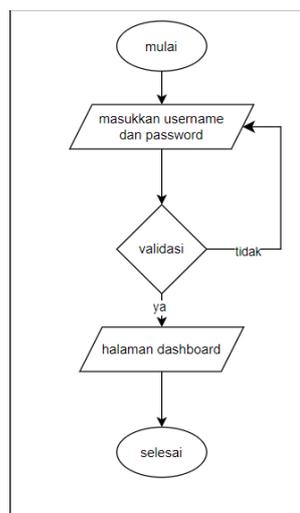
Tabel 4. 246 *Blackbox testing* konfirmasi pesanan

Tes faktor	Hasil	Keterangan
<i>Pembeli</i> menunggu pesanan dikirim dari penjual dan ketika pesanan telah dikirim, <i>pembeli</i> akan mengkonfirmasi pesannya.	✓	Status pesanan akan berubah.
Screenshot		
		

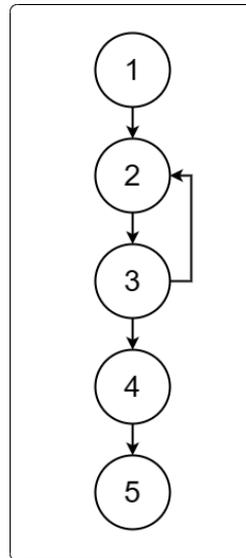
2. *White box*

a. *White box testing* kesalahan *Email* dan *password*

1) *Flowchart*



Gambar 4. 56 *Flowchart* kesalahan *Email* dan *password*

2) *Flowgraph*

Gambar 4. 57 *Flowgraph* kesalahan *Email* dan *password*

Berdasarkan gambar 4. 58 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

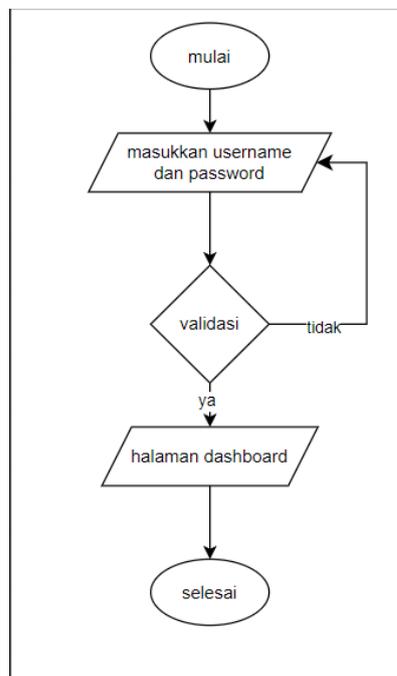
(4) Grafik matriks kesalahan *email* dan *password*

Tabel 4. 257 Grafik matriks kesalahan email dan *password*

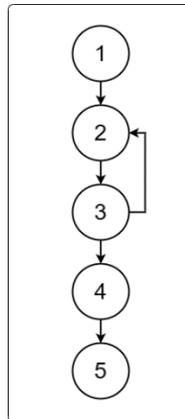
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

b. *White box testing login* berhasil

1) *Flowchart*



Gambar 4. 58 *Flowchart login* berhasil

2) *Flowgraph*

Gambar 4. 59 *Flowgraph login* berhasil

Berdasarkan gambar 4. 60 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

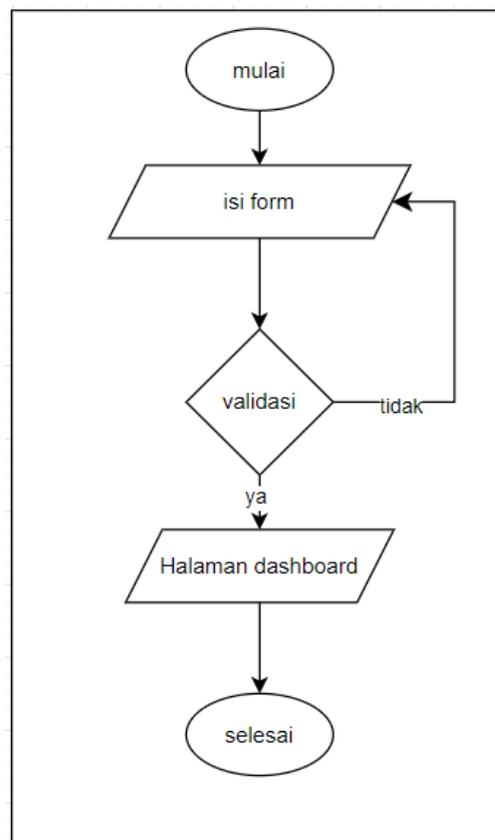
(4) Grafik matriks *login* berhasil

Tabel 4. 26 Grafik matriks *login* berhasil

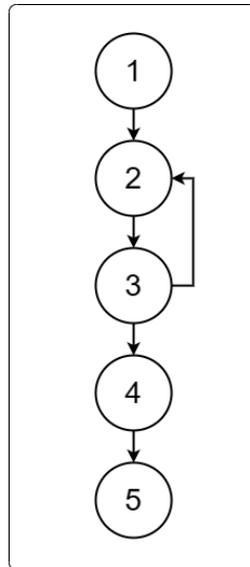
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

c. White box testing Register

1) *Flowchart*



Gambar 4. 60 *Flowchart Register*

2) *Flowgraph***Gambar 4. 61** *Flowgraph*

Berdasarkan gambar 4. 60 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 - 2 - 3 - 2

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

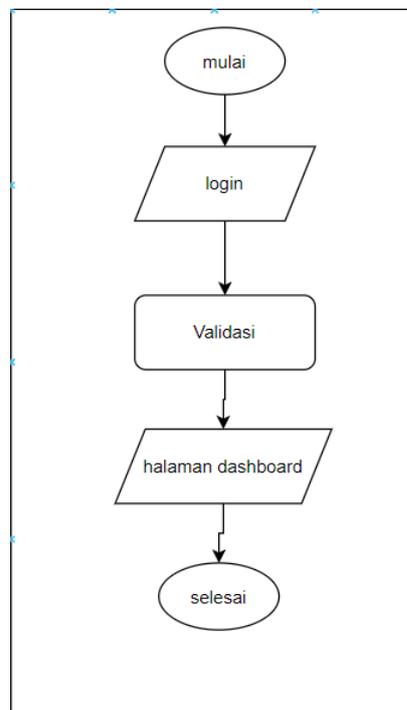
(4) Grafik matriks *Register* berhasil

Tabel 4. 27 Grafik matriks *Register* berhasil

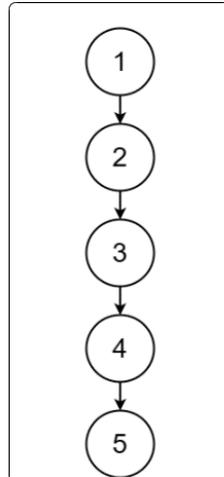
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

d. *White box testing* halaman *dashboard* utama

1) *Flowchart*



Gambar 4. 62 *Flowchart* halaman *dashboard* utama

Flowgraph**Gambar 4. 63** *Flowgraph dashboard utama*

Berdasarkan gambar 4. 62 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

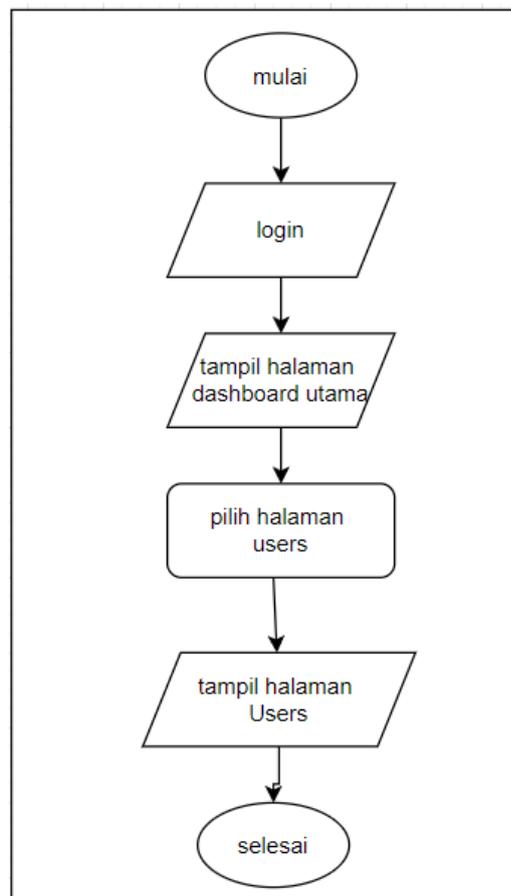
(4) Grafik matriks *halaman dashboard utama*

Tabel 4. 28 Grafik matriks halaman dashboard utama

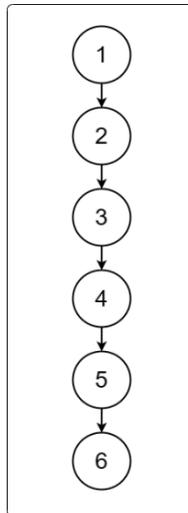
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

e. White box testing halaman *Users*

1) *Flowchart*



Gambar 4. 64 *Flowchart* halaman *User*

2) *Flowgraph*

Gambar 4. 65 *Flowgraph* halaman *Users*

Berdasarkan gambar 4. 64 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

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

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

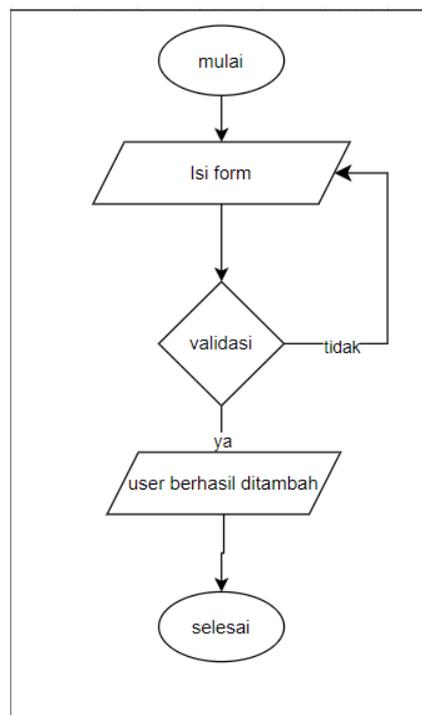
(4) Grafik matriks halaman *Users*

Tabel 4. 29 Grafik matriks halaman *Users*

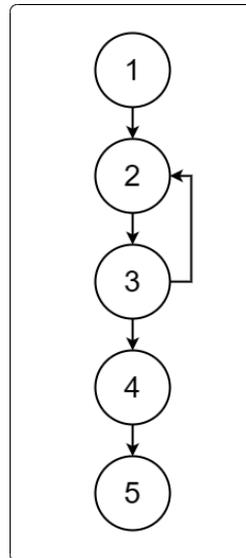
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

f. *White box testing* tambah *User*

1) *Flowchart*



Gambar 4. 66 *Flowchart* tambah *User*

2) *Flowgraph*

Gambar 4. 67 *Flowgraph* tambah *User*

Berdasarkan gambar 4. 66 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 - 2 - 3 - 2

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

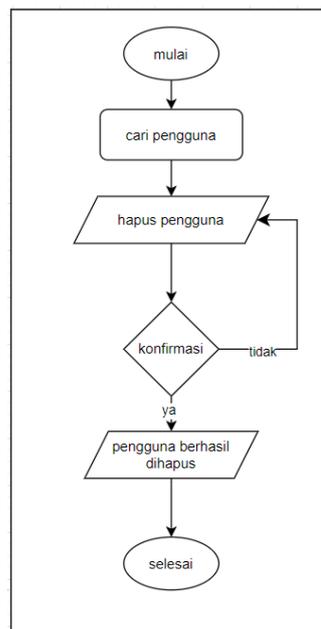
(4) Grafik matriks tambah *User*

Tabel 4. 30 Grafik matriks tambah *User*

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

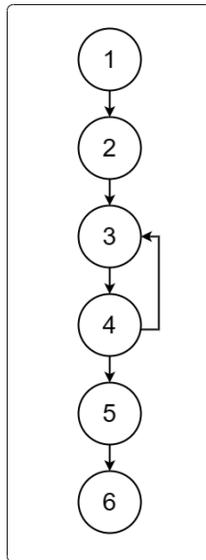
g. *White box testing* hapus *User*

1) *Flowchart*



Gambar 4. 68 *Flowchart* hapus *User*

Flowgraph



Gambar 4. 69 Flowgraph hapus User

Berdasarkan gambar 4. 70 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

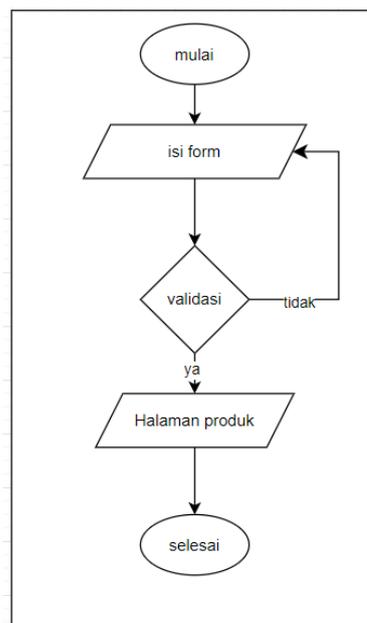
(4) Grafik matriks hapus *User*

Tabel 4. 31 Grafik matriks hapus *User*

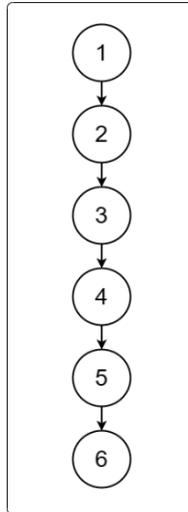
	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4			1		1		$2 - 1 = 1$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$1 + 1 = 2$

h. *White box testing* halaman jasa

1) *Flowchart*



Gambar 4. 70 *Flowchart* halaman tambah jasa

Flowgraph**Gambar 4. 71** *Flowgraph* halaman tambah jasa

Berdasarkan gambar 4. 72 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

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

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 1

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

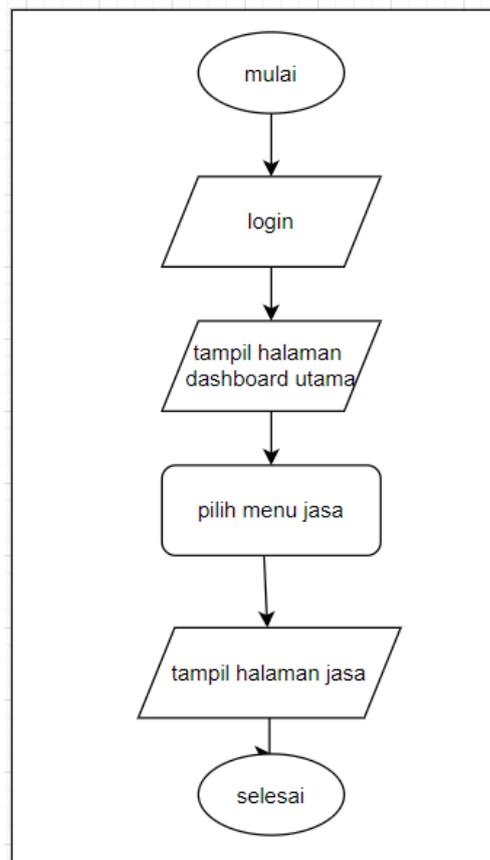
(4) Grafik matriks halaman tambah jasa

Tabel 4. 32 Grafik matriks halaman jasa

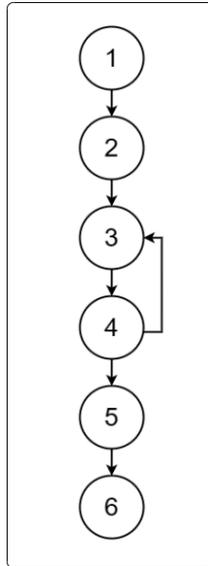
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

i. *White box testing* jasa

1) *Flowchart*



Gambar 4. 72 *Flowchart* jasa

Flowgraph**Gambar 4. 73** *Flowgraph* jasa

Berdasarkan gambar 4. 74 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 - 2 - 3 - 2

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

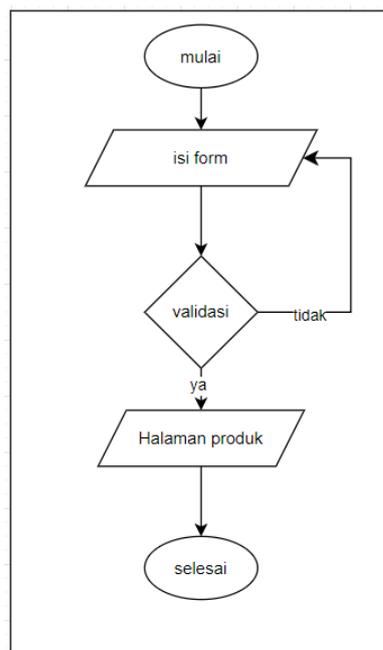
(4) Grafik matriks jasa

Tabel 4. 33 Grafik matriks jasa

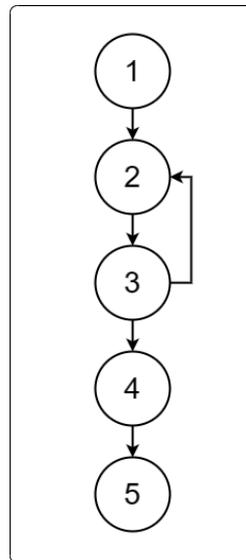
	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4			1		1		$2 - 1 = 1$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$1 + 1 = 2$

j. *White box testing* edit jasa

1) *Flowchart*



Gambar 4. 74 *Flowchart* edit jasa

Flowgraph**Gambar 4. 75** *Flowgraph* edit jasa

Berdasarkan gambar 4. 76 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

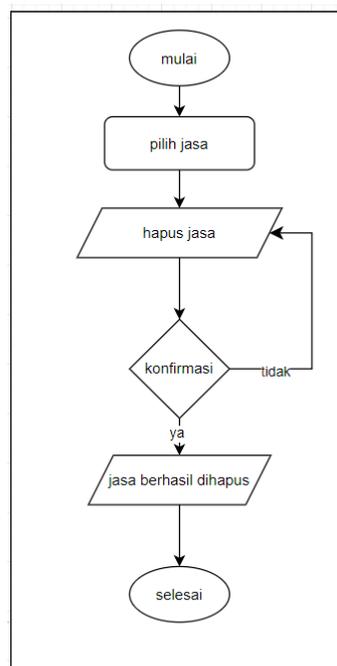
(4) Grafik matriks ubah jasa

Tabel 4. 34 Grafik matriks edit jasa

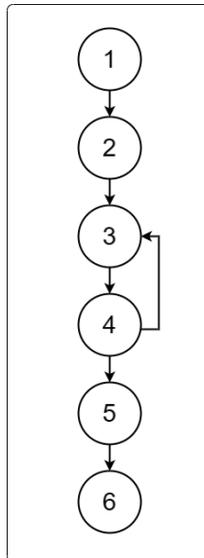
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

k. *White box testing* hapus jasa

1) *Flowchart*



Gambar 4. 76 *Flowchart* hapus jasa

2) *Flowgraph***Gambar 4. 77** *Flowgraph* hapus jasa

Berdasarkan gambar 4. 78 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

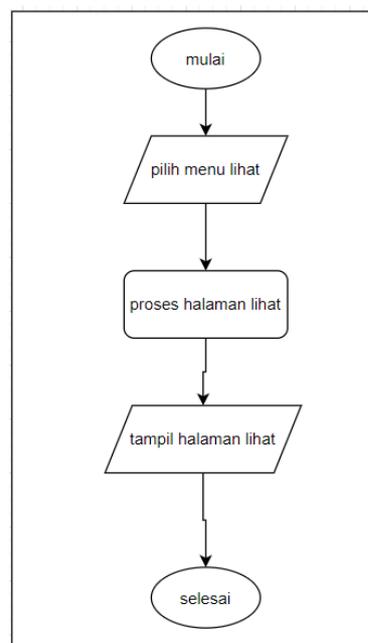
(4) Grafik matriks hapus jasa

Tabel 4. 35 Grafik matriks hapus jasa

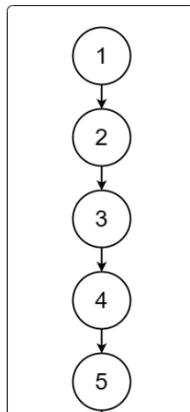
	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4			1		1		$2 - 1 = 1$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$1 + 1 = 2$

1. *White box testing* halaman informasi jasa

1) *Flowchart*



Gambar 4. 78 *Flowchart* halaman informasi jasa

Flowgraph

Gambar 4. 79 *Flowgraph* halaman informasi jasa

Berdasarkan gambar 4. 80 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

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

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 1

(3) *Independent path* pada *Flowgraph* yaitu:

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

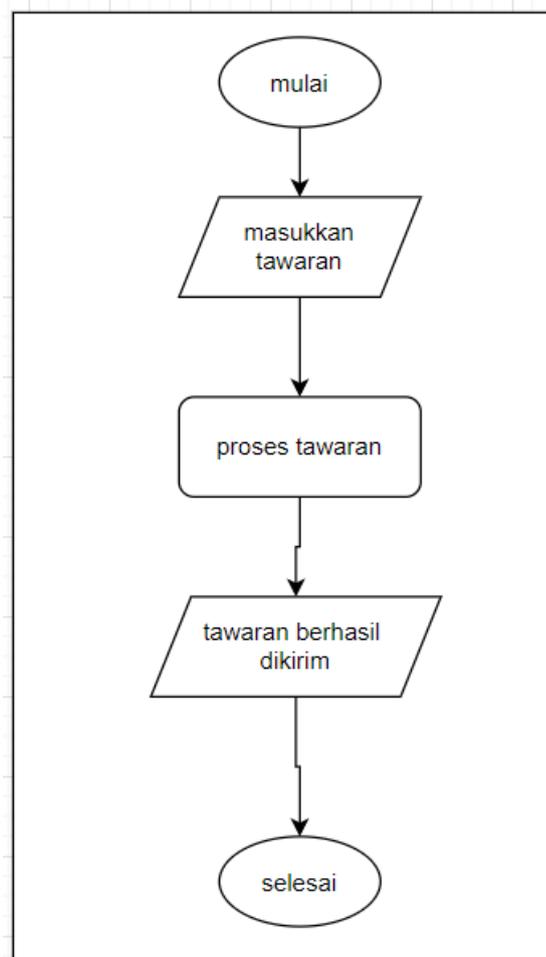
(4) Grafik matriks halaman informasi jasa

Tabel 4. 36 Grafik matriks halaman informasi jasa

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

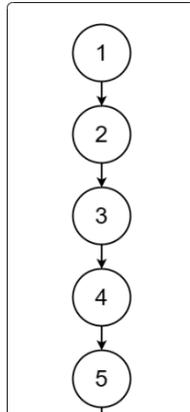
m. *White box testing* tawar harga

1) *Flowchart*



Gambar 4. 80 *Flowchart* tawar harga

Flowgraph



Gambar 4. 81 *Flowgraph* tawar harga

Berdasarkan gambar 4. 82 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

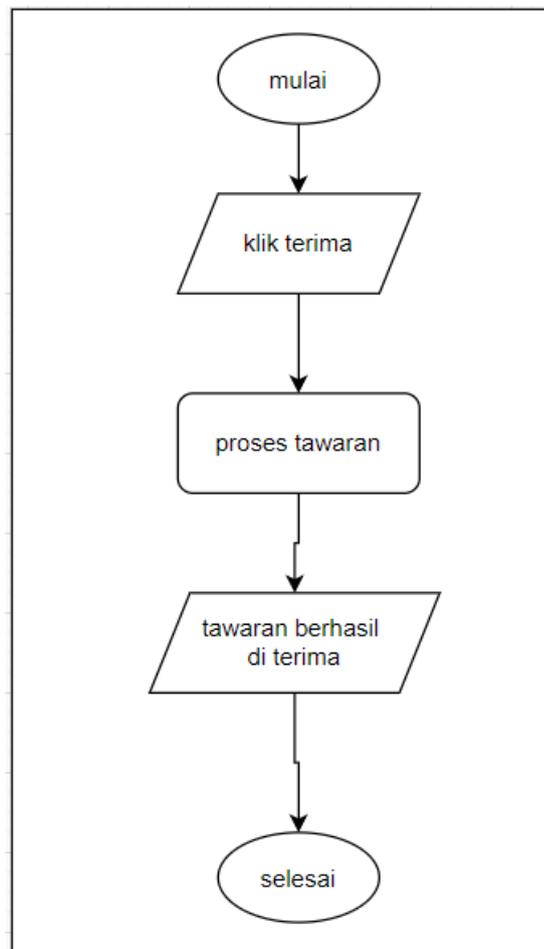
(4) Grafik matriks tambah beasiswa

Tabel 4. 37 Grafik matriks tawar harga

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

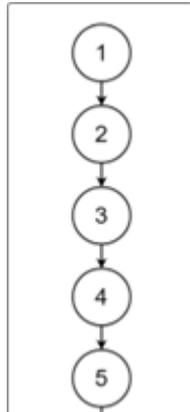
n. *White box testing* verifikasi tawaran

1) *Flowchart*



Gambar 4. 82 *Flowchart* verifikasi tawaran

Flowgraph



Gambar 4. 83 *Flowgraph* terima tawaran

Berdasarkan gambar 4. 84 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

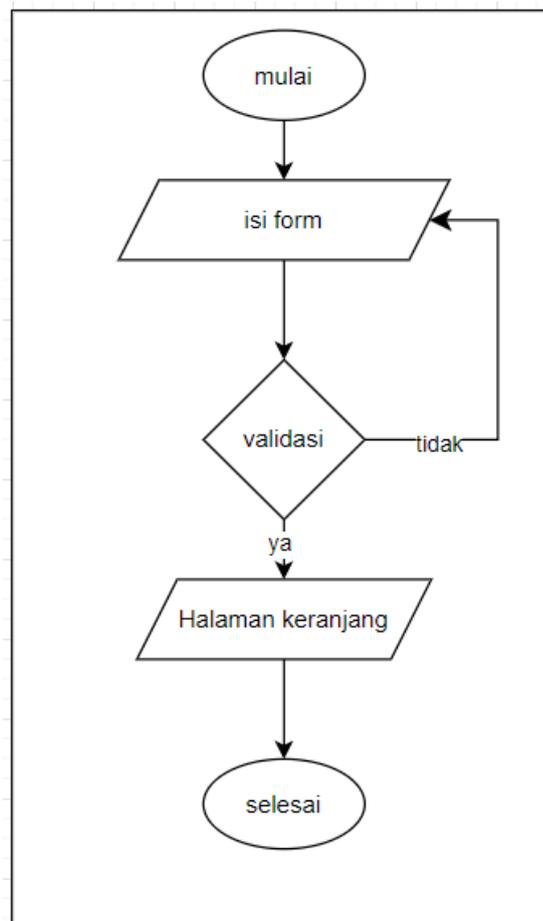
(4) Grafik matriks terima tawaran

Tabel 4. 38 Grafik matriks terima tawaran

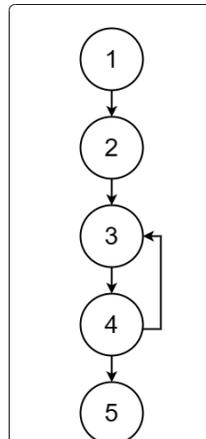
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

o. *White box testing* tambah keranjang

1) *Flowchart*



Gambar 4. 84 *Flowchart* tambah keranjang

2) *Flowgraph***Gambar 4. 85** *Flowgraph* tambah keranjang

Berdasarkan gambar 4. 86 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 3

Path 2 = 1 – 2 – 3 – 4 – 5

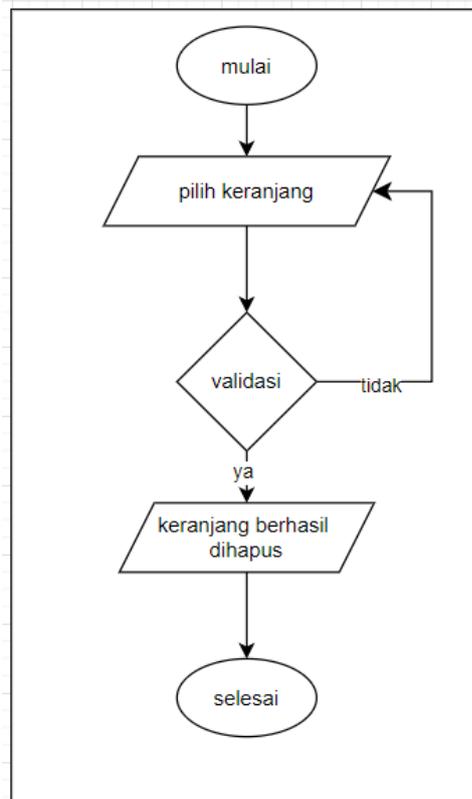
(4) Grafik matriks tambah keranjang

Tabel 4. 39 Grafik matriks tambah keranjang

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

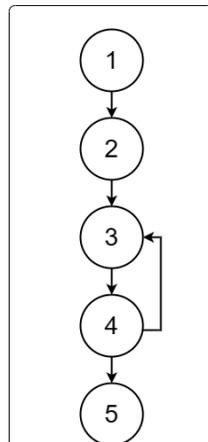
p. *White box testing* hapus keranjang

1) *Flowchart*



Gambar 4. 86 *Flowchart* halaman hapus keranjang

Flowgraph



Gambar 4. 87 Flowgraph halaman hapus keranjang

Berdasarkan gambar 4. 88 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

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

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari Flowgraph diatas

memiliki *region* = 1

(3) *Independent path* pada Flowgraph yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

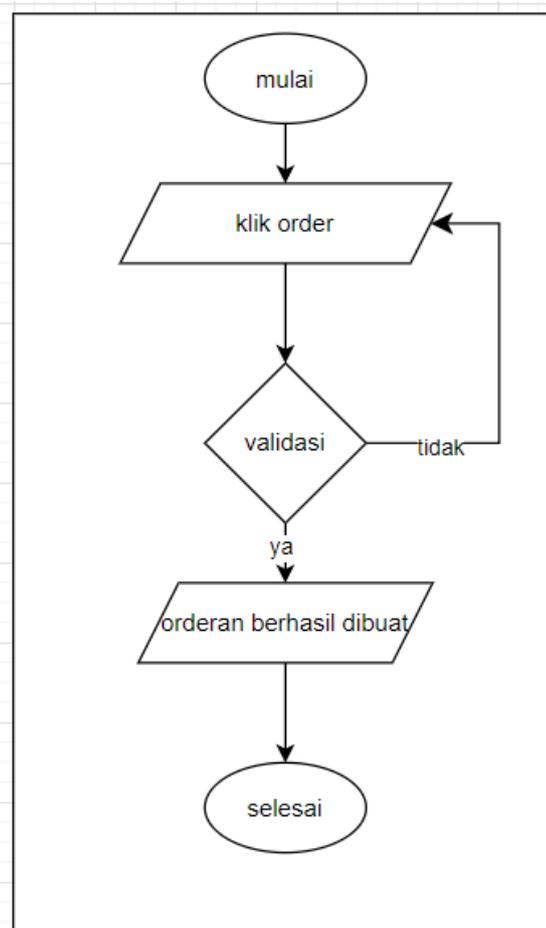
(4) Grafik matriks halaman hapus keranjang

Tabel 4. 40 Grafik matriks halaman hapus keranjang

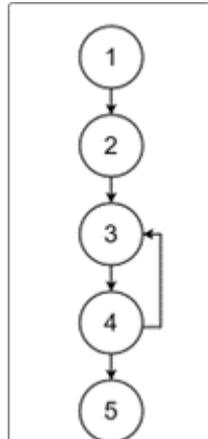
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
SUM (E + 1)						$0 + 1 = 1$

q. *White box testing* tambah orderan

1) *Flowchart*



Gambar 4. 88 *Flowchart* halaman tambah orderan

2) *Flowgraph*

Gambar 4. 89 *Flowgraph* halaman tambah orderan

Berdasarkan gambar 4. 90 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

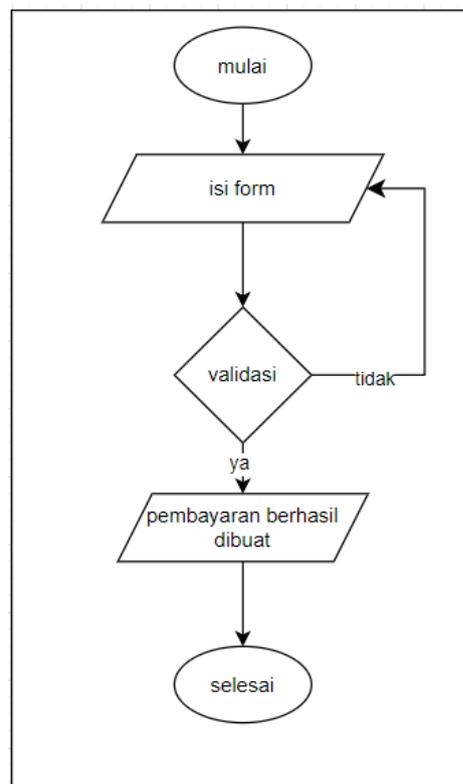
(4) Grafik matriks halaman tambah orderan

Tabel 4. 41 Grafik matriks halaman tambah orderan

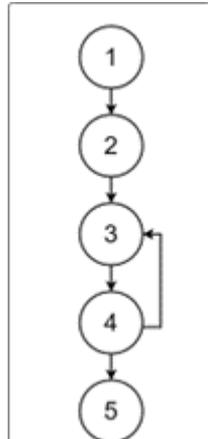
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

r. *White box testing* buat pembayaran

1) *Flowchart*



Gambar 4. 90 *Flowchart* halaman buat pembayaran

2) *Flowgraph*

Gambar 4. 91 *Flowgraph* halaman buat pembayaran

Berdasarkan gambar 4. 92 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

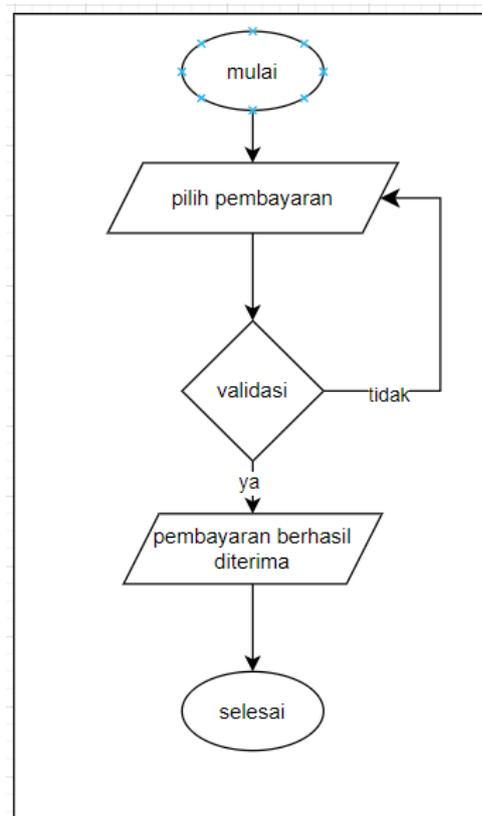
(4) Grafik matriks halaman buat pembayaran

Tabel 4. 42 Grafik matriks halaman buat pembayaran

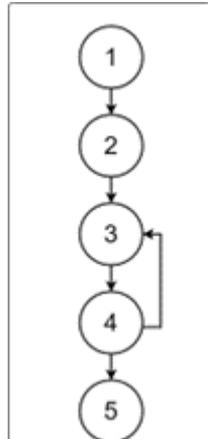
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

s. *White box testing* terima pembayaran

1) *Flowchart*



Gambar 4. 92 *Flowchart* halaman terima pembayaran

2) *Flowgraph*

Gambar 4. 93 *Flowgraph* halaman terima pembayaran

Berdasarkan gambar 4. 94 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

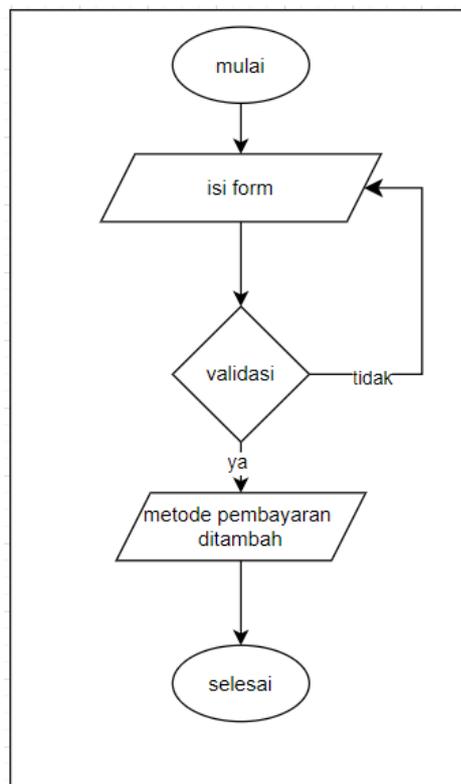
(4) Grafik matriks halaman terima pembayaran

Tabel 4. 43 Grafik matriks halaman terima pembayaran

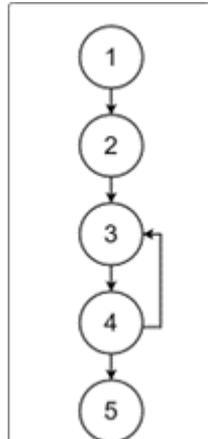
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

t. *White box testing* tambah metode pembayaran

1) *Flowchart*



Gambar 4. 94 *Flowchart* halaman tambah metode pembayaran

2) *Flowgraph*

Gambar 4. 95 *Flowgraph* halaman tambah metode pembayaran

Berdasarkan gambar 4. 96 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

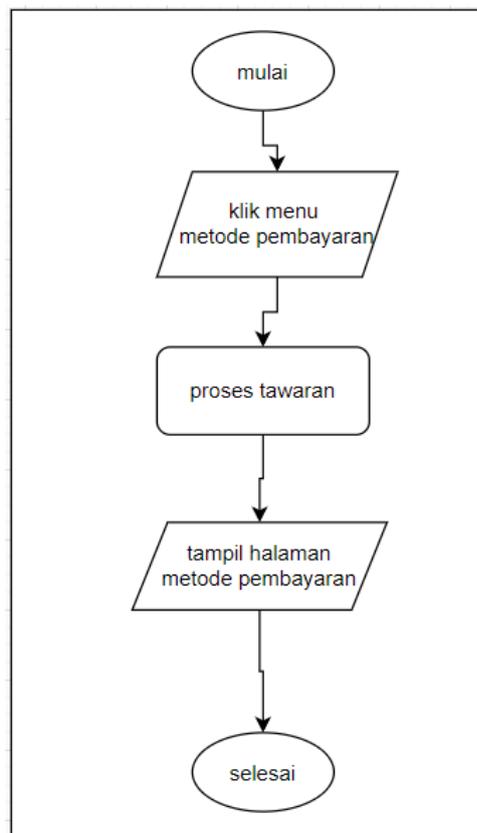
(4) Grafik matriks halaman tambah metode pembayaran

Tabel 4. 44 Grafik matriks halaman tambah metode pembayaran

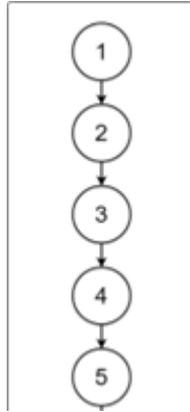
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

u. *White box testing* metode pembayaran

1) *Flowchart*



Gambar 4. 96 *Flowchart* metode pembayaran

2) *Flowgraph*

Gambar 4. 97 *Flowgraph* metode pembayaran

Berdasarkan gambar 4. 98 diatas, dilakukan perhitungan sebagai berikut:

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

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

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

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

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

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

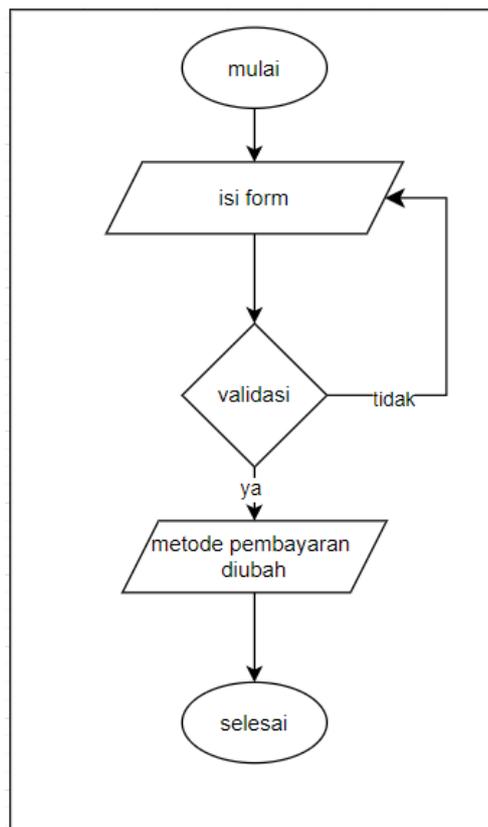
(4) Grafik matriks metode pembayaran

Tabel 4. 45 Grafik matriks metode pembayaran

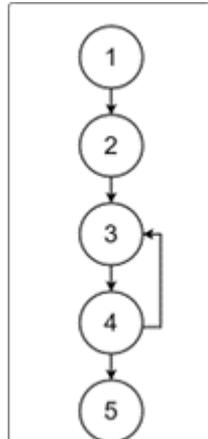
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

v. *White box* edit metode pembayaran

1) *Flowchart*



Gambar 4. 98 *Flowchart* halaman edit metode pembayaran

2) *Flowgraph*

Gambar 4. 99 *Flowgraph* halaman edit metode pembayaran

Berdasarkan gambar 4. 100 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

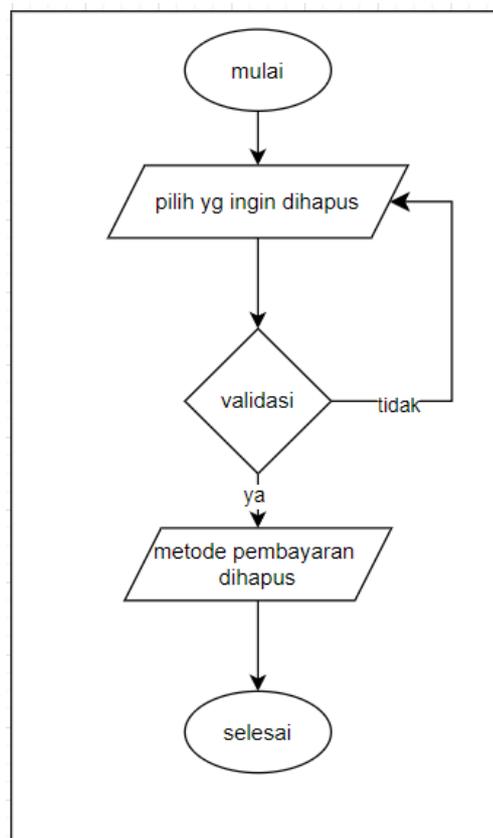
(4) Grafik matriks halaman edit metode pembayaran

Tabel 4. 46 Grafik matriks halaman edit metode pembayaran

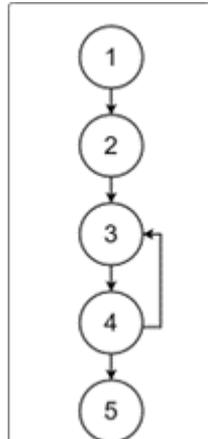
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

w. *White box testing* hapus metode pembayaran

1) *Flowchart*



Gambar 4. 100 *Flowchart* halaman hapus metode pembayaran

2) *Flowgraph*

Gambar 4. 101 *Flowgraph* halaman hapus metode pembayaran

Berdasarkan gambar 4. 102 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

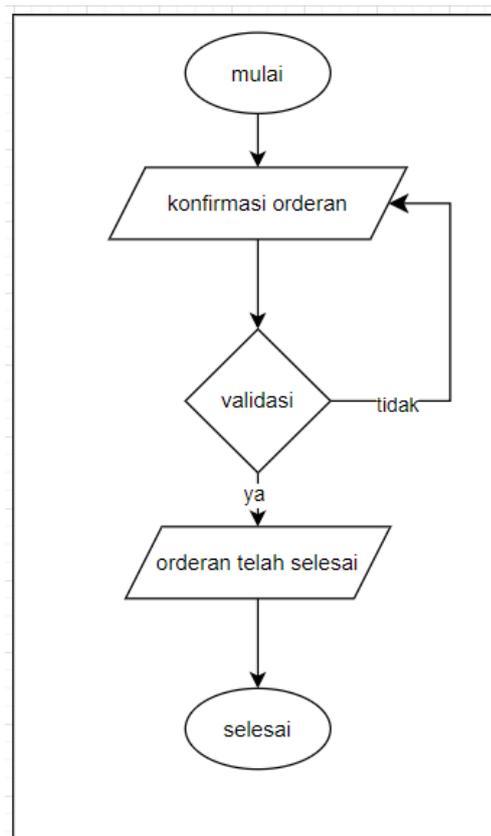
(4) Grafik matriks halaman hpus metode pembayaran

Tabel 4. 47 Grafik matriks hapus edit metode pembayaran

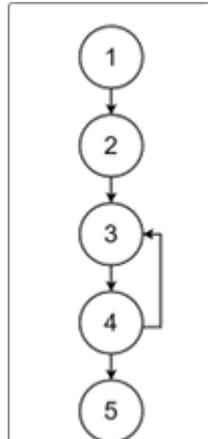
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

x. *White box testing* Konfirmasi orderan

1) *Flowchart*



Gambar 4. 102 *Flowchart* halaman konfirmasi orderan

2) *Flowgraph*

Gambar 4. 103 *Flowgraph* halaman konfirmasi orderan

Berdasarkan gambar 4. 104 diatas, dilakukan perhitungan sebagai berikut:

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

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

$$E \text{ (edge)} = 4$$

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

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

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 2$$

$$= 0 + 2$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas

memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

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

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

(4) Grafik matriks halaman konfirmasi orderan

Tabel 4. 48 Grafik matriks konfirmasi orderan

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

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini bertujuan untuk mengembangkan aplikasi e-marketplace desain grafis berbasis web yang dapat membantu para desainer grafis dan konsumen dalam berinteraksi dan menawarkan/jual layanan desain grafis. Aplikasi ini diharapkan dapat menjembatani kebutuhan desainer grafis yang ingin mempromosikan jasanya dan konsumen yang mencari layanan desain berkualitas. Metode pengembangan yang digunakan adalah Rapid Application Development (RAD) dengan tahapan perencanaan kebutuhan, lokakarya desain RAD, dan implementasi. Pendekatan ini dipilih karena memungkinkan pengembangan aplikasi yang cepat dan fleksibel, sesuai dengan kebutuhan pasar yang dinamis.

Penelitian dilaksanakan di Universitas Muhammadiyah Parepare menggunakan laptop Asus dengan spesifikasi minimal, menunjukkan bahwa aplikasi ini dapat dikembangkan dengan sumber daya yang terjangkau. Hasil penelitian menunjukkan bahwa aplikasi ini dapat membantu mahasiswa mengelola bantuan yang diterima secara lebih efektif dan efisien, meskipun beberapa fitur aplikasi masih memerlukan penyempurnaan. Hal ini mengindikasikan potensi aplikasi untuk digunakan dalam konteks akademis dan profesional.

Penelitian ini diharapkan dapat menjadi dasar untuk pengembangan lebih lanjut dalam meningkatkan fungsionalitas dan kegunaan aplikasi pengelolaan

bantuan mahasiswa. Pengembangan di masa depan dapat mencakup integrasi dengan platform media sosial, sistem pembayaran yang lebih canggih, dan fitur kolaborasi real-time antara desainer dan klien. Secara umum, dapat disimpulkan bahwa aplikasi e-marketplace desain grafis berbasis web ini dapat membantu dalam meningkatkan efisiensi pengelolaan dan pemanfaatan layanan desain grafis, serta mempermudah interaksi antara desainer dan konsumen. Dengan terus dikembangkan, aplikasi ini berpotensi untuk mengubah cara industri desain grafis beroperasi, membuka peluang baru bagi desainer freelance dan memperluas akses konsumen ke berbagai layanan desain.

B. Saran

Pengembangan lanjutan aplikasi e-marketplace desain grafis berbasis web ini merupakan langkah krusial untuk memastikan efektivitas dan kegunaan yang optimal dalam skala yang lebih luas. Implementasi dan pengujian dalam berbagai konteks dan lingkungan yang beragam sangat diperlukan untuk mengidentifikasi potensi masalah dan area perbaikan yang mungkin tidak terlihat dalam pengujian terbatas. Proses ini akan membantu dalam mengumpulkan umpan balik yang berharga dari berbagai pengguna dengan kebutuhan dan preferensi yang berbeda-beda. Selain itu, beberapa fitur aplikasi masih memerlukan penyempurnaan untuk meningkatkan fungsionalitas dan kegunaannya. Hal ini mencakup perbaikan antarmuka pengguna untuk meningkatkan pengalaman pengguna, optimalisasi kinerja sistem untuk menangani volume transaksi yang lebih besar, dan penambahan fitur-fitur inovatif yang dapat membedakan aplikasi ini dari

kompetitor di pasar. Penyempurnaan ini juga harus mempertimbangkan aspek keamanan dan privasi data pengguna, yang merupakan faktor kritis dalam membangun kepercayaan pengguna. Dengan melakukan pengembangan lanjutan dan penyempurnaan fitur secara berkelanjutan, aplikasi ini diharapkan dapat menjadi solusi komprehensif yang memenuhi kebutuhan dinamis industri desain grafis dan memberikan nilai tambah yang signifikan bagi para penggunanya.

DAFTAR PUSTAKA

- Abdul Kahfi Assidiq, (2020) dalam kamus Biologi, laboratorium, Jakatra: MediaKompitindo
- Ahmad Sidqi (2020) program studi teknik informatika – program sarjana fakultas teknologi industry Universitas islam indonesia, Melakukan Penelitian Yang Berjudul “Sistem Monitoring Hafalan Al-Qur’an Santri Mts Pondok Pesantren Sunan Pandanaran”
- Alda, M., Krisna Maulana, D., Abdillah, M. D., Hidayat, R., Studi, P., Informasi, S., Sains, F., Teknologi, D., Negeri, I., & Utara, S. (2024). Membangun Aplikasi Pencarian Wisata Top di Sumatera Utara Berbasis Mobile Menggunakan Kodular. *Jurnal Pendidikan Tambusai*, 8(1), 2936–2947.
- Astuti, Wati Aris, and A. R. A. H. (2023). *Cibogohilir vellagers products and marketplace Adeh Ratna Komala et al. www.ieeca.org/journal 136. 10(1), 135–145. www.ieeca.org/journal*
- Birkle, C., Pendlebury, D. A., Schnell, J., & Adams, J. (2020). Web of science as a data source for research on scientific and scholarly activity. *Quantitative Science Studies*, 1(1), 363–376. https://doi.org/10.1162/qss_a_00018
- Deng, X., Gu, Y., Zheng, B., & Chen, S. (2023). M IND 2W EB : Towards a Generalist Agent for the Web. *Nips, NeurIPS*.
- Granitzer, M., Voigt, S., Fathima, N. A., Golasowski, M., Guetl, C., Hecking, T., Hendriksen, G., Hiemstra, D., Martinovič, J., Mitrović, J., Mlakar, I., Moiras, S., Nussbaumer, A., Öster, P., Potthast, M., Srdič, M. S., Megi, S., Slaninová, K., Stein, B., ... Zerhoudi, S. (2024). Impact and development of an Open Web Index for open web search. *Journal of the Association for Information Science and Technology*, 75(5), 512–520. <https://doi.org/10.1002/asi.24818>
- Pranoto, S., Sutiono, S., Sarifudin, & Nasution, D. (2024). Penerapan UML Dalam Perancangan Sistem Informasi Pelaporan Dan Evaluasi Pembangunan Pada Bagian Administrasi Pembangunan Sekretariat Daerah Kota Tebing Tinggi. *Surplus: Jurnal Ekonomi Dan Bisnis*, 2(2), 384–401. <https://qjurnal.my.id/index.php/sur/article/view/866>
- Putri, A. N., Al Hidayah, H., Hidayatullah, H. M., Asnidar, A., & Nasir, N. (2024). (Putri et al., 2024). *Jurnal Kajian Penelitian Pendidikan Dan Kebudayaan (JKPPK)*, 2(2). <https://doi.org/10.59031/jkppk.v2i2.335>
- Ridwan, R., & Mutia, C. (2022). Pengembangan Aplikasi Web dan Mobile Pada

Monitoring dan Evaluasi Program Tahfidz Quran. *Circuit: Jurnal Ilmiah Pendidikan Teknik Elektro*, 6(1), 47. <https://doi.org/10.22373/crc.v6i1.11700>

Syukroni, M. F., Sains, J., Informasi, S., Ruang, P., Bagi, P., Virtual, P., Pada, O., Belanja, S., Untuk, O., Swalayan, P., Web, B., Stok, P. S., Sistem, P., Keputusan, P., Menggunakan, B., Weighted, M., Dan, P., Additive, S., Berbasis, W., ... Jurnal, R. (2017). Rancang Bangun Knowledge Management Sistem Berbasis Web Pada Madrasah Muallimin Al-Islamiyah Uteran Geger Madiun. *Teknik Informatika Universitas Muhammadiyah Ponorogo*, 7–35. <http://eprints.umpo.ac.id/3019/>

Vallabhaneni, R., Vadakkethil Somanathan Pillai, S. E., Vaddadi, S. A., Addula, S. R., & Ananthan, B. (2024). Secured web application based on CapsuleNet and OWASP in the cloud. *Indonesian Journal of Electrical Engineering and Computer Science*, 35(3), 1924–1932. <https://doi.org/10.11591/ijeecs.v35.i3.pp1924-1932>

WIDIARTO, H. (2023). Rancangan Mock Up Sistem Monitoring Panel Tegangan Rendah Berbasis Transmission Control Protocol / Internet Protocol (Tcp/Ip). *KNOWLEDGE: Jurnal Inovasi Hasil Penelitian Dan Pengembangan*, 3(1), 42–50. <https://doi.org/10.51878/knowledge.v3i1.2136>