BAB I

PENDAHULUAN

A. Latar Belakang

Arbas *Cafe* merupakan salah satu kafe yang ada di kota Enrekang. Kafe ini menyediakan berbagai macam menu makanan dan minuman yang digemari oleh masyarakat. Saat ini, Arbas *Cafe* masih menggunakan sistem pemesanan menu secara manual, yaitu dengan memanggil pelayan untuk memesan menu. Sistem pemesanan menu secara manual ini memiliki beberapa kelemahan, yaitu:

1. Membutuhkan waktu yang lama untuk melakukan pemesanan, terutama jika kafe sedang ramai.
2. Kesalahan dalam mencatat pesanan pelanggan dapat terjadi.
3. Pelayan harus bolak-balik ke dapur untuk menyampaikan pesanan pelanggan.

Untuk mengatasi kelemahan-kelemahan tersebut, Arbas *Cafe* perlu menerapkan sistem pemesanan menu yang lebih efisien. Salah satu solusi yang dapat diterapkan adalah menggunakan *multi channel*.

Menggunakan *Multi channel* untuk pemesanan menu di *Cafe* adalah sistem yang memungkinkan pelanggan untuk memesan menu dari berbagai saluran, seperti situs web atau kode QR. Sistem ini dapat memberikan pengalaman pelanggan yang lebih baik dengan memungkinkan pelanggan memesan menu dengan cara yang paling nyaman bagi mereka.

Berikut adalah beberapa manfaat dari implementasi sistem *Multi channel* untuk pemesanan menu di *cafe*:

1. Menjangkau pelanggan yang lebih luas.
2. Meningkatkan pengalaman pelanggan.
3. Meningkatkan efisiensi
4. Meningkatkan penjualan

Berdasarkan hal tersebut, penulis tertarik untuk melakukan penelitian dengan judul Implementasi Sistem Pemesanan Menu Menggunakan metode *Multi channel* Pada Arbas *Cafe*. Penelitian ini bertujuan untuk menerapkan sistem pemesanan menu menggunakan *Multi channel* di Arbas *Cafe*.

B. Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka permasalahan yang muncul pada penelitian ini adalah:

1. Bagaimana cara menerapkan sistem pemesanan menu menggunakan metode *Multi channel* di Arbas *Cafe*?
2. Apakah sistem pemesanan menu menggunakan metode *Multi channel* dapat meningkatkan efisiensi waktu pemesanan di Arbas *Cafe*?

C. Batasan Masalah

Mengingat keterbatasan kemampuan peneliti serta menghindari keluasan dan agar tidak menyimpang dari ruang lingkup penelitian, maka peneliti membatasi masalah sebagai berikut:

a. Penelitian ini hanya dilakukan pada Arbas *Cafe* sebagai objek penelitian.

b. Penelitian hanya berfokus pada pemesanan menu.

D. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah menghasilkan sistem yang memungkinkan pelanggan untuk dengan mudah mengakses menu kafe dan melakukan pemesanan melalui perangkat mereka sendiri.

E. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini yaitu:

1. Bagi penulis.

Dapat sebagai penerapan ilmu pengetahuan yang telah didapat selama berada di bangku perkuliahan terutama pada bidang Teknik Informatika , serta juga sebagai syarat dari kelulusan program Strata I Program studi Teknik Informatika , Fakultas Teknik, Universitas Muhammadiyah Parepare.

2. Bagi Pengguna

Arbas *Cafe* akan dapat memberikan layanan yang lebih baik kepada pelanggan dengan memungkinkan mereka untuk dengan cepat mengakses menu dan melakukan pemesanan melalui perangkat mereka sendiri. Mereka tidak perlu menunggu menu fisik atau mengajukan pertanyaan kepada pelayan, sehingga menghemat waktu.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Aplikasi

Aplikasi merupakan program siap pakai yang dapat dijalankan perintah dari pengguna aplikasi yang dimaksudkan untuk memperoleh suatu hasil mana yang lebih akurat tergantung tujuan aplikasinya artinya menggunakan metode untuk memecahkan suatu masalah pengolahan data aplikasi yang sering mengandalkan komputasi ekspektasi dan pengolahan data yang diantisipasi.

Pengertian umum aplikasi adalah alat aplikasi yang digunakan dalam beberapa cara Istimewa dan terintegrasi sesuai fungsinya, aplikasi tersebut peralatan komputer tersedia untuk pengguna.

Pengertian aplikasi menurut para ahli:

1.) Suhardi & Fatkhurrahman (2020), Aplikasi adalah perangkat lunak yang dibuat untuk membantu pengguna melakukan tugas tertentu, baik di perangkat desktop maupun perangkat mobile, dan biasanya memiliki antarmuka pengguna yang interaktif.

2.) Wahyuni & Hidayat (2021), Aplikasi merupakan program komputer yang dirancang untuk menjalankan fungsi tertentu, seperti pengolahan data, komunikasi, maupun hiburan, dan biasanya dijalankan melalui sistem operasi tertentu.

3.) Andriani et al. (2022), Aplikasi adalah kumpulan fitur digital yang diintegrasikan dalam satu sistem guna mendukung pengguna dalam menyelesaikan permasalahan atau kebutuhan tertentu dengan memanfaatkan teknologi informasi.

4.) Maulana & Widodo (2023), Aplikasi merupakan perangkat lunak berbasis mobile atau web yang dibuat untuk menyelesaikan pekerjaan secara otomatis, efisien, dan dapat diakses dengan mudah melalui perangkat elektronik.

2. Implementasi

Dalam bahasa implementasi berarti pelaksanaan, penerapan. Secara umum, implementasi adalah tindakan atau pelaksanaan rencana yang telah disiapkan secara matang, hati-hati dan teliti. Implementasi adalah proses pelaksanaan atau penerapan suatu rencana, kebijakan, atau program. Implementasi merupakan tahap penting dalam proses manajemen, karena keberhasilan suatu rencana atau kebijakan ditentukan oleh efektivitas implementasi. Secara etimologis realisasi berasal dari bahasa Inggris yaitu *to realisasi* yang artinya melaksanakan. Implementasi menyediakan sarana untuk mencapai sesuatu yang mempunyai dampak atau konsekuensi pada seseorang. Dalam konteks manajemen, implementasi dapat diartikan sebagai proses penerjemahan rencana atau kebijakan menjadi tindakan nyata. Implementasi melibatkan berbagai pihak, mulai dari pembuat kebijakan, pelaksana kebijakan, hingga masyarakat sasaran. Tujuan implementasi adalah untuk mewujudkan tujuan dari rencana atau kebijakan yang telah ditetapkan. Implementasi yang efektif akan menghasilkan perubahan yang diinginkan sesuai dengan tujuan yang ditetapkan.

3. Sistem

Sistem adalah jaringan prosedur yang saling terkait yang disatukan untuk melakukan suatu aktivitas atau mencapai tujuan tertentu. Penjelasan di atas menjelaskan bahwa suatu sistem beroperasi dalam suatu jaringan prosedur yang saling berhubungan untuk mencapai tujuan dan sasaran yang dimaksudkan.

Sistem mempunyai beberapa arti. Menurut Sutrisno & Hidayat (2020), "Sistem adalah sekumpulan elemen yang saling terhubung dan bekerja sama secara terpadu untuk mencapai tujuan tertentu melalui proses yang terstruktur dan terorganisir. Ramdani & Lestari (2021), kemudian menjelaskan bahwa Sistem merupakan serangkaian prosedur atau metode yang saling berinteraksi dan digunakan untuk menjalankan suatu fungsi atau tujuan tertentu. Sedangkan menurut Indrawan & Prasetyo (2022), Sistem adalah mekanisme atau tata cara kerja yang terdiri dari komponen-komponen yang saling mendukung dalam suatu kesatuan proses yang bertujuan menghasilkan output tertentu."

4. Pemesanan Menu

Pemesanan menu adalah proses yang dilakukan oleh pelanggan untuk memesan makanan dan minuman yang tersedia di daftar menu. Proses pemesanan menu biasanya dilakukan di tempat makan, baik secara langsung maupun melalui telepon atau aplikasi pemesanan makanan daring. Pemesanan menu merupakan salah satu aspek penting dalam bisnis restoran. Proses pemesanan menu yang efisien dan efektif dapat meningkatkan kepuasan pelanggan dan mendorong penjualan restoran. Definisi pemesanan menu menurut para ahli yaitu Sari & Nugroho (2020) "Pemesanan menu adalah proses pemilihan dan pencatatan

makanan atau minuman oleh pelanggan melalui sistem yang disediakan, baik secara manual maupun berbasis digital, untuk kemudian diproses oleh penyedia layanan."

5. *Multi channel*

Menurut Handayani & Prasetyo (2023), Multi channel merupakan strategi integratif yang memungkinkan konsumen melakukan interaksi melalui berbagai saluran yang tersedia seperti aplikasi, web, call center, dan media sosial. *Multi channel* di pemesanan menu adalah sebuah strategi yang memungkinkan pelanggan untuk memesan menu dari berbagai saluran, baik secara *online* maupun *offline*. Hal ini memberikan kemudahan bagi pelanggan untuk memesan menu sesuai dengan kenyamanan mereka.

Secara umum, *Multi channel* di pemesanan menu dapat didefinisikan sebagai berikut:

a. Sebuah strategi yang digunakan oleh restoran untuk menawarkan opsi pemesanan menu yang beragam kepada pelanggan.
b. Opsi pemesanan menu yang ditawarkan dapat mencakup saluran *online* maupun *offline*.
c. Tujuan dari strategi *Multi channel* adalah untuk memberikan kemudahan bagi pelanggan dan meningkatkan penjualan restoran.

Strategi *Multi channel* di pemesanan menu memiliki berbagai manfaat, antara lain:

a. Kemudahan bagi pelanggan

Pelanggan dapat memesan menu dari berbagai saluran sesuai dengan kenyamanan mereka. Misalnya, pelanggan yang ingin memesan menu dalam jumlah besar dapat memesan melalui telepon.

b. Peningkatan penjualan

Strategi *multichannel* dapat membantu restoran menjangkau lebih banyak pelanggan dan meningkatkan penjualan mereka. Hal ini karena restoran dapat menawarkan berbagai opsi pemesanan menu yang sesuai dengan kebutuhan pelanggan.

c. Peningkatan efisiensi

Strategi *Multi channel* dapat membantu restoran meningkatkan efisiensi operasional mereka. Hal ini karena restoran dapat mengintegrasikan berbagai saluran pemesanan menu ke dalam satu sistem.

6. Website

Menurut Hakim Lukmanul, definisi situs web adalah sarana internet yang menghubungkan (antar) berbagai dokumen dalam lingkup lokal maupun jarak jauh. Dokumen yang ada di situs web disebut halaman web. Terkait kontennya, hyperlink pada situs web memungkinkan pengguna situs web untuk berpindah dari satu halaman ke halaman lainnya.

7. Visual Studio *Code*

Editor teks Visual Studio *Code* dapat digunakan di berbagai sistem operasi, seperti Linux, Mac, dan Windows. Editor ini menawarkan dukungan untuk JavaScript, Typescript, dan Node.js, beserta berbagai bahasa pemrograman lain yang dapat ditambahkan melalui plug-in yang tersedia di pasar Visual Studio *Code*, termasuk C++, C#, Python, Go, Java, dan banyak lagi.

Visual Studio *Code* menawarkan berbagai fitur seperti *Intellisense*, Integrasi Git, Debugging, dan ekstensi yang meningkatkan *fungsionalitas* editor teks, yang semuanya akan menerima pembaruan saat versi baru dirilis. Yang membedakan VS *Code* dari editor teks lainnya adalah pembaruan bulanan rutin yang dilakukannya. Sebagai sumber daya sumber terbuka, pengguna memiliki kesempatan untuk mengakses kode sumber editor teks VS *Code* dan berkontribusi pada 27 pengembangannya. Selain itu, kode sumber dapat dilihat melalui tautan di Github. Kemampuan untuk terlibat dalam pengembangan VS *Code* di masa mendatang berkontribusi signifikan terhadap popularitasnya di kalangan pengembang aplikasi.

8. PHP

Sebagaimana dinyatakan oleh Prasetyo & Firmansyah (2020), PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang digunakan untuk membangun dan mengelola halaman web dinamis serta berinteraksi dengan database. Peran utama PHP dalam pengembangan situs web adalah pemrosesan data dalam basis data. Melalui PHP, data dari situs web dapat dimasukkan ke dalam basis data, dimodifikasi, dihapus, dan ditampilkan di situs. Bahasa pemrograman

berbasis skrip ini dijalankan di server, dan bersifat sumber terbuka, sehingga dapat digunakan bersama HTML. Akibatnya, sintaksis diproses sepenuhnya di server, dengan hanya hasil yang dikirimkan ke browser.

9. MySQL

Fadhilah & Wicaksono (2022), "MySQL adalah perangkat lunak database yang digunakan untuk menyimpan, mengelola, dan memanipulasi data secara terstruktur. MySQL juga mendukung bahasa SQL sebagai perintah utama dalam pengelolaan datanya." MySQL bersifat *open source* dan menggunakan SQL (*Structured Query Language*).

10. Xampp

Definisi XAMPP merupakan perangkat lunak yang paling umum digunakan untuk menjalankan server *Apache* dan mengembangkan situs web berbasis PHP. XAMPP sebenarnya merupakan gabungan dari beberapa perangkat lunak yang berhubungan dengan server, situs web, dan pengembangannya. Definisi XAMPP merupakan akronim yang berasal dari singkatan masing-masing program, yaitu X (*Cross Operating System*), A (*Apache*), M (MySQL), P (PHP), dan P (Perl).

11. *Bootstrap*

Bootstrap merupakan *framework* HTML, CSS, Javascript populer untuk membangun situs web yang *responsive*. Dengan *Bootstrap*, *developer* website dapat membuat *responsive* website dan dapat berjalan sempurna pada browser-browser populer seperti Chrome, Firefox, Opera, dan Internet Explorer serta pada perangkat-perangkat handphone maupun PC desktop. *Responsive* Web

Design adalah desain website yang secara otomatis akan menyesuaikan dengan kondisi plat*form* yang digunakan dari sisi tampilan serta segala isinya yang ada pada website tersebut.

Bootstrap terdiri dari satu set file CSS, Javascript, dan Jquery. *Bootstrap* menyediakan *Class-Class* dalam membuat berbagai jenis komponen-komponen website seperti menu *buttons*, *card*, *form*, *jumbotron*, media *Object*, *navbar*, *pagination*, *progress*, *spinners*, dan lainnya.

12. JavaScript

JavaScript adalah bahasa pemrograman berbasis skrip yang banyak digunakan oleh programmer web dalam pengembangan web karena dapat membuat halaman web yang dinamis dan didukung oleh banyak browser web. JavaScript berjalan di dalam kode HTML dan memungkinkan aplikasi sisi klien untuk membuat halaman web yang interaktif.

13. HTML

Menurut Nugroho & Lestari (2022), HTML adalah bahasa markup berbasis teks yang memungkinkan pengembang menyusun elemen-elemen halaman web, seperti paragraf, judul, form, dan media lainnya, agar dapat ditampilkan dan dipahami oleh browser. Sedangkan menurut Rizky & Maulana (2024), HTML adalah bahasa markup yang digunakan untuk mendeskripsikan struktur halaman web, bersifat deklaratif, dan berfungsi sebagai kerangka utama dalam pengembangan antarmuka web.

14. CSS

CSS (*Cascading Style Sheets*) CSS merupakan bahasa yang digunakan untuk mengontrol tampilan dari sebuah dokumen yang ditulis dengan bahasa markup/ markup *language*, jika kita berbicara dalam konteks web, dapat diartikan secara bebas sebagai: CSS merupakan bahasa yang digunakan untuk mengontrol tampilan/desain dari sebuah halaman HTML.

CSS dapat mengubah jenis huruf, ukuran huruf, warna dan format huruf, mengatur ukuran layout, lebar, tinggi dan warna elemen, mengubah tampilan tampilan, membuat halaman website yang *responsif* dan masih banyak lagi. Untuk mendesain huruf dapat dilakukan dengan mendefinisikan jenis huruf, untuk mengatur warna dapat menggunakan *color*, *margin* digunakan untuk mengatur jarak pada elemen eksternal tertentu. mengatur warna atau gambar pada latar belakang dapat menggunakan "*background*". mengatur ukuran huruf menggunakan "*font size*", jenis huruf menggunakan "*font-family*" dan masih banyak lagi.

15. UML *(Unified Modelling Language)*

Fahmi & Wibowo (2021), *Unified Modelling Language (UML)* adalah alat bantu analisis dan perancangan sistem yang digunakan untuk menggambarkan proses bisnis, alur data, serta interaksi antar komponen dalam sistem informasi secara terstruktur. UML menyediakan standar untuk merancang model sistem. UML merupakan singkatan dari *Unified Modeling Language*. UML juga merupakan cara untuk memfasilitasi pengembangan aplikasi yang berkelanjutan.

Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan

mempelajari kode program UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang dikembangkan oleh satu pengembang ke pengembang lainnya. Tidak hanya kalangan *developer* dan pebisnis saja, namun semua orang bisa memahami sistem UML.

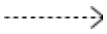
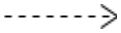
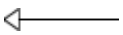
UML dibuat oleh *Object Management Group* mulai versi 1.0 pada bulan Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal: *Object, Class, Abstraction, Encapsulation, Inheritance dan Polymorphism.*

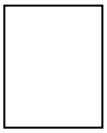
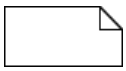
UML sendiri mempunyai beberapa diagram yaitu:

a. *Use Case* Diagram

Diagram *Use Case* menggambarkan fungsionalitas yang diharapkan dari sistem. Penekanannya adalah pada "apa" yang dilakukan sistem, bukan "bagaimana". Sebuah *Use Case* mewakili interaksi antara operator dan sistem. *Use Case* adalah tugas spesifik, seperti masuk ke sistem, membuat daftar belanja, dll. Aktor adalah orang atau komputer yang berinteraksi dengan suatu sistem untuk melakukan tugas tertentu. *Use Case* adalah serangkaian interaksi yang saling terkait antara sistem dan aktor. *Use Case* diimplementasikan dengan mendeskripsikan jenis interaksi antara pengguna program (sistem) dan sistem itu sendiri.

Tabel 2. 1 *Use Case* Diagram

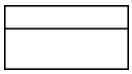
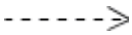
NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri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

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

b. *Class* diagram

Diagram kelas adalah representasi visual dari struktur sistem perangkat lunak dalam tipe-tipe penyusunnya. Diagram kelas adalah aliran *database* dalam suatu sistem. Diagram kelas merupakan penjelasan proses *database* suatu program. Diagram kelas ini diperlukan dalam laporan sistem.

Tabel 2. 2 Simbol *Class* Diagram

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

c. *Activity* diagram

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

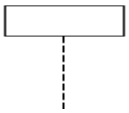
Tabel 2. 3 Simbol *Activity* Diagram

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

d. *Sequence* Diagram

Sequence Diagram adalah diagram yang menjelaskan bagaimana objek berinteraksi dan menunjukkan (memberi tanda atau instruksi) interaksi antar objek tersebut.

Tabel 2. 4 Simbol *Sequence* diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.

2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktifitas yang terjadi

16. *Flowchart*

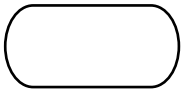
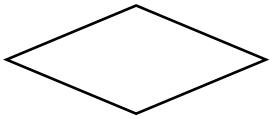
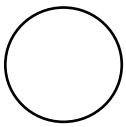
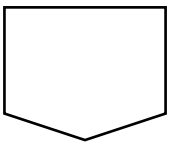
Menurut Wulandari & Ramadhan (2021), Flowchart merupakan representasi grafis dari suatu proses atau prosedur yang disusun secara terurut dan menggunakan simbol-simbol standar untuk memudahkan analisis sistem. *Flowchart* (diagram alir) dapat digunakan sebagai alternatif untuk menyajikan algoritma.

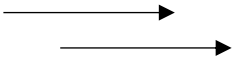
Ada lima macam bagan alir, di antaranya:

a. Bagan Alir Sistem (*system Flowchart*) merupakan diagram yang menunjukkan alur kerja umum sistem.
b. Bagan alir dokumen, disebut juga bagan alur formulir, adalah bagan alur yang memperlihatkan alur laporan dan formulir, termasuk salinannya.
c. Bagan Alir Skematik (*schematic Flowchart*) adalah diagram alir yang menggambarkan prosedur suatu sistem dengan menggunakan simbol dan gambar diagram alir sistem dan perangkat lain yang digunakan dalam sistem.
d. Bagan Alir Program (program *Flowchart*) merupakan diagram yang menjelaskan secara rinci langkah-langkah dalam proses pemrograman.

e. Bagan Alir Proses (*process Flowchart*) adalah diagram alur yang banyak digunakan dalam teknik industri untuk menggambarkan proses proses.

Tabel 2. 5 Simbol-Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		Terminal	Menyatakan permulaan atau akhir suatu program
2		*Input/Multi channel*	Menyatakan proses *Input* atau *Multi channel* tanpa tergantung jenis peralatanya
3		*Process*	Menyatakan suatu Tindakan (Proses) yang dilakukan oleh computer
4		*Decision*	Menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemunkinan jawaban:ya / tidak
5		*Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang sama
6		*Offline Connector*	Menyatakan sambungan dari proses ke proses lainya dalam halaman yang berbeda

No	Simbol	Nama	Fungsi
7		*Prefefned Process*	Menyatakan penyediaan tempat penyimpanan suatu pengolaan untuk memberi harga awal
8		*Punched card*	Menyatakan *Input* berasal dari kartu atau *Multi channel* ditulis ke kartu
9		*Document*	Mencetak keluaran dalam bentuk dokumen(Melalui printer)
10		*Flow*	Menyatakan jalannya arus suatu proses

B. Metode Pengujian

Beberapa kasus uji yang mewakili penggunaan sistem yang umum, kritis, atau tidak normal harus dilakukan dengan strategi, pertanyaan, atau jalur navigasi yang berbeda. Hal utama dalam pengembangan sistem adalah memilih kelompok pengujian yang sesuai, sekecil mungkin dan cepat untuk memeriksa kinerja sistem. Pengujian harus mencakup pengujian unit yang memverifikasi validasi prosedur dan fungsi secara *independen* dari komponen sistem lainnya. Setelah itu harus dilakukan pengujian terhadap modul-modul untuk mengetahui apakah kombinasi beberapa unit dalam satu modul berfungsi dengan baik, termasuk kinerja beberapa modul yang saling berhubungan untuk melihat apakah bekerja sesuai dengan karakteristik sistem yang diinginkan.

Setelah struktur kontrol lintas modul terbukti baik, pengujian unit juga sama pentingnya. Pengujian unit digunakan untuk menguji setiap modul untuk memastikan bahwa setiap modul berfungsi dengan benar. Ada dua cara untuk menguji suatu unit:

1. *White box Testing*

White box Testing merupakan suatu cara pengujian suatu aplikasi atau perangkat lunak dengan melihat suatu modul untuk memeriksa dan menganalisis apakah kode program yang dihasilkan salah atau tidak. Jika modul ini menghasilkan keluaran yang tidak memenuhi persyaratan, kode dikompilasi ulang dan diperiksa ulang hingga memenuhi harapan. Singkatnya, *White box Testing* menguji dengan melihat kode murni dari aplikasi atau perangkat lunak yang diuji, tanpa memperhatikan tampilan atau antarmuka pengguna aplikasi tersebut.

Untuk membuat pengujian *white box*, penguji harus memiliki keterampilan pemrograman, untuk memahami kode sumber yang diselidiki, ia juga harus memiliki pandangan *global* tentang cara kerja aplikasi, elemen penyusunnya, dan tentu saja kode sumbernya. Berbeda dengan pengujian "kotak hitam", penguji memiliki profil "pengembang", bukan "profil pengguna". Dengan membuat pengujian kotak putih, penguji dapat melihat baris kode mana yang dipanggil untuk setiap fungsi. Ini memungkinkan Anda menguji aliran data dan menangani pengecualian dan kesalahan. Ketergantungan sumber daya serta logika dan kebenaran kode *internal* juga dieksplorasi. Oleh karena itu, pengujian ini sangat berguna selama pengembangan aplikasi, meskipun pengujian ini dapat dilakukan

pada berbagai tahapan proyek. Metode pengujian *White box* dapat diterapkan pada (terutama) pengujian *unit*, pengujian integrasi, dan pengujian sistem.

Untuk membuat *White box Testing*, penguji harus memiliki keterampilan pemrograman, untuk memahami kode sumber yang diselidiki, ia juga harus memiliki pandangan *global* tentang cara kerja aplikasi, elemen-elemen yang menyusunnya, dan tentu saja kode sumbernya. Berbeda dengan pengujian "*Black box*", penguji memiliki profil "pengembang", bukan "profil pengguna".

Dengan membuat *White box Testing*, penguji dapat melihat baris kode mana yang dipanggil untuk setiap fungsi. Ini memungkinkan Anda menguji aliran data dan menangani pengecualian dan kesalahan. Ketergantungan sumber daya serta logika dan kebenaran kode *internal* juga dieksplorasi. Oleh karena itu, pengujian ini sangat berguna selama pengembangan aplikasi, meskipun pengujian ini dapat dilakukan pada berbagai tahapan proyek. Metode *White box Testing* dapat diterapkan pada (terutama) pengujian unit, pengujian integrasi, dan pengujian sistem.

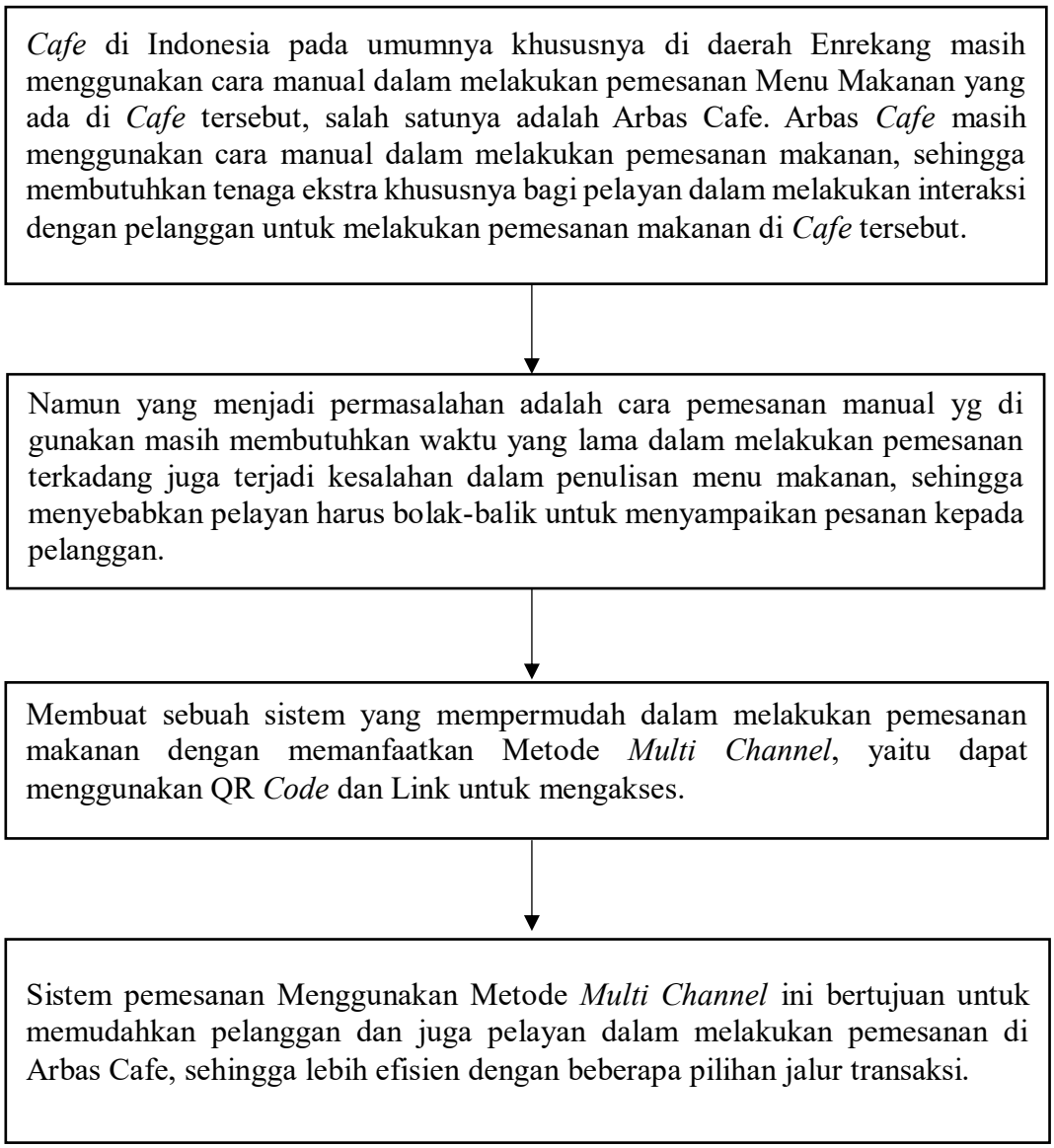
2. *Black box*

Berdasarkan hasil pengujian yang dilakukan, tidak terdapat kesalahan pada seluruh navigasi dan tombol-tombol lain pada program, serta pada proses yang dilakukan, namun aplikasi mempunyai aturan yang ditetapkan dan harus dipatuhi, karena jika mereka tidak; Diamati, sistem akan menolak perintah yang tidak sesuai, misalnya *error* jika pengguna belum memasukkan informasi yang harus dimasukkan sesuai aturan sistem operasi, dan sistem memberikan informasi kepada pengguna karena informasi yang diproses tidak lengkap atau tidak. memenuhi

persyaratan proses selanjutnya Menguji fungsionalitas perangkat lunak itu sendiri, *mis*. prosedur program (jalur utama) atau proses perulangan (*looping*) yang berfokus pada efisiensi aplikasi yang dimaksud.

C. Kerangka Pikir

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



Gambar 2. 1 Kerangka Pikir

D. Kajian Hasil Penelitian Terdahulu

Dalam melakukan sebuah penelitian atau acuan terhadap penelitian yang telah dilakukan sebelumnya, melalui berbagai buku, jurnal maupun skripsi sehingga dapat memudahkan penelitian dalam menentukan kerangka awal sebagai konsep untuk meneliti. Sehingga dalam penelitian kali ini akan dicantumkan dari hasil penelitian sebelumnya :

1. Suharianto, Lukman Bahar Agung Pambudi, Angga Rahagiyanto, Gandu Eko Julianto Suyoso, "**Implementasi QR *Code* untuk Efisiensi Waktu Pemesanan Menu Makanan dan Minuman di Restoran maupun Kafe".** Yang membedakan antara penelitian ini dengan penelitian saya adalah penelitian yang saya menggunakan metode *Multi channel* yaitu dapat melakukan transaksi dengan beberapa cara, tidak hanya bergantung pada QR *Code*.
2. Wiratama, A., "**Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android**.". Perbedaan penelitian saya dengan penelitian ini yaitu penelitian saya menggunakan metode *Multi channel* sedangkan penelitian ini hanya berfokus pada pengembangan aplikasi berbasis Android saja, tanpa menerapkan pendekatan *multi* kanal, serta belum mendukung *integrasi* antar *platform* pemesanan.
3. Muh.Jayadi, Andi Irmayani Pawelloi, Wahyuddin, "**APLIKASI PEMESANAN MENU PADA WARKOP SHERLOCK BERBASIS ANDROID** ". Yang membedakan antara penelitian ini dengan penelitian saya adalah penelitian terdahulu ini menggunakan metode yang berorientasi pada pengembangan aplikasi android sebagai solusi utama. Sedangkan metode yang saya gunakan yaitu metode

Multi channel yaitu adalah sebuah strategi yang memungkinkan pelanggan untuk memesan menu dari berbagai saluran, baik secara *online* maupun *offline*.

BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini merupakan penelitian eksperimental yang bersifat aplikatif sehingga dari ruang lingkup masalah dapat dilakukan dengan metode studi pustaka (*library research*), metode pengumpulan data (*field research*).

B. Waktu Dan Tempat Penelitian

Rencana waktu yang digunakan untuk penelitian ini berlangsung selama ± 3 bulan dan bertempat di kota Enrekang, Sulawesi Selatan, Indonesia.

C. Alat Dan Bahan Penelitian

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

1. Perangkat Keras (*Hardware*)

   Perangkat keras yang digunakan selama proses penelitian yaitu Laptop HP dengan spesifikasi sebagai berikut :

   a. Processor : AMD Athlon Gold 3150U with Radeon Graphics 2.40 GHz
   b. RAM : 4,00 GB
   c. SSD : 500 GB
   d. Monitor : 14 Inch

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pembuatan aplikasi adalah sebagai berikut :

a. Sistem Operasi : Windows 10 Ultimate 64 bit
b. Editor : *Visual studio code*
c. *Database* : Mysql

D. Tahapan Penelitian

Metode penelitian dilakukan dengan lima tahap penelitian, uraian dari lima tahap tersebut adalah sebagai berikut :

1. Persiapan penelitian

Pada tahap ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapakan buku-buku, artikel-artikel tentang topik penelitian serta *Software* yang digunakan selama penelitian.

2. Pengumpulan data

Pada tahap ini peneliti melakukan wawancara dengan pemilik *Cafe* terkait proses pemesanan menu serta mengambil data menu yang terdapat pada cafe.

3. Analisis

Pada tahap analisis peneliti melakukan analisa pada sistem yang diterapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecehan masalah.

4. Perancangan

Peneliti kemudian merancang apalikasi yang berdasarkan pemecahan masalah.

5. Pembuatan Program

Setelah melakukan pengumpulan data, analisis, dan perancangan, maka peneliti akan membangun Program berdasarkan algoritma yang telah ada menggunakan bahasa pemrograman php.

6. Pengujian

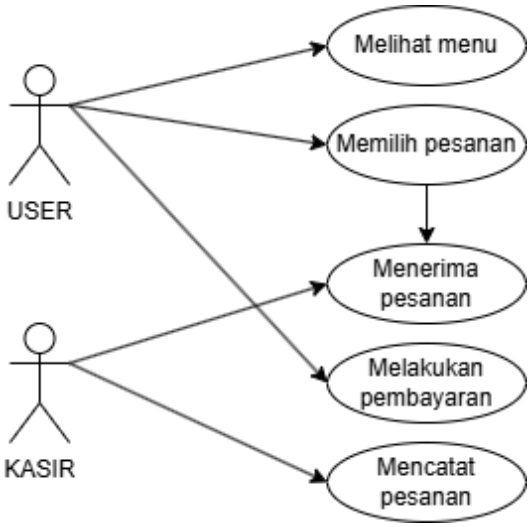
Program yang telah dibangun sebelumnya akan di uji tingkat keberhasilannya. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

7. Implementasi

Setelah tidak terdapat kekurangan pada program maka aplikasi siap digunakan oleh *user*.

E. Desain Sistem

1. Sistem yang dijalankan

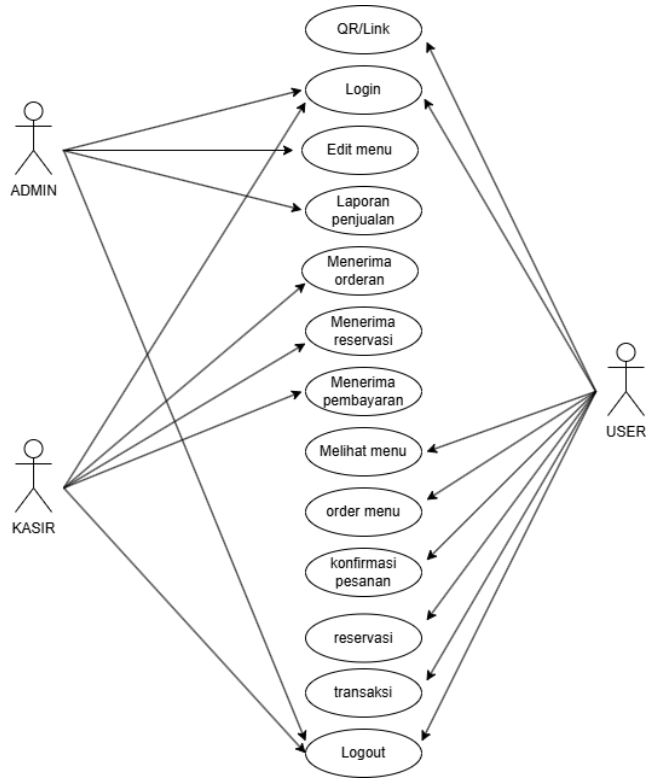


Gambar 2. 2 sistem yang berjalan

Proses pemesanan menu diawali dengan pelanggan datang ke *Cafe* dan melihat daftar menu. Setelah pelanggan menentukan pilihan pesanan kasir mencatat pesanan yang dipilih oleh pelanggan, lalu pelanggan membayar harga total yang diinformasikan oleh kasir. Setelah itu *staff* menyajikan pesanan pelanggan.

2. Sistem yang diusulkan

Pada gambar di bawah ini adalah sebuah tampilan dari proses *system* yang di usulkan atau alur dari aplikasi yang akan di bangun.



Gambar 2. 3 *Use Case* sistem yang diusulkan

Ketika pelanggan akan melakukan pemesanan, maka pelanggan bisa memilih apa yang akan di gunakan untuk memesan makanan pada *café* tersebut, bisa menggunakan Link ataupun QR *Code*, kedua pilihan tersebut akan mengarahkan ke halaman menu makanan dan minuman setelah meng*Input* pesanan dan mengkonfirmasi pesanan kemudian akan dikirimkan ke kasir. Pengguna juga dapat melakukan reservasi tempat, transaksi pembayaran melalui aplikasi tersebut.

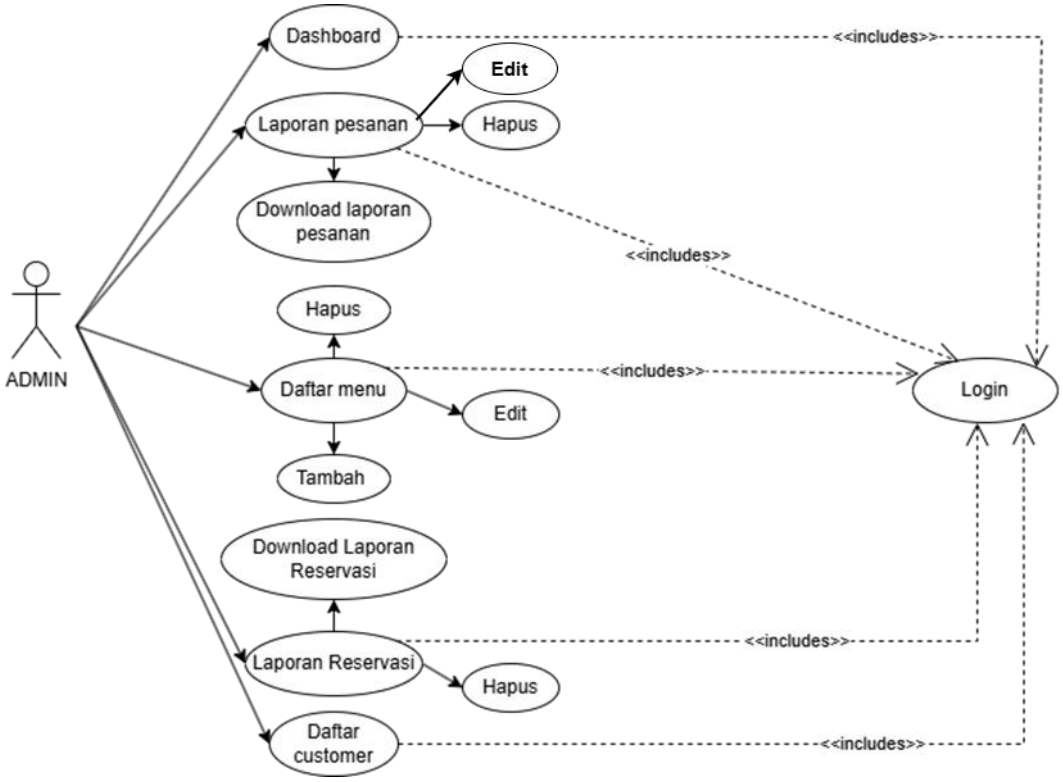
BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data dengan UML

1. *Use Case* diagram

a. Aktor admin

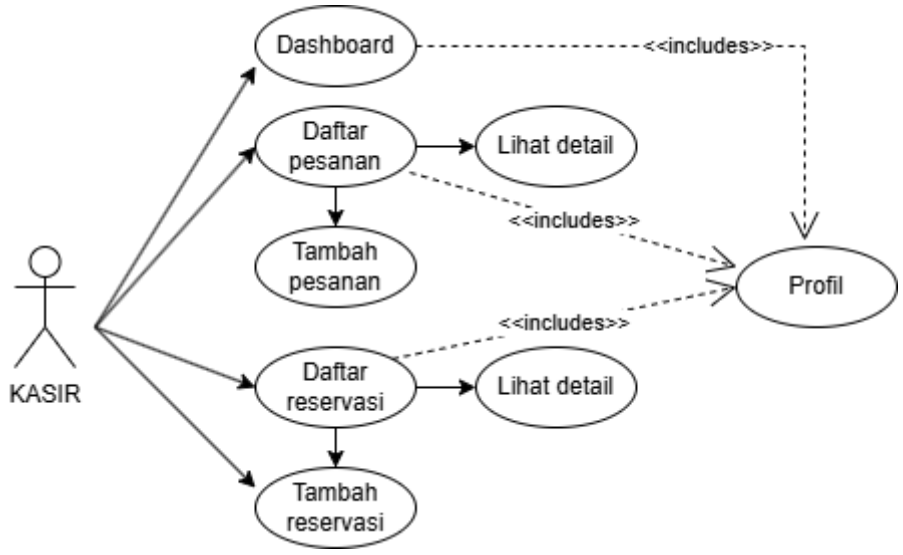


Gambar 3. 1 *Use Case* diagram admin

Tabel 3. 1 Penjelasan *Use Case* aktor admin

Nama *use case*	Deskripsi *use case*
Login	Admin dapat *login*
Dashboard	Admin dapat melihat in*form*asi terkait penjualan
Laporan pesnan	Admin dapat melihat, menghapus dan men*download* laporan pesanan
Daftar menu	Admin dapat melihat, menambah, mengedit dan menghapus daftar menu
Laporan reservasi	Admin dapat melihat, menghapus dan men*download* laporan reservasi
Daftar *customer*	Admin dapat melihat daftar *Customer*

b. Aktor kasir

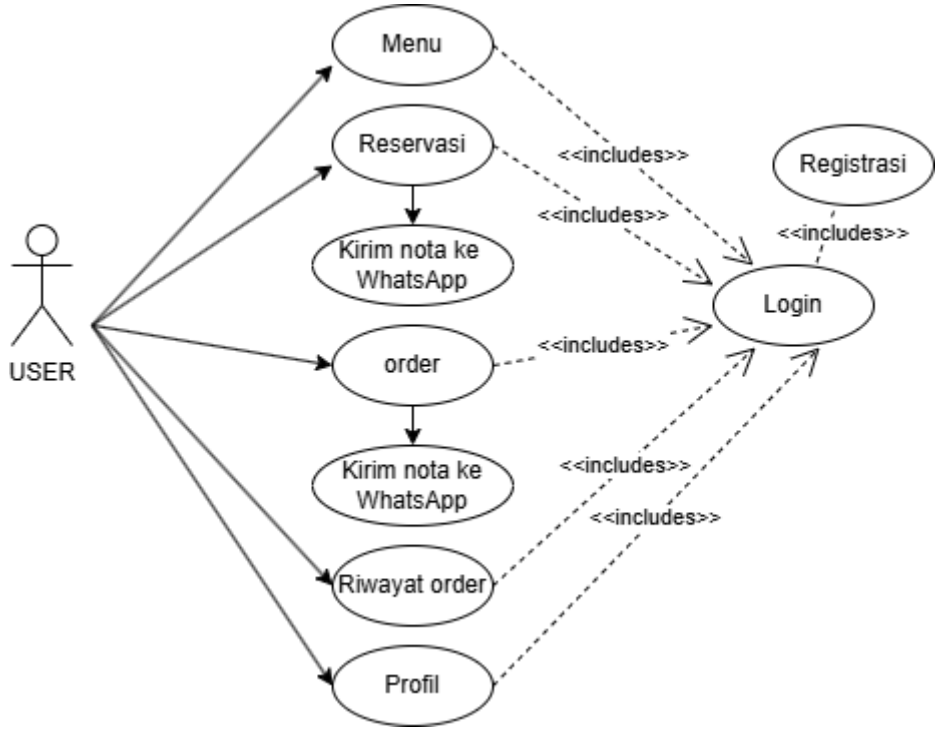


Gambar 3. 2 *Use Case* diagram kasir

Tabel 3. 2 Penjelasan *Use Case* aktor kasir

Nama *use case*	Deskripsi *use case*
Login	Kasir dapat *login*
Dashboard	Kasir dapat melihat in*form*asi terkait penjualan
Daftar pesnan	Kasir dapat melihat, menambah dan melihat detail pesanan
Daftar reservasi	Kasir dapat melihat, menambah dan melihat detail pesanan

c. Aktor *user*



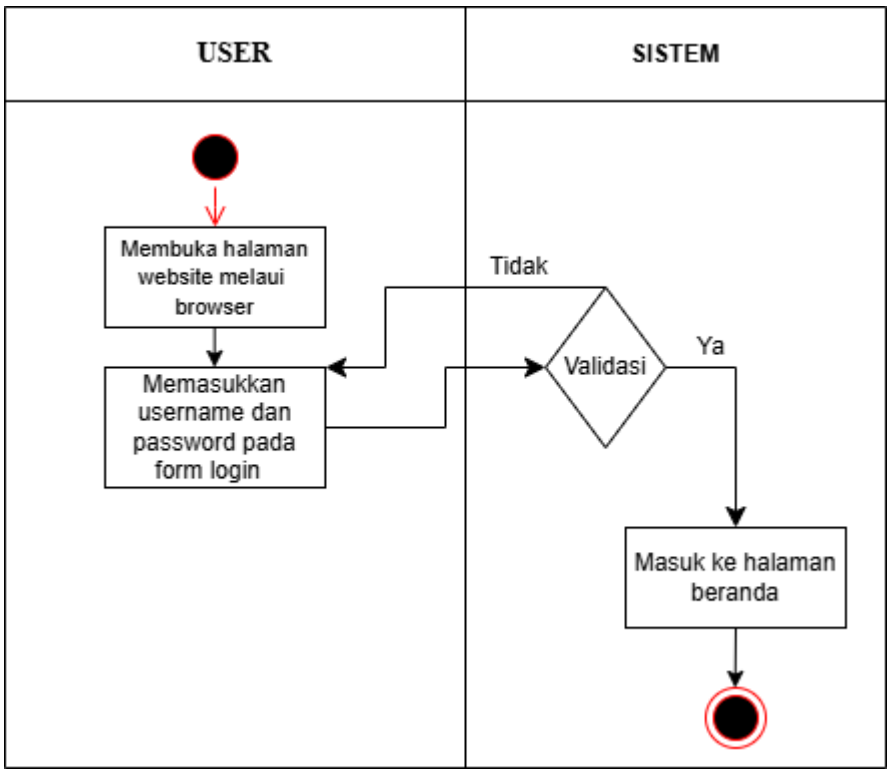
Gambar 3. 3 *Use Case* diagram *user*

Tabel 3. 3 Penjelasan *Use Case* aktor *user*

Nama *use case*	Deskripsi *use case*
Registrasi	*Customer* melakukan registrasi untuk dapat *login*
Login	*Customer* dapat *login*
Menu	*Customer* dapat melihat daftar makanan dan minuman yang tersedia di *cafe*
Reservasi	*Customer* dapat memesan meja di *Cafe* melalui fitur reservasi
Order	*Customer* dapat memesan makanan atau minuman
Riwayat order	*Customer* dapat melihat riwayat order yang pernah dilakukan
Profi	*Customer* dapat mengubah sandi

2. *Activity* Diagram

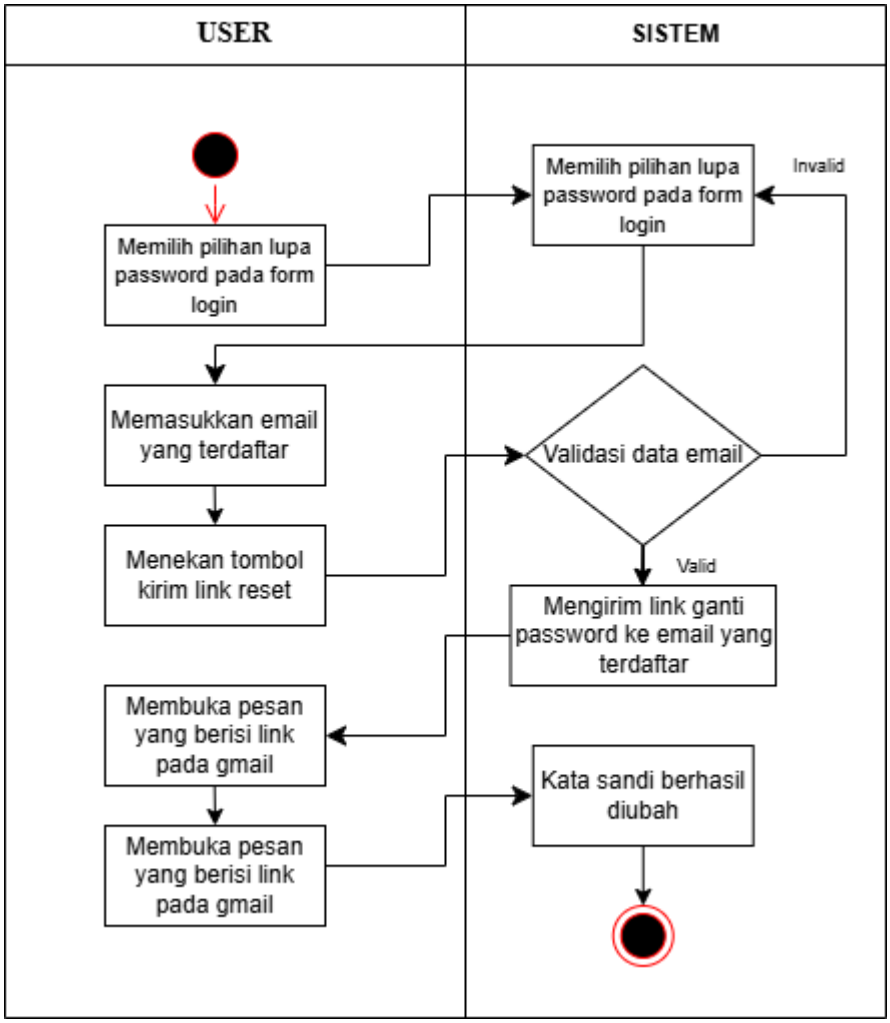
1) Halaman *login*



Gambar 4. 1 *Activity* diagram *login* admin

pengguna membuka halaman situs web aplikasi melalui browser. Lalu pada *form login* pengguna memasukkan *username* dan kata sandi lalu menekan tombol masuk. Kemudian sistem memeriksa apakah datanya benar. Jika sudah benar pengguna akan diarahkan ke halaman beranda, jika tidak akan diarahkan kembali ke halaman *login*.

2) Halaman lupa *password*



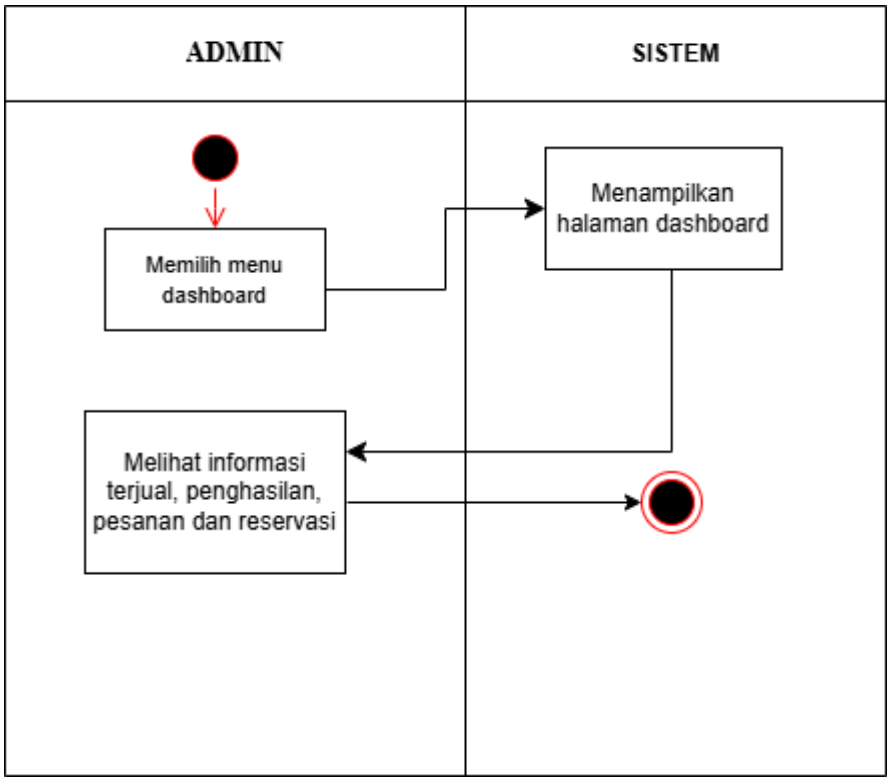
Gambar 4. 2 *Activity* diagram lupa *password*

Pengguna memilih opsi lupa *password* pada *form login*. kemudian sistem akan menampilkan halaman lupa *password* yang berisi *form* email. pengguna memasukkan alamat email terdaftar dan menekan tombol kirim link reset. Sistem kemudian menjalankan proses validasi untuk memeriksa apakah alamat email terdaftar dalam *database*. Apabila benar terdaftar maka sistem akan mengirim pesan yang berisi tautan ke halaman ganti *password* dan kode untuk mengubah

password. pengguna membuka email lalu mengeklik tautan lalu memasukkan kata sandi baru beserta kode yang diterimanya, lalu mengeklik tombol simpan, kata sandi Anda berhasil diubah. Jika email tidak terdaftar, proses validasi email akan in*valid* dan sistem akan mengarahkan kembali ke halaman *form* lupa *password*.

a. Admin

1.) Halaman *dashboard* admin

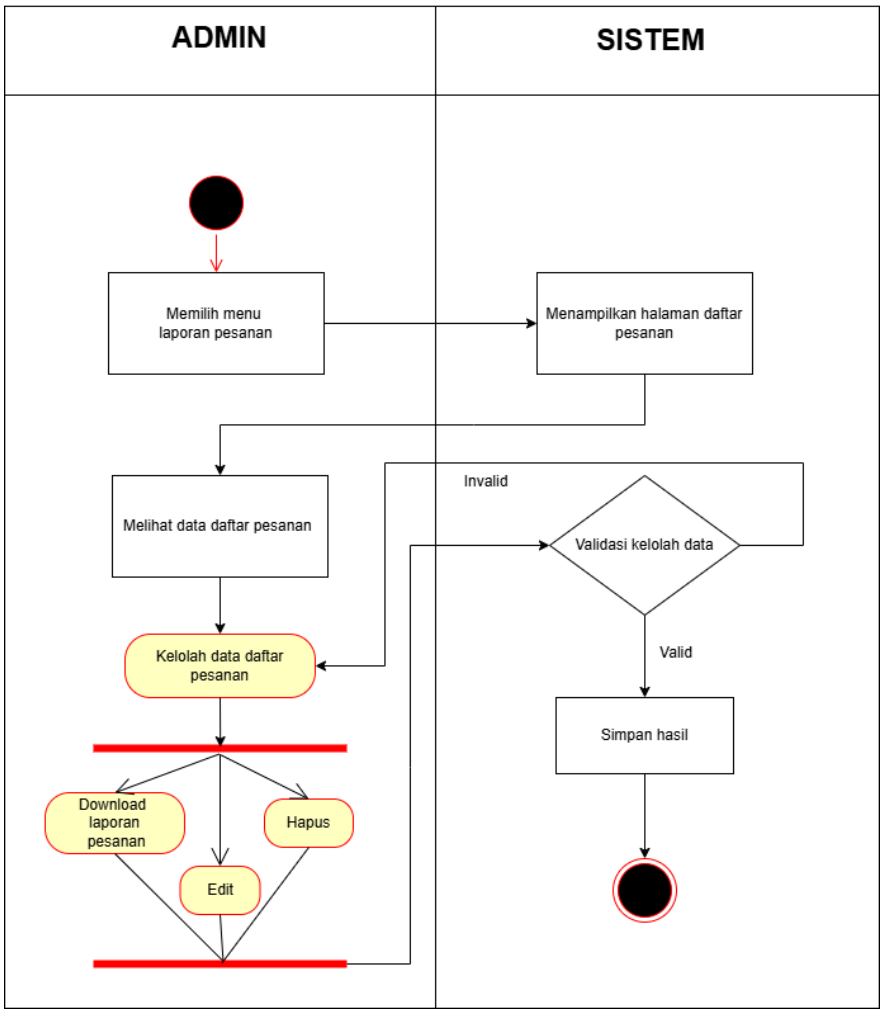


Gambar 4. 3 *Activity* diagram *dashboard* admin

Admin memilih menu *dashboard* dan sistem akan menampilkan halaman informasi terkait penjualan.

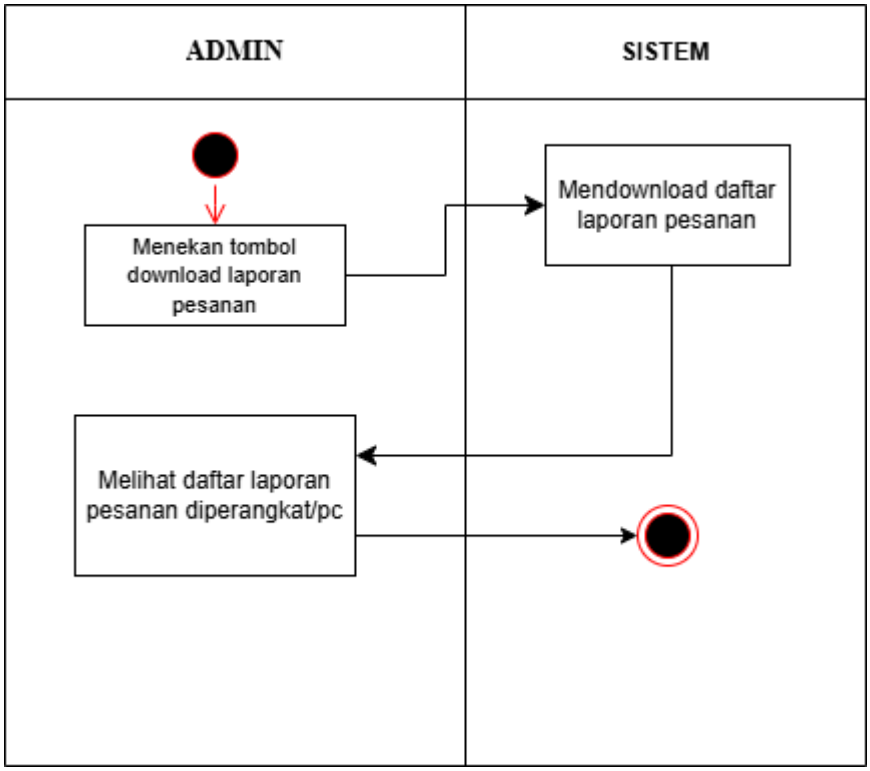
2.) Halaman laporan pesanan admin

Admin memilih menu laporan Pesanan, sistem kemudian menampilkan halaman daftar pesanan yang berisi *customer* yang telah melakukan pemesanan. Kemudian admin melihat data dan mengelola data dengan opsi menghapus, dan mengunduh laporan pesanan. Setelah admin memilih tindakan, hasilnya disimpan oleh sistem.



Gambar 4. 4 *Activity* diagram laporan pesanan admin

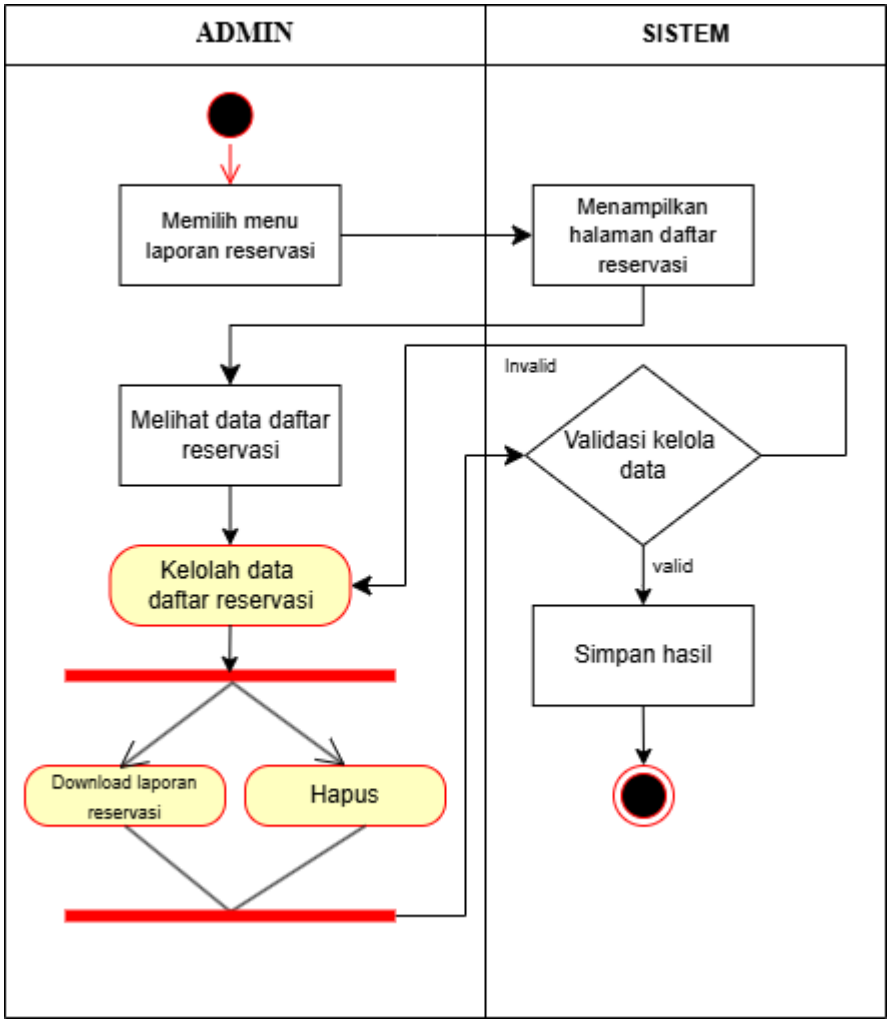
3.) Halaman *download* laporan pesanan admin



Gambar 4. 5 *Activity* diagram *download* laporan pesanan admin

Admin menekan tombol *download* laporan pesanan dan sistem akan mengunduh daftar laporan pesanan. Setelah itu admin dapat melihat di perangkat hasil *download* daftar laporan pesanan.

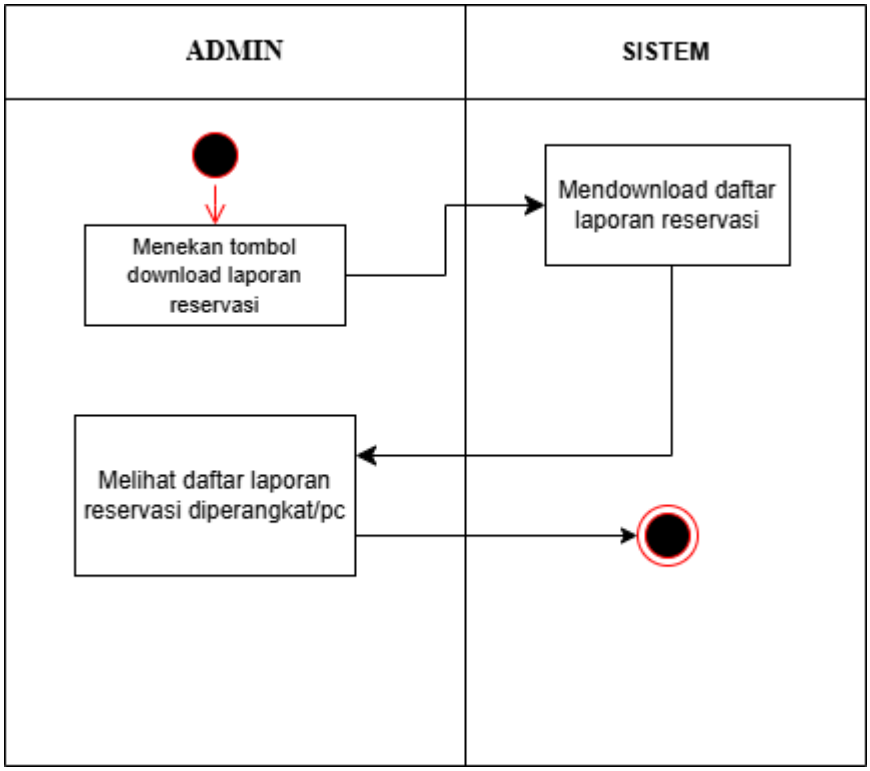
4.) Halaman laporan reservasi admin



Gambar 4. 6 *Activity* diagram laporan reservasi admin

Admin memilih menu laporan reservasi sistem kemudian menampilkan halaman daftar reservasi yang berisi *customer* yang telah melakukan reservasi. Kemudian admin melihat data dan mengelola data dengan opsi menghapus, dan mengunduh laporan reservasi. Setelah admin memilih tindakan, hasilnya disimpan oleh sistem.

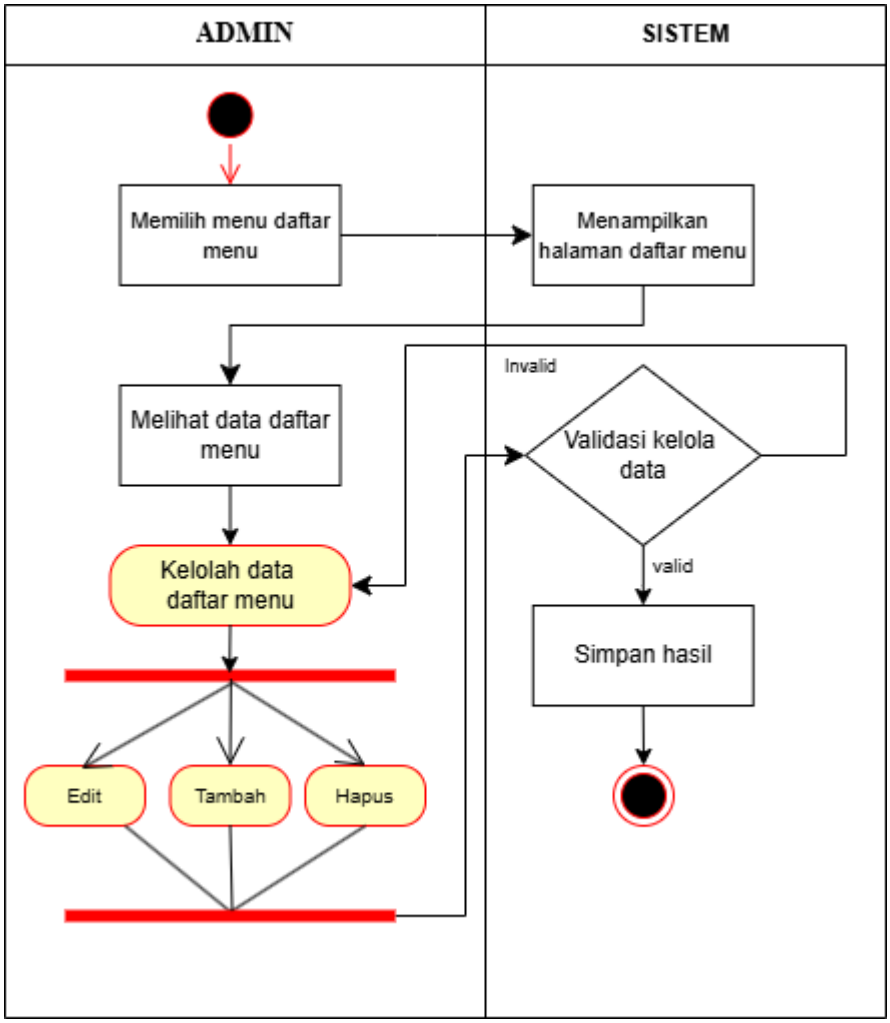
5.) Halaman *download* laporan reservasi admin



Gambar 4. 7 *Activity* diagram laporan reservasi admin

Admin menekan tombol *download* laporan reservasi dan sistem akan mengunduh daftar laporan reservasi. Setelah itu admin dapat melihat di perangkat hasil *download* daftar laporan reservasi.

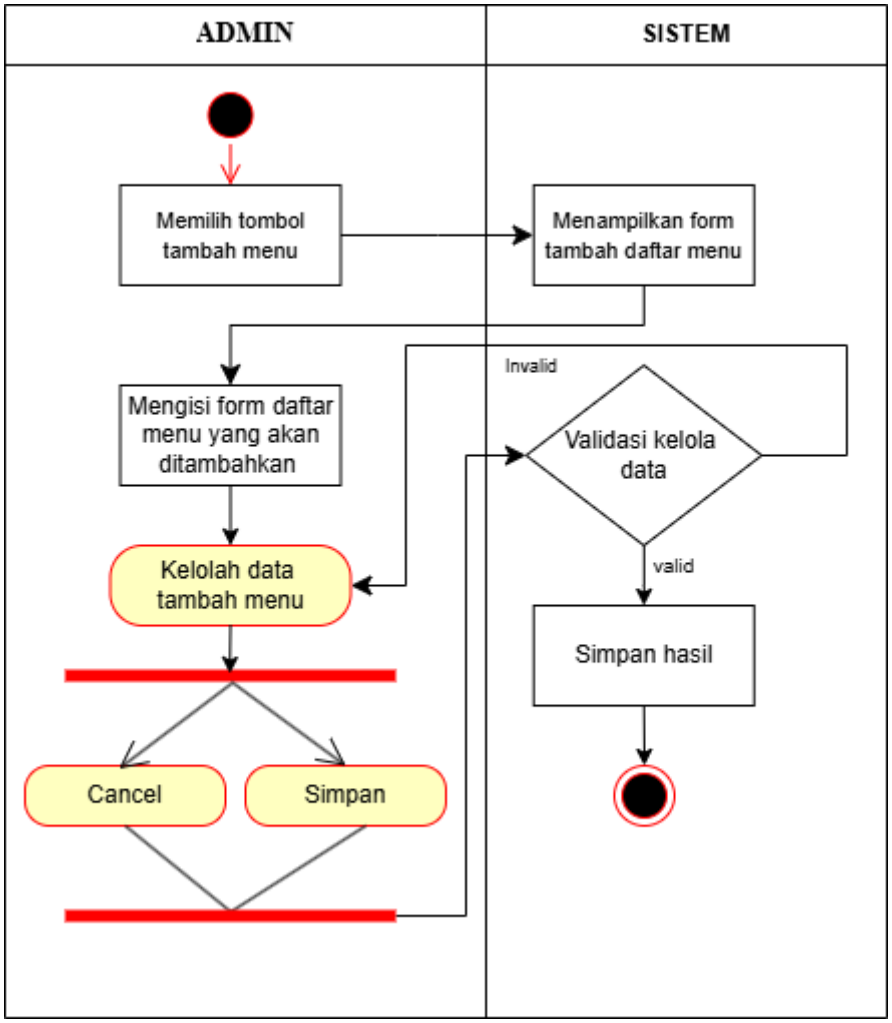
6.) Halaman daftar menu admin



Gambar 4. 8 *Activity* diagram daftar menu admin

Admin memilih menu daftar menu kemudian sistem akan menampilkan halaman daftar menu yang berisi daftar menu makanan dan minuman. kemudian admin melihat data dan mengelolah data dengan opsi menghapus, mengedit dan tambah. Setelah admin memilih tindakan hasilnya disimpan oleh sistem.

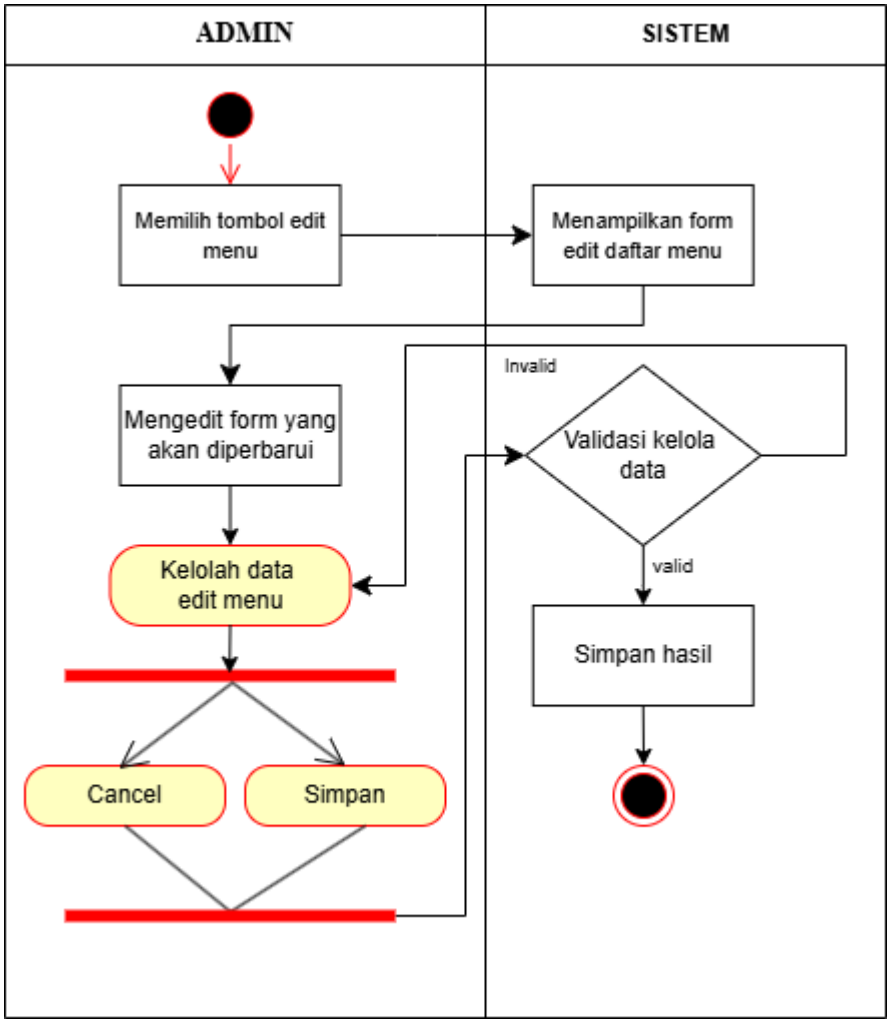
7.) Halaman tambah menu admin



Gambar 4. 9 *Activity* diagram tambah menu admin

Admin menekan tombol tambah menu kemudian sistem akan menampilkan *form* tambah daftar menu lalu setelah mengisih *form* lalu menekan tombol simpan kemudian sistem akan memvalidasi data tersebut, jika data *valid* sistem akan menyimpan data, jika gagal sistem akan mengarahkan kembali ke *form* tambah daftar menu.

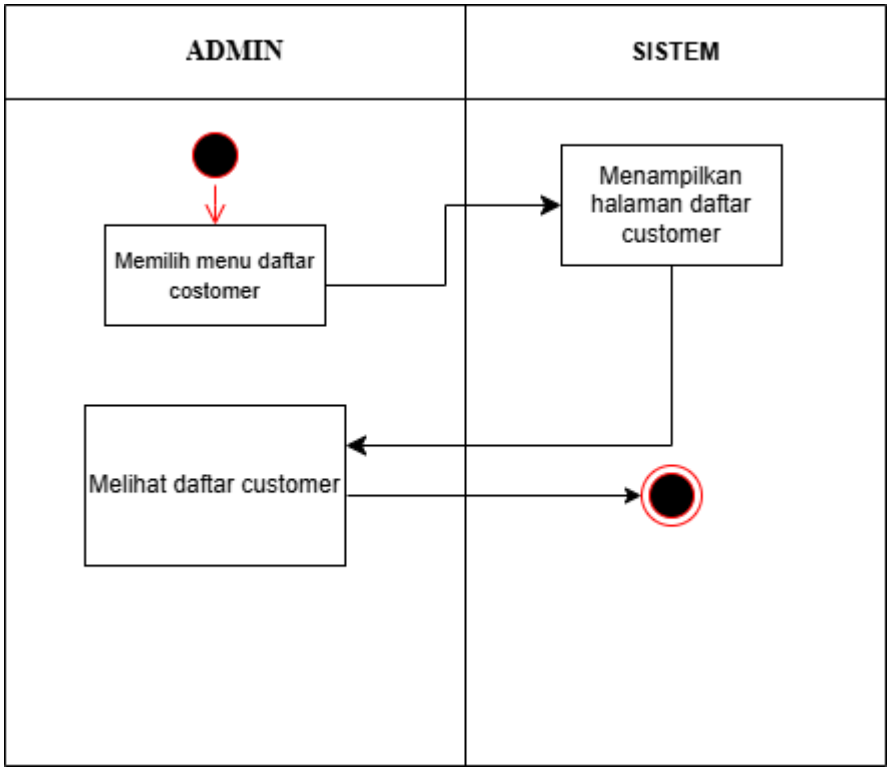
8.) Halaman edit menu admin



Gambar 4. 10 *Activity* diagram edit menu admin

Admin menekan tombol edit menu kemudian sistem akan menampilkan *form* edit menu setelah mengubah *form* menu lalu menekan tombol simpan kemudian sistem akan memvalidasi data tersebut, jika data *valid* sistem akan menyimpan data, jika gagal sistem akan mengarahkan kembali ke *form* edit menu.

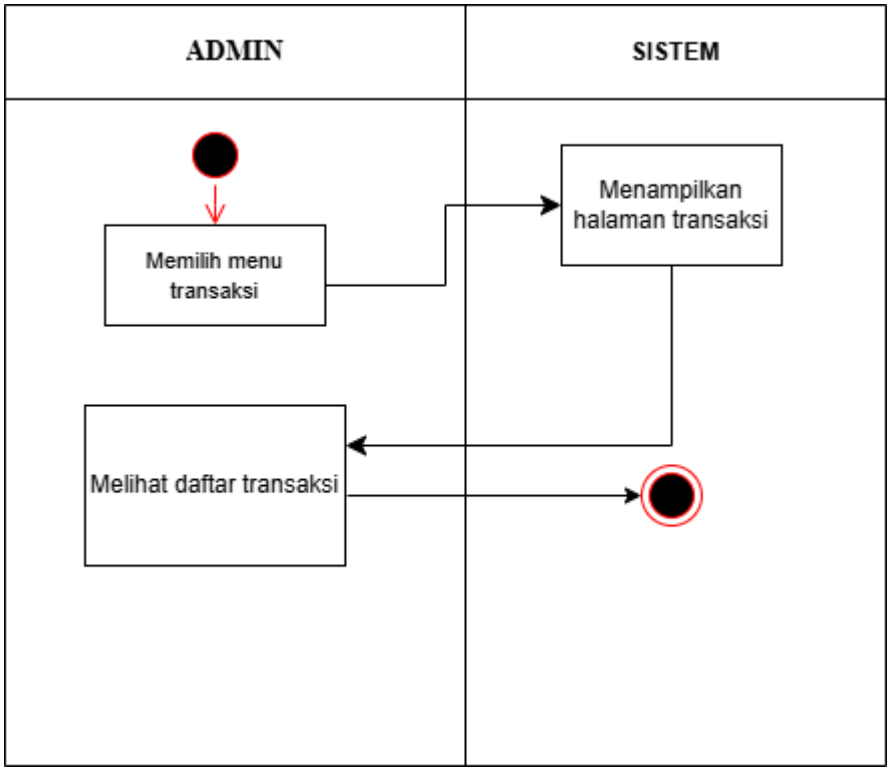
9.) Halaman daftar *customer* admin



Gambar 4. 11 *Activity* diagram daftar *customer* admin

Admin memilih daftar *customer* dan sistem akan menampilkan halaman daftar *customer*.

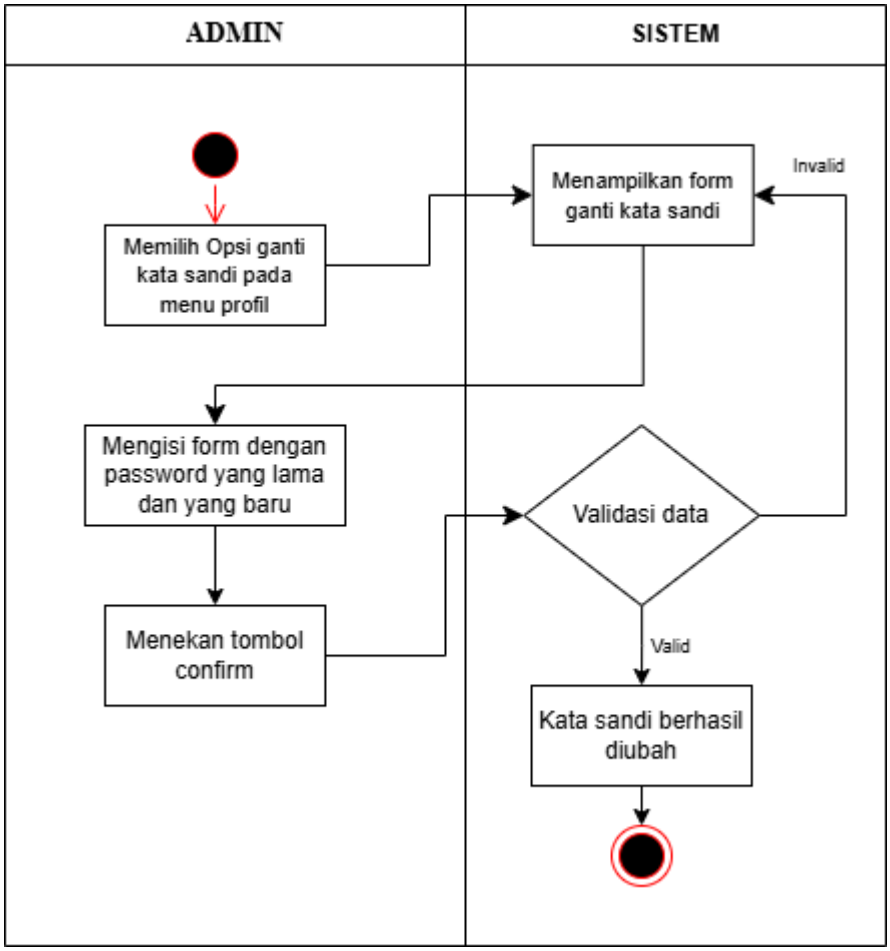
10.) Halaman transaksi admin



Gambar 4. 12 *Activity* diagram transaksi admin

Admin memilih menu transaksi dan sistem akan menampilkan halaman transaksi.

11.) Halaman ganti sandi admin

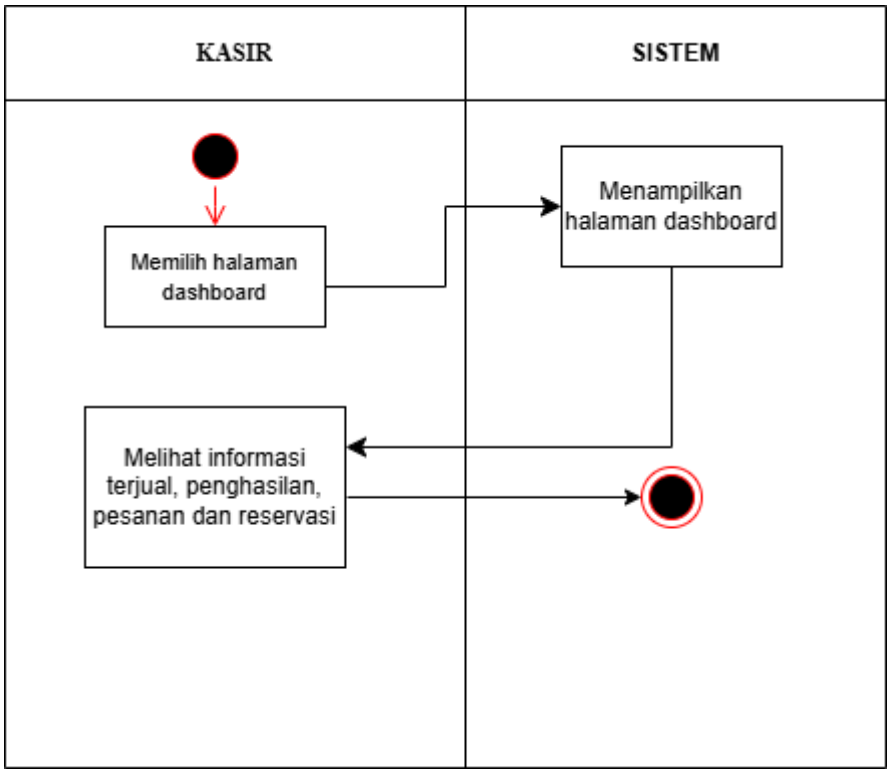


Gambar 4. 13 *Activity* diagram ganti sandi admin

Admin memilih opsi ubah kata sandi di menu profil. Kemudian sistem akan menampilkan *form* untuk mengubah kata sandi. Setelah Admin memasukkan kata sandi baru dan menekan tombol Konfirmasi, Sistem akan memvalidasi data. Jika *valid*, kata sandi berhasil diubah. Jika gagal, sistem akan menampilkan *form* perubahan kata sandi lagi.

b. Kasir

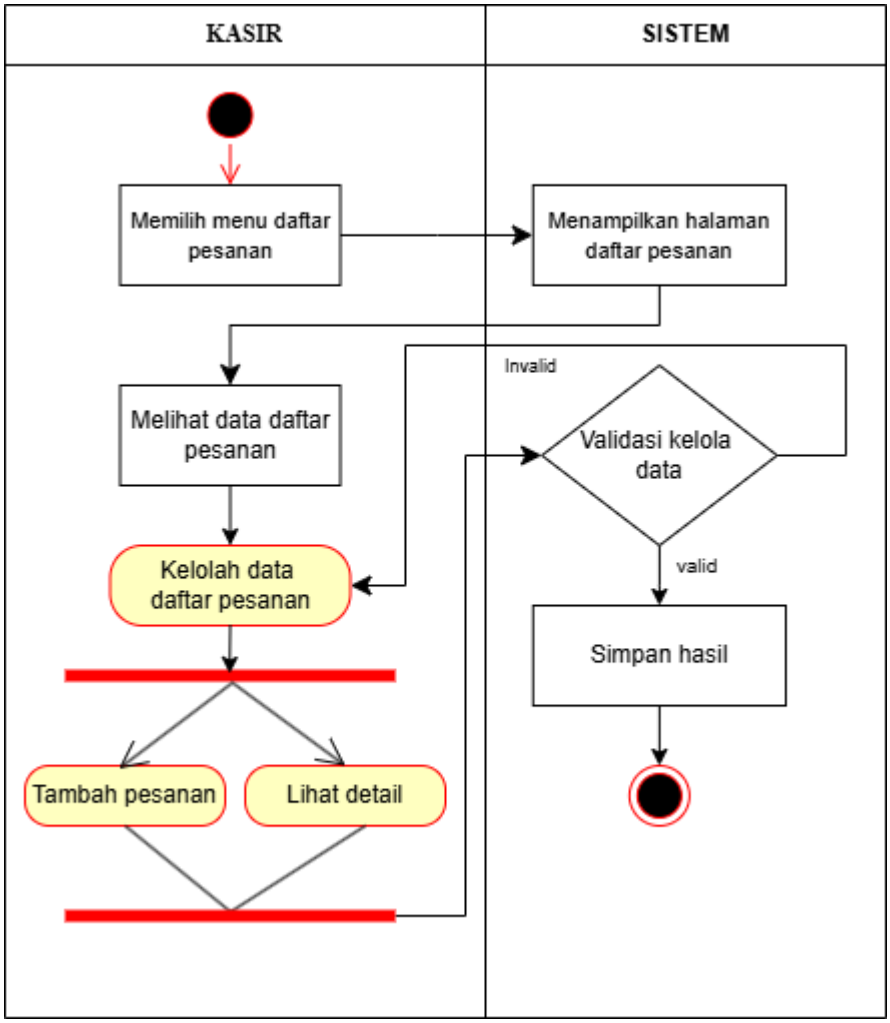
1) Halaman *dashboard* kasir



Gambar 4. 14 *Activity* diagram halaman *dashboard* kasir

Kasir memilih menu *dashboard* dan sistem akan menampilkan halaman in*form*asi terkait penjualan.

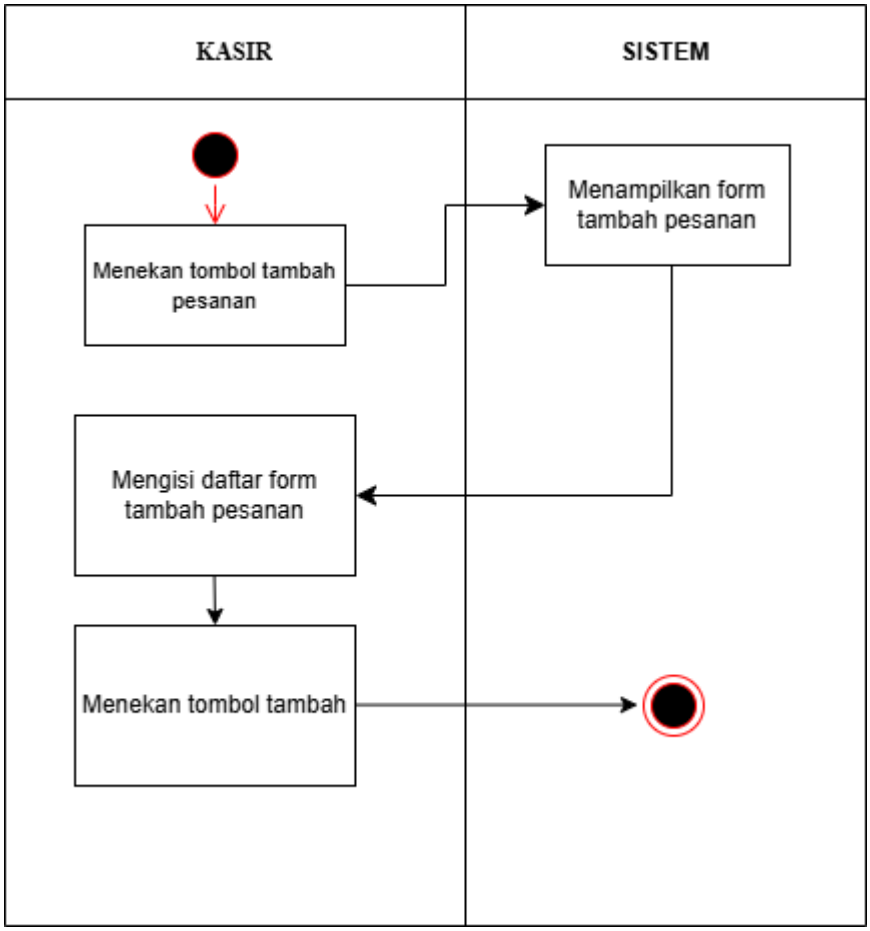
2) Halaman daftar pesanan kasir



Gambar 4. 15 *Activity* diagram daftar pesanan kasir

Kasir memilih menu daftar pesanan kemudian sistem akan menampilkan halaman daftar pesanan yang berisi daftar *customer* yang telah melakukan pemesanan. kemudian kasir melihat data dan mengelolah data dengan opsi tambah pesanan dan lihat detail. Setelah admin memilih tindakan hasilnya disimpan oleh sistem.

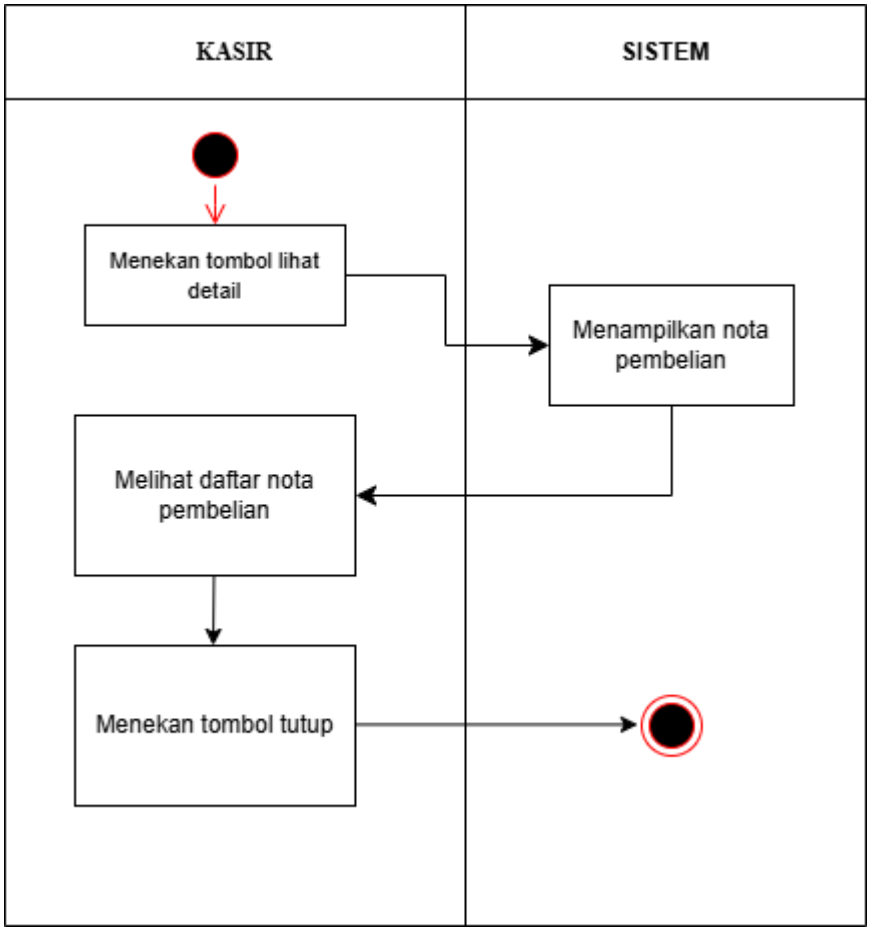
3) Halaman tambah pesanan kasir



Gambar 4. 16 *Activity* diagram tambah pesanan kasir

Kasir menekan tombol tambah pesanan kemudian sistem akan menampilkan *form* tambah pesanan lalu setelah mengisih *form* lalu menekan tombol tambah, kemudian sistem akan menyimpan data.

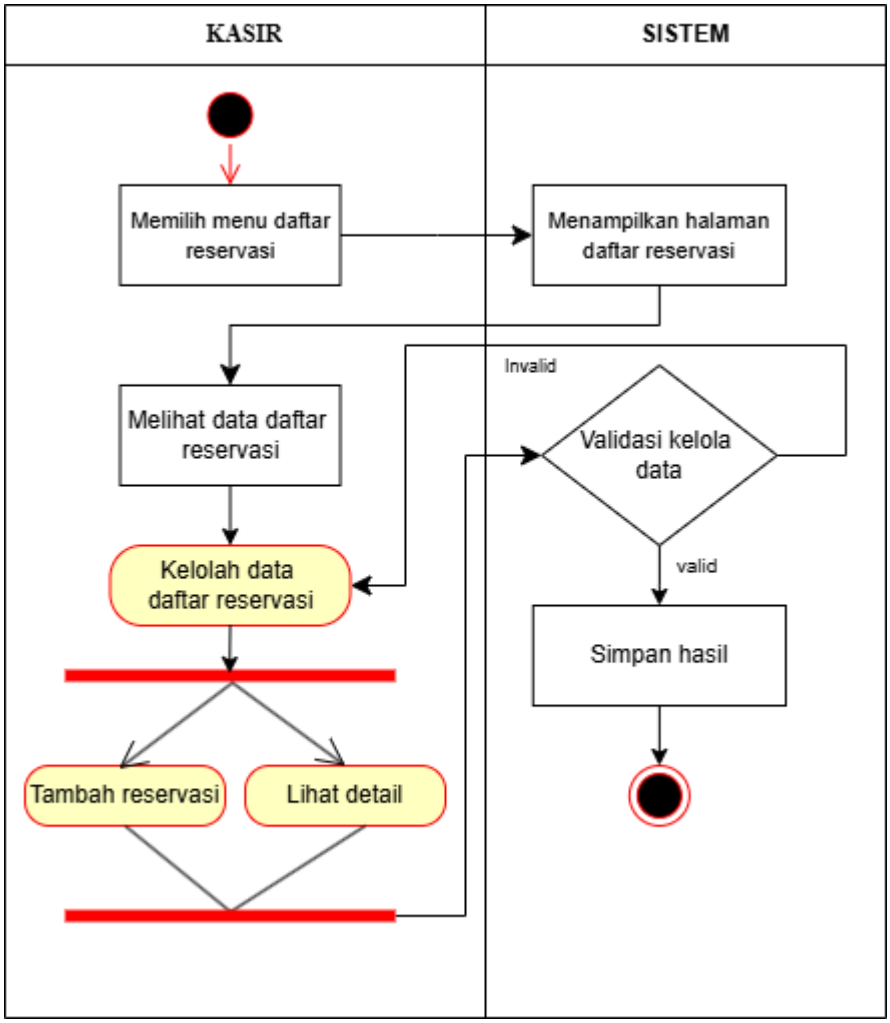
4) Halaman lihat detail pesanan kasir



Gambar 4. 17 *Activity* diagram lihat detail pesanan kasir

Kasir menekan tombol lihat detail kemudian sistem akan menampilkan nota pembelian dari *customer*.

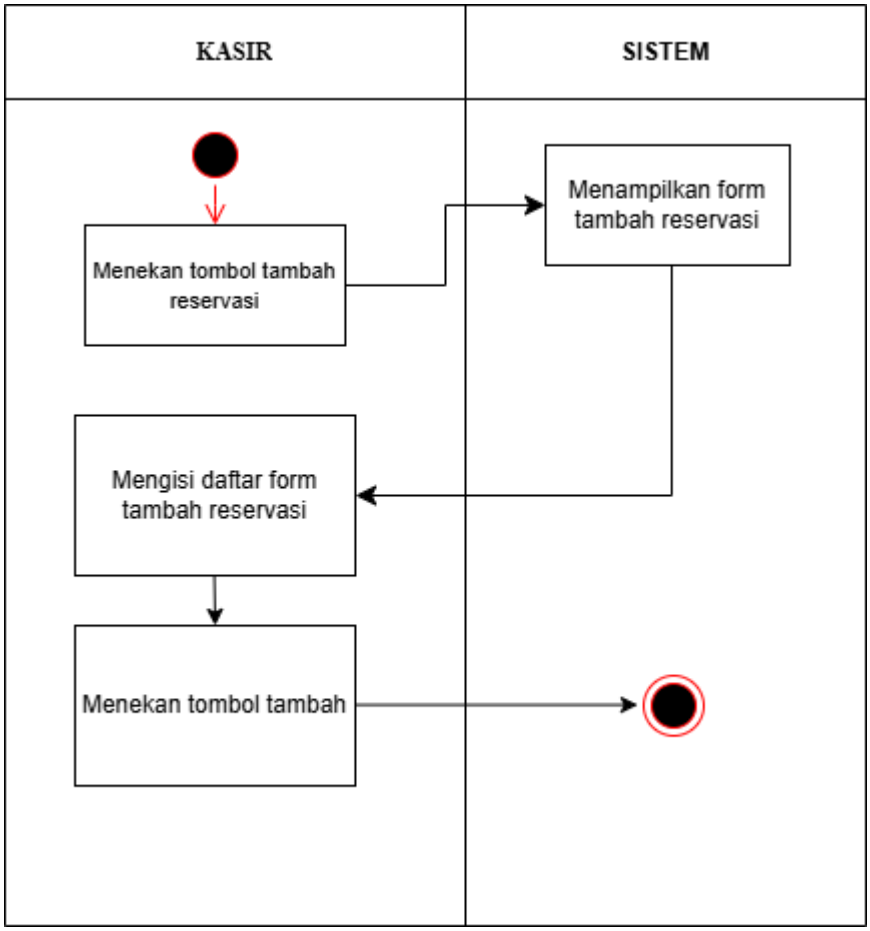
5) Halaman daftar reservasi kasir



Gambar 4. 18 *Activity* diagram daftar reservasi kasir

Kasir memilih menu daftar reservasi kemudian sistem akan menampilkan halaman daftar reservasi yang berisi daftar *customer* yang telah melakukan pemesanan reservasi. kemudian kasir melihat data dan mengelolah data dengan opsi tambah reservasi dan lihat detail. Setelah admin memilih tindakan hasilnya disimpan oleh sistem.

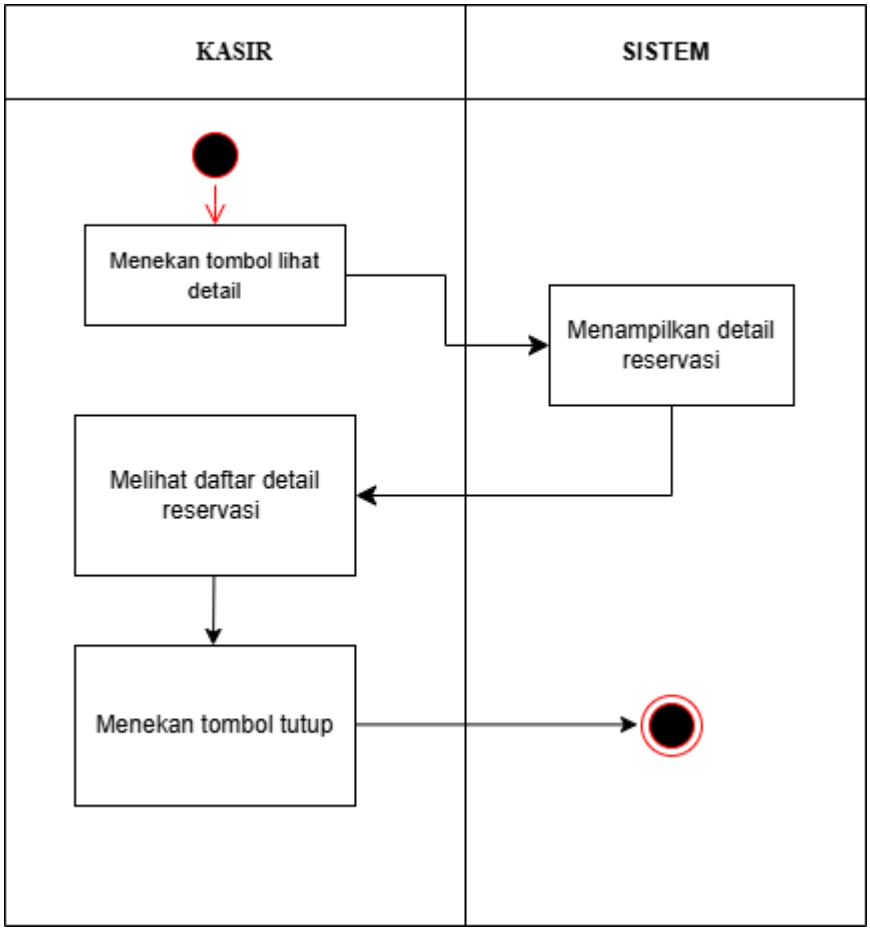
6) Halaman tambah reservasi kasir



Gambar 4. 19 *Activity* diagram tambah reservasi kasir

Kasir menekan tombol tambah reservasi kemudian sistem akan menampilkan *form* tambah reservasi lalu setelah mengisih *form* lalu menekan tombol tambah, kemudian sistem akan menyimpan data.

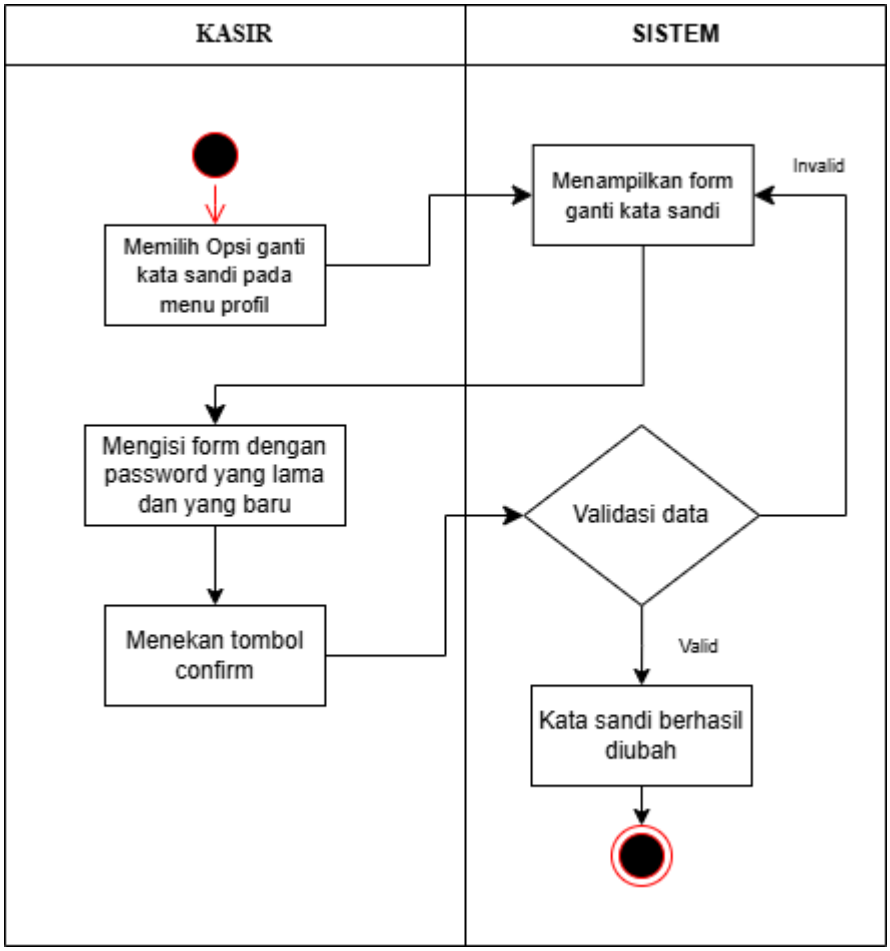
7) Halaman lihat detail reservasi kasir



Gambar 4. 20 *Activity* diagram lihat detail reservasi kasir

Kasir menekan tombol lihat detail kemudian sistem akan menampilkan detail pemesanan reservasi dari *customer*.

8) Halaman ganti sandi kasir

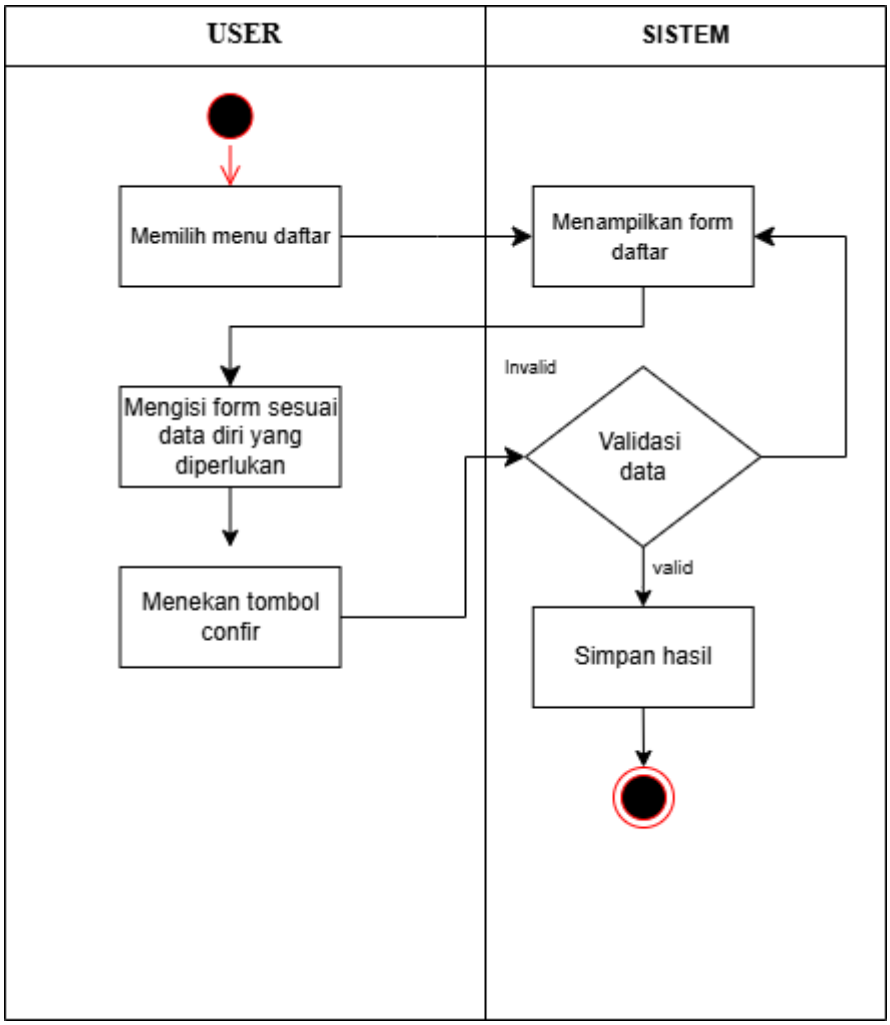


Gambar 4. 21 *Activity* diagram lihat detail reservasi kasir

Kasir memilih opsi ubah kata sandi di menu profil. Kemudian sistem akan menampilkan *form* untuk mengubah kata sandi. Setelah kasir memasukkan kata sandi baru dan menekan tombol Konfirmasi, Sistem akan memvalidasi data. Jika *valid*, kata sandi berhasil diubah. Jika gagal, sistem akan menampilkan *form* perubahan kata sandi lagi.

c. *Customer*

1) Halaman daftar *customer*

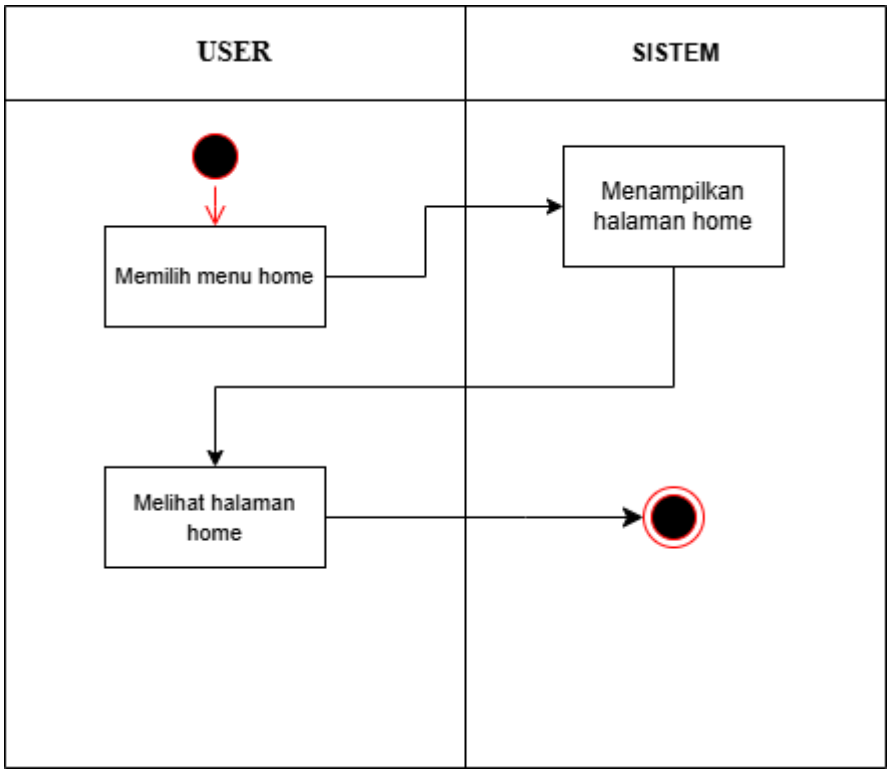


Gambar 4. 22 *Activity* diagram daftar *customer*

Apabila *customer* belum memiliki akun maka *customer* perlu membuat akun dengan memilih menu daftar lalu sistem akan menampilkan *form* daftar, kemudian *customer* mengisi *form* lalu menekan tombol *confirm* setelah itu sistem akan memvalidasi data tersebut. Jika data sudah sesuai maka validasi berhasil dan

customer sudah bisa melakukan *login*, tetapi apabila validasi gagal maka sistem akan mengarahkan kembali ke halaman *form* daftar.

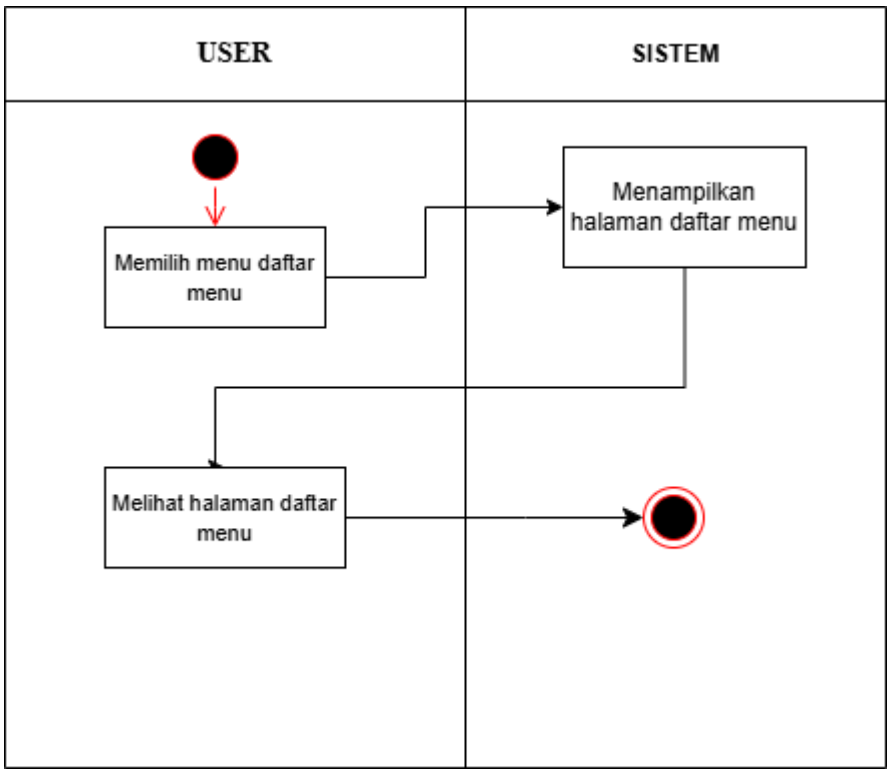
2) Halaman *home customer*



Gambar 4. 23 *Activity* diagram halaman *home customer*

Customer memilih menu *home* lalu sistem akan menampilkan halaman *home*.

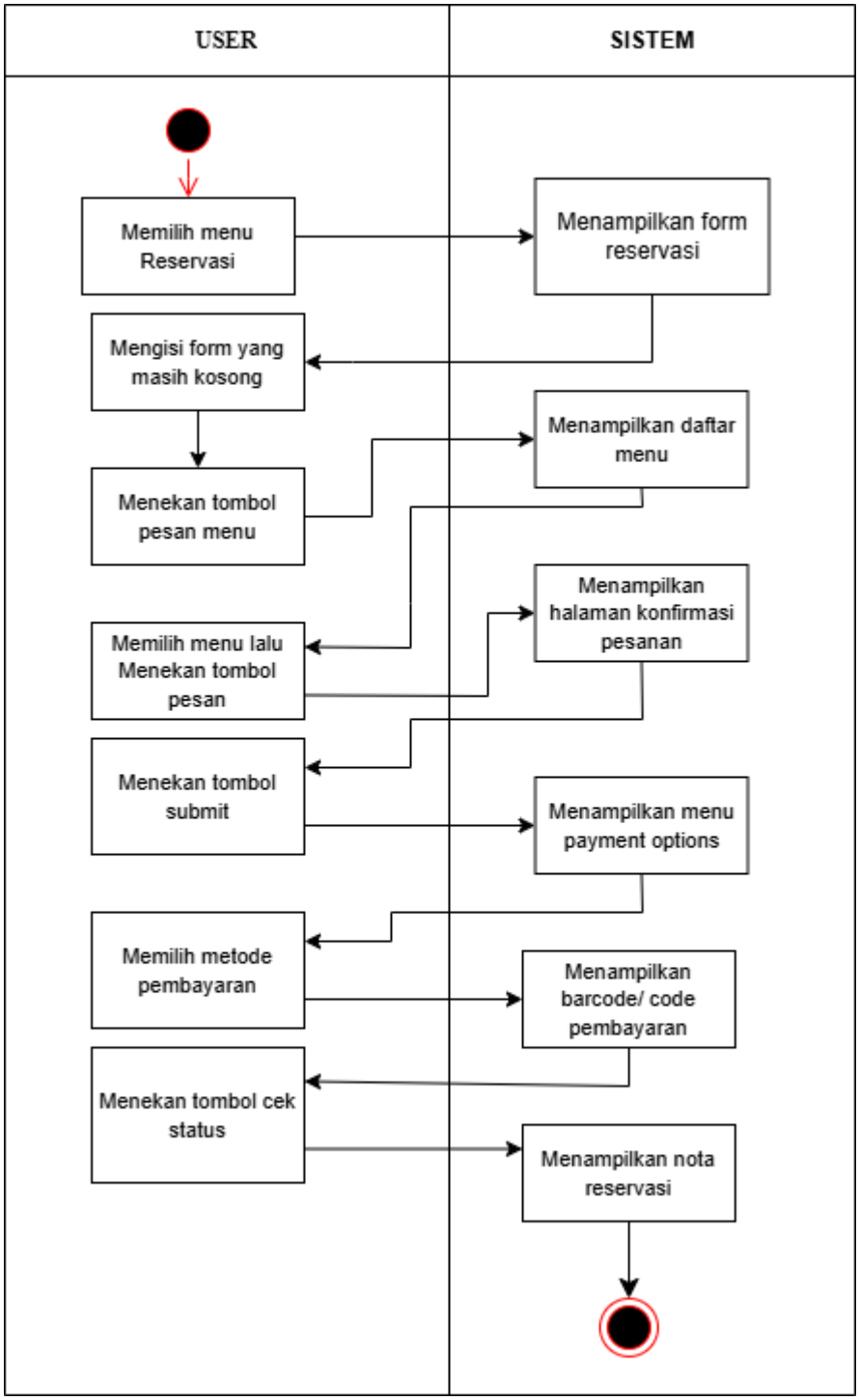
3) Halaman daftar menu *customer*



Gambar 4. 24 *Activity* diagram halaman daftar menu *customer*

Customer memilih menu daftar Menu lalu sistem akan menampilkan halaman daftar menu yang berisi daftar makanan dan minuman.

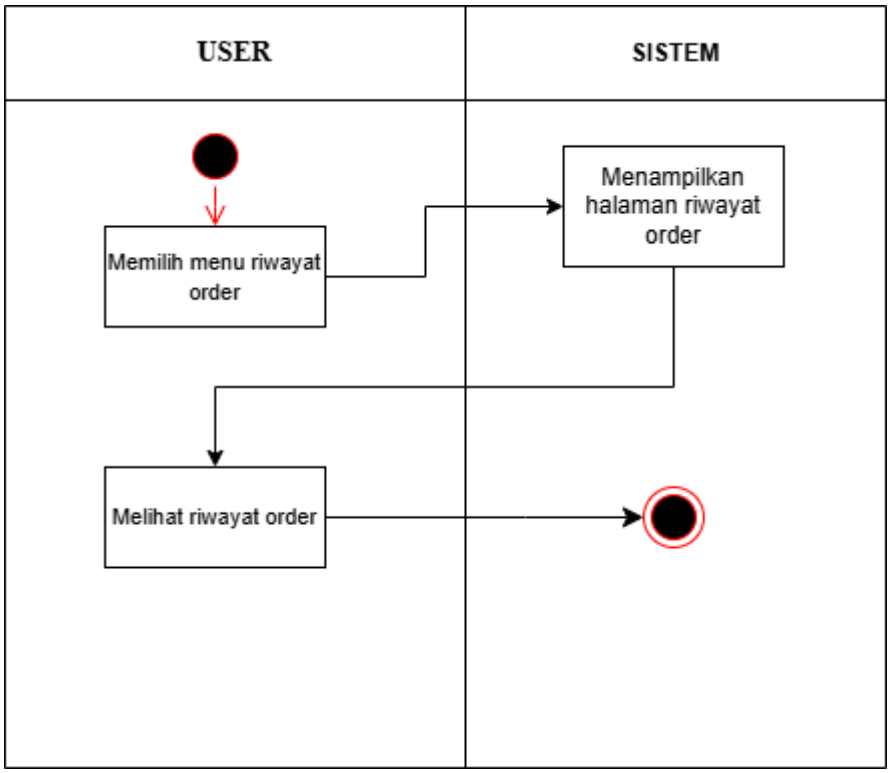
4) Halaman reservasi *customer*



Gambar 4. 25 *Activity* diagram reservasi *customer*

Customer memilih menu reservasi lalu sistem akan menampikan halaman *form* reservasi, *customer* mengisi *form* dan memesan menu, setelah *customer* mengkonfirmasi pesanan sistem akan menampilkan menu *payment options*, *customer* memilih metode pembayaran, setalah melakukan pembayaran sistem akan menampilkan nota pemesanan reservasi.

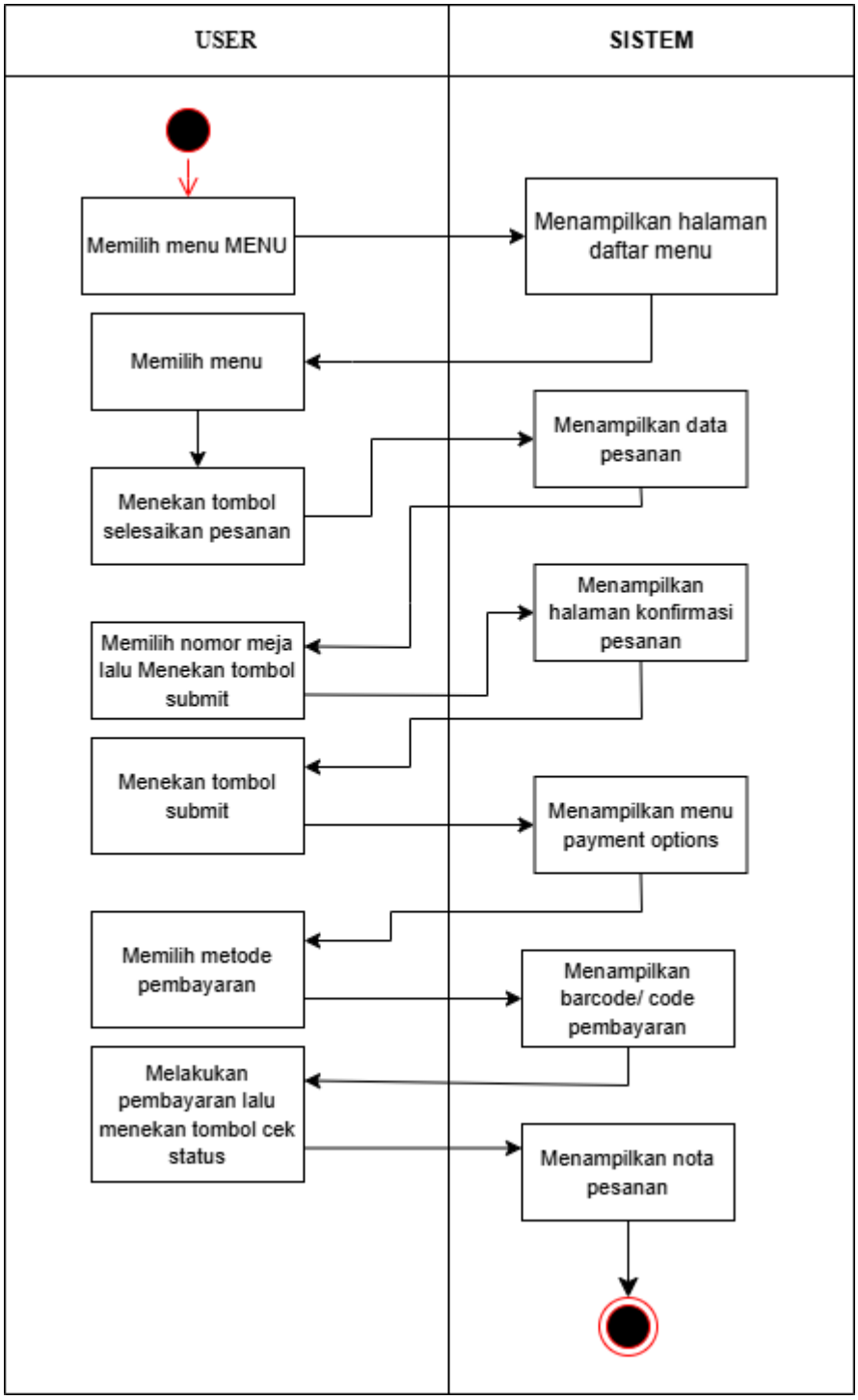
5) Halaman riwayat order *customer*



Gambar 4. 26 *Activity* diagram riwayat order *customer*

Customer memilih menu riwayat order lau sistem akan menampilkan halaman riwayat order.

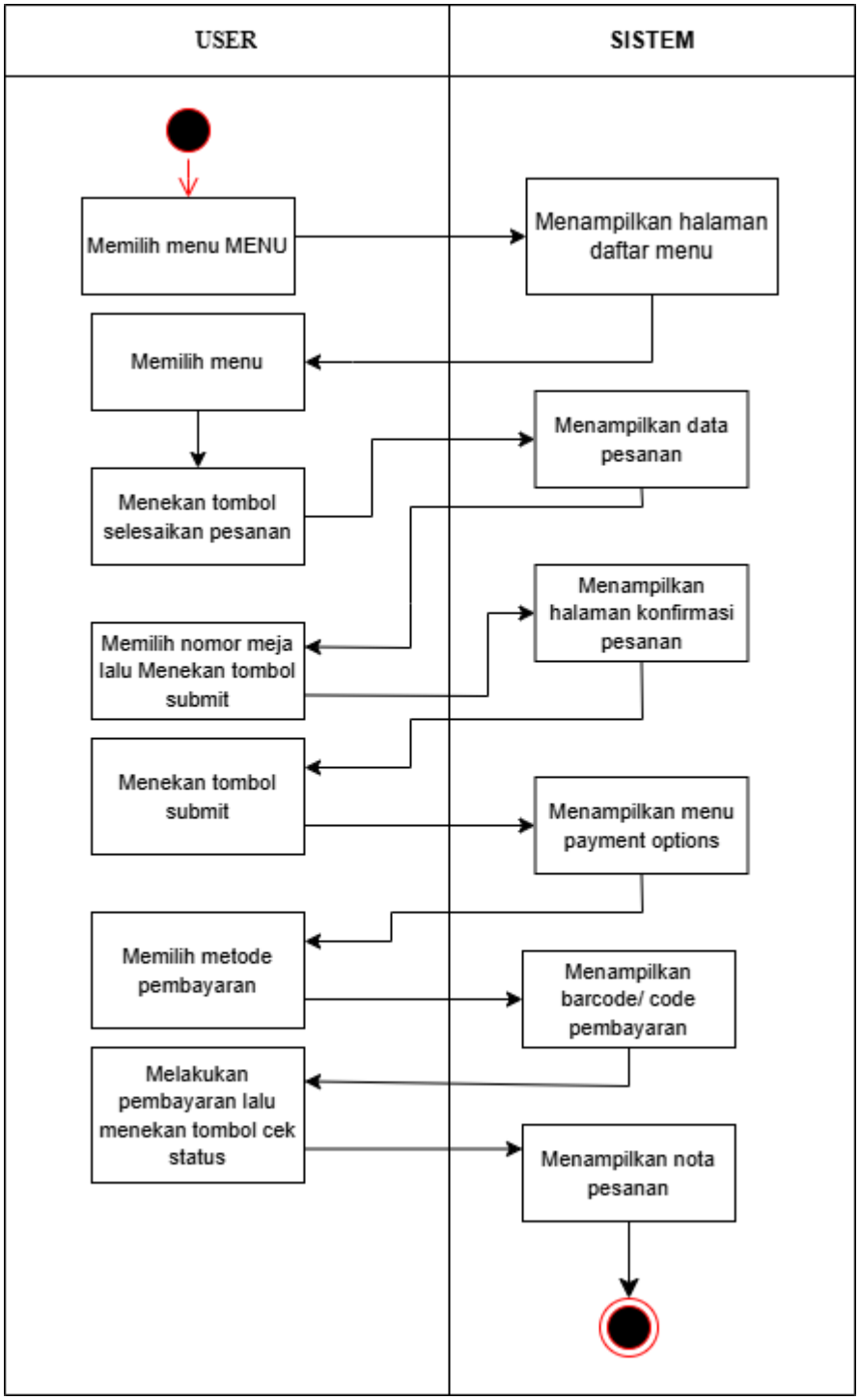
6) Halaman order *customer*



Gambar 4. 27 *Activity* diagram halaman order *customer*

Customer memilih menu “menu” lalu sistem akan menampikan halaman daftar menu, setelah *customer* mengkonfirmasi pesanan sistem akan menampilkan menu *payment options*, *customer* memilih metode pembayaran, setalah melakukan pembayaran sistem akan menampilkan nota pesanan.

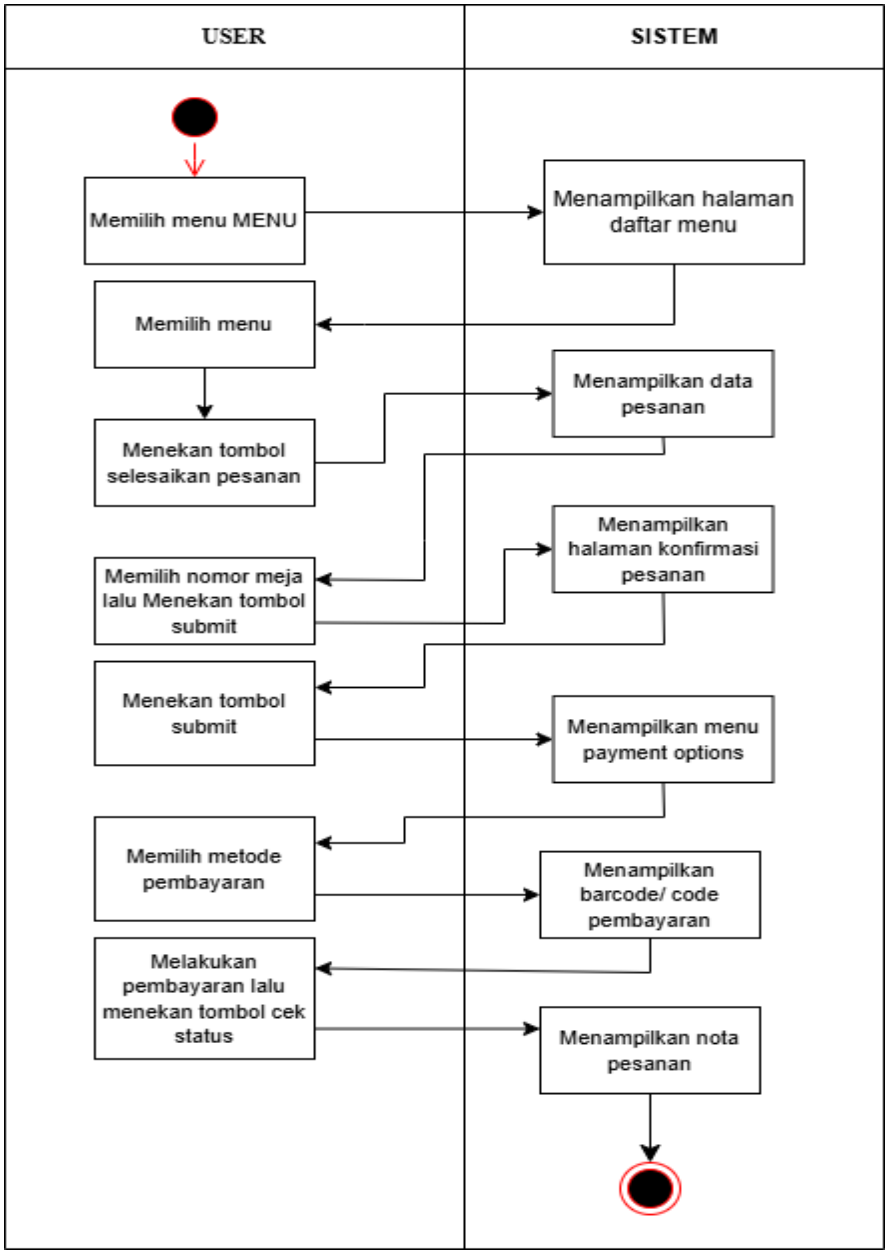
7) Halaman konfirmasi pesanan *customer*



Gambar 4. 28 *Activity* diagram halaman konfirmasi pesanan *customer*

Customer memilih menu "menu" lalu sistem akan menampikan halaman daftar menu, *customer* memilih menu dan meja lalu menekan tombol *submit*, kemudian sistem akan menampilkan halaman konfirmasi pesanan.

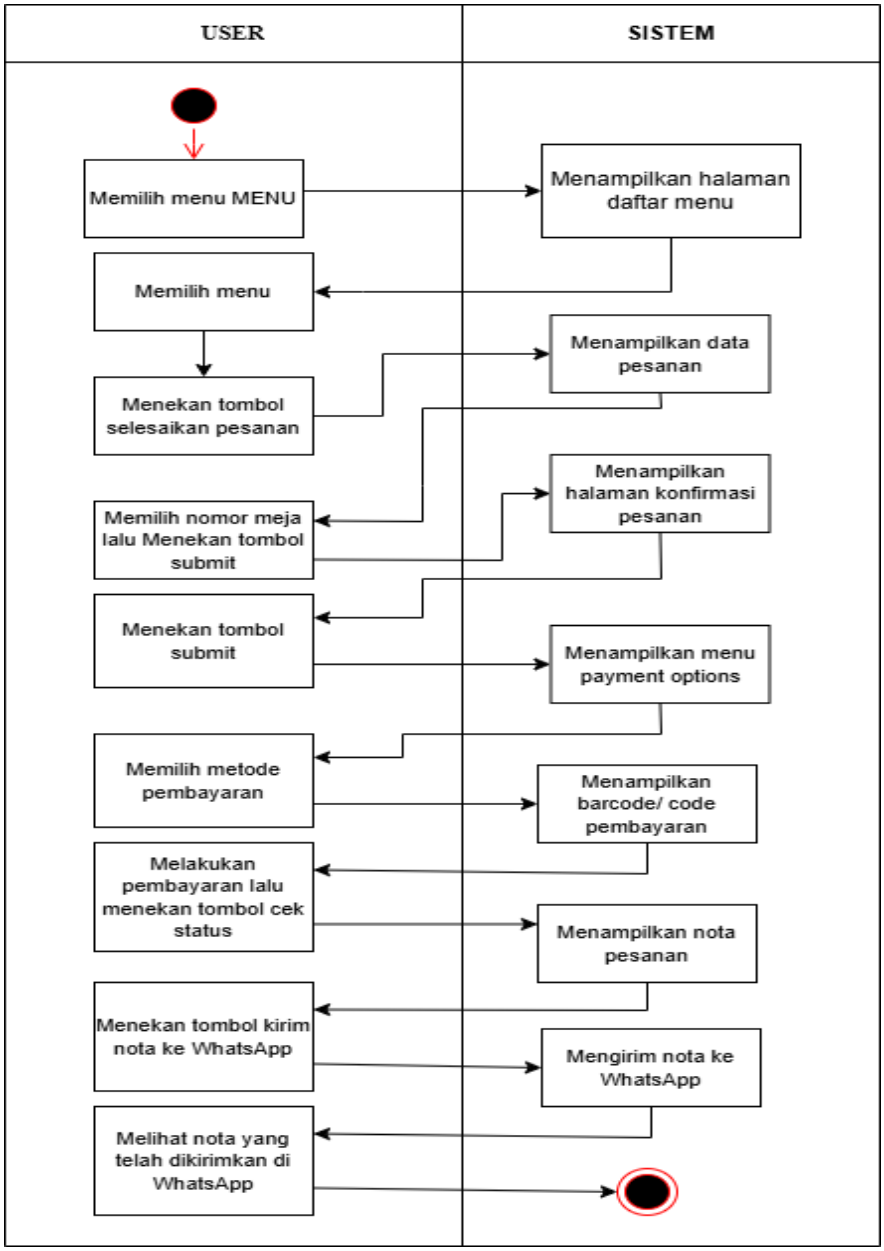
8) Halaman nota pembelian *customer*



Gambar 4. 29 *Activity* diagram halaman nota pembelian *customer*

Customer memilih menu “menu” lalu sistem akan menampikan halaman daftar menu, setelah *customer* mengkonfirmasi pesanan sistem akan menampilkan menu *payment options*, *customer* memilih metode pembayaran, setalah melakukan pembayaran sistem akan menampilkan nota pembelian *customer*.

9) Halaman nota pembelian WhatsApp *customer*



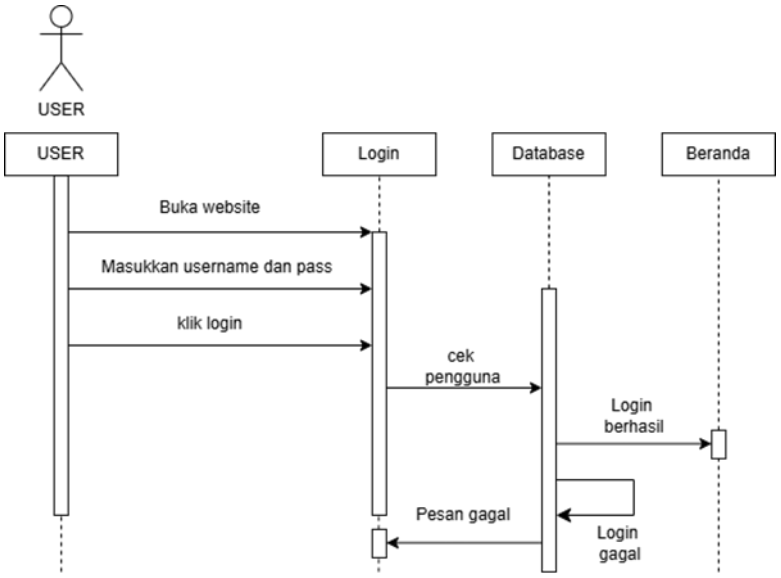
Gambar 4. 30 *Activity* diagram nota pembelian WhatsApp *customer*

Customer memilih menu "menu" lalu sistem akan menampikan halaman daftar menu, setelah *customer* mengkonfirmasi pesanan sistem akan menampilkan menu

payment options, *customer* memilih metode pembayaran, setalah melakukan pembayaran sistem akan menampilkan nota pesanan, *customer* memekan tombol kirim nota ke WhatsApp selanjutnya sistem akan mengirim nota ke WhatsApp *customer*.

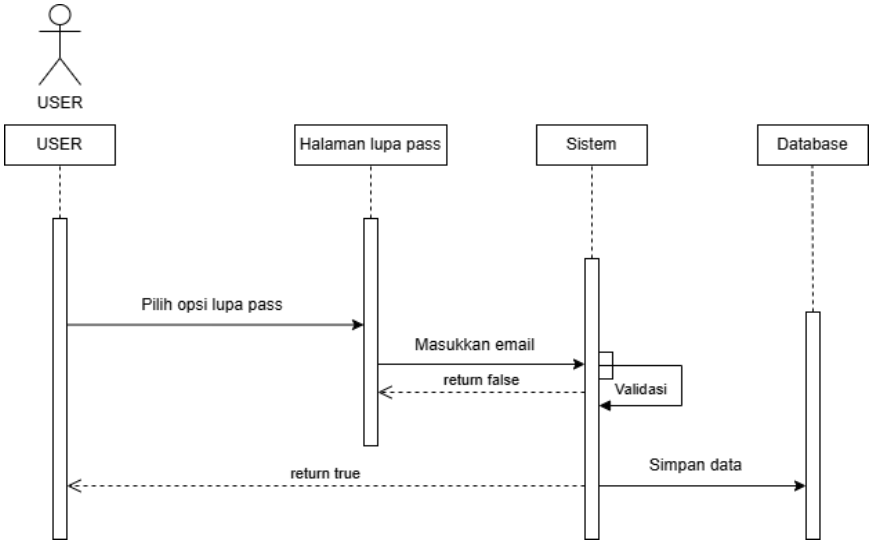
3. Sequence Diagram

1. Halaman *login*



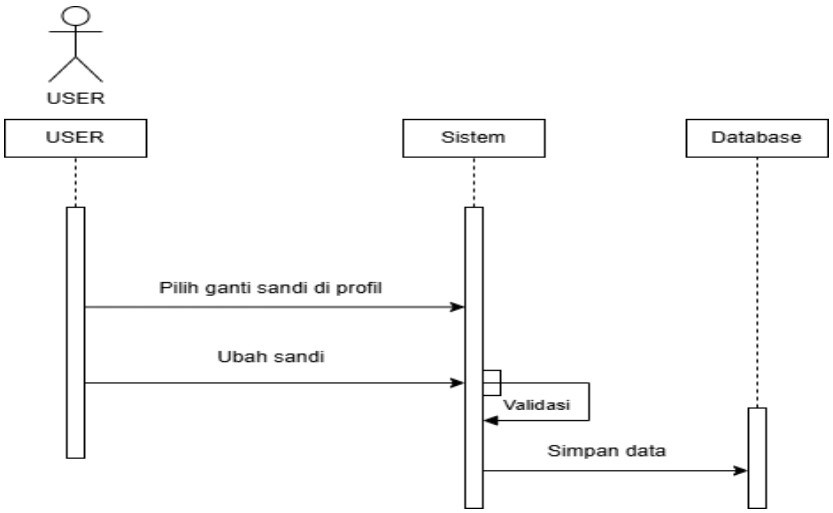
Gambar 4. 31 *Sequence* diagram *login*

2. Halaman lupa *password*



Gambar 4. 32 *Sequence* diagram lupa *password*

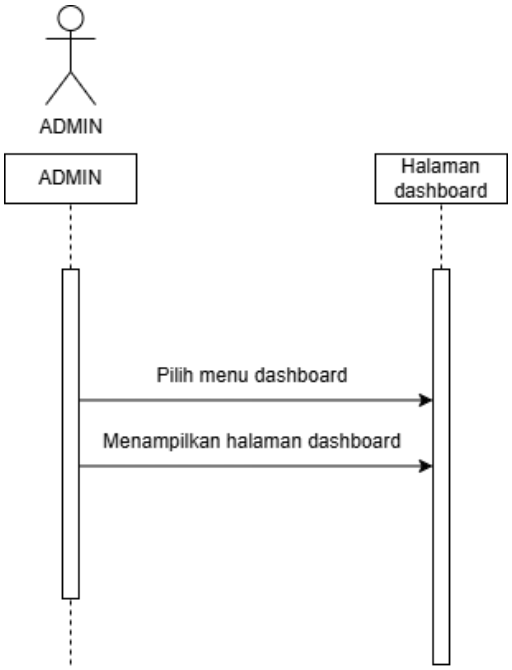
3. Halaman ganti sandi



Gambar 4. 33 *Sequence* diagram ganti sandi

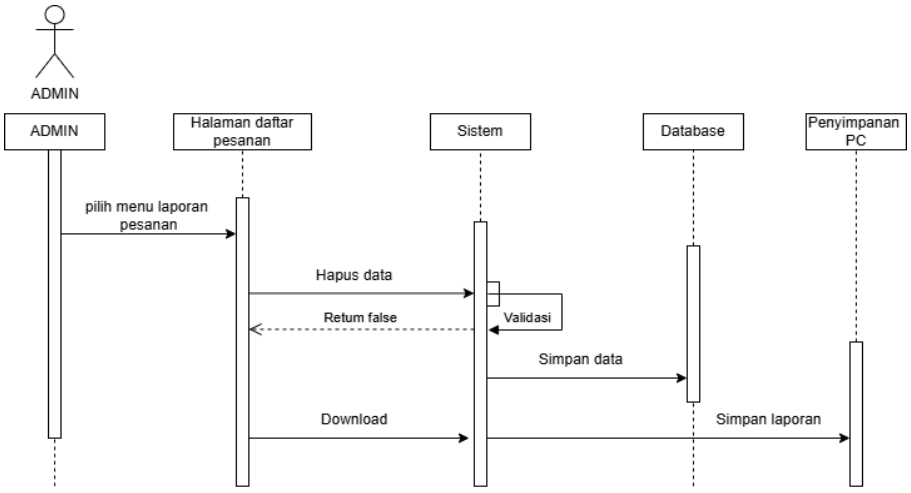
a. Admin

1. Halaman *dashboard* admin



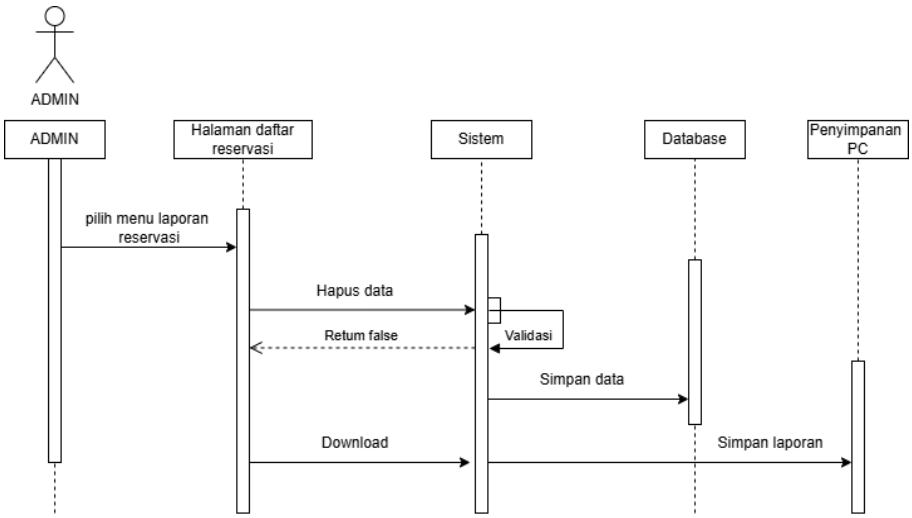
Gambar 4. 34 *Sequence* diagram halaman *dashboard* admin

2. Halaman laporan pesanan admin



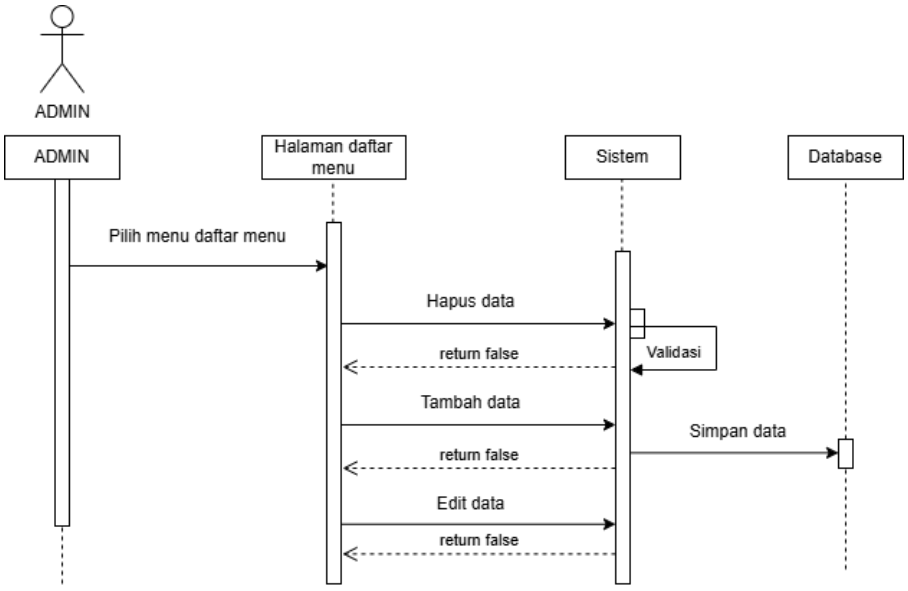
Gambar 4. 35 *Sequence* diagram laporan pesanan admin

3. Halaman laporan reservasi admin



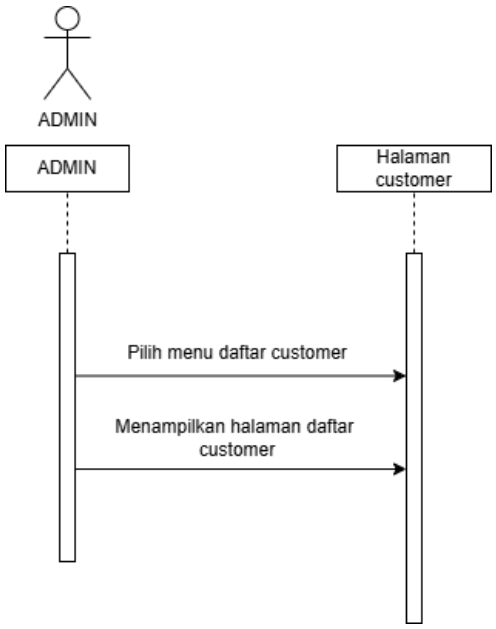
Gambar 4. 36 *Sequence* diagram laporan reservasi admin

4. Halaman daftar menu admin



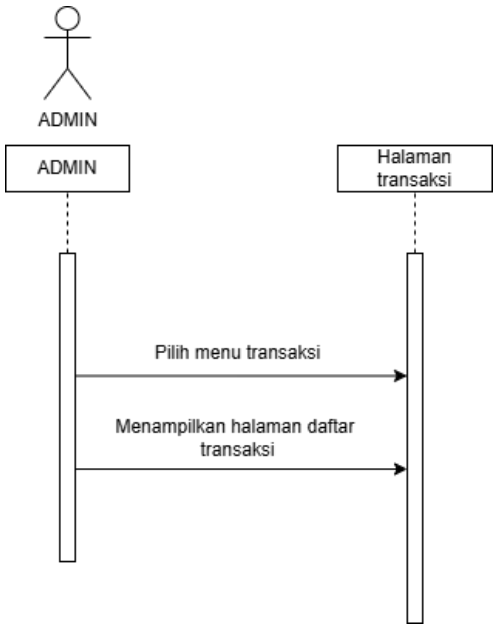
Gambar 4. 37 *Sequence* diagram daftar menu admin

5. Halaman daftar *customer* admin



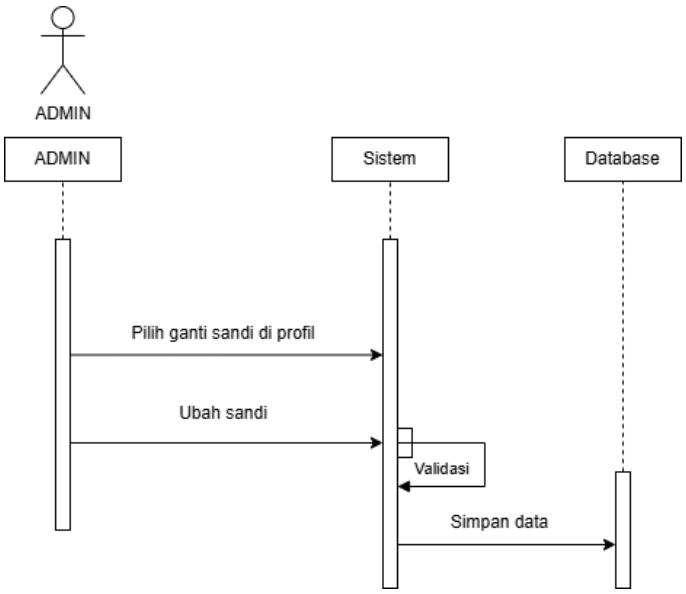
Gambar 4. 38 *Sequence* diagram daftar *customer* admin

6. Halaman transaksi admin



Gambar 4. 39 *Sequence* diagram transaksi admin

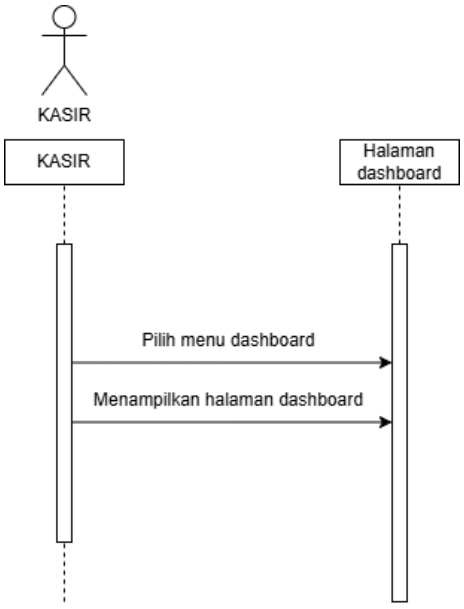
7. Halaman ganti sandi admin



Gambar 4. 40 *Sequence* diagram ganti sandi admin

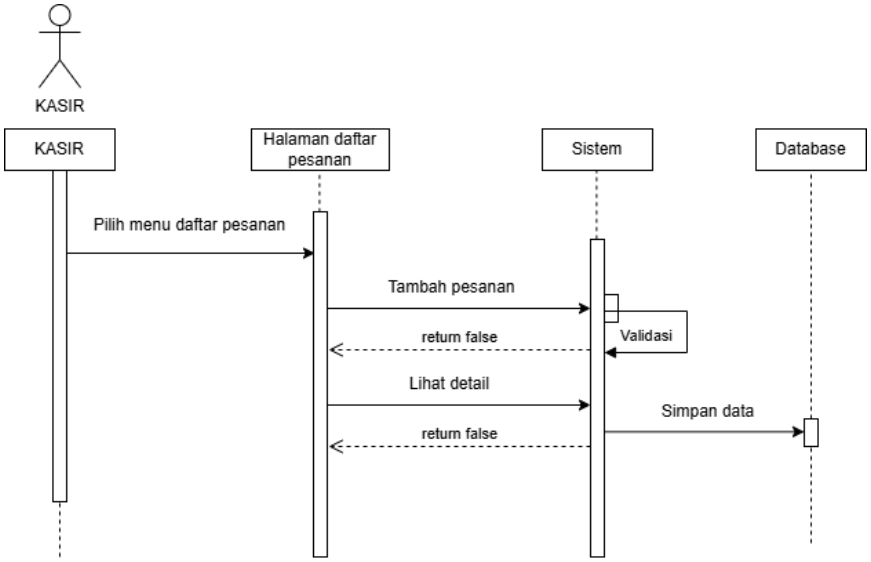
b. Kasir

1) Halaman *dashboard* kasir



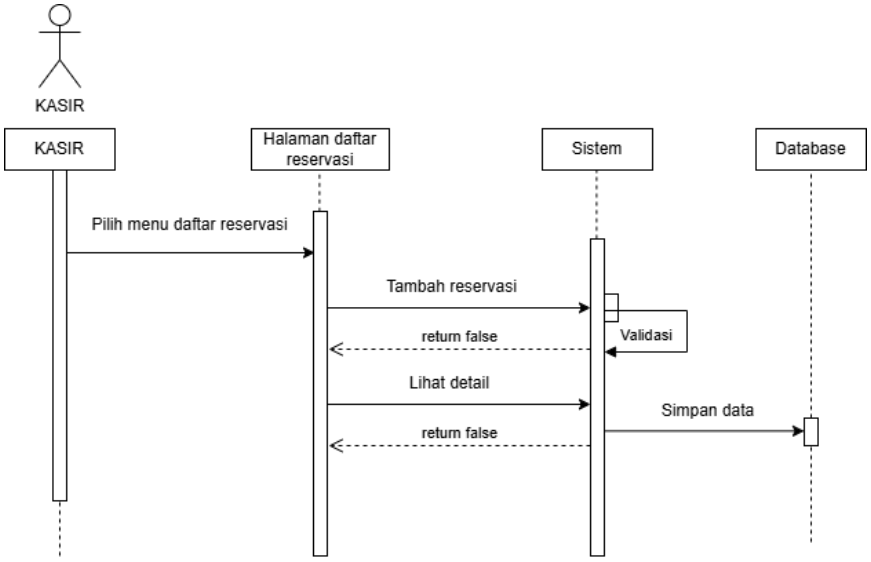
Gambar 4. 41 *Sequence* diagram halaman *dashboard* kasir

2) Halaman daftar pesanan kasir



Gambar 4. 42 *Sequence* diagram daftar pesanan kasir

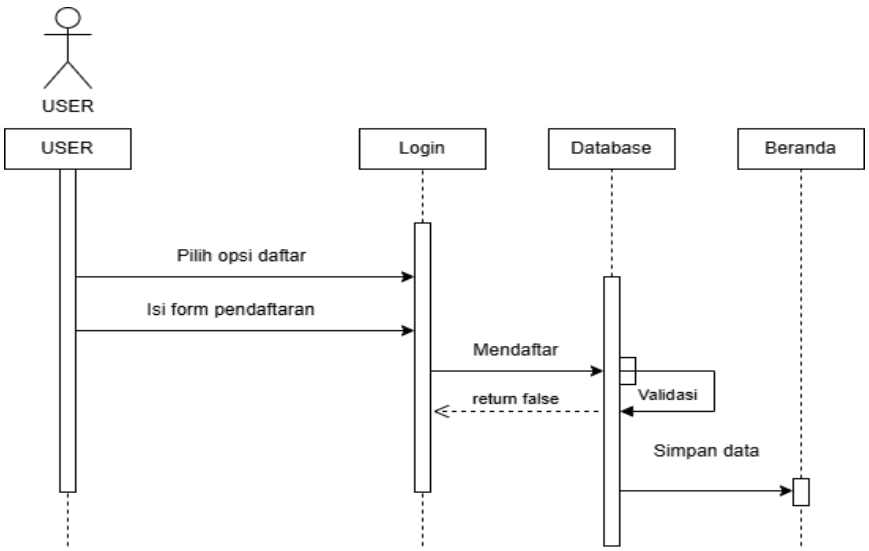
3) Halaman daftar reservasi kasir



Gambar 4. 43 *Sequence* diagram daftar reservasi kasir

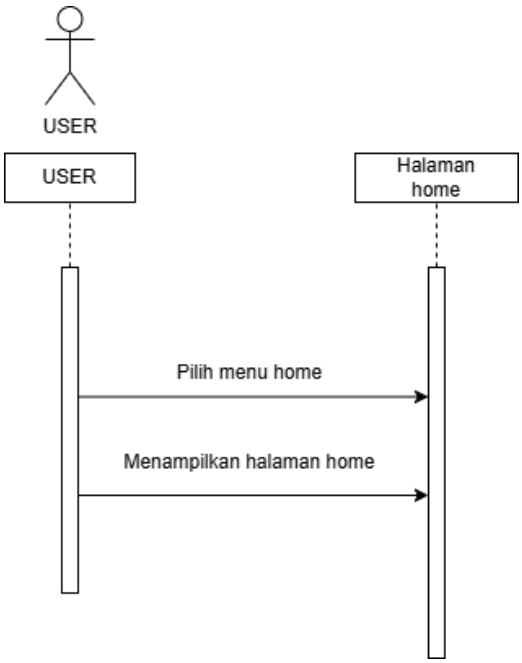
c. *Customer*

1) Halaman daftar *customer*



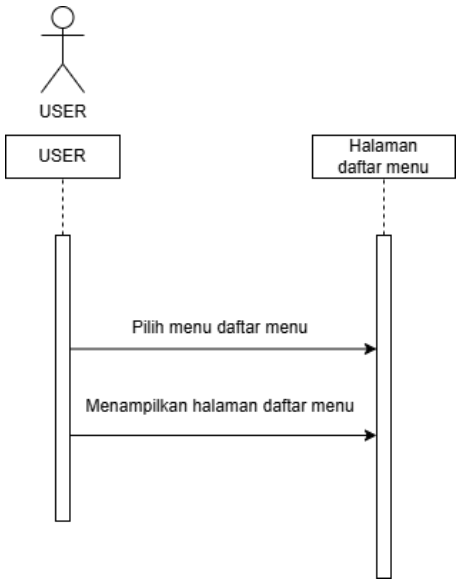
Gambar 4. 44 *Sequence* diagram daftar *customer*

2) Halaman *home customer*



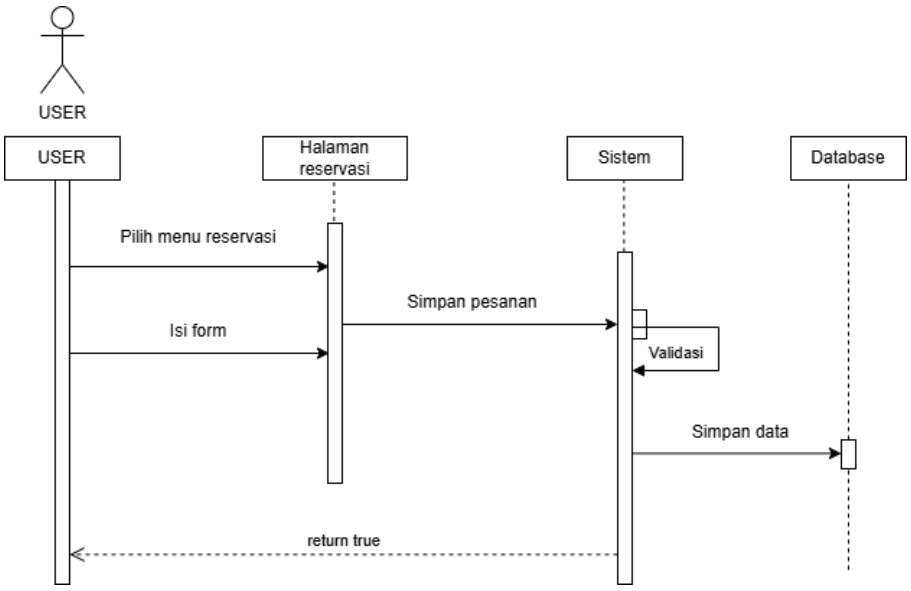
Gambar 4. 45 *Sequence* diagram halaman *home customer*

3) Halaman daftar menu *customer*



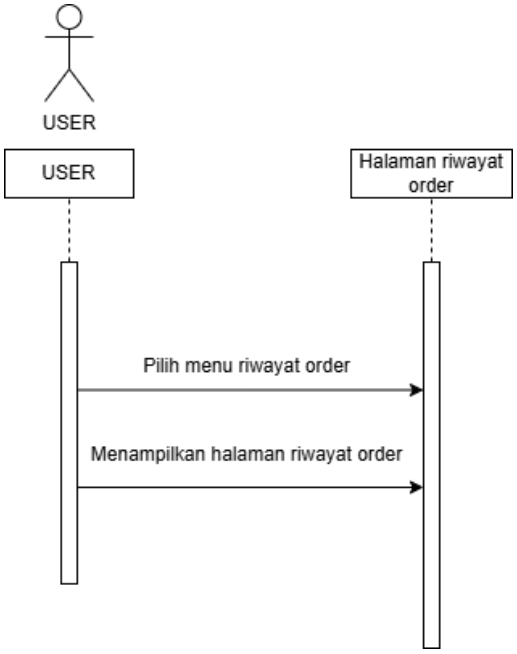
Gambar 4. 46 *Sequence* diagram daftar menu *customer*

4) Halaman reservasi *customer*



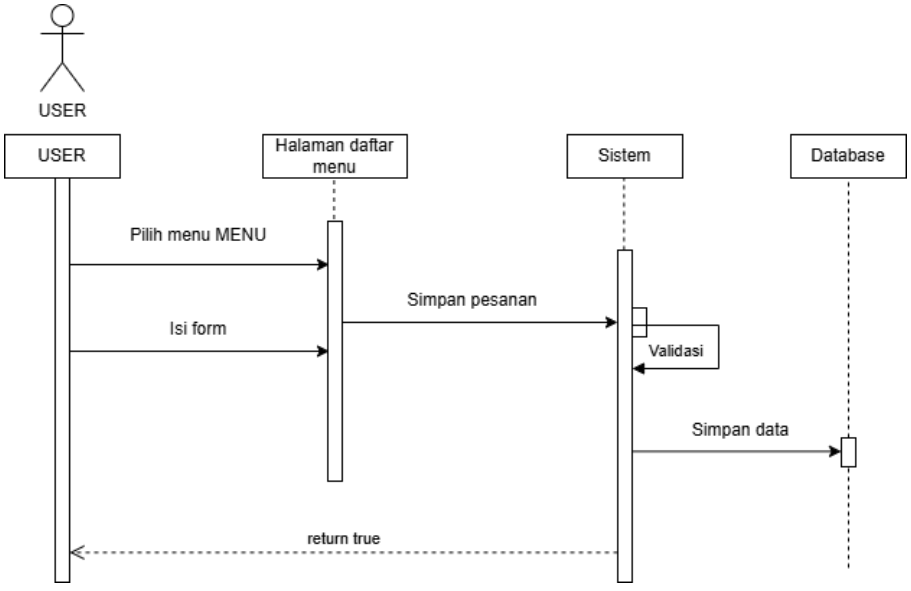
Gambar 4. 47 *Sequence* diagram reservasi *customer*

5) Halaman riwayat order *customer*



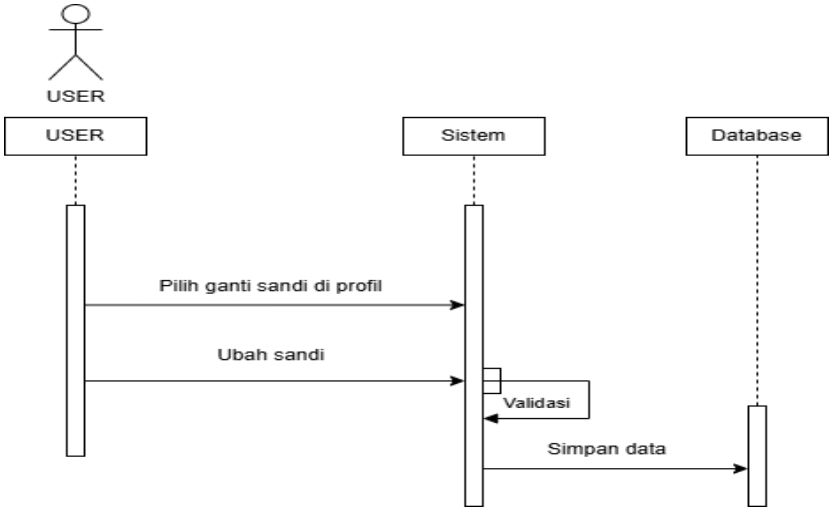
Gambar 4. 48 *Sequence* diagram riwayat order *customer*

6) Halaman order *customer*



Gambar 4. 49 *Sequence* diagram order *customer*

7) Halaman ganti sandi *customer*



Gambar 4. 50 *Sequence* diagram ganti sandi *customer*

4. Rancangan *Database*

Berikut ini rancangan *database* yang digunakan dalam pembuatan aplikasi sistem pemesanan menu menggunakan metode *Multi channel* pada arbas.

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
tbl_menu	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	33	InnoDB	utf8mb4_general_ci	16.0 KB	-
tbl_reservasi	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	10	InnoDB	utf8mb4_general_ci	16.0 KB	-
tbl_users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
tbl_user_order	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	11	InnoDB	utf8mb4_general_ci	16.0 KB	-
4 tabel	Jumlah	59	InnoDB	utf8mb4_general_ci	64.0 KB	0 B

Gambar 4. 51 Struktur *database*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama_menu	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
3	price	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya
4	images	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
5	jenis	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
6	stok	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya

Gambar 4. 52 Struktur tabel menu

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐	1	id	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	2	nama	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	3	email	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	4	whatsapp	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	5	tanggal	date			Ya	*NULL*			Ubah Hapus Lainnya
☐	6	waktu	time			Ya	*NULL*			Ubah Hapus Lainnya
☐	7	jumlah_org	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	8	listmeja	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	9	id_menu	varchar(50)	utf8mb4_general_ci		Ya	*NULL*			Ubah Hapus Lainnya
☐	10	quantity	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	11	order_at	datetime			Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	12	status	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya

Gambar 4. 53 Struktur tabel reservasi

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐	1	id_users	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT	Ubah Hapus Lainnya
☐	2	nama	varchar(100)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	3	email	varchar(100)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	4	password	varchar(100)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	5	no_hp	varchar(100)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	6	alamat	varchar(300)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	7	kode	int(1)			Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	8	reset_token	varchar(255)	utf8mb4_general_ci		Ya	*NULL*			Ubah Hapus Lainnya

Gambar 4. 54 Struktur tabel *user*

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
☐	1	id	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	2	nama	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	3	email	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	4	no_wa	varchar(20)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	5	alamat	text	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	6	meja	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	7	order_at	datetime			Ya	*NULL*			Ubah Hapus Lainnya
☐	8	item	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	9	quantity	varchar(50)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
☐	10	status	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya

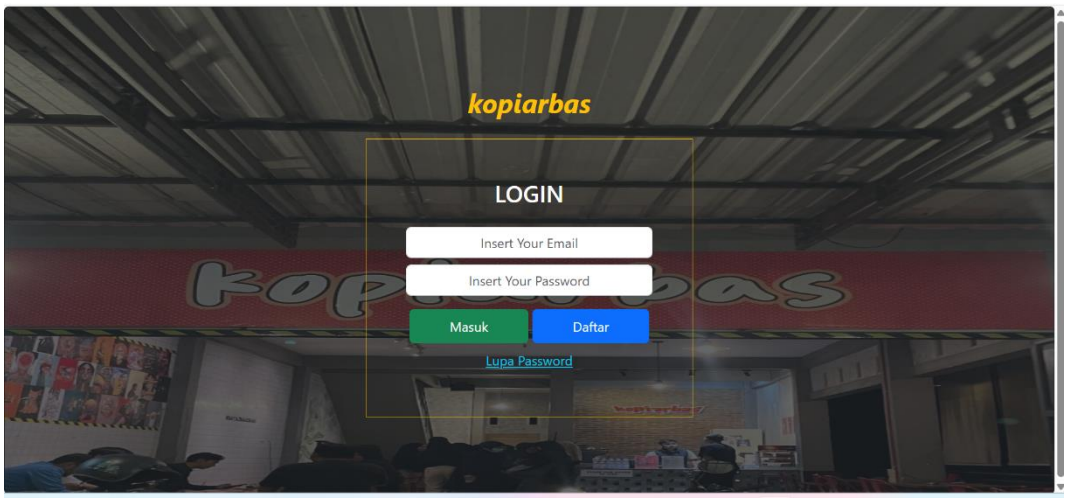
Gambar 4. 55 Struktur tabel *user* order



Gambar 4. 56 *Entity relationship* diagram

B. Detail Sistem

a.) Halaman *login*



Gambar 4. 57 Halaman *login*

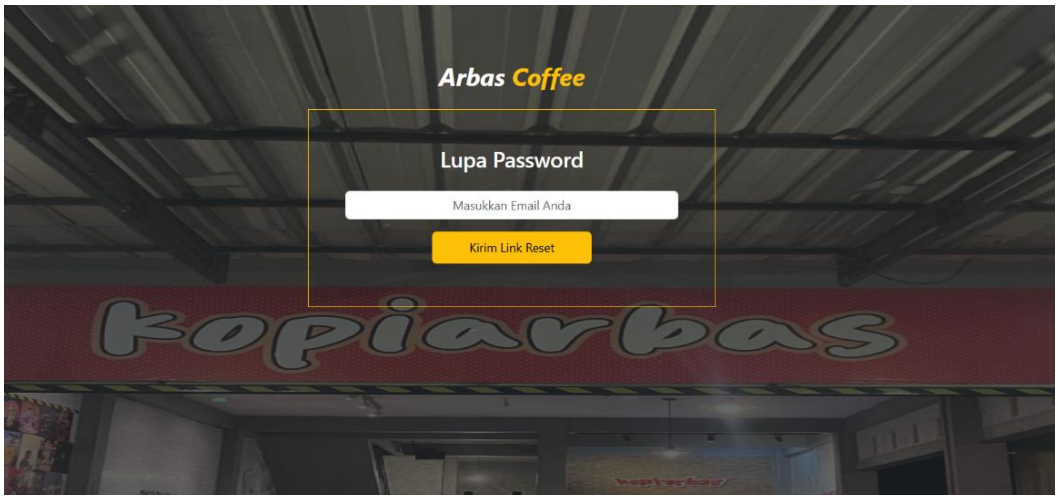
1. Penjelasan kode halaman *login*

(<!DOCTYPE html> dan <html lang="en">) dengan bahasa utama Inggris. Tag <meta charset="utf-8"> memastikan karakter ditampilkan dengan benar, dan <meta name="viewport"> membuat halaman responsif di berbagai perangkat. Bagian login dibungkus dalam <section id="login"> dan tampilan kartu dibuat dengan . Gambar latar diberi efek transparan (opacity: 30%) dan teks "Arbas Coffee" ditampilkan tebal, miring, serta warna kuning pada kata "Coffee".

PHP digunakan untuk menampilkan notifikasi seperti sukses, peringatan, atau error berdasarkan kondisi tertentu menggunakan isset($msg) dan htmlspecialchars() untuk keamanan (mencegah XSS).

Tombol login menggunakan btn btn-success (warna hijau), dan tombol daftar menggunakan btn btn-primary (warna biru). **CSS** mengatur reset margin/padding bawaan, membatasi tinggi maksimal (max-height: 100vh), dan membuat gambar latar transparan, penuh layar, serta berubah saat hover (opacity: 1) untuk efek interaktif.

b.) Halaman lupa *password*



Gambar 4. 58 Halaman lupa *password*

1. Penjelasan kode halaman lupa *password*

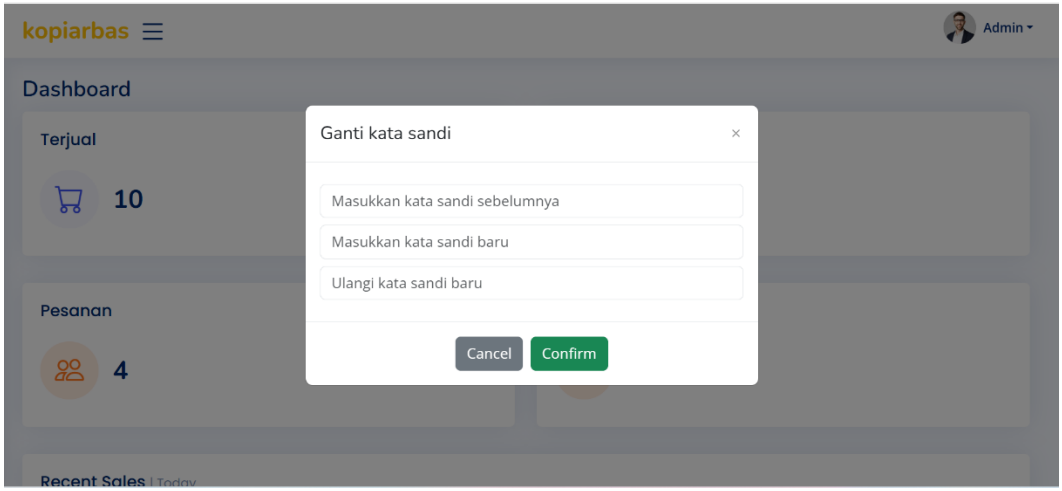
 adalah elemen utama kartu dengan latar gelap, tanpa jarak antar elemen (g-0), dan teks rata tengah. <img class="card-img" style="opacity: 30%"> menampilkan gambar latar transparan agar teks tetap terlihat. meletakkan konten di atas gambar dengan posisi absolut, lebar minimal 22rem, dan jarak atas 5rem.

Grid Bootstrap digunakan dengan dan agar konten berada di tengah. Elemen <h2> berisi teks

"Arbas Coffee" yang tebal (fw-bold), miring (fst-italic), dan bagian "Coffee" diberi warna kuning (text-warning). membungkus isi kartu dengan padding besar dan garis tepi kuning. Formulir menggunakan metode POST. Input email bersifat wajib (required). Tombol kirim menggunakan btn btn-warning (kuning) dan padding (my-2 px-5 py-2) agar lebih nyaman dan menarik.

CSS menghapus padding/margin bawaan (* selector), mengatur tinggi maksimal body ke 100vh untuk mencegah scroll. Gambar latar diset transparan (opacity: 0.5), lebar penuh (width: 100%), dan tinggi layar penuh (height: 100vh). Saat hover, gambar jadi tampak jelas (opacity: 1) untuk efek visual yang menarik.

c.) Halaman ganti sandi



Gambar 4. 59 Halaman ganti sandi

1. Penjelasan kode halaman ganti sandi

Elemen div dengan kelas modal fade digunakan untuk membuat modal dengan efek transisi memudar. Atribut id digunakan sebagai penanda unik. data-bs-backdrop dengan nilai static dan data-bs-keyboard dengan nilai false mencegah modal ditutup melalui klik di luar atau tombol Esc. tabindex="-1" mencegah fokus otomatis. Atribut aria-labelledby dan aria-hidden mendukung aksesibilitas. Kelas modal-dialog modal-dialog-centered menempatkan modal di tengah layar. modal-content menjadi wadah utama isi modal dengan teks gelap melalui text-dark.

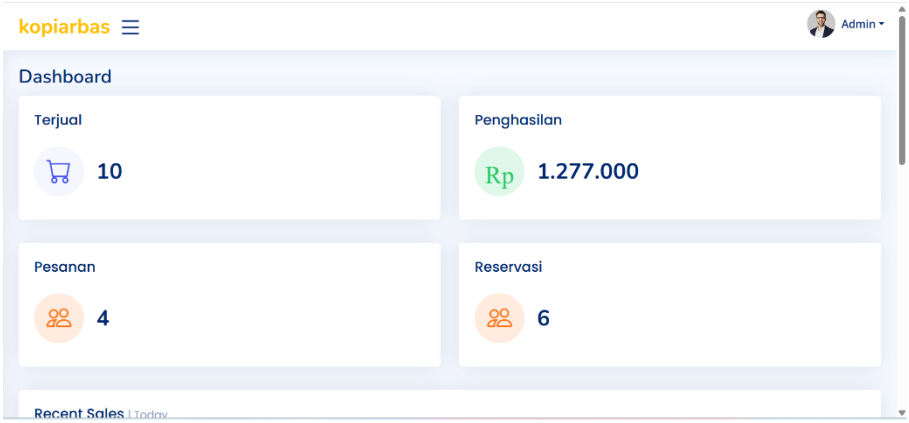
Bagian modal-header berisi judul dan tombol tutup. Judul menggunakan modal-title dan ukuran font kecil, sedangkan tombol tutup memakai btn-close yang langsung menutup modal saat diklik. Bagian modal-body berisi input bertipe password. Kelas form-control my-2 memberikan gaya standar dan jarak atas bawah. Atribut required memastikan input harus diisi. Bagian modal-footer berisi tombol Cancel dan Confirm. Tombol Cancel menggunakan btn btn-secondary dengan warna abu-abu. Tombol Confirm menggunakan btn btn-success berwarna hijau. justify-content-center membuat tombol berada di tengah.

CSS untuk modal mengatur tampilan dasar, posisi, ukuran, latar, dan efek bayangan. Bagian header, body, dan footer masing-masing diatur padding dan batasnya. Kelas tombol mengatur warna, ukuran, dan transisi. Kelas form-control digunakan untuk tampilan input agar konsisten dan responsif.

1. Admin

a. Halaman *dashboard* admin

Halaman *dashboard* berisi info penjualan yang memuat jumlah barang terjual, total penghasilan, jumlah pesanan, dan jumlah reservasi.



Gambar 4. 60 Halaman *dashboard* admin

1. Penjelasan kode halaman *dashboard* admin

Elemen <section> digunakan untuk mengelompokkan konten khusus seperti dashboard, dengan kelas section dashboard yang memberi gaya khusus. memakai grid Bootstrap agar elemen tersusun fleksibel dalam satu baris. Pada card terjual, digunakan col-xxl-4 dan col-md-6 untuk menentukan lebar kolom sesuai ukuran layar. dengan kelas card info-card sales-card membentuk kartu dengan desain Bootstrap. Elemen <h5> menampilkan judul "Terjual", sedangkan menampilkan ikon keranjang belanja dari Bootstrap dengan gaya tengah menggunakan d-flex align-items-center justify-

content-center. <h6> menampilkan jumlah barang terjual yang dihitung dari dua tabel dengan status berhasil.

Card penghasilan memiliki struktur grid yang sama. Elemen berfungsi sebagai kartu info penghasilan dengan kelas revenue-card. <h5> menampilkan judul "Penghasilan", dan ikon menggunakan bi-currency-dollar dengan tambahan rupiah-icon agar sesuai format Rupiah. <h6> menampilkan total pendapatan dengan fungsi number_format untuk memudahkan pembacaan angka.

Pada card pesanan, <h5> menampilkan judul "Pesanan" dan ikon menggunakan bi-people untuk menggambarkan jumlah pelanggan. <h6> menampilkan jumlah pesanan yang diproses, dihitung dengan mysqli_num_rows(). Card reservasi menampilkan judul "Reservasi" pada <h5>, dan <h6> menampilkan jumlah reservasi yang berhasil dari tbl_reservasi. <?php include("daftarpesanan.php"); ?> menyisipkan file eksternal daftar pesanan. Variabel $query mengambil data pendapatan bulanan dari dua tabel dan mengelompokkannya berdasarkan bulan. ORDER BY bulan mengurutkan data berdasarkan bulan terbaru. Fungsi number_format memformat angka pendapatan.

CSS pada .rupiah-icon menggunakan ::before untuk menambahkan teks "Rp" di depan ikon dengan font-family: inherit. Pada bagian general styles, :root mengaktifkan scrolling halus, font-family diatur ke "Open Sans", background berwarna biru keabu-abuan, dan color teks menjadi abu-abu gelap. Warna link default biru dan berubah saat hover. Heading menggunakan font "Nunito".

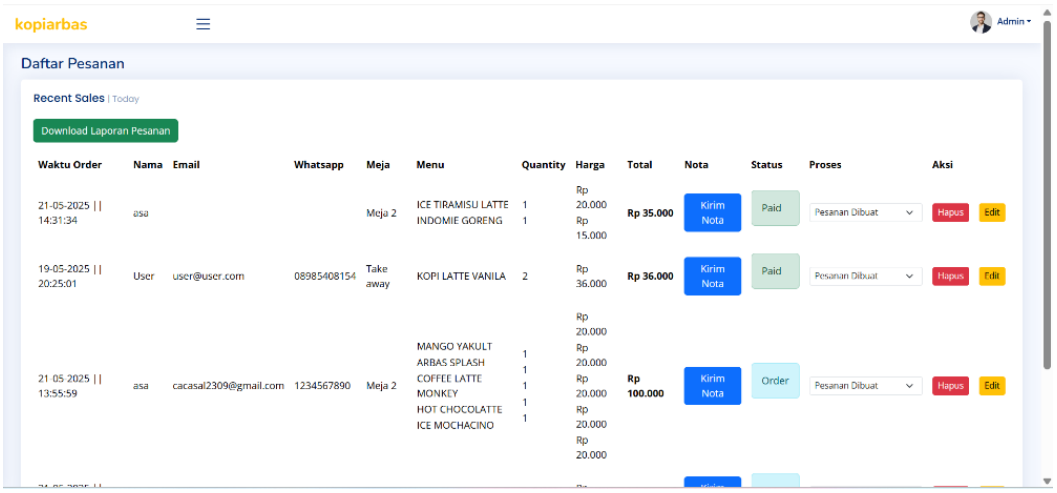
Card styles mengatur margin bawah 30px, tanpa border, radius 5px, dan bayangan lembut. .card-header dan .card-footer memiliki border abu terang dan latar putih. Teksnya abu kebiruan dan padding 15px. .card-title diberi padding atas bawah, font-size 18px, font-weight 500, warna biru tua, dan font "Poppins". .card-body diberi padding 20px.

.info-card diberi margin bawah 10px. .info-card h6 memiliki font-size 28px, warna biru tua, dan font-weight 700. .card-icon berukuran 64px, font-size 32px, dan tidak berubah ukuran karena flex-shrink dan flex-grow bernilai 0. Warna ikon disesuaikan untuk tiap kartu: .sales-card .card-icon berwarna biru tua dengan latar biru muda, .revenue-card .card-icon berwarna hijau dengan latar hijau muda, dan .customers-card .card-icon berwarna oranye dengan latar oranye muda.

Flex utilities seperti .d-flex, .align-items-center, dan .justify-content-center digunakan untuk menyusun elemen secara responsif dan terpusat. .ps-3 memberi padding kiri. .rupiah-icon menambahkan teks "Rp" dan mengikuti font elemen induk.

B. Halaman laporan pesanan admin

Halaman laporan pesanan berisi daftar pesanan yang telah dilakukan oleh *customer*, pada halaman ini admin dapat men*download* laporan pesanan, menghapus, dan melihat *customer* yang sudah melakukan pembayaran dan belum.



Gambar 4. 61 Halaman laporan pesanan admin

Halaman *download* laporan pesanan admin

	A	B	C	D	E	F	G	H	I	J	K
1	LAPORAN DAFTAR PESANAN ARBAS COFFEE										
2											
3	Waktu Order	Nama	Email	Whatsapp	Meja	Menu	Quantity	Price	Total	Status	
4	05/12/2024 07:37	User	user@user.com	85298940782	Meja 2	KOPI LATTE VANILA	2	18000	36000	Done	
5	05/12/2024 07:47	User	user@user.com	85298940782	Meja 3	KOPI LATTE VANILA	4	18000	72000	Done	
6	05/12/2024 08:23	User	user@user.com	85298940782	Meja 3	HAZELNUT FRAPPUCINO	2	20000	40000	Done	
7	06/12/2024 05:40	User	user@user.com	85298940782	Meja 4	MOCHACINO	2	18000	36000	Done	
8	07/12/2024 10:39	tasya	219280110.taliatasalsyabilla@gmail.com	82296332982	Meja 1	KOPI SUSU GULA AREN	1	15000	15000	Done	
9	14/12/2024 18:59	User	user@user.com	85298940782		KOPI SUSU GULA AREN	1	15000	15000	Done	
10	16/12/2024 20:15	ugiy				AMERICANO, BON-BON	1, 1	15000, 12000	27000	Order	
11	29/12/2024 18:55	User	user@user.com	85298940782	Meja 1				0	Order	
12	29/12/2024 18:55	User	user@user.com	85298940782	Meja 1				0	Order	
13	29/12/2024 18:55	User	user@user.com	85298940782	Meja 1				0	Order	
14	09/01/2025 08:28	User	user@user.com	85298940782	Meja 1				0	Done	
15	10/01/2025 14:25	User	user@user.com	85298940782	Meja 2	ESPRESSO, KOPI SUSU GULA AREN, MOCHACINO	1, 1, 1	10000, 15000, 18000	43000	Done	
16	20/01/2025 05:59	User	user@user.com	85298940782	Meja 1	KOPI SUSU GULA AREN	1	15000	15000	Done	
17	20/01/2025 05:59	User	user@user.com	85298940782	Meja 1	KOPI SUSU GULA AREN	1	15000	15000	Order	
18											
19											

Gambar 4. 62 Halaman *download* laporan pesanan admin

1. Penjelasan kode halaman pesanan admin

PHP proses penghapusan data digunakan untuk menghapus pesanan berdasarkan ID yang dikirim melalui form. Variabel $pesan menyimpan hasil proses. Saat metode request adalah POST dan parameter hapus_id valid, sistem menjalankan query SQL DELETE FROM tbl_user_order WHERE id = ?. Jika berhasil, $pesan diisi "sukses", jika gagal "error". Bagian utama tampilan berada dalam <section id="daftar_pesanan"> dengan struktur grid Bootstrap menggunakan dan . Di dalamnya terdapat yang menampilkan daftar pesanan. Bagian memuat judul <h5> bertuliskan "Recent Sales | Today" dan form tombol untuk mengunduh laporan ke export_pesanan.php.

Jika $pesan berisi "sukses", muncul notifikasi hijau "Data berhasil dihapus", jika gagal muncul notifikasi merah "Terjadi kesalahan saat menghapus data". Bagian utama daftar menggunakan <table class="table table-borderless">. Bagian <thead> memuat kolom seperti Waktu Order, Nama, Email, WhatsApp, Meja, Menu, Quantity, Harga, Total, Nota, Status, dan Aksi. Data ditampilkan dengan while dari tabel tbl_user_order menggunakan PHP seperti <?= $row["nama"]; ?>. Nomor WhatsApp diubah dari 0 ke 62. Menu ditampilkan dengan foreach dari tabel tbl_menu, dan total harga dihitung dari jumlah pesanan dikali harga menu.

Link WhatsApp dibentuk melalui variabel $whatsappLink yang menyertakan nomor dan isi nota. Pada bagian aksi, tombol hapus dibungkus dalam

form POST dan disertai konfirmasi JavaScript onclick="return confirm('Yakin ingin menghapus data ini?')".

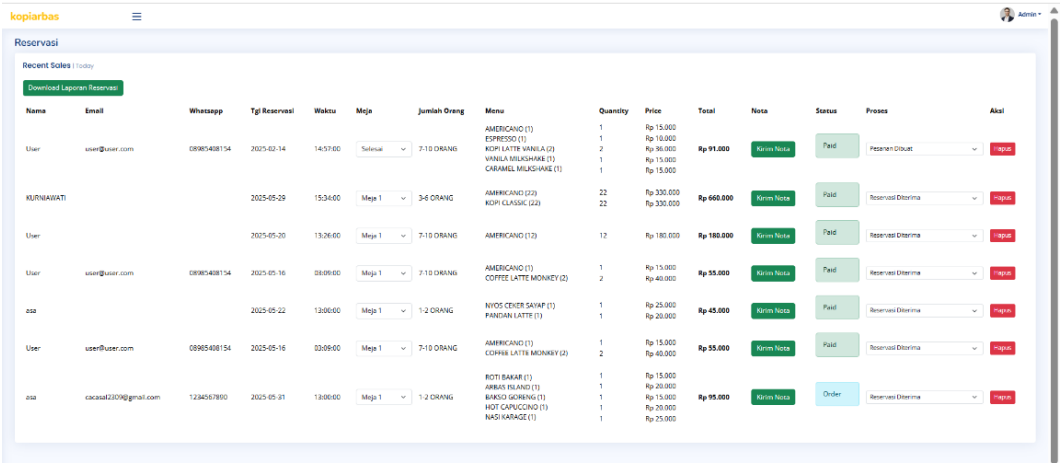
Gaya .table mengatur agar tabel memenuhi lebar kontainer, memiliki margin bawah, warna teks abu tua, dan garis batas rapi. Bagian .table thead mengatur teks bawah sel, garis bawah tebal, dan latar ungu muda. .table tbody menambahkan garis bawah tiap baris, kecuali baris terakhir. Elemen <td> dan <th> memiliki padding dan teks rata atas. .table-borderless menghapus garis antar sel.

Notifikasi .alert memiliki padding dan margin nyaman, sudut membulat, dengan warna sesuai jenis: hijau untuk .alert-success, merah untuk .alert-danger, dan biru untuk .alert-info.

Tombol menggunakan kelas .btn. Tombol hijau .btn-success untuk aksi positif seperti "Download Laporan", biru .btn-primary untuk aksi utama seperti "Kirim Nota", dan merah .btn-danger untuk hapus. Ukuran tombol kecil menggunakan .btn-sm. Gaya .mb-3 memberi jarak bawah antar elemen sebesar 1rem dan tidak bisa ditimpa karena menggunakan !important.

C. Halaman laporan reservasi

Halaman laporan reservasi berisi daftar reservasi yang telah dilakukan oleh *customer*, pada halaman ini admin dapat men*download* laporan reservasi, menghapus, dan melihat *customer* yang sudah melakukan pembayaran dan belum.



Gambar 4. 63 Halaman laporan reservasi admin

Halaman *download* laporan reservasi admin

	A	B	C	D	E	F	G	H	I	J	K
1	LAPORAN DAFTAR RESERVASI ARBAS COFFEE										
2											
3	Nama	Email	Whatsapp	Tgl Reservasi	Waktu	Jumlah Org	Menu	Quantity	Price	Total	Status
4	User	user@user.com	85298940782	19/12/2024	16:08:00	1-2 ORANG	ESPRESSO	3	10000	30000	Done
5	User	user@user.com	85298940782	19/12/2024	16:08:00	1-2 ORANG	ESPRESSO	3	10000	30000	Done
6							AMERICANO	1	15000		
7	User	user@user.com	85298940782	19/12/2024	16:08:00	1-2 ORANG	ESPRESSO	1	10000	25000	Done
8	Talia tasalsyabilla			01/01/2025	20:52:00	3-6 ORANG	AMERICANO	12	15000	180000	Order
9	User	user@user.com	85298940782	0000-00-00	00:00:00					0	Done
10	User	user@user.com	85298940782	0000-00-00	00:00:00					0	Done
11							KOPI LATTE VANILA	1	18000		
12							VANILA FRAPPUCINO	1	20000		
13	User	user@user.com	85298940782	0025-09-23	20:00:00	3-6 ORANG	PEACH TEA	1	15000	53000	Done
14	User	user@user.com	85298940782	0000-00-00	00:00:00		ESPRESSO	1	10000	10000	Order
15	User	user@user.com	85298940782	0000-00-00	00:00:00		ESPRESSO	1	10000	10000	Order
16	User	user@user.com	85298940782	0000-00-00	00:00:00		ESPRESSO	1	10000	10000	Order

Gambar 4. 64 Halaman *download* laporan reservasi admin

1. Penjelasan kode halaman laporan reservasi

Kode PHP digunakan untuk menghapus data dari tabel tbl_reservasi berdasarkan ID yang dikirim lewat form. $_SERVER['REQUEST_METHOD'] === 'POST' memastikan permintaan berasal dari metode POST. intval($_POST['hapus_id']) mengubah ID menjadi angka untuk mencegah SQL Injection. Query penghapusan dipersiapkan, parameter diikat dengan bind_param, lalu dijalankan dengan execute. Hasil penghapusan disimpan dalam variabel $pesan. <section id="daftar_pesanan"> adalah kontainer utama halaman. Bootstrap digunakan untuk membuat layout responsif dengan class row, col-12, dan card-body. Notifikasi ditampilkan berdasarkan isi $pesan, dengan warna hijau untuk sukses dan merah untuk error. Form <form action="component/export_reservasi.php" method="post"> digunakan untuk mengekspor data ke file export_reservasi.php dalam format Excel atau CSV.

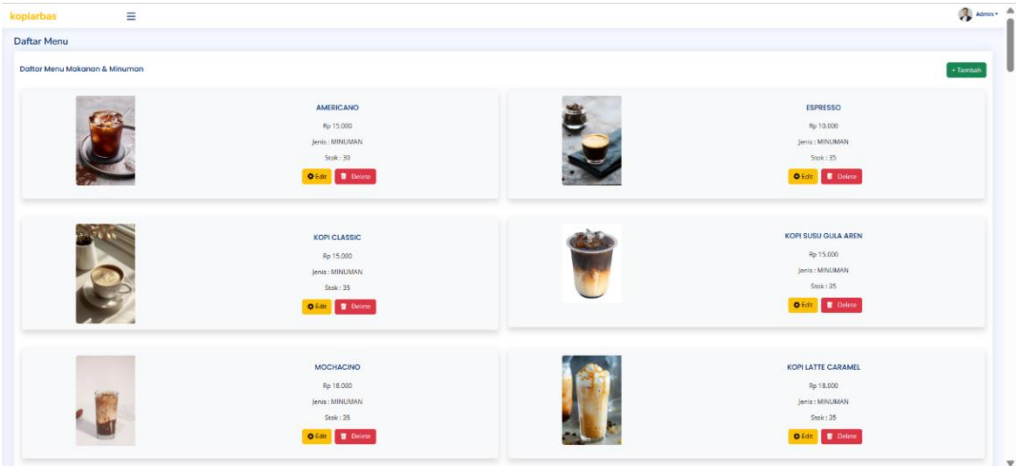
Tabel menampilkan data reservasi seperti Nama, Email, WhatsApp, Tanggal, Menu, Status, dan Aksi. Data diambil dengan $conn->query("SELECT * FROM tbl_reservasi") lalu ditampilkan dengan while. Nomor WhatsApp diformat ke format internasional (62) dan digunakan untuk membuat link kirim nota ke WhatsApp melalui $whatsappLink. Tombol "Kirim Nota" membuka chat WhatsApp dengan pesan otomatis. Tombol hapus menggunakan input tersembunyi <input type="hidden" name="hapus_id" value="<?= $row['id']; ?>"> dan konfirmasi melalui onclick="return confirm(...)".

Tabel menggunakan gaya width: 100%, teks abu gelap, dan border-collapse: collapse. Header tabel (thead) memakai vertical-align: bottom, garis bawah tebal, dan latar ungu muda. Baris (tbody) diberi garis bawah kecuali baris terakhir. Sel diberi padding agar mudah dibaca. Alert (notifikasi) memiliki padding, border transparan, dan sudut membulat. Warna hijau untuk sukses (alert-success), merah untuk kesalahan (alert-danger), dan biru untuk informasi (alert-info).

Tombol (button) diberi warna dasar biru (btn-primary), hijau untuk aksi sukses (btn-success), merah untuk hapus data (btn-danger), dan efek transisi saat hover. Ukuran kecil (btn-sm) digunakan dalam tabel. Form input bergaya penuh (width: 100%), teks abu gelap, padding nyaman, dan saat aktif tampil dengan border biru dan efek bayangan. Label diberi warna biru tua dan margin bawah agar rapi.

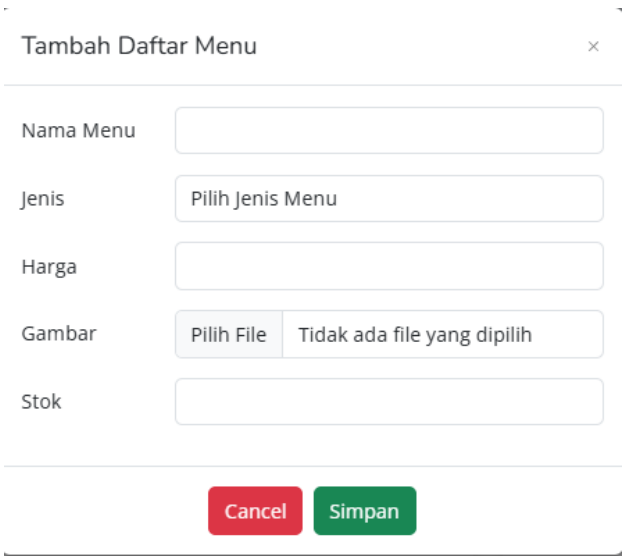
D. Halaman daftar menu admin

Halaman ini berisi daftar menu makanan dan minuman. Pada halaman ini admin dapat menambahkan menu baru, mengedit menu sebelumnya, dan menghapus menu.



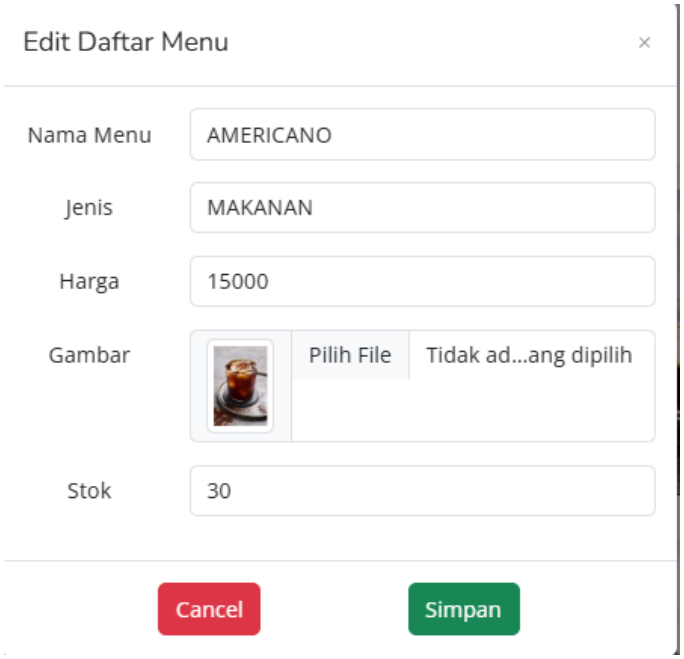
Gambar 4. 65 Halaman daftar menu admin

Halaman tambah menu admin



Gambar 4. 66 Tambah menu admin

Halaman edit menu admin



Gambar 4. 67 Edit menu admin

1. Penjelasan kode halaman daftar menu admin

Bagian daftar menu mengambil data dari tabel tbl_menu menggunakan $conn, menjalankan "SELECT * FROM tbl_menu" untuk menampilkan data dalam bentuk kartu melalui <?php while ($row = $data->fetch_assoc()) : ?>. Setiap menu ditampilkan dalam , dengan gambar menu ditampilkan menggunakan <img src="../assets/images/<?= $row['images']; ?>" class="rounded-start" width="150px">. Informasi seperti nama, harga, jenis, dan stok menu ditampilkan menggunakan elemen seperti <h5 class="card-title"><?= $row['nama_menu']; ?></h5> dan Harga: <?= rupiah($row['price']); ?> .

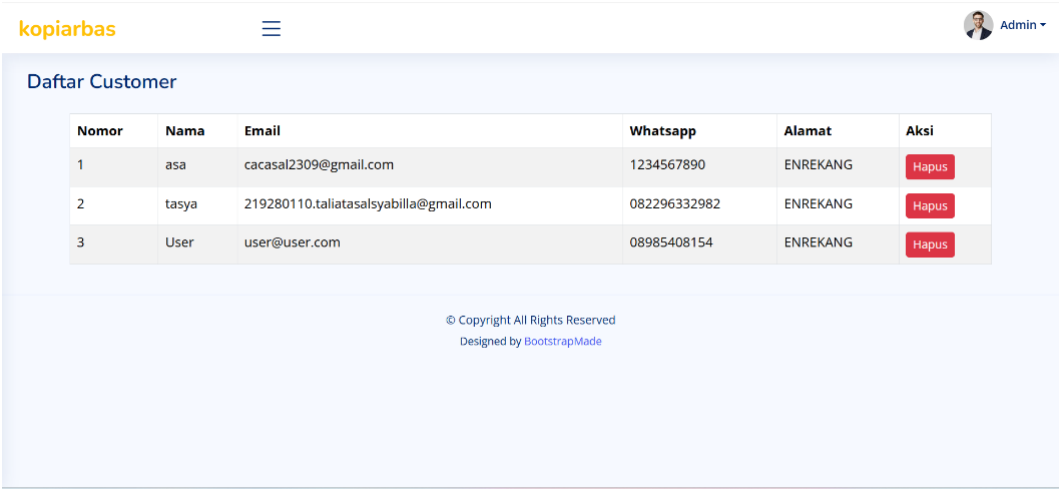
Tombol "+ Tambah" dibuat dengan <button class="btn btn-success" data-bs-toggle="modal" data-bs-target="#modalTambah"> + Tambah </button>. Modal berisi form POST ke action.php dengan input seperti <input type="text" class="form-control" name="nama_menu" required />, <select class="form-control" name="jenis_menu" required>, <input type="text" class="form-control" name="harga_menu" required />, <input type="file" class="form-control" name="gambar_menu" required />, dan <input type="number" class="form-control" name="stok" min="1" required />. Tombol simpan menggunakan <button class="btn btn-success">Simpan</button>, sedangkan batal menggunakan <button class="btn btn-danger" data-bs-dismiss="modal">Cancel</button>.

Tiap kartu menu dilengkapi tombol Edit dan Delete. Tombol Edit: <button class="btn btn-warning"></button>, dan Delete: <button class="btn btn-danger"></button>.

CSS mengatur tampilan tombol, modal, gambar, dan bayangan. Kelas .btn memberi warna dasar biru dan teks putih, dengan variasi warna sesuai fungsi seperti .btn-success, .btn-danger, .btn-warning, serta ukuran kecil .btn-sm. Layout dengan flexbox memakai .justify-content-between dan .mx-2. Modal diatur dengan .modal-content (tanpa border, radius 0.5rem, dan shadow), .modal-header dan .modal-footer memiliki garis dan padding, dan judul modal memakai .modal-title. Gambar diatur dengan .img-thumbnail, .rounded-start, dan efek bayangan .shadow. Warna latar dengan .bg-body-tertiary.

E. Halaman daftar *customer* admin

Halaman ini berisi data daftar *customer* yang telah melakukan daftar registrasi sebelumnya.



Gambar 4. 68 Halaman daftar *customer* admin

1. Penjelasan kode halaman daftar *customer* admin

<section id="daftar_pesanan"> digunakan untuk membuat bagian khusus dalam halaman web dengan identitas unik daftar_pesanan. Elemen ini biasanya digunakan untuk mengelompokkan konten logis dengan heading. membuat baris dalam sistem grid, biasanya menggunakan Bootstrap. Di dalamnya, mengatur elemen agar menempati seluruh lebar satu baris (12 kolom penuh). membentuk sebuah kartu yang digunakan untuk menampilkan daftar penjualan atau pesanan. Kelas card adalah komponen visual Bootstrap, recent-sales mungkin digunakan untuk styling khusus, dan overflow-auto memungkinkan scrollbar

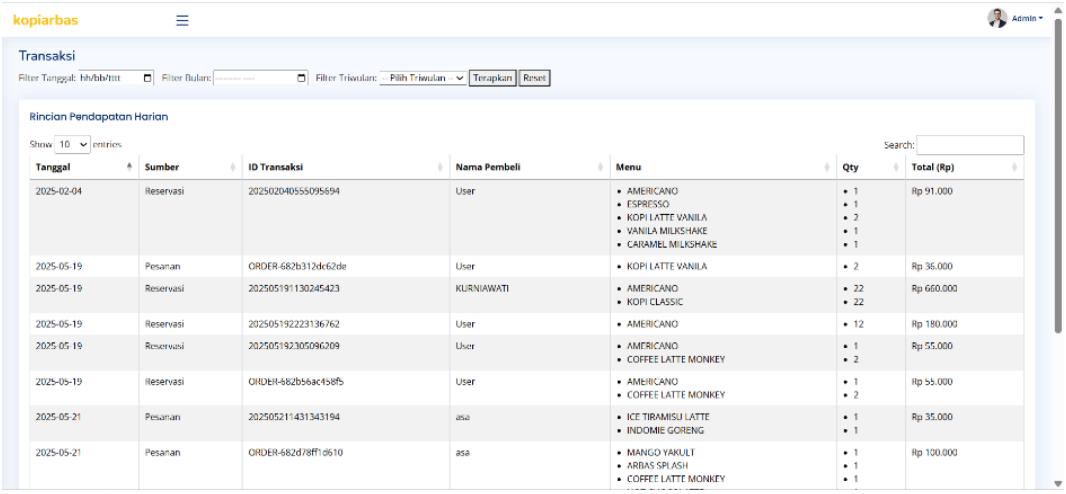
muncul jika isi melebihi tinggi kartu. adalah wadah isi utama kartu, biasanya berisi teks, tabel, atau form.

<h5 class="card-title"> digunakan untuk menampilkan judul di dalam kartu dengan gaya khusus. <form action="action.php" method="post"> adalah form yang mengirimkan data ke file action.php menggunakan metode POST. <table class="table table-borderless"> membuat tabel tanpa garis pinggir. Di dalamnya ada <thead> untuk header kolom dan <tbody class="align-middle"> untuk isi data dengan teks sejajar vertikal di tengah.

PHP $n = 1; digunakan untuk membuat penomoran urut. $query = $conn->query(...) menjalankan query untuk mengambil data dari database. while ($row = mysqli_fetch_array($query)) menampilkan isi data baris per baris. <?= $n++ ?> menampilkan nomor urut. <?= $row["nama"]; ?> menampilkan nama dari database. strtoupper($row["alamat"]); mengubah teks alamat menjadi huruf kapital.

Kode CSS .card mengatur tampilan kartu dengan margin bawah, tanpa border, sudut melengkung, dan bayangan lembut. .card-body memberi padding di sisi kanan, kiri, dan bawah. .card-title memiliki padding atas, bawah, dan ukuran font 18px dengan warna biru tua. .table mengatur lebar penuh, warna teks gelap, dan border disatukan agar rapi. .table thead th mengatur posisi teks di bawah sel dan memberi garis bawah. .table-borderless menghapus semua garis tabel.
.recent-sales mengatur font menjadi 14px. .recent-sales .table thead memberi latar belakang biru muda dan menghapus border pada header.

F. Halaman transaksi admin



Gambar 4. 69 Halaman transaksi admin

1. Penjelasan kode halaman transaksi admin

Pastikan koneksi ke database ($conn) sudah dilakukan sebelum menjalankan query. Query pesanan mengambil data dari tabel tbl_user_order dengan status = 1, mencakup bulan-tahun dalam format YYYY-MM, ID item yang dipesan, dan jumlahnya. Query reservasi mengambil data serupa dari tabel tbl_reservasi dengan status = 1. Hasil query disimpan dalam variabel $result_pesanan dan $result_reservasi.

Inisialisasi dilakukan pada array dan variabel berikut: $pendapatan_bulanan untuk menyimpan pendapatan per bulan, $total_pesanan_semua untuk total pendapatan dari pesanan, $total_reservasi_semua untuk reservasi, dan $total_pendapatan_semua sebagai total keseluruhan.

Fungsi prosesData digunakan untuk mengambil data dari hasil query ($result), memecah item dan quantity menjadi array, mengambil harga dari tabel tbl_menu, menghitung total harga tiap item, dan menyimpannya dalam $pendapatan_bulanan berdasarkan bulan dan sumber data (order atau reservasi). Fungsi ini dijalankan dua kali: prosesData($result_pesanan, ...) untuk pesanan dan prosesData($result_reservasi, ...) untuk reservasi. Fungsi krsort($pendapatan_bulanan) mengurutkan array berdasarkan bulan secara menurun (dari terbaru ke terlama).

Bagian HTML terdapat pada <section id="pendapatan_bulanan"> yang berisi tabel <table class="table table-bordered">. Struktur tabel terdiri dari <thead> dengan kolom Bulan, Pendapatan Pesanan, Pendapatan Reservasi, dan Total Pendapatan. Data ditampilkan di <tbody> termasuk total keseluruhan. Fungsi number_format(...) digunakan untuk memformat angka menjadi format mata uang Rupiah.

CSS untuk elemen .card mengatur: margin-bottom: 30px; agar tidak menempel ke elemen bawah, border: none;, border-radius: 5px; untuk sudut membulat, dan box-shadow: 0px 0 30px rgba(1, 41, 112, 0.1); sebagai efek bayangan. Elemen .card-body memiliki padding 0 20px 20px 20px. Pada judul .card-title diberikan padding 20px 0 15px 0, ukuran font 18px, ketebalan 500, dan warna #012970.

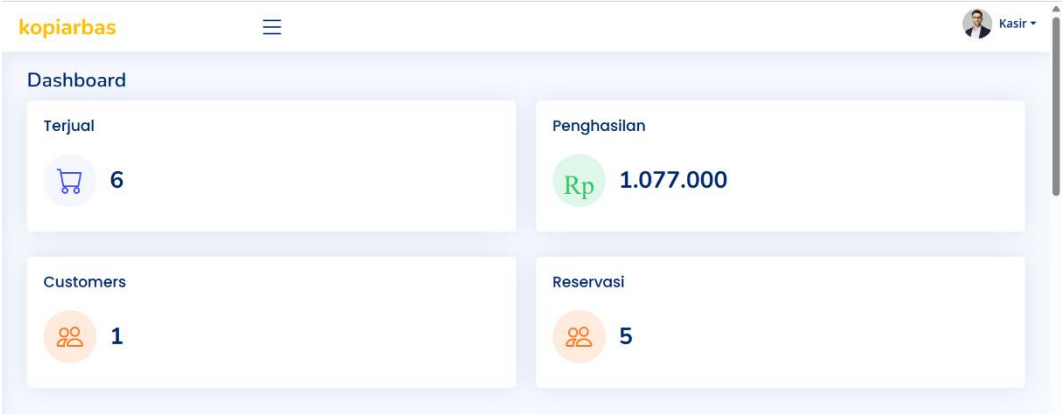
Untuk tabel, kelas .table mengatur lebar penuh width: 100%, margin bawah 1rem, warna teks #212529, dan border-collapse: collapse. Pada .table thead

th, teks rata bawah vertical-align: bottom dan garis bawah 2px solid #dee2e6. Seluruh elemen .table th dan .table td diberi padding 0.75rem, teks sejajar atas vertical-align: top, dan garis atas 1px solid #dee2e6. Untuk tabel dengan border, elemen .table-bordered th dan .table-bordered td diberi garis 1px solid #dee2e6.

2. Kasir

a. Halaman *dashboard* kasir

Halaman *dashboard* berisi info penjualan yang memuat jumlah barang terjual, total penghasilan, jumlah pesanan, dan jumlah reservasi.



Gambar 4. 70 Halaman *dashboard* kasir

1. Penjelasan tentang kode halaman *dashboard* kasir

<section *Class*="section *dashboard*">: Ini adalah bagian utama dari halaman *dashboard*. *Class* section dan *dashboard* digunakan untuk *styling* dan layout. <div *Class*="row">: Membuat baris (row) untuk mengatur tata letak grid. <div *Class*="col-lg-12">: Kolom dengan lebar penuh (12 kolom) pada layar besar (large screen). <div *Class*="row">: Baris kedua untuk mengatur kartu-kartu di dalamnya.

Pada *card* terjual memiliki col-xxl-4 col-md-6: Kolom dengan lebar 4 pada layar ekstra besar (xxl) dan 6 pada layar medium (md). *card* info-*card* sales-*card*: Kartu dengan *Class* khusus untuk styling. *card*-*body*: Bagian tubuh kartu yang berisi konten. *card*-title: Judul kartu ("Terjual"). *card*-icon: Ikon kartu (dalam hal ini, ikon keranjang belanja bi-cart). ps-3: *Padding* kiri sebesar 1rem untuk teks di sebelah ikon. h6: Menampilkan jumlah pesanan yang berhasil diambil dari *database*.

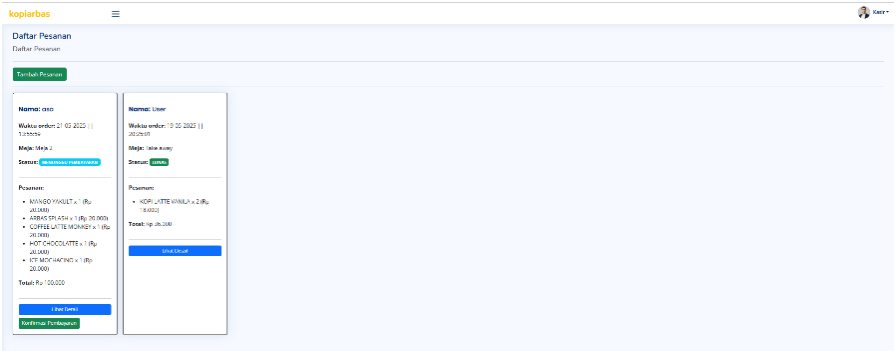
Card penghasilan menggunakan *Class* khusus revenue-*card*, rupiah-icon: *Class* untuk mengganti ikon dollar dengan teks "Rp". number_*form*at($total, 0, ',', '.'): Menampilkan total penghasilan dalam *form*at Rupiah. Untuk *card customer* juga menggunakan *Class* khusus *customer*s-*card* bi-people: Ikon untuk representasi pelanggan. Dan pada *card* reservasi mirip dengan *card customer*, tetapi menampilkan jumlah reservasi.

Pada daftar pesanan menggunakan col-12 untuk Kolom dengan lebar penuh untuk menampilkan daftar pesanan. include("daftarpesanan.php"): Memasukkan file PHP yang berisi daftar pesanan.

Kode css untuk ikon rupiah, content: "Rp";: Mengganti ikon dollar dengan teks "Rp". font-*family*: inherit;: Menggunakan *font* yang sama dengan elemen induk. CSS untuk *card* terjual menggunakan margin-bottom: 30px;: untuk memberikan jarak antar kartu. border-radius: 5px;: untuk membuat sudut kartu melengkung. box-shadow: untuk memberikan efek bayangan pada kartu. dan *card*-icon: untuk mengatur ukuran dan posisi ikon di dalam kartu. CSS layout dan grid menggunkan display: flex;: untuk mengatur tata letak menggunakan flexbox. col-xxl-4: untuk mengatur Kolom dengan lebar 33.33% pada layar ekstra besar. col-md-6: kolom dengan lebar 50% pada layar medium. col-12: Kolom dengan lebar penuh pada semua layar.

b. Halaman daftar pesanan kasir

Halaman daftar pesanan berisi daftar pesanan yang telah dilakukan oleh *customer*, pada halaman ini kasir dapat menambah pesanan dan melihat detail pesanan.



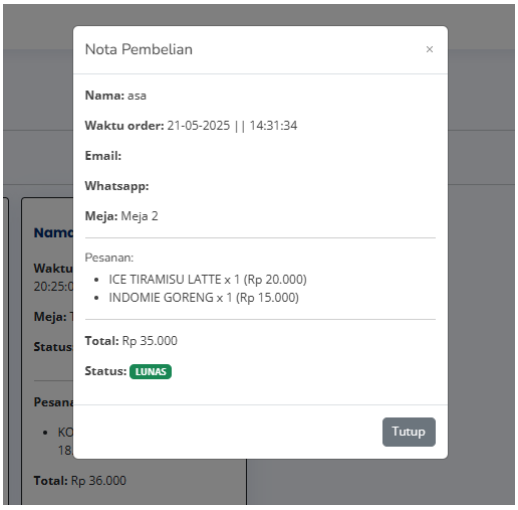
Gambar 4. 71 Halaman daftar pesanan kasir

Halaman tambah pesanan kasir



Gambar 4. 72 Halaman tambah pesanan kasir

Halaman lihat detail pesanan kasir



Gambar 4. 73 Halaman detail pesanan kasir

1. Penjelasan tentang kode halaman daftar pesanan kasir

Kode ini dimulai dengan elemen <section> dengan id="daftar_pesanan" sebagai wadah utama daftar pesanan. Di dalamnya terdapat

<h5> untuk judul dan <hr> sebagai garis pemisah. Tombol "Tambah Pesanan" menggunakan <button> dengan class="btn btn-success" dan atribut data-bs-toggle="modal" serta data-bs-target="#tambahPesananModal" untuk membuka modal saat diklik.

Data pesanan diambil dari database tbl_user_order menggunakan PHP dan ditampilkan dalam elemen dengan class card. Perulangan while digunakan untuk menampilkan tiap pesanan, dengan pengolahan $status, $item, dan $quantity. Format tanggal diatur dengan $format_tgl.

Setiap pesanan ditampilkan dalam card dengan informasi seperti nama, waktu, email, nomor WhatsApp, meja, status, dan daftar pesanan. Elemen <p> dan <ul> digunakan untuk menampilkan data, dan for digunakan untuk menampilkan item dan kuantitas. Fungsi rupiah($total) memformat total ke bentuk Rupiah. Tombol "Lihat Detail" membuka modal dengan informasi lebih lengkap.

Setiap modal memiliki id dinamis berdasarkan ID pesanan (id="notaModal<?= $row['id']; ?>") dan class="modal fade". Modal menampilkan detail yang sama dengan tampilan card. Modal tambah pesanan menampilkan form dengan input nama, email, nomor WhatsApp, meja, serta daftar menu dari tbl_menu. Checkbox digunakan untuk memilih menu dan input number untuk menentukan jumlahnya. Data dikirim ke tambah_pesanan.php via POST.

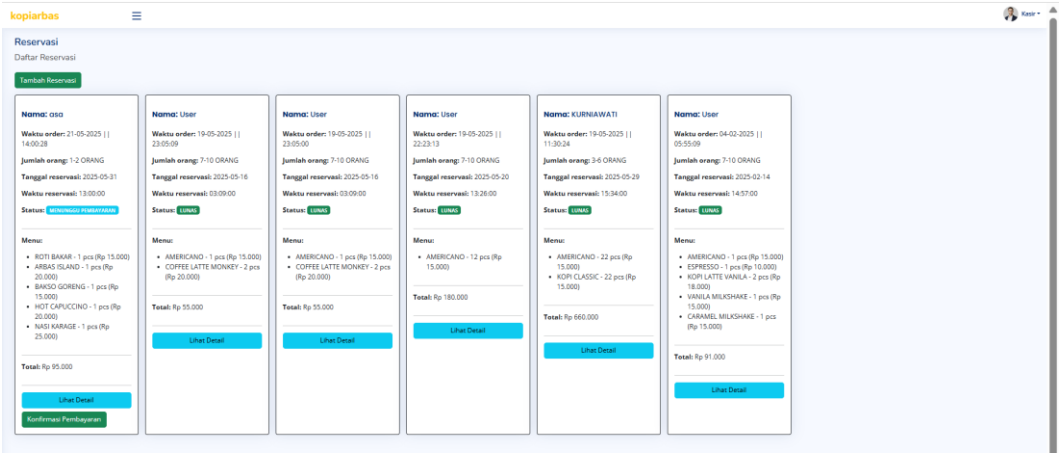
Kode body styling mengatur tampilan dasar <body> dengan margin, padding, dan background abu muda. Container diatur melalui class .container

dengan lebar maksimal 800px, margin auto, latar putih, padding 20px, bayangan lembut, dan sudut membulat. Heading <h2> ditata agar teks rata tengah dan berwarna abu gelap.

Table styling mengatur agar tabel penuh lebar, border rapi, sel dengan padding, header biru dan teks putih, baris genap dengan warna berbeda, dan efek hover. Tombol dengan class .button ditata dengan padding, latar hijau, teks putih, tanpa border, sudut melengkung, dan efek hover lebih gelap.

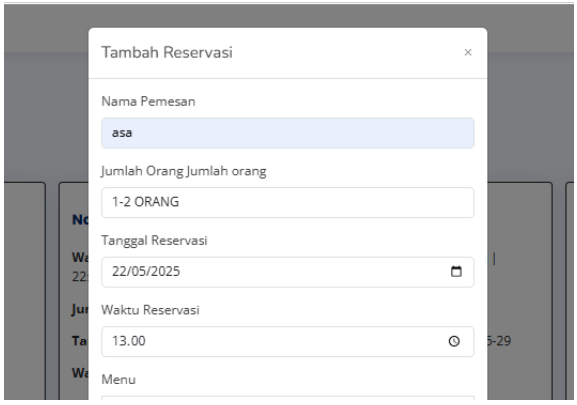
c. Halaman daftar reservasi kasir

Halaman daftar reservasi berisi daftar reservasi yang telah dilakukan oleh *customer*, pada halaman ini kasir dapat menambah reservasi, melihat detail reservasi.



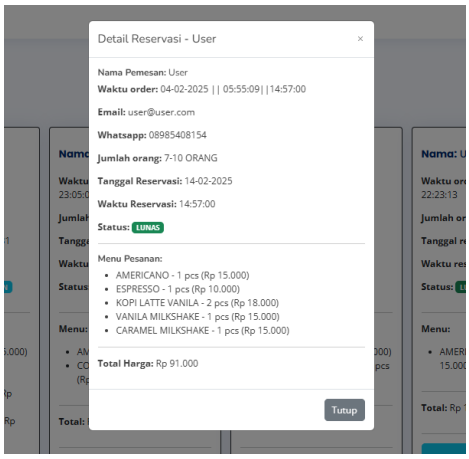
Gambar 4. 74 Halaman daftar reservasi kasir

Halaman tambah reservasi kasir



Gambar 4. 75 Halaman tambah reservasi kasir

Halaman lihat detail reservasi kasir



Gambar 4. 76 Halaman lihat detail reservasi kasir

1. Penjelasan tentang kode halaman daftar reservasi kasir

<section id="daftar_reservasi"> adalah wadah utama untuk daftar reservasi. dan mengatur layout menggunakan grid Bootstrap. <h5 class="mb-4"> menampilkan judul dengan margin bawah. Tombol tambah reservasi memakai class="btn btn-success mb-3",

dengan data-bs-toggle dan data-bs-target untuk membuka modal id="tambahReservasiModal". Daftar reservasi menggunakan class="d-flex flex-wrap gap-3" agar tampil fleksibel. Setiap kartu reservasi memakai class="card border border-dark p-3" dengan style="width: 300px;". digunakan untuk daftar menu dan <hr> sebagai pemisah.

Status reservasi menggunakan class="badge bg-info". Tombol "Lihat Detail" memakai class="btn btn-info" dengan data-bs-target="#modalDetail_1". Modal detail memiliki class="modal fade", id="modalDetail_1", dan tombol btn-close untuk menutup. Modal tambah reservasi id="tambahReservasiModal" mengirim data ke tambah_reservasi.php. Form berisi input nama, email, jumlah orang (select), tanggal (type="date"), waktu (type="time"), checkbox menu, dan input jumlah (type="number").

CSS section daftar reservasi memakai padding: 20px;, font-weight: bold;, dan margin-bottom: 20px;. Tombol tambah reservasi menggunakan background-color: #28a745;, border-color: #28a745;, dan efek hover warna #218838. Kartu reservasi diatur dengan display: flex;, flex-wrap: wrap;, gap: 15px;. Teks dalam kartu memakai font-weight: bold;, margin: 5px 0;, dan padding-left: 20px;. Kartu berukuran 300px, border: 1px solid #000;, border-radius: 8px;, padding: 15px;, background-color: #fff;, dan box-shadow ringan.

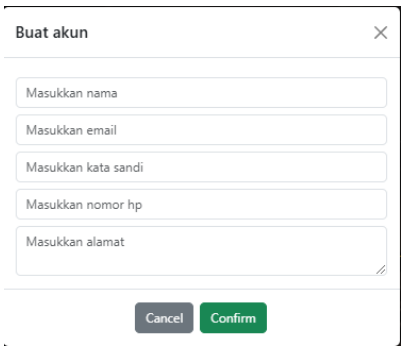
Badge status memakai warna bg-info (Order), bg-success (Diterima), bg-danger (Dibatalkan). Tombol "Lihat Detail" menggunakan background-color: #007bff;, border-color: #007bff;, color: white;, dan hover #0056b3. Modal

memakai border-radius: 10px;, background-color: #f8f9fa;, dengan border-bottom dan border-top sebagai pemisah.

3. Customer

a. Halaman daftar *customer*

Ketika *customer* ingin melakukan *login customer* harus daftar terlebih dahulu. Data yang harus disiapkan yaitu nama, email, sandi, nomor hp, dan alamat setelah itu *customer* menekan tombol confim. Setelah mendaftar *customer* dapat *login* menggunakan email dan kata sandi yang telah di daftarkan tadi.



Gambar 4. 77 Halaman daftar *customer*

1. Penjelasan tentang kode halaman daftar *customer*

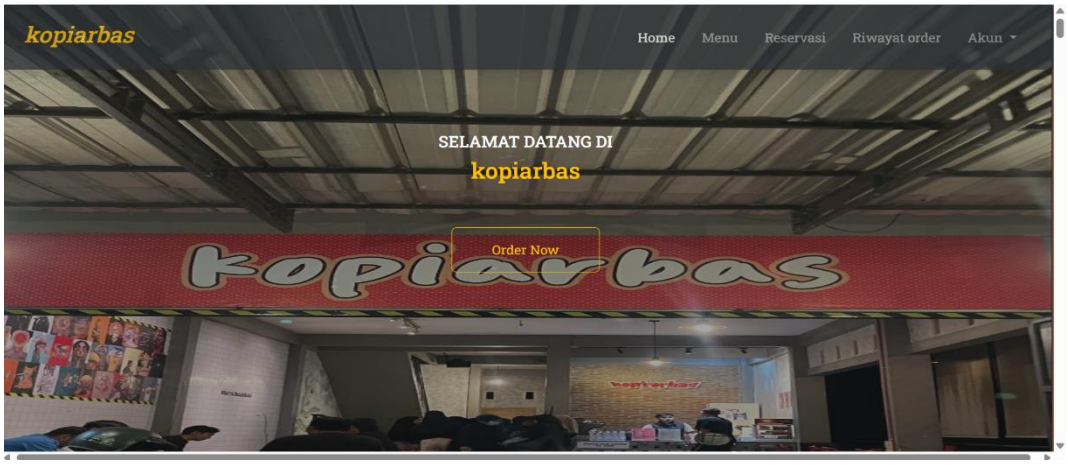
Kode HTML formulir pendaftaran ditampilkan sebagai modal, yaitu jendela pop-up di atas halaman. Modal ini mengumpulkan data pengguna seperti nama, email, kata sandi, nomor handphone, dan alamat menggunakan metode POST, namun atribut action pada tag <form> masih kosong. Modal tidak dapat ditutup dengan klik luar atau tombol Esc karena data-bs-backdrop="static" dan

data-bs-keyboard="false". Warna teks menggunakan class text-dark agar kontras dengan latar terang.

Di dalam modal terdapat input field required: input teks untuk nama, email, password, nomor handphone, dan textarea untuk alamat. Semua menggunakan class form-control dan my-2 untuk tampilan konsisten dan jarak antar elemen. Modal dilengkapi judul "Buat akun" dan tombol tutup di sudut kanan atas. Bagian footer modal berisi tombol "Cancel" dengan class btn btn-secondary dan "Confirm" dengan class btn btn-success.

CSS mengatur tampilan dasar halaman. Selector universal (*) menghapus padding dan margin default. Selector body memakai max-height: 100vh agar tidak melebihi tinggi layar. Selector img mengatur gambar agar full layar dengan width: 100% dan height: 100vh. Opacity: 0.5 membuat gambar setengah transparan, dan saat dihover (img:hover), opacity berubah menjadi 1 agar tampak jelas, menambah efek visual interaktif.

b. Halaman *home customer*



Gambar 4. 78 Halaman *home customer*

1. Penjelasan tentang kode halaman *home customer*

Navigasi berada di atas halaman dan tetap terlihat saat digulir (fixed-top), menggunakan tema gelap (data-bs-theme="dark") dengan latar belakang transparan (opacity-75), warna dasar Bootstrap (bg-body-tertiary), dan teks "Arbas" berwarna kuning (text-warning). Navbar responsif dengan tombol toggle pada layar kecil. Hero section menampilkan gambar latar belakang beropasitas 60% dan teks selamat datang; teks "Arbas Cafe" berwarna kuning dan tombol "Order Now" memakai border kuning (btn-outline-warning). Bagian menu, reservasi, dan riwayat order memuat konten dari file eksternal (menu.php, reservasi.php, riwayat.php) menggunakan PHP untuk modularitas.

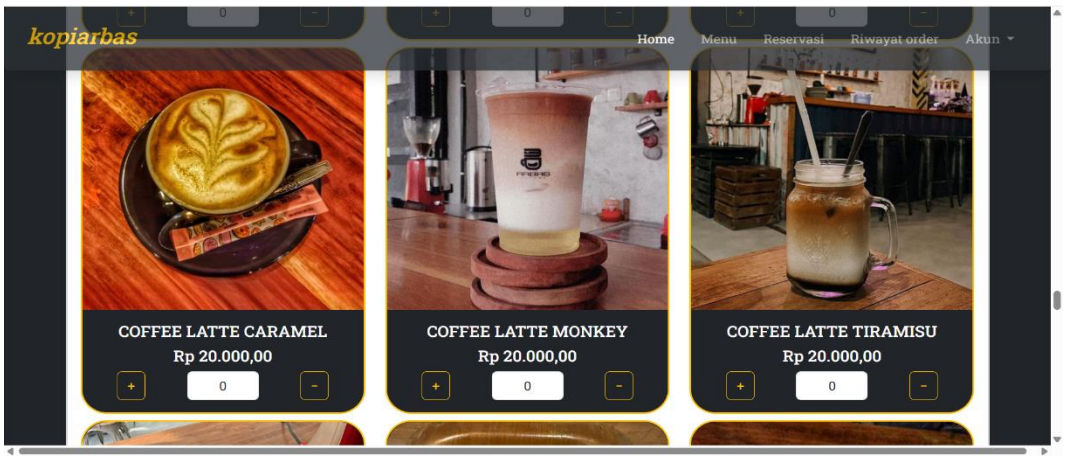
Footer berlatar belakang gelap (bg-dark), teks terang (text-light), dan warna kuning (text-warning) untuk judul seperti "Opening Hours" dan "Follow Us", serta menampilkan jam operasional dan tautan media sosial. Kode <link

href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat font Roboto Slab yang diterapkan ke seluruh elemen menggunakan selector *,

serta menghilangkan margin dan padding default. Kode <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap 5.3.3, dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"> memuat ikon Bootstrap Icons. Aturan CSS <body> mengatur latar belakang coklat tua (#6b4031), dan <#footer> mengatur tinggi minimal footer 40% dari tinggi layar.

c. Halaman daftar menu *customer*

Halaman ini berisi daftar menu makanan dan minuman, *customer* dapat menambah menu ke daftar order.



Gambar 4. 79 Halaman daftar menu *customer*

1. Penjelasan tentang kode halaman daftar menu *customer*

Halaman ini menampilkan daftar menu makanan dan minuman yang dapat dipesan. Data diambil dari database menggunakan PHP dan dibagi menjadi dua kategori: $minuman (jenis "Minuman") dan $makanan (jenis "Makanan"). Informasi pengguna login disimpan dalam $users dari session. Tampilan menggunakan Bootstrap agar responsif dan menarik. Bagian Card Header berisi judul "DAFTAR MENU" dengan margin atas untuk menghindari tumpang tindih dengan navbar. Jika URL memiliki parameter msg, sistem menampilkan alert status pemesanan. Form pemesanan mengirim data ke orderDetail.php dan memiliki dua tab: Minuman dan Makanan. Setiap tab menampilkan daftar menu dalam bentuk card berisi gambar, nama, harga, dan tombol + atau - untuk mengatur jumlah pesanan, yang disimpan dalam input value[]. Setelah memilih, pengguna klik "Selesaikan Pesanan" untuk membuka modal berisi form data diri (nama, email, WhatsApp, meja). Warna kuning digunakan pada text-warning, btn-outline-warning, btn-warning; gelap untuk latar card (text-bg-dark); putih/terang untuk teks (text-light).Kode<link

href="https://fonts.googleapis.com/css2?

family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat font Roboto Slab dan diterapkan ke semua elemen dengan selector *, serta menghapus margin dan padding bawaan. Kode <link

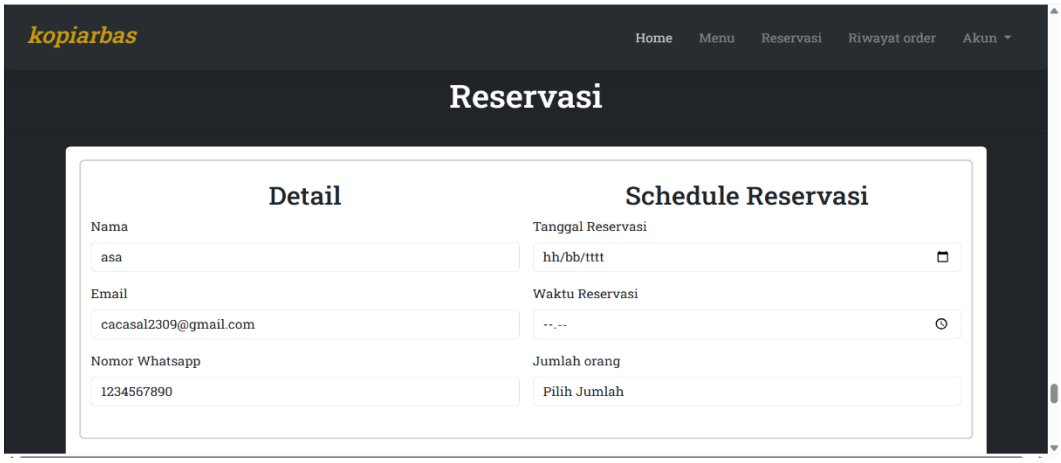
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap 5.3.3, dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-

icons@1.11.3/font/bootstrap-icons.min.css"> memuat ikon Bootstrap. Aturan <body> memberi latar coklat tua (#6b4031), sedangkan <#footer> mengatur tinggi minimal 40% dari tinggi layar.

d. Halaman reservasi *customer*

Halaman reservasi berisi data *customer* dan schedule reservasi, *customer* mengisi *form* yang kosong apabila ingin reservasi tempat.



Gambar 4. 80 Halaman reservasi *customer*

1. Penjelasan tentang kode halaman reservasi *customer*

Struktur utama halaman ini menggunakan *Bootstrap* untuk membuat tampilan yang responsif dan menarik. *Card* Header bagian ini menampilkan judul "Reservasi" dengan *margin* atas yang disesuaikan untuk menghindari tumpang tindih dengan *navbar*. *Card Body* bagian ini berisi *form* untuk mengisi detail reservasi. *Form* ini dibagi menjadi dua kolom: Detail dan Schedule Reservasi. Kolom Detail kolom ini digunakan untuk mengisi in*form*asi pribadi pengguna, seperti nama, email, dan nomor WhatsApp. Data ini diambil dari session pengguna

yang sedang *login*. Kolom Schedule Reservasi kolom ini digunakan untuk mengisi jadwal reservasi, termasuk tanggal, waktu, dan jumlah orang. Setelah mengisi detail reservasi, pengguna dapat mengklik tombol "Pesan Menu" untuk membuka modal yang berisi daftar menu. Modal ini di-include dari file modal_daftar_menu.php. Kuning: btn-*warning* (untuk tombol "Pesan Menu"). Gelap: *text*-bg-*dark* (untuk latar belakang *card* utama). Putih/terang: *text*-light (untuk teks di atas latar belakang gelap).

Kode CSS dan <link> yang digunakan dalam HTML ini berfungsi untuk mengatur tampilan halaman web dengan beberapa fitur utama. Pertama kode <linkhref="https://fonts.googleapis.com/css2?*family*=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat *font* Roboto Slab dari Google Fonts, sehingga *font* ini bisa digunakan dalam halaman web. *Font* ini kemudian diterapkan ke seluruh elemen dengan aturan CSS Selector * (Universal Selector) Aturan ini menetapkan bahwa semua elemen (* adalah selector universal) akan menggunakan *font* Roboto Slab. Selain itu, properti margin: 0; dan padding: 0; menghilangkan jarak bawaan pada semua elemen, yang biasanya ada di beberapa elemen HTML seperti <*body*> atau <h1>.

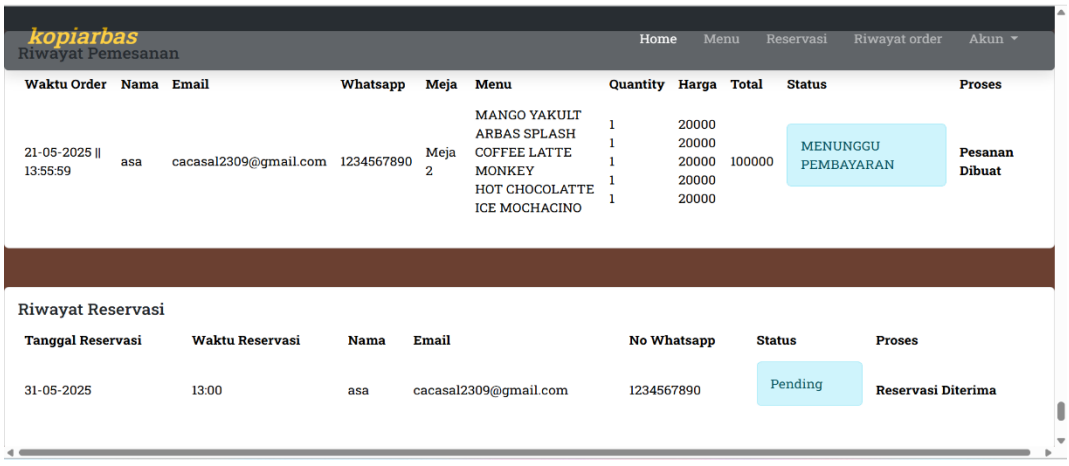
Kedua kode <link href= "https: //cdn.jsdelivr.net/ npm/ *Bootstrap*@5.3.3/dist/css/*Bootstrap*.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous"> memuat file CSS dari *Bootstrap* versi 5.3.3, yang memberikan berbagai gaya bawaan dan komponen seperti tombol, grid sistem, dan elemen responsif. Ketiga kode <link rel="stylesheet" href= "https:

//cdn.jsdelivr.net/ npm/ *Bootstrap*-icons@1.11.3/font/*Bootstrap*-icons.min.css"> memuat ikon dari *Bootstrap* Icons, yang memungkinkan penggunaan ikon dengan kelas seperti <i *Class*="bi bi-alarm"></i>.

Setelah itu ada aturan CSS <*body*> Aturan ini mengatur warna latar belakang halaman web menjadi coklat tua (#6b4031). Warna ini akan diterapkan pada seluruh area *body* dari halaman web. Lalu aturan <#footer> digunakan untuk mengatur elemen dengan id="footer" agar memiliki tinggi minimal 40% dari tinggi layar (*viewport height*). Ini memastikan bahwa footer tidak terlalu kecil meskipun kontennya sedikit.

e. Halaman riwayat order *customer*

Halaman riwayat order berisi riwayat pembelian *customer* sebelumnya.



Gambar 4. 81 Halaman riwayat order *customer*

1. Penjelasan tentang kode halaman riwayat order *customer*

Kode PHP digunakan untuk menampilkan pesan notifikasi setelah proses edit atau hapus reservasi, ditampilkan dalam alert Bootstrap. Section Riwayat Pemesanan menampilkan tabel dari data tbl_user_order yang difilter berdasarkan email pengguna login, berisi waktu order, nama, email, WhatsApp, meja, menu, quantity, harga, total, dan status. Section Riwayat Reservasi menampilkan data dari tbl_reservasi berdasarkan email pengguna login, berisi tanggal, waktu reservasi, nama, email, WhatsApp, dan status. Tombol edit membuka form untuk mengubah data reservasi, sedangkan tombol hapus digunakan untuk menghapus data. Kode <link

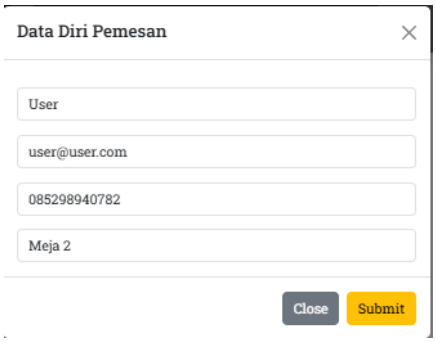
href="https://fonts.googleapis.com/css2?

family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat font Roboto Slab dan diterapkan ke semua elemen menggunakan selector *, serta menghapus margin dan padding default. Kode <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap 5.3.3, dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"> memuat ikon Bootstrap Icons. Aturan <body> mengatur latar belakang coklat tua (#6b4031), sedangkan <#footer> menetapkan tinggi minimal 40% dari tinggi layar.

f. Halaman order *customer*

Halaman ini berisi data *customer* yang telah memilih menu dan *customer* bisa memilih nomor meja yang di inginkan lalu menkan tombol *submit*.



Gambar 4. 82 Halaman order *customer*

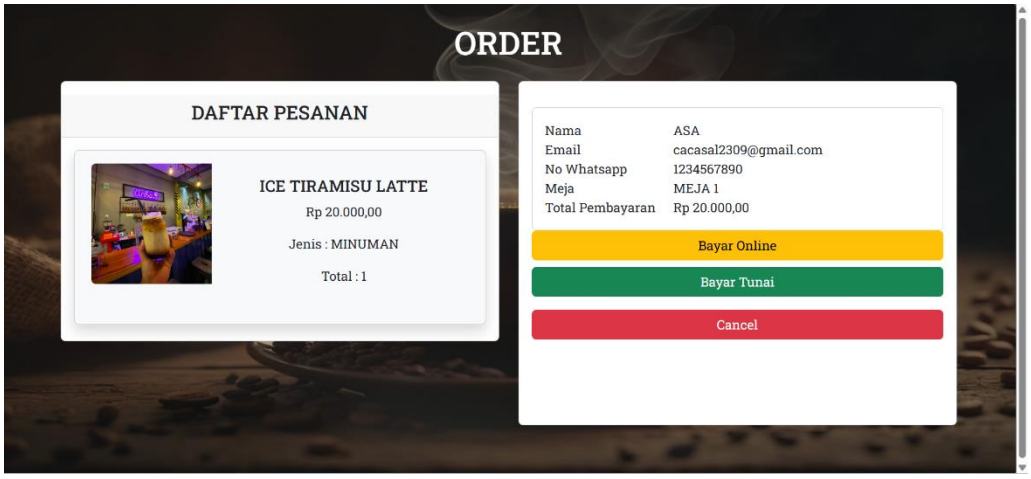
1. Penjelasan tentang kode halaman order *customer*

Kode ini adalah modal Bootstrap dengan class="modal fade" dan id="Order", muncul dengan animasi fade-in. data-bs-backdrop="static" mencegah modal tertutup saat klik luar, data-bs-keyboard="false" mencegah tertutup dengan tombol ESC, dan aria-hidden="true" menyembunyikan modal secara default. Class="modal-header" menampilkan judul "Data Diri Pemesan" dan tombol btn-close untuk menutup modal. Form input menampilkan data default dari variabel $users; jika nomor handphone kosong, akan terisi "-". Dropdown meja menyediakan pilihan meja dan opsi "Take away". Tombol "Close" menutup modal tanpa menyimpan, sedangkan tombol "Submit" mengirim data ke server. Kode <link

href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat font Roboto Slab dan diterapkan ke seluruh elemen dengan selector *, serta menghapus margin dan padding bawaan. Kode <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap 5.3.3, dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"> memuat ikon Bootstrap. Aturan <body> memberi latar belakang coklat tua (#6b4031), dan <#footer> mengatur tinggi minimal 40% dari tinggi layar.

g. Halaman konfirmasi pesanan *customer*



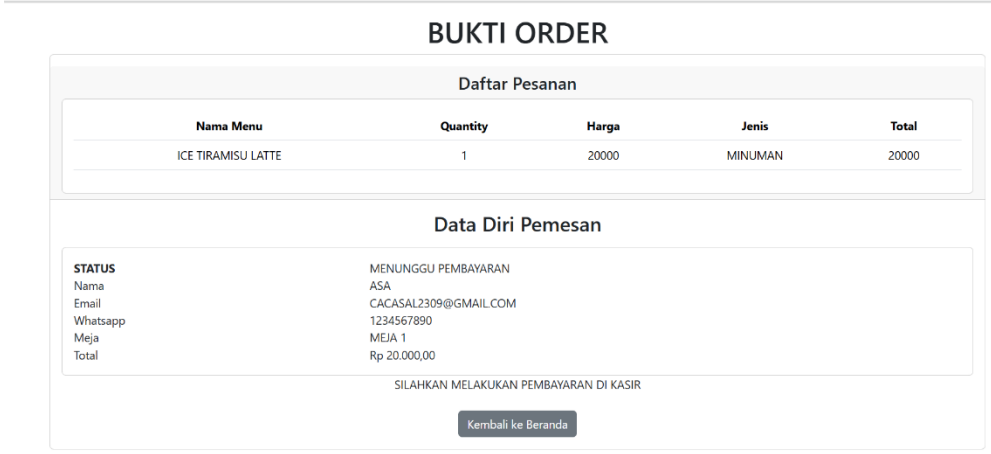
Gambar 4. 83 Halaman konfirmasi pesanan *customer*

1. Penjelasan tentang kode halaman konfirmasi pesanan *customer*

Kode pada halaman konfirmasi pesanan menggunakan Bootstrap untuk tampilan responsif. Daftar pesanan ditampilkan dalam card list, dengan looping PHP pada array $arr untuk menampilkan menu berdasarkan ID dan jumlah pesanan. Data diambil lewat query SQL dan dimasukkan ke HTML, harga diformat dengan fungsi rupiah(), dan total pembayaran dijumlahkan. Class="card-body datadiri text-center" menampilkan info pemesan seperti Nama, Email, WhatsApp, Meja, dan Total Pembayaran dari variabel PHP. Hidden input digunakan untuk mengirim data ke server saat tombol "Submit" ditekan. Fungsi JavaScript validateOrder() digunakan untuk validasi sebelum form dikirim, sedangkan tombol "Cancel" mengarah ke halaman utama. Kode <link href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> memuat font Roboto Slab dan diterapkan ke semua elemen dengan selector *, serta menghapus margin dan padding bawaan. Kode <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap 5.3.3, dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"> memuat ikon Bootstrap. Aturan <body> mengatur latar belakang coklat tua (#6b4031), dan <#footer> memberi tinggi minimal 40% dari tinggi layar.

h. Halaman nota pembelian *customer*

Halaman ini berisi nota pembelian dari *customer*.



Gambar 4. 84 Halaman nota pembelian *customer*

1. Penjelasan tentang kode halaman nota pembelian *customer*

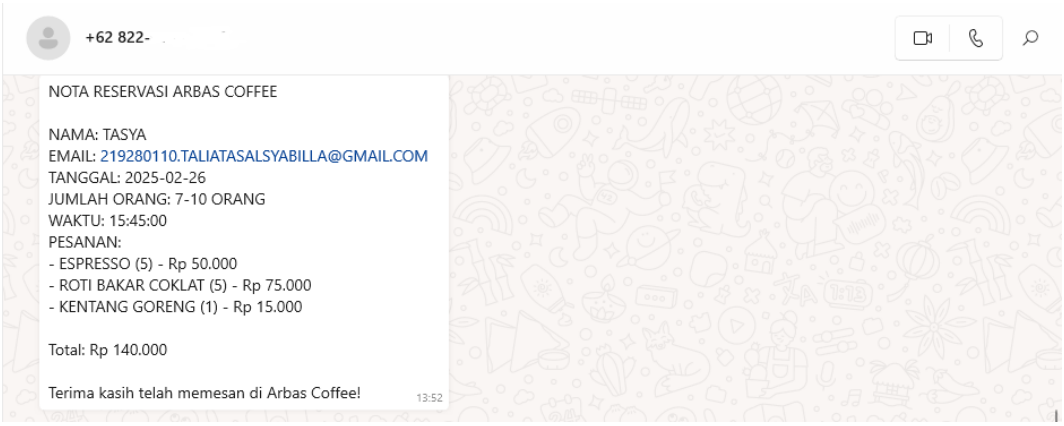
Elemen <h1> menampilkan judul halaman di bagian atas dengan teks berwarna gelap dan rata tengah. Tabel digunakan untuk menampilkan daftar pesanan seperti nama menu, jumlah, harga satuan, jenis menu, dan total per item. Bagian PHP dalam <tbody> menampilkan data dari database tbl_menu menggunakan looping for pada array $id_menu dan $quantity, serta query SQL berdasarkan id_menu. Total harga dijumlahkan dan ditampilkan. Bagian data diri menampilkan nama, email, WhatsApp, nomor meja, dan total harga diformat dengan fungsi rupiah($total). Elemen <p> memberi pesan agar pelanggan mengambil screenshot sebagai bukti. Tombol "Kembali ke Beranda" mengarah ke index.php.

Dua elemen <link rel="preconnect"> mempercepat koneksi ke fonts.googleapis.com dan fonts.gstatic.com. Kode <link

href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@100..900&display=swap" rel="stylesheet"> mengimpor font Roboto Slab. <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="..." crossorigin="anonymous"> memuat CSS Bootstrap dan <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css"> memuat Bootstrap Icons.
Dua tag <script> memuat Alpine.js untuk interaktivitas dengan atribut defer agar dijalankan setelah konten dimuat, dan Snap.js dari Midtrans untuk pembayaran online menggunakan data-client-key sebagai autentikasi.

Halaman nota pembelian WhatsApp *customer*



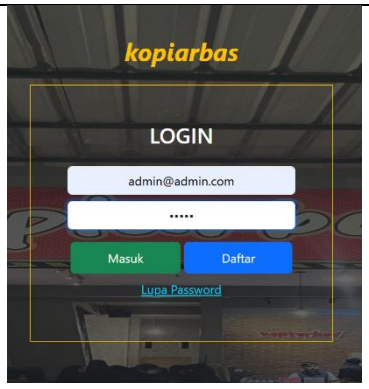
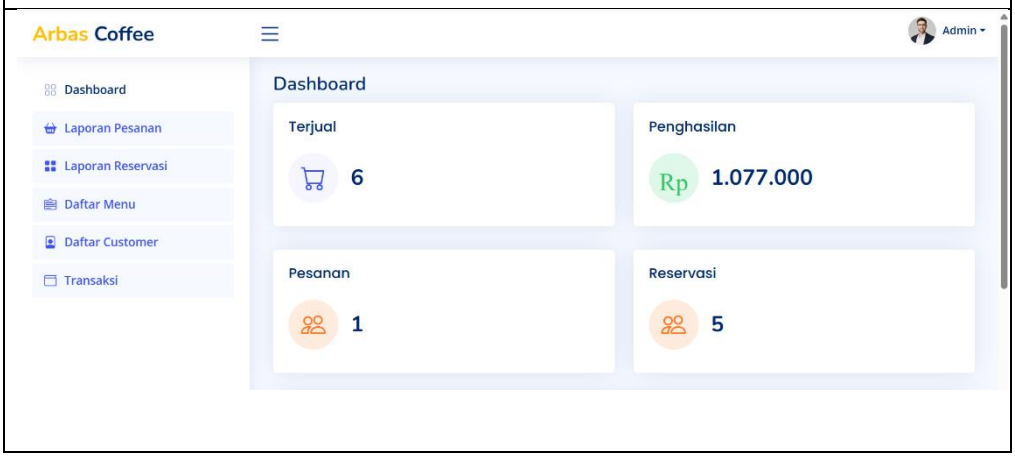
Gambar 4. 85 nota pembelian WhatsApp *customer*

C. Pengujian Sistem

1. Black box Testing

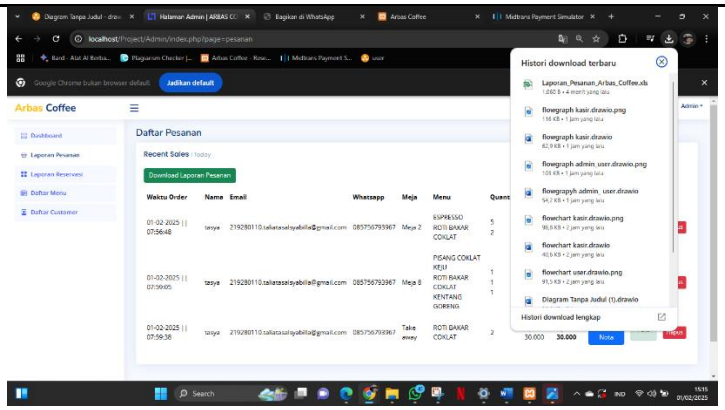
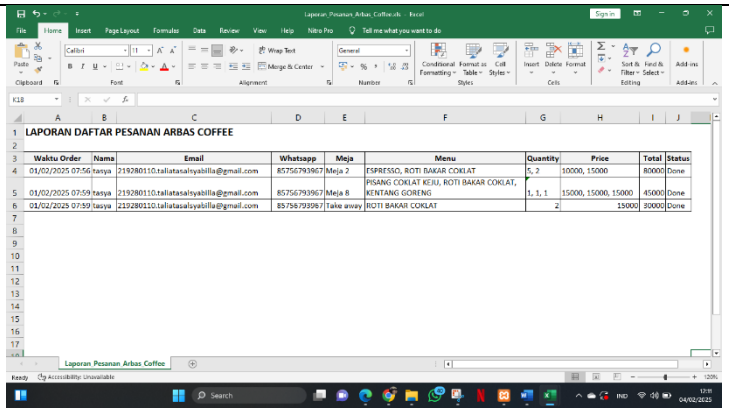
a. Pengujian *form login* admin

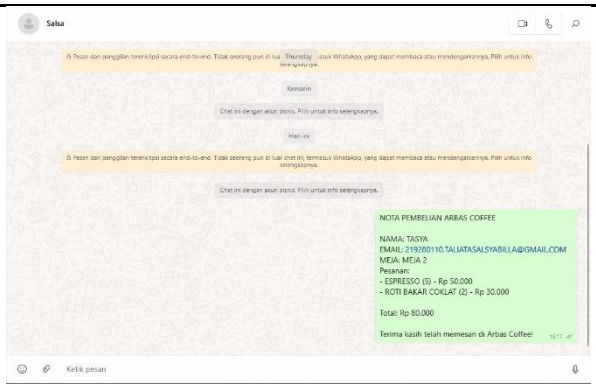
Tabel 4. 1 Pengujian *form login* admin

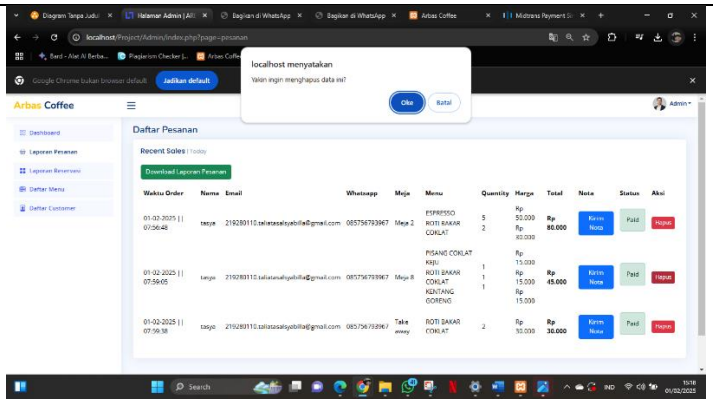
Tes Faktor	Hasil	Keterangan
Admin memasukkan *username* dan *password* yang terdaftar pada sistem	✔	Jika *username* dan *password* tidak sesuai maka pengguna tetap berada pada halaman *login*. Jika *username* dan *password* sesuai maka pengguna diarahkan ke halaman *dashboard*.
Screen Shoot		
 		

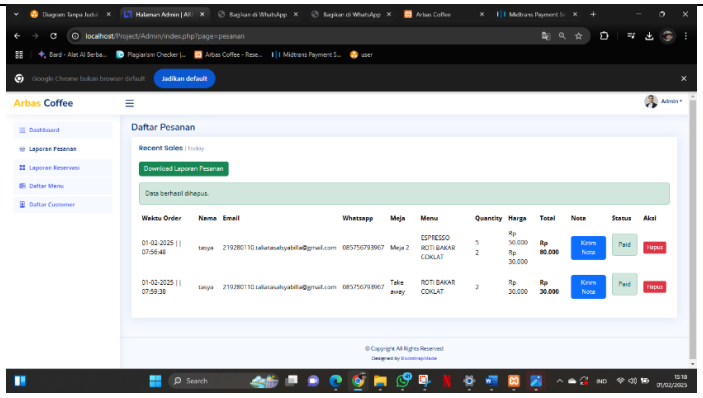
b. Pengujian halaman laporan pesanan admin

Tabel 4. 2 Pengujian halaman laporan pesanan admin

Tes Faktor	Hasil	Keterangan
Kelolah data halaman laporan pesanan meliputi, melihat, men*download*, menghapus dan mengirim nota ke WhatsApp *customer*.	✔	1. Berhasil menampilkan halaman saat menu ditekan. 2. Berhasil men*download* data. 3. Berhasil mengirim nota ke *customer*. 4. Berhasil menghapus data yang dipilih.
Screen Shoot		
		
		

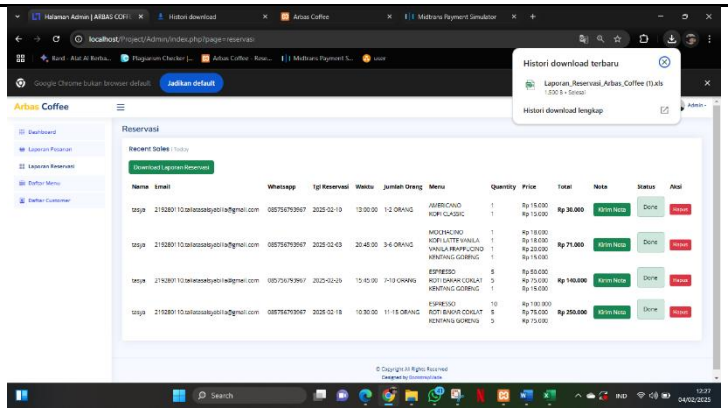
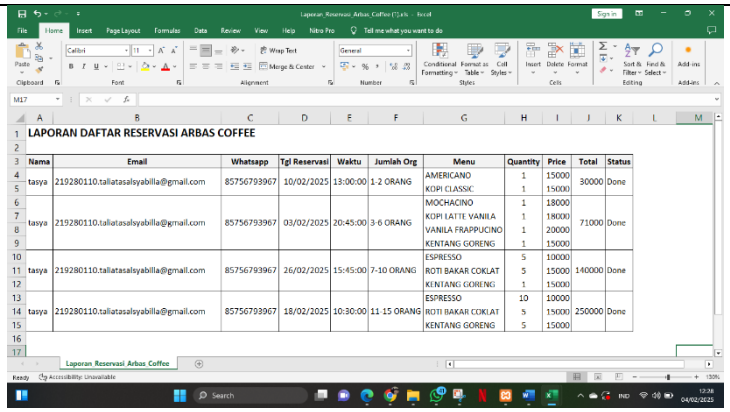

Salsa
Kemarin
Hari ini
NOTA PEMBELIAN ARBAS COFFEE
NAMA: TASYA
EMAIL: 219280110.TALIATASALSYABILLA@GMAIL.COM
MEJA: MEJA 2
Pesanan:
- ESPRESSO (5) - Rp 50.000
- ROTI BAKAR COKLAT (2) - Rp 30.000
Total: Rp 80.000
Terima kasih telah memesan di Arbas Coffee!
Ketik pesan


localhost menyatakan
Yakin ingin menghapus data ini?
Oke
Batal
Arbas Coffee
Daftar Pesanan
Recent Sales
Download Laporan Pesanan


Arbas Coffee
Daftar Pesanan
Recent Sales
Download Laporan Pesanan
Data berhasil dihapus.

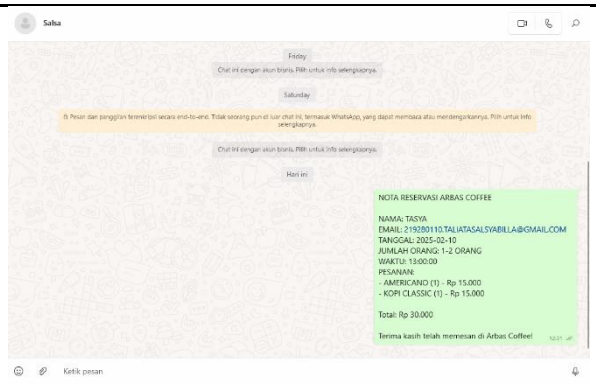
c. Pengujian halaman laporan reservasi admin

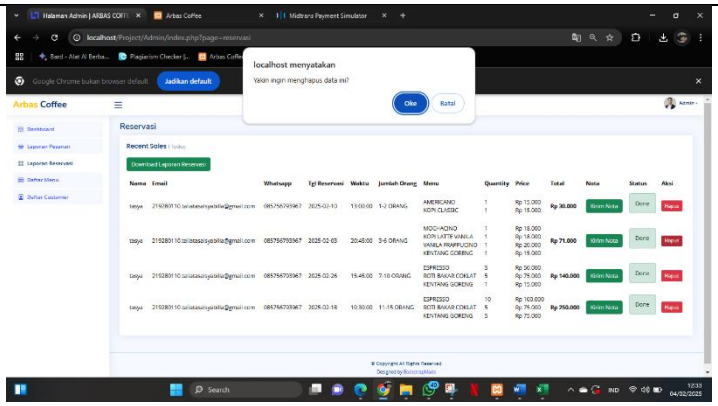
Tabel 4. 3 Pengujian halaman laporan reservasi admin

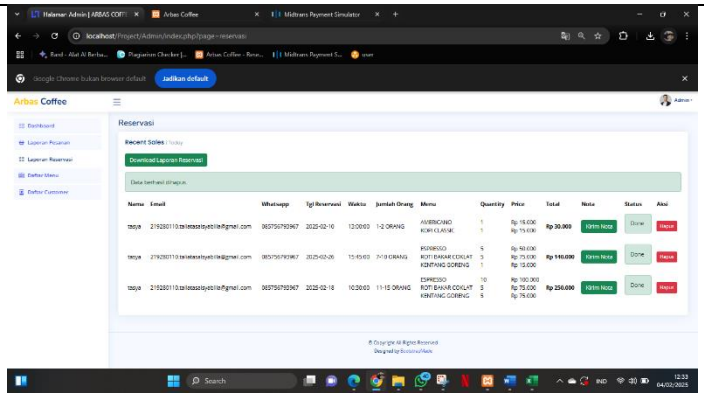
Tes Faktor	Hasil	Keterangan
Kelolah data halaman laporan reservasi meliputi melihat, men*download*, menghapus dan mengirim nota ke WhatsApp *customer*.	✔	1. Berhasil menampilkan halaman saat menu ditekan. 2. Berhasil men*download* data. 3. Berhasil mengirim nota ke *customer*. . 4. Berhasil menghapus data yang dipilih.
Screen Shoot		
		
		

LAPORAN DAFTAR RESERVASI ARBAS COFFEE

Nama	Email	Whatsapp	Tgl Reservasi	Waktu	Jumlah Org	Menu	Quantity	Price	Total	Status
tasya	219280110.tallatasalsyabilla@gmail.com	85756793967	10/02/2025	13:00:00	1-2 ORANG	AMERICANO	1	15000	30000	Done
						KOPI CLASSIC	1	15000		
tasya	219280110.tallatasalsyabilla@gmail.com	85756793967	03/02/2025	20:45:00	3-6 ORANG	MOCHACINO	1	18000	71000	Done
						KOPI LATTE VANILA	1	18000		
						VANILA FRAPPUCINO	1	20000		
						KENTANG GORENG	1	15000		
tasya	219280110.tallatasalsyabilla@gmail.com	85756793967	26/02/2025	15:45:00	7-10 ORANG	ESPRESSO	5	10000	140000	Done
						ROTI BAKAR COKLAT	5	15000		
						KENTANG GORENG	1	15000		
tasya	219280110.tallatasalsyabilla@gmail.com	85756793967	18/02/2025	10:30:00	11-15 ORANG	ESPRESSO	10	10000	250000	Done
						ROTI BAKAR COKLAT	5	15000		
						KENTANG GORENG	5	15000		

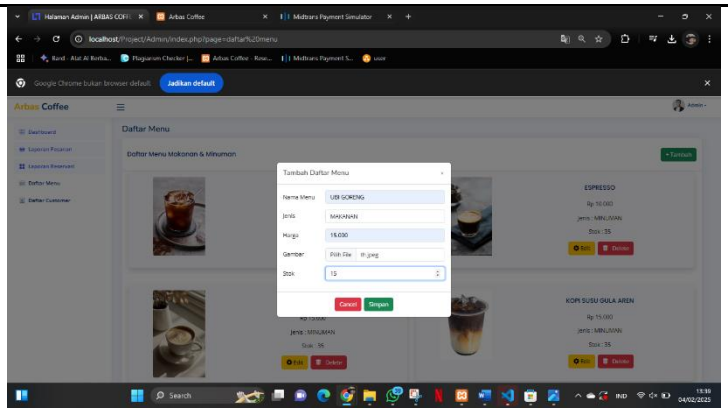
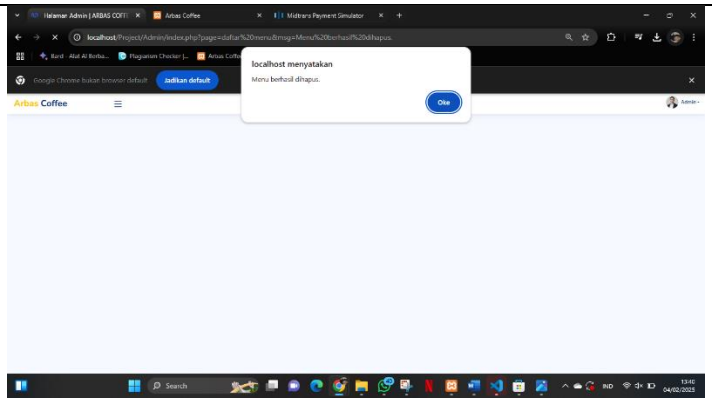

Salsa
Friday
Chat ini dengan akun bisnis. Pilih untuk info selengkapnya.
Saturday
Pesan dan panggilan terenkripsi secara end-to-end. Tidak seorang pun di luar chat ini, termasuk WhatsApp, yang dapat membaca atau mendengarkannya. Pilih untuk info selengkapnya.
Chat ini dengan akun bisnis. Pilih untuk info selengkapnya.
Hari ini
NOTA RESERVASI ARBAS COFFEE
NAMA: TASYA
EMAIL: 2192801110.TALIATASALSYABILLA@GMAIL.COM
TANGGAL: 2025-02-10
JUMLAH ORANG: 1-2 ORANG
WAKTU: 13:00:00
PESANAN:
- AMERICANO (1) - Rp 15.000
- KOPI CLASSIC (1) - Rp 15.000
Total: Rp 30.000
Terima kasih telah memesan di Arbas Coffee!
Ketik pesan

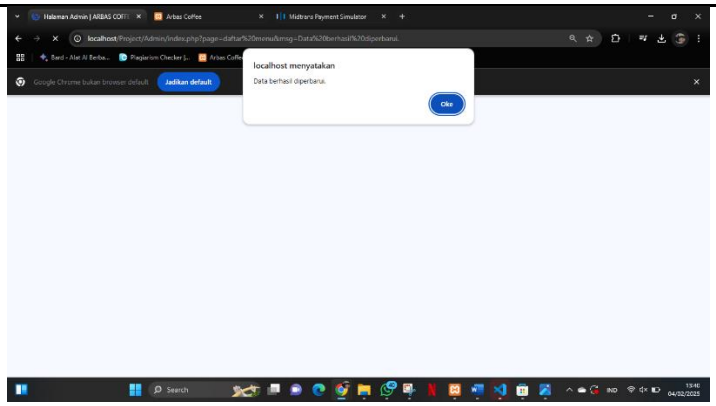

localhost menyatakan
Yakin ingin menghapus data ini?
Oke
Batal
Arbas Coffee
Reservasi
Recent Sales
Download Laporan Reservasi


Arbas Coffee
Reservasi
Recent Sales
Download Laporan Reservasi

d. Pengujian halaman daftar menu admin

Tabel 4. 4 Pengujian halaman daftar menu admin

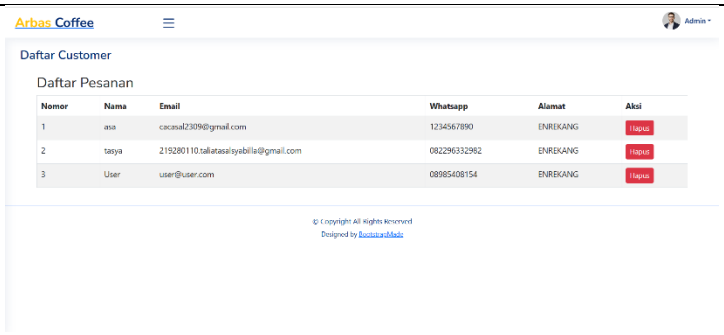
Tes Faktor	Hasil	Keterangan
Kelolah data halaman daftar menu meliputi melihat, menambah, menghapus, dan mengedit data mengubah jumlah stok.	✔	1. Berhasil menampilkan halaman saat menu ditekan. 2. Berhasil menambah data. 3. Berhasil meghapus data. 4. Berhasil mengedit data.
Screen Shoot		
		
		



e. Pengujian halaman daftar *customer* admin

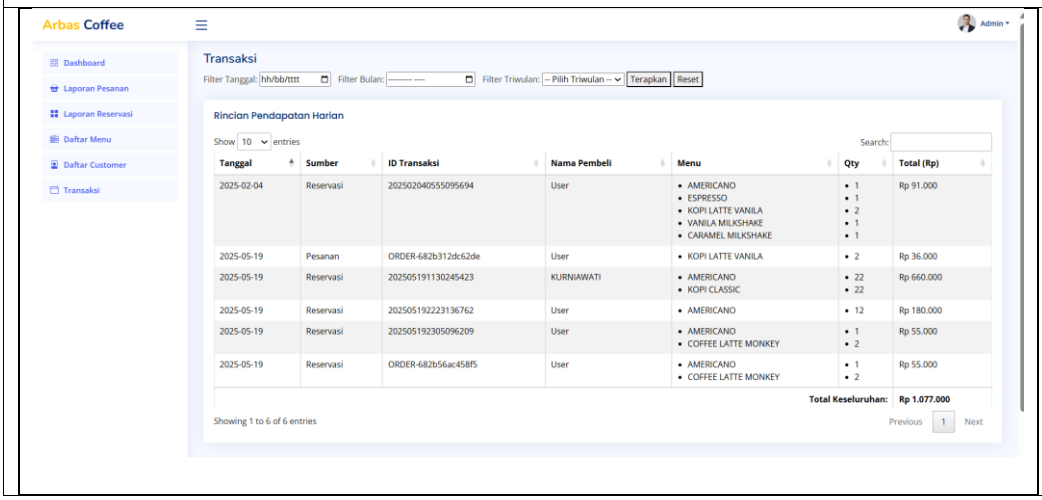
Tabel 4. 5 Pengujian halaman daftar *customer* admin

Tes Faktor	Hasil	Keterangan
Admin memilih menu daftar *customer*	✔	Berhasil menampilkan daftar *customer*.
Screen Shoot		



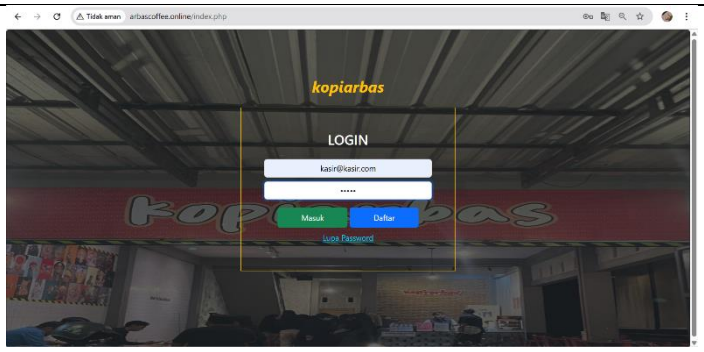
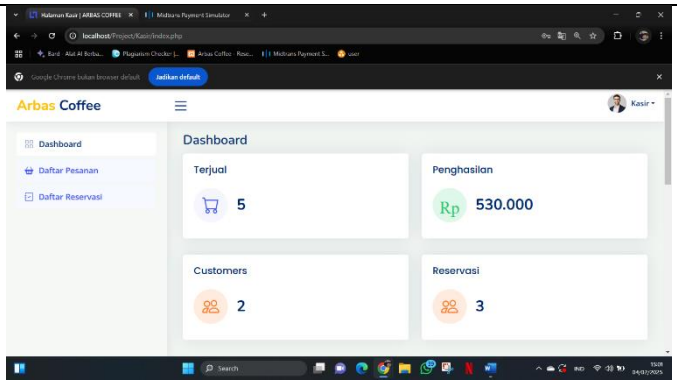
f. Pengujian halaman transaksi admin

Tabel 4. 6 Pengujian halaman transaksi admin

Tes Faktor	Hasil	Keterangan
Admin memilih menu transaksi	✔	Berhasil menampilkan halaman transaksi .
Screen Shoot		
		

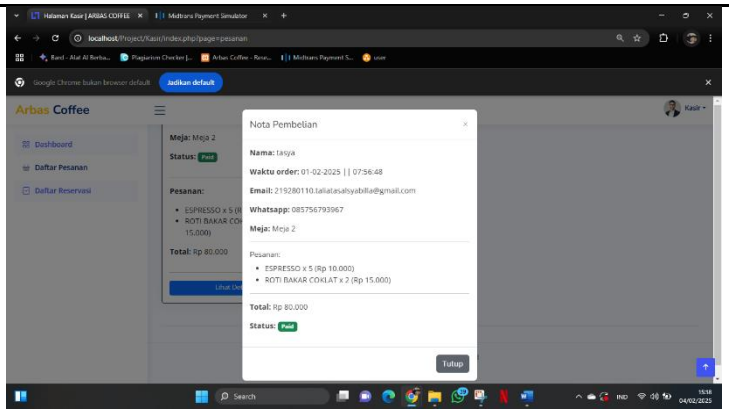
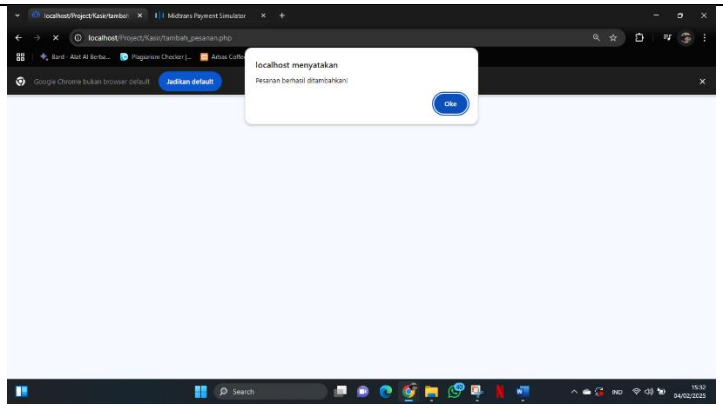
g. Pengujian *form login* kasir

Tabel 4. 7 Pengujian *form login* kasir

Tes Faktor	Hasil	Keterangan
Kasir memasukkan *username* dan *password* yang terdaftar pada sistem	✔	1. Jika *username* dan *password* tidak sesuai maka pengguna tetap berada pada halaman *login*. 2. Jika *username* dan *password* sesuai maka pengguna diarahkan ke halaman *dashboard*.
Screen Shoot		
		
		

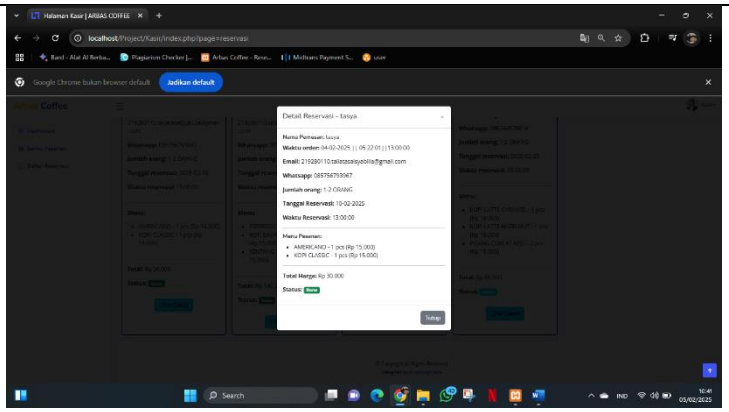
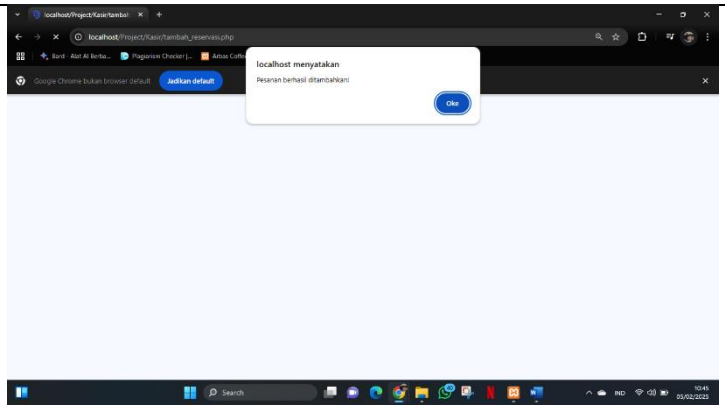
h. Pengujian halaman daftar pesanan kasir

Tabel 4. 8 Pengujian halaman daftar pesanan kasir

Tes Faktor	Hasil	Keterangan
Kelolah data halaman daftar pesanan meliputi melihat detail pesanan dan menambah pesanan	✔	1. Berhasil menampilkan halaman saat menu ditekan. 2. Berhasil melihat detail pesanan. 3. Berhasil menambah pesanan.
Screen Shoot		
		
		

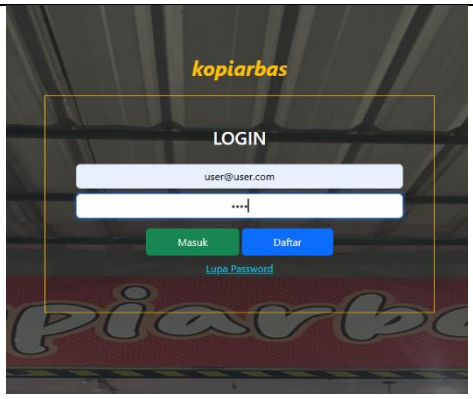
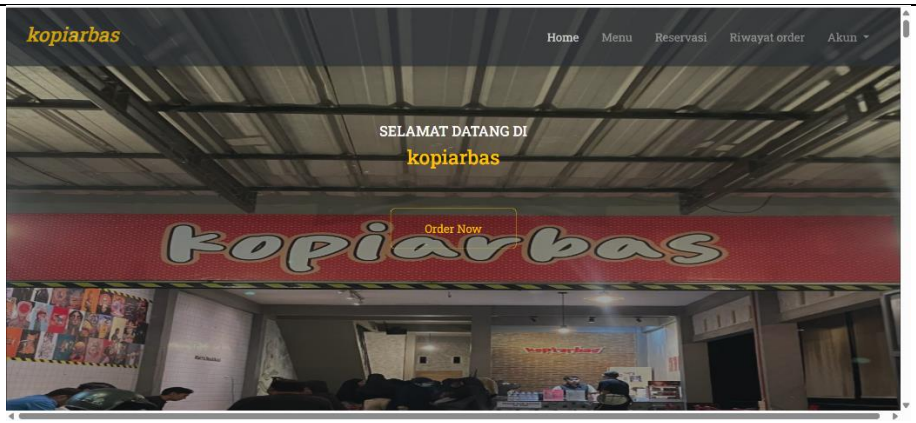
i. Pengujian halaman daftar reservasi kasir

Tabel 4. 9 Pengujian halaman daftar reservasi kasir

Tes Faktor	Hasil	Keterangan
Kelolah data halaman daftar reservasi meliputi melihat detail pesanan dan menambah reservasi	✔	1. Berhasil menampilkan halaman saat menu ditekan. 2. Berhasil melihat detail pesanan. 3. Berhasil menambah reservasi.
Screen Shoot		
		
		

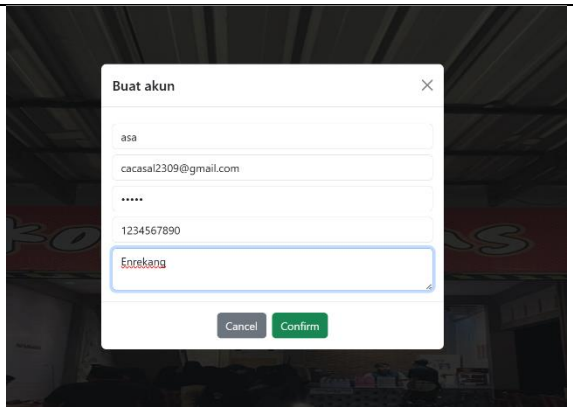
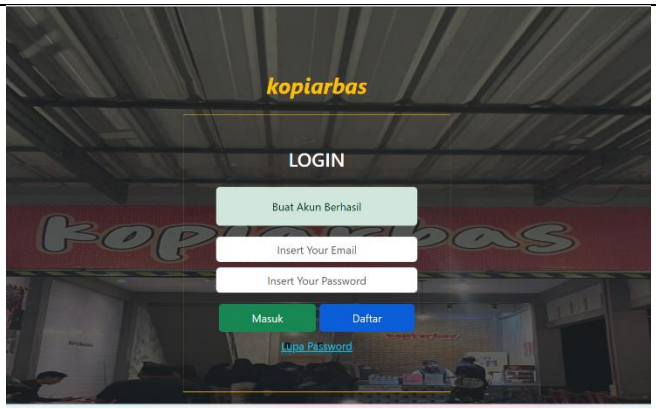
j. Pengujian *form login customer*

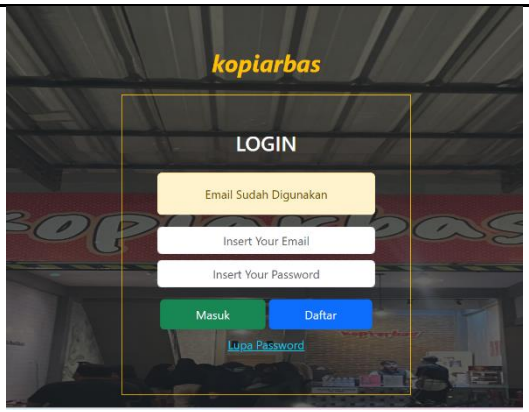
Tabel 4. 10 Pengujian *form login customer*

Tes Faktor	Hasil	Keterangan
Customer memasukkan *username* dan *password* yang terdaftar pada sistem	✔	1. Jika *username* dan *password* tidak sesuai maka pengguna tetap berada pada halaman *login*. 2. Jika *username* dan *password* sesuai maka pengguna diarahkan ke halaman *home*.
Screen Shoot		
		
		

k. Pengujian halaman registrasi *customer*

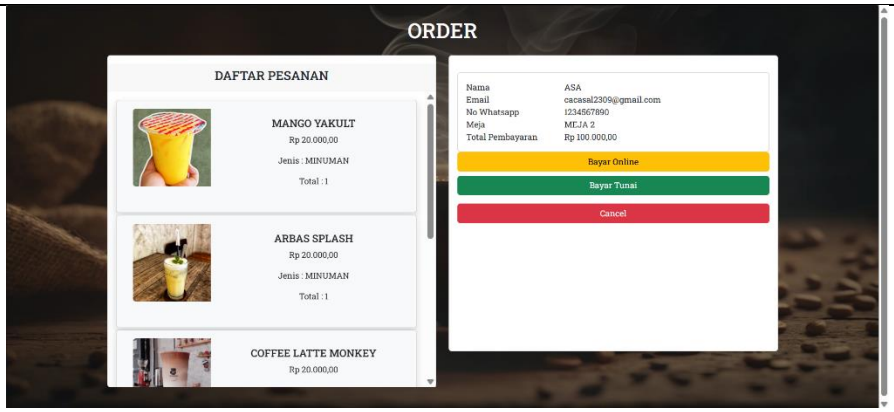
Tabel 4. 11 Pengujian halaman registrasi *customer*

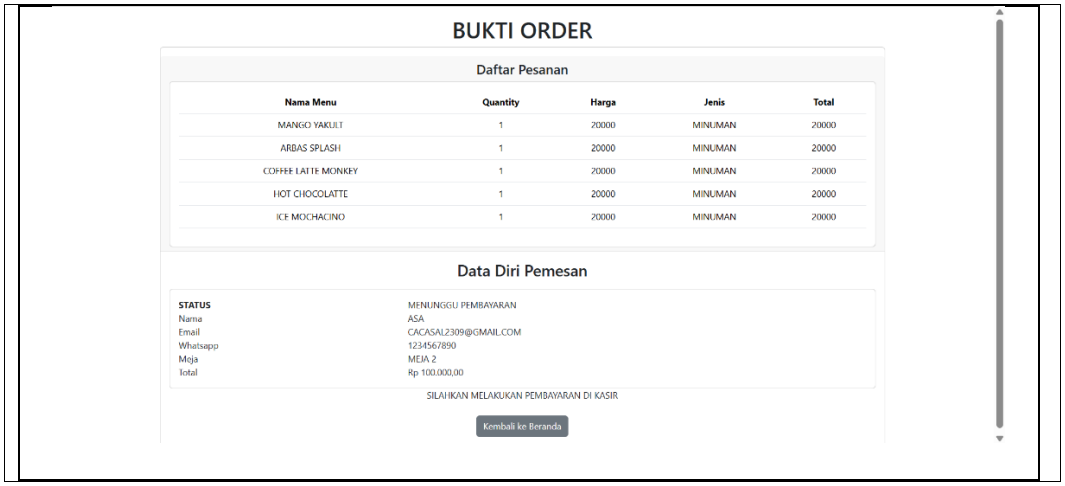
Tes Faktor	Hasil	Keterangan
Customer memasukkan data sesuai dengan *form*	✔	1. Jika data yang dimasukkan telah sesuai maka akan muncul pemberitahuan berhasil. 2. Jika *customer* telah mendaftarkan email sebelumnya maka registrasi gagal dan akan muncul pemberitahuan gagal karena satu email hanya bisa mendaftar satu kali.
Screen Shoot		
		
		



l. Pengujian halaman order menu *customer*

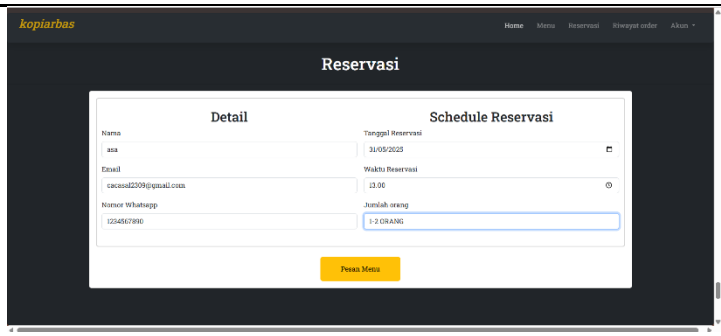
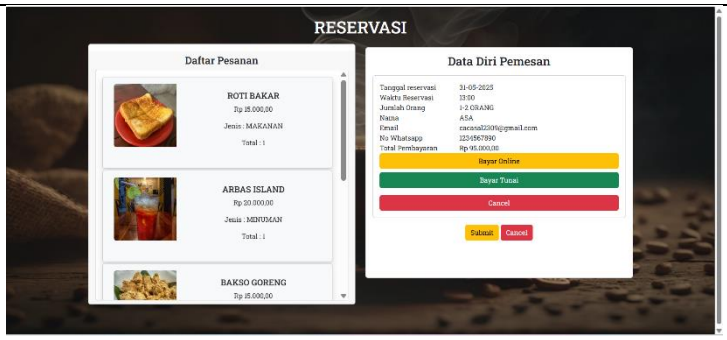
Tabel 4. 12 Pengujian halaman order menu *customer*

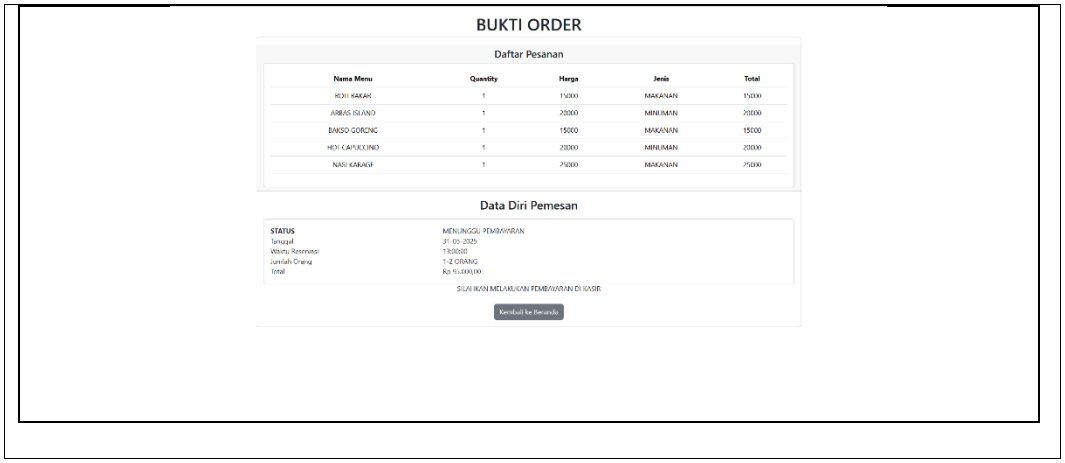
Tes Faktor	Hasil	Keterangan
Customer memilih order now	✔	Jika sudah memilih menu dan nomor meja maka akan muncul konfirmasi pesanan, *customer* dapat melakukan pembayaran dan akan muncul nota pembelian.
Screen Shoot		
		


BUKTI ORDER
Daftar Pesanan
Nama Menu
Quantity
Harga
Jenis
Total
MANGO YAKULT
1
20000
MINUMAN
20000
ARBAS SPLASH
1
20000
MINUMAN
20000
COFFEE LATTE MONKEY
1
20000
MINUMAN
20000
HOT CHOCOLATTE
1
20000
MINUMAN
20000
ICE MOCHACINO
1
20000
MINUMAN
20000
Data Diri Pemesan
STATUS
MENUNGGU PEMBAYARAN
Nama
ASA
Email
CACASAL2309@GMAIL.COM
Whatsapp
1234567890
Meja
MEJA 2
Total
Rp 100.000,00
SILAHKAN MELAKUKAN PEMBAYARAN DI KASIR
Kembali ke Beranda

m. Pengujian halaman reservasi *customer*

Tabel 4. 13 Pengujian halaman reservasi *customer*

Tes Faktor	Hasil	Keterangan
Customer memilih reservasi	✔	Jika sudah mengisi *form* dan memilih menu maka akan muncul konfirmasi pesanan, *customer* dapat melakukan pembayaran dan akan muncul nota pembelian.
Screen Shoot		
		
		



BUKTI ORDER

Daftar Pesanan

Nama Menu	Quantity	Harga	Jenis	Total
HEXI KAKAK	1	15000	MAKANAN	15000
ARBAS ISLAND	1	20000	MINUMAN	20000
BAKSO GORENG	1	15000	MAKANAN	15000
HEXI CAPUCCINO	1	20000	MINUMAN	20000
NASI KARAGE	1	25000	MAKANAN	25000

Data Diri Pemesan

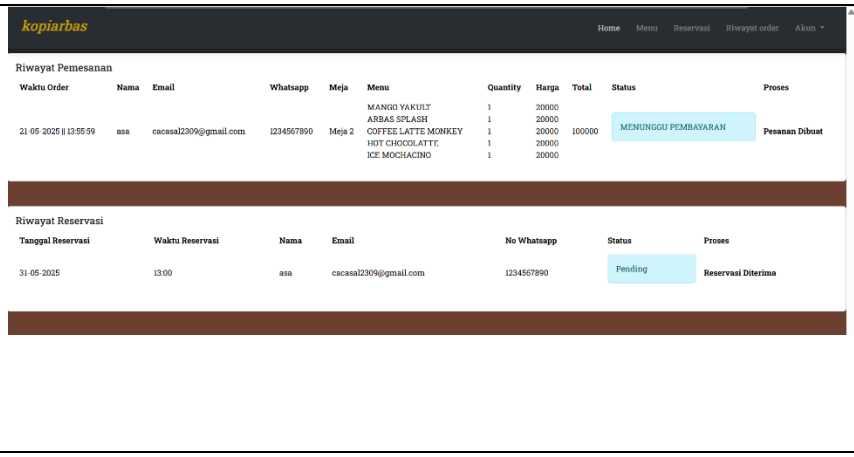
STATUS	MENUNGGU PEMBAYARAN
Tanggal	31-05-2025
Waktu Reservasi	13:30:00
Jumlah Orang	1-2 ORANG
Total	Rp 55.000,00

SILAHKAN MELAKUKAN PEMBAYARAN DI KASIR

Kembali ke Beranda

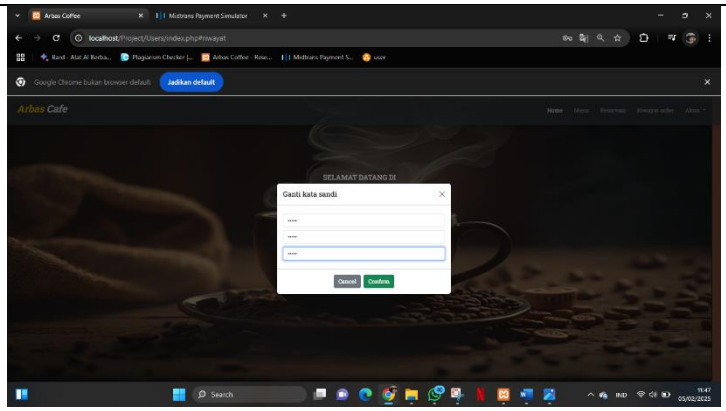
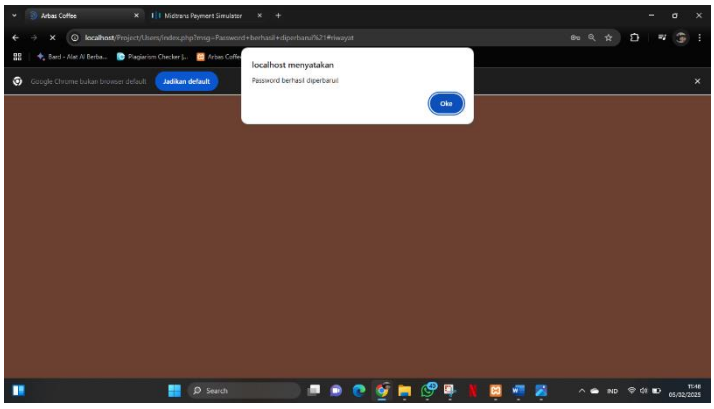
n. Pengujian halaman riwayat order *customer*

Tabel 4. 14 Pengujian halaman riwayat order *customer*

Tes Faktor	Hasil	Keterangan
Customer memilih riwayat order.	✔	Berhasil menampilkan riwayat order dari *customer*.
Screen Shoot		
		

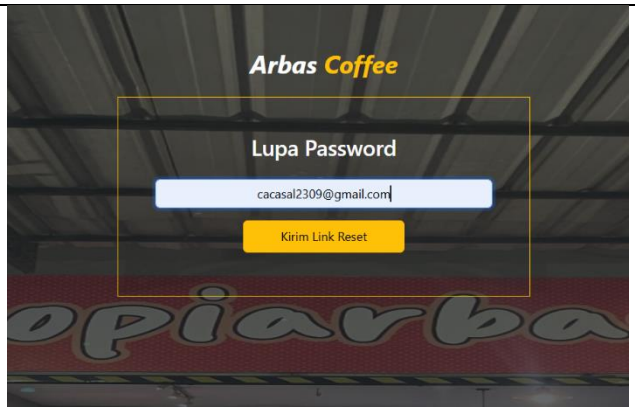
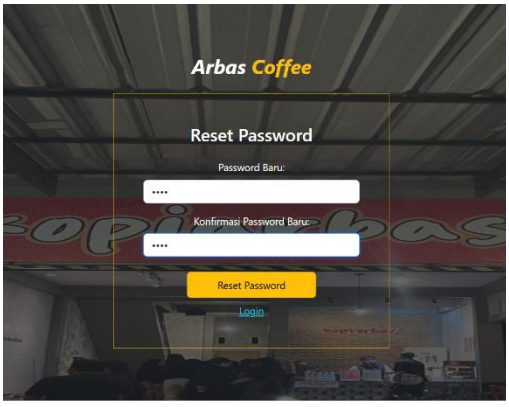
o. Pengujian halaman ganti sandi *customer*

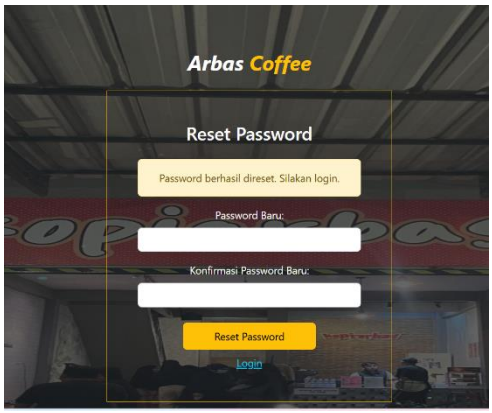
Tabel 4. 15 Pengujian halaman ganti sandi *customer*

Tes Faktor	Hasil	Keterangan
Customer memasukkan kata sandi lama dan kata sandi baru.	✔	Berhasil mengubah kata sandi ditandai dengan muncul *password* berhasil diperbaharui.
Screen Shoot		
		
		

p. Pengujian halaman lupa *password customer*

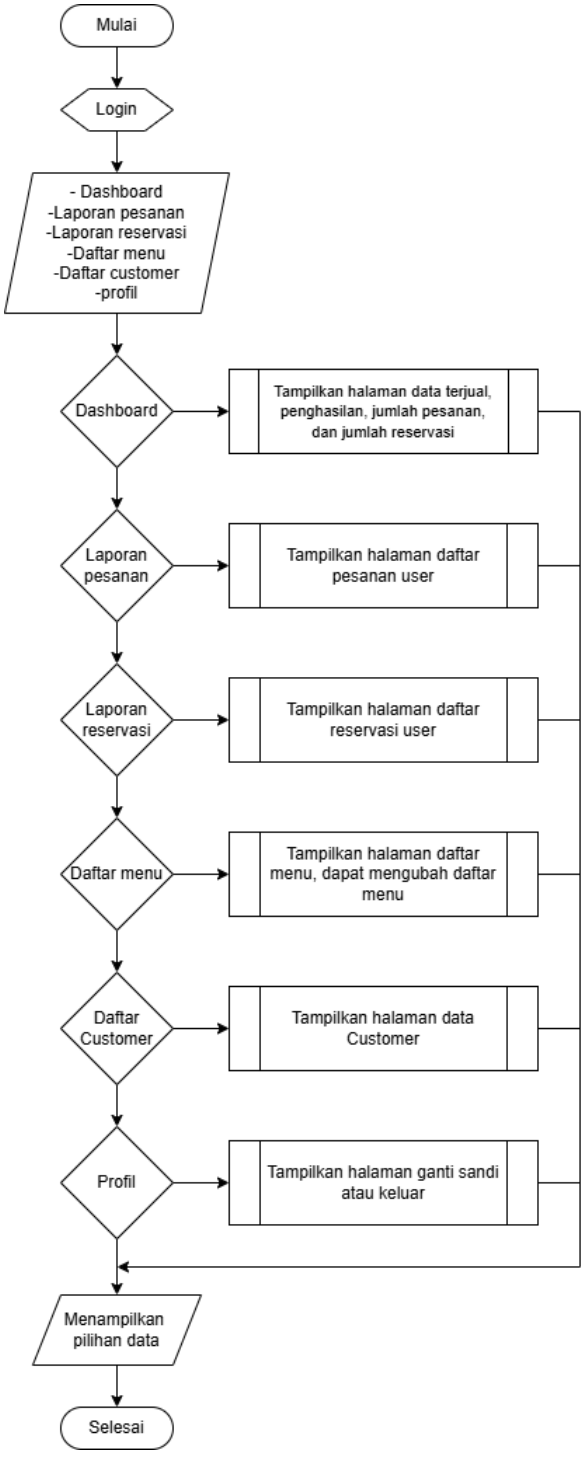
Tabel 4. 16 Pengujian halaman lupa *password customer*

Tes Faktor	Hasil	Keterangan
Customer memasukkan email yang terdaftar	✔	1. jika email ditemukan maka sistem akan mengirimkan link untuk mengubah *password* pengguna. 2. *Customer* mengecek pesan email yang berisi link dan diarahkan ke halaman reset *password* lalu memasukkan *password* yang baru.
Screen Shoot		
		
		

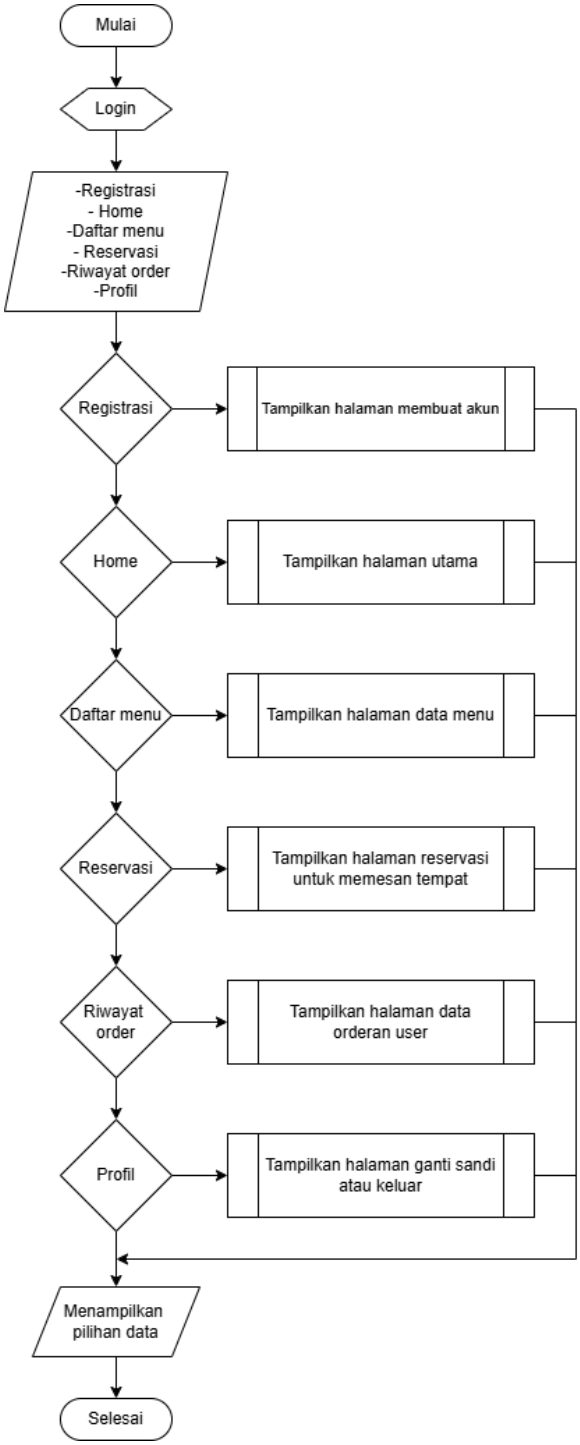

Arbas Coffee
Reset Password
Password berhasil direset. Silakan login.
Password Baru:
Konfirmasi Password Baru:
Reset Password
Login

2. White box Testing

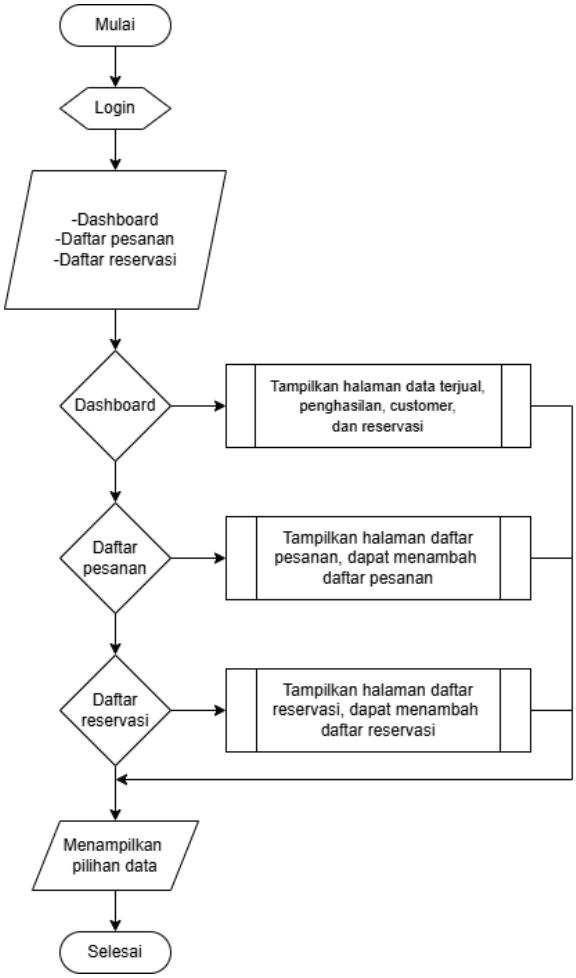
a. *Flowchart* dan *flowgraph*



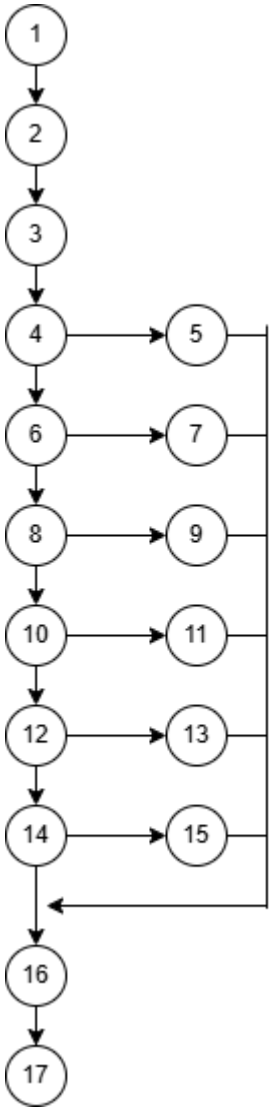
Gambar 4. 86 *Flowchart* aktor admin



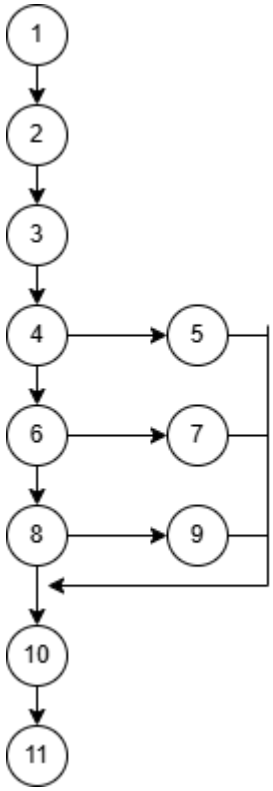
Gambar 4. 87 *Flowchart* aktor *user*



Gambar 4. 88 *Flowchart* aktor kasir



Gambar 4. 89 *Flowgraph* aplikasi admin dan *user*



Gambar 4. 90 *Flowgraph* aplikasi kasir

1) Dari *flowgraph* di atas dapat dilakukan proses perhitungan sebagai berikut:

Menghitung *Cylomatic Complexity* V(G) dari *edge* dan *Node* dengan rumus pada *flow*graph admin dan *user* :

V (G) = E – N + 2

N (node) = 17

E (edge) = 22

Penyelesaian :

V(G) = E – N + 2

= 22 – 17 + 2

= 7

Menghitung *Cylomatic Complexity* V(G) dari *edge* dan *Node* dengan rumus pada *flow*graph kasir :

V (G) = E – N + 2

N (node) = 11

E (edge) = 13

Penyelesaian :

V(G) = E – N + 2

= 13 – 11 + 2

= 4

2) Angka 8 dari hasil perhitungan *Cylomatic Complexity* menunjukkan jumlah *independent path* dari basis *path testing,* atau dengan kata lain menunjukkan jumlah pengujian yang harus dijalankan minimal seklai (semua *statement* telah diuji).
3) Hasil *independent path* dari *flowgraph* admin dan *user* di atas dapat dijabarkan sebagai berikut :

Path 1 : 1, 2, 3, 4, 5, 16, 17.

Path 2 : 1, 2, 3, 4, 6, 7, 16, 17.

Path 3 : 1, 2, 3, 4, 6, 8, 9, 16, 17.

Path 4 : 1, 2, 3, 4, 6, 8, 10, 11, 16, 17.

Path 5 : 1, 2, 3, 4, 6, 8, 10, 12, 13, 16, 17.

Path 6 : 1, 2, 3, 4, 6, 8, 10, 12, 14, 15, 16, 17.

Path 7 : 1, 2, 3, 4, 6, 8, 10, 12, 14, 16, 17.

Hasil *independent path* dari *flowgraph* kasir di atas dapat dijabarkan sebagai berikut :

Path 1 : 1, 2, 3, 4, 5, 10, 11.

Path 2 : 1, 2, 3, 4, 6, 7, 10, 11.

Path 3 : 1, 2, 3, 4, 6, 8, 9, 10, 11.

Path 4 : 1, 2, 3, 4, 6, 8, 10, 11.

Tabel 4. 17 Grafik matriks admin dan *user*

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

Tabel 4. 18 Grafik matriks kasir

	1	2	3	4	5	6	7	8	9	10	11	E = 1
1		1										0
2			1									0
3				1								0
4					1	1						1
5										1		0
6							1	1				1
7										1		0
8									1	1		1
9										1		0
10											1	0
SUM (E + 1)												4

3. Pengujian kegunaan

Pengujian kegunaan (*usability testing*) adalah metode evaluasi yang digunakan untuk menguji seberapa mudah dan intuitif suatu produk atau layanan digunakan oleh pengguna sebenarnya. Tujuan utama dari *usability testing* adalah untuk mengidentifikasi masalah dalam desain produk dan menemukan area di mana pengguna mengalami kesulitan atau kebingungan. Pengujian kegunaan pada aplikasi ini dilakukan dengan memberikan kuesioner ke 5 orang responden yang berstatus pelanggan

Menurut Kinnear dalam Husein Umar (2009), Skala *Likert* mencakup pernyataan seperti setuju-tidak setuju, senang-tidak senang, dan baik-tidak baik. Studi ini menggunakan skala *likert* yang berkisar dari 1-4.

Tabel 4. 19 Kriteria pembobotan jawaban

Pilihan Jawaban	**Bobot Skor**	**Skor Seluruhnya**
Sangat setuju	4	20
Setuju	3	15
Tidak setuju	2	10
Sangat tidak setuju	1	5

Tabel 4. 20 Kriteria interpretasi skor

Angka	**Keterangan**
0%-19,99%	Sangat tidak setuju
20%-39,99%	Tidak setuju
40%-59,99%	Cukup setuju
60%-79,99%	Setuju
80%-100%	Sangat setuju

Berikut hasil persentase masing-masing jawaban dari 10 pertanyaan yang diberikan kepada 5 responden:

a. Tampilan aplikasi menarik dan mudah dipahami

Tabel 4. 21 Hasil perhitungan pertanyaan pertama

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	3	12	(18/20)*100 = 90%
Setuju	3	2	6	
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	18	

b. Navigasi dalam aplikasi mudah digunakan dan tidak membingungkan

Tabel 4. 22 Hasil perhitungan pertanyaan kedua

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	2	8	(17/20)*100
Setuju	3	3	9	= 85%
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	17	

c. Informasi menu (nama, harga, gambar) mudah dibaca

Tabel 4. 23 Hasil perhitungan pertanyaan ketiga

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	3	12	(18/20)*100
Setuju	3	2	6	= 90%
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	18	

d. QR Code/link untuk mengakses menu dapat digunakan dengan mudah.

Tabel 4. 24 Hasil perhitungan pertanyaan keempat

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	1	4	(16/20)*100 = 80%
Setuju	3	4	12	
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	16	

e. Proses pemesanan menu melalui aplikasi mudah dilakukan

Tabel 4. 25 Hasil perhitungan pertanyaan kelima

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	-	-	(14/20)*100 = 70%
Setuju	3	4	12	
Tidak setuju	2	1	2	
Sangat tidak setuju	1	-	-	
Jumlah		5	14	

f. Waktu yang dibutuhkan untuk memesan makanan melalui aplikasi lebih cepat dibanding pemesanan manual

Tabel 4. 26 Hasil perhitungan pertanyaan keenam

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	-	-	(15/20)*100 = 75%
Setuju	3	5	15	
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	15	

g. Aplikasi memungkinkan saya untuk melihat riwayat pesanan dengan mudah

Tabel 4. 27 Hasil perhitungan pertanyaan ketujuh

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	-	-	(15/20)*100 = 75%
Setuju	3	5	15	
Tidak setuju	2	-	-	
Sangat tidak setuju	1	-	-	
Jumlah		5	15	

h. Fitur reservasi tempat di aplikasi mudah digunakan.

Tabel 4. 28 Hasil perhitungan pertanyaan kedelapan

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	1	4	(15/20)*100 = 75%
Setuju	3	3	9	
Tidak setuju	2	1	2	
Sangat tidak setuju	1	-	-	
Jumlah		5	15	

i. Kecepatan respon aplikasi saat berpindah halaman cukup baik.

Tabel 4. 29 Hasil perhitungan pertanyaan kesembilan

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	-	-	(14/20)*100 = 70%
Setuju	3	4	12	
Tidak setuju	2	1	2	
Sangat tidak setuju	1	-	-	
Jumlah		5	14	

j. Aplikasi kompatibel dan berfungsi dengan baik di perangkat yang saya gunakan.

Tabel 4. 30 Hasil perhitungan pertanyaan Kesepuluh

Jawaban	Skor	Responden	Jumlah skor	Nilai persentase
Sangat Setuju	4	1	4	(14/20)*100 = 70%
Setuju	3	2	6	
Tidak setuju	2	2	4	
Sangat tidak setuju	1	-	-	
Jumlah		5	14	

BAB V

PENUTUP

A. Kesimpulan

Sistem pemesanan menu *Multi channel* yang dirancang berhasil diimplementasikan menggunakan kode *QR* dan tautan sebagai media akses, sehingga memungkinkan pelanggan memesan makanan dan minuman dengan lebih cepat dan lebih efisien. Implementasi sistem ini akan membantu mengurangi kesalahan pencatatan pesanan yang sering terjadi pada sistem manual. Selain itu, hal ini dapat meningkatkan pengalaman pelanggan dengan menyediakan proses yang lebih nyaman. Sistem ini terbukti menghemat waktu bagi pelanggan dan staf kafe, terutama selama jam sibuk, karena proses pemesanan terintegrasi langsung ke kasir.

B. Saran

Untuk pengembangan aplikasi web ini dapat menambahkan fitur *delivery* agar pelanggan dapat menikmati layanan tanpa harus datang langsung ke lokasi dan pelacakan status pemesanan agar pelanggan dapat mengetahui estimasi waktu penyajian.

DAFTAR PUSTAKA

Andriani, R., Nugroho, B., & Yuliana, S. (2022). Rancang Bangun Aplikasi Mobile Pemesanan Produk UMKM. Prosiding Seminar Nasional Teknologi dan Rekayasa, 4(1), 98–105.

Fadhilah, R., & Wicaksono, A. (2022). Implementasi MySQL dalam Pengembangan Sistem Informasi Penjualan. Jurnal Teknologi dan Informatika, 9(3), 105–112.

Fahmi, R., & Wibowo, T. (2021). Penggunaan UML dalam Analisis dan Perancangan Sistem Informasi Manajemen. Jurnal Sistem dan Teknologi Informasi, 9(2), 102–110.

Handayani, D., & Prasetyo, A. (2023). Transformasi Digital melalui Strategi Multi channel dalam Industri Retail. Jurnal Bisnis Digital, 4(1), 77–85.

Indrawan, H., & Prasetyo, D. (2022). Sistem Pendukung Keputusan Pemilihan Supplier Menggunakan Metode SAW. Jurnal Teknologi dan Informatika, 9(2), 102–110.

Maulana, T. H., & Widodo, S. (2023). Penerapan Progressive Web App pada Sistem Informasi Inventaris. Jurnal Teknologi dan Sistem Informasi, 11(3), 201–210.

Nugroho, B., & Lestari, D. (2022). Perancangan Web Profil Perusahaan Menggunakan HTML dan Bootstrap. Jurnal Teknologi dan Informasi, 10(3), 88–95.

Pambudi, L. B. A., Rahagiyanto, A., & Suyoso, G. E. J. (2020). Implementasi QR *Code* untuk efisiensi waktu pemesanan menu makanan dan minuman di restoran maupun kafe. BIOS: Jurnal Teknologi In*form*asi dan Rekayasa Komputer, 1(1), 35-39.

Prasetyo, A., & Firmansyah, R. (2020). Penerapan PHP dan MySQL dalam Pembuatan Sistem Informasi Akademik. Jurnal Ilmiah Komputer dan Informatika, 8(1), 45–52.

Ramdani, R., & Lestari, D. (2021). Analisis dan Perancangan Sistem Informasi Penggajian. Jurnal Sistem Informasi dan Komputer Terapan Indonesia, 3(2), 50–59.

Rizky, A., & Maulana, I. (2024). Pembuatan Website Informasi Produk Menggunakan HTML dan Tailwind CSS. Jurnal Pengembangan Teknologi Informasi, 12(2), 66–74.

Saputra, D., Arafat, M., & Saputro, H. (2022). MEMBANGUN WEBSITE PADA PT SURYA BINTANG INDONESIA MENGGUNAKAN PHP DAN MYSQL. *JTIM: Jurnal Teknik Mahakarya*, *5*(1), 17-24.

Sari, M., & Nugroho, A. (2020). Sistem Pemesanan Menu Berbasis Android pada Restoran Cepat Saji. Jurnal Teknologi Informasi dan Ilmu Komputer, 7(3), 190–198.

Suhardi, A., & Fatkhurrahman, M. (2020). Perancangan Aplikasi Mobile Edukasi Berbasis Android. Jurnal Teknologi Informasi dan Ilmu Komputer, 7(2), 123–130.

Sutrisno, A., & Hidayat, R. (2020). Perancangan Sistem Informasi Inventaris Barang pada Sekolah Berbasis Web. Jurnal Teknologi dan Sistem Informasi, 6(1), 22–30.

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

Wahyuni, D., & Hidayat, R. (2021). Pengembangan Aplikasi Informasi Akademik Berbasis Web. JICSI (Jurnal Ilmu Komputer dan Sistem Informasi), 9(1), 45–53.

Wiratama, A. (2020). RANCANG BANGUN APLIKASI PEMESANAN MENU MAKANAN RESTORAN BERBASIS ANDROID (Doctoral dissertation, Unversitas Islam Negeri Sultan Syarif Kasim Riau).

Wulandari, D., & Ramadhan, F. (2021). Analisis Sistem Informasi Penjualan dengan Pendekatan Flowchart dan DFD. Jurnal Teknologi Informasi dan Rekayasa Sistem, 9(1), 45–53.