

Implementasi Payment Gateway Pada Pengembangan Sistem Pemesanan Menu Kafe Berbasis Mobile

Muhammad Mundzir Tamam^{*1}, Rodhiyah Mardhiyyah²

¹Informatika, Universitas Teknologi Yogyakarta

²Teknik Komputer, Universitas Teknologi Yogyakarta

E-mail : ¹muhammad.5210411390@student.uty.ac.id, ²rodhiyah@staff.uty.ac.id

*Corresponding author

Received 23 April 2025; Revised 7 May 2025; Accepted 13 May 2025

Abstrak - Pemesanan menu pada suatu kedai kopi, umumnya dilakukan dengan pelanggan datang langsung ke lokasi, memilih menu yang diinginkan, dan melakukan pemesanan melalui kasir. Cara tersebut seringkali dijumpai sebagai satu-satunya cara untuk memesan menu kafe. Sistem tersebut juga diterapkan pada Kopi Gambus Krupyak, Yogyakarta dan masih terdapat beberapa kelemahan, seperti waktu yang dibutuhkan apabila antrean melonjak dan kesalahan saat mencatat pesanan pelanggan. Penelitian ini bertujuan untuk mengembangkan sistem pemesanan menu berbasis mobile yang memungkinkan pelanggan untuk memesan dan membayar langsung dari ponsel dan mengurangi antrean dan waktu tunggu. Metode yang digunakan melibatkan perancangan sistem menggunakan Unified Modelling Language (UML) dan pengembangan aplikasi pemesanan berbasis mobile sekaligus sistem pengelolaan kafe berbasis website. Aplikasi pemesanan dibangun dengan menggunakan framework React Native, sedangkan Sistem pengelolaan menggunakan sebuah framework bahasa PHP, yakni CodeIgniter. Sistem yang dibangun ini dilengkapi dengan fitur Payment Gateway dari Midtrans yang memungkinkan pelanggan untuk membayar pesanannya secara cashless langsung di dalam aplikasi sehingga dapat mengurangi antrean. Hasil penelitian menunjukkan bahwa sesudah pengimplementasian aplikasi, pelanggan tidak perlu repot untuk memesan, membayar, dan menunggu di kafe sehingga beberapa antrean pesanan pelanggan dapat dikerjakan secara bersamaan.

Keywords - Aplikasi Mobile, Payment Gateway, Pemesanan Menu, Sistem berbasis website, Unified Modelling Language.

Abstract - *Menu ordering at a coffee shop, is generally done with the customers visit the location, select their desired items, and place orders through a cashier, often the only available way to order. This system is also implemented at Kopi Gambus Krupyak, Yogyakarta, and still faces several challenges, such as extended waiting times during peak hours and errors in recording customer orders. This study aims to develop a mobile-based menu ordering system that allows customers to order and pay directly from their phones, reducing queues and waiting times. The method involves designing the system using Unified Modeling Language (UML) and developing both a mobile ordering application and a web-based café management system. The ordering application is built using the React Native framework, while the management system utilizes CodeIgniter, a PHP-based framework. The system is equipped with a Midtrans Payment Gateway feature, enabling customers to make cashless payments directly within the app, helping to minimize queues. The results show that, after the application's implementation, customers no longer need to order, pay, and wait at the café, allowing multiple orders to be processed simultaneously.*

Keywords - *Menu ordering, Mobile application, Payment Gateway, Unified Modeling Language, Website-based system.*

1. PENDAHULUAN

Pemesanan menu yang terjadi pada suatu kedai kopi umumnya berlangsung sesuai kebiasaan, di mana pelanggan datang langsung ke lokasi, memilih menu yang diinginkan, dan melakukan pemesanan melalui kasir. Pada sistem ini, pelanggan biasanya memesan, membayar, dan menunggu pesanan diproses di tempat. Sistem pemesanan seperti ini seringkali mengandalkan interaksi langsung antara pelanggan dan staf, dan bergantung pada kecepatan layanan agar pelanggan tidak menunggu terlalu lama.

Di Kopi Gambus, sebuah kafe di Krapyak, Bantul, proses pemesanan manual yang dilakukan oleh staf kasir rata-rata memerlukan waktu 2-5 menit. Namun ketika antrean melonjak, proses ini dapat memakan waktu yang lebih lama yaitu sekitar 5-15 menit. Selain itu, kesalahan dalam pencatatan pesanan juga sering terjadi, yang pada akhirnya mengurangi kepuasan pelanggan dan menurunkan kualitas pelayanan. Hal ini berdampak negatif bagi reputasi dan kepercayaan pelanggan terhadap kafe. Berdasarkan data dari Studi Forrester "The State of Retail 2023", konsumen modern menginginkan pengalaman transaksi yang efisien dan personal di berbagai saluran, dan lebih memilih bertransaksi di toko yang memberikan pengalaman berbelanja yang dipersonalisasi.

Untuk mengatasi masalah ini, diperlukan sebuah aplikasi *mobile* yang dapat menangani sistem pemesanan yang rumit. Aplikasi *mobile* sendiri adalah aplikasi yang dirancang secara eksklusif untuk platform *mobile* (misalnya iOS, Android, atau Windows Mobile) [1]. Aplikasi *mobile* umumnya membantu penggunaannya terhubung dengan layanan internet yang biasanya diakses melalui PC, serta memudahkan penggunaannya menggunakan aplikasi internet pada perangkat yang portabel. Istilah "*mobile*" adalah kata sifat yang merujuk pada kemampuan untuk bergerak atau berpindah dengan mudah. [2]. Aplikasi pemesanan *mobile* memungkinkan pelanggan untuk memesan dan membayar langsung dari ponsel dan mengurangi waktu tunggu. Pelanggan dapat melihat menu, memilih item sesuai keinginan, dan melakukan pembayaran secara *online*, tanpa perlu mengantre di kasir. Sistem ini juga mendukung personalisasi pesanan, memungkinkan pelanggan untuk menambahkan catatan khusus atau memilih variasi menu sesuai preferensi karena tergolong aplikasi berjenis *e-commerce*. Secara definisi sendiri, *E-commerce* adalah teknologi internet yang digunakan oleh konsumen dan pelaku bisnis guna membeli dan menjual barang [3]. *E-Commerce* meliputi distribusi, penjualan, pembelian, pemasaran dan pelayanan dari sebuah produk yang dilakukan melalui sistem elektronik seperti internet atau bentuk jaringan komputer yang lain [4]. Meningkatnya bisnis *e-commerce* memungkinkan individu untuk memesan makanan dengan mudah melalui aplikasi khusus di smartphone atau perangkat genggam [5]. Dengan demikian, pengembangan sistem ini diharapkan dapat meningkatkan kenyamanan pelanggan.

Dalam pengembangannya, sebuah sistem berupa aplikasi tidaklah luput dari pemrograman berbasis objek. Pemrograman berorientasi objek (PBO) adalah sebuah konsep pemrograman yang mendefinisikan suatu proses penyelesaian masalah dalam program dengan menganalogikannya sebagai objek yang saling berinteraksi satu sama lain [6]. Selain itu, diperlukan sebuah konsep yang bernama MVC atau *Model-View-Controller* guna memisahkan pengembangan aplikasi berdasarkan komponen primer yang membentuk sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan elemen pengendali aplikasi [7]. Sementara itu, data sebagai komponen penting jalannya konsep MVC diperoleh dari sebuah *web service* bernama RESTful API yang merupakan sebuah perangkat lunak independen dari platform, arsitektur, atau bahasa pemrograman, yang menyediakan layanan dan metode untuk pertukaran data dan dapat diakses melalui jaringan. [8]. Di dalam API ini, terdapat fungsi CRUD yang berjalan melalui protokol HTTP dengan metode *GET*, *POST*, *PUT* dan *DELETE* [9].

Sementara itu untuk bahasa pemrograman yang digunakan aplikasi sendiri ialah Node.js. Node.js adalah *software* yang digunakan untuk membangun aplikasi web dan dikembangkan menggunakan bahasa pemrograman JavaScript [10]. Node.js sendiri memiliki beberapa macam

framework atau kerangka kode yang dapat digunakan untuk mempermudah pengembangan aplikasinya. Salah satu *framework* yang mendukung pengembangan sistem berbasis *mobile* ini ialah React Native. React Native adalah kerangka kerja pengembangan perangkat lunak berbasis JavaScript yang memungkinkan pengembang menciptakan aplikasi *mobile* yang responsif dan dapat digunakan di berbagai platform [9]. React Native memungkinkan pengembang membuat antarmuka seluler yang interaktif menggunakan komponen deklaratif. Aplikasi yang dibangun dengan React Native bukanlah aplikasi web seluler, karena *framework* ini memanfaatkan elemen UI dasar yang sama seperti yang digunakan pada aplikasi iOS dan Android [11].

Kemudian aplikasi pemesanan *mobile* ini memiliki fitur berupa pembayaran di dalam aplikasi berbasis *Payment Gateway*. *Payment Gateway* sendiri adalah layanan pihak ketiga yang menjembatani *merchant* dengan bank. Dengan adanya layanan ini, *merchant* dapat menyediakan pembayaran *online* di situs belanja dengan menghubungkan *website* ke *payment gateway* melalui API atau *Application Program Interface* [12]. *Payment gateway* sangat diperlukan karena biaya awal dan pemeliharaan yang tinggi untuk menghubungkan dengan bank, serta perlunya sistem yang dapat menghubungkan rekening bank pelanggan dan *merchant* [13]. Salah satu *payment gateway* yang mendukung banyak metode pembayaran ialah Midtrans. Midtrans merupakan platform pembayaran *online* yang menyediakan integrasi dengan beragam metode pembayaran di Indonesia [14]. Midtrans menyediakan berbagai metode pembayaran, termasuk kartu kredit, transfer bank, debit langsung, dompet digital, dan pembayaran tunai. Selain itu, API Midtrans dapat diintegrasikan dengan *framework* React Native dan Node.js [15].

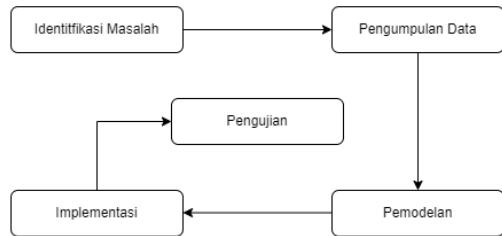
Pembangunan sebuah sistem yang dapat menunjang pemesanan sebuah kafe juga memperhatikan sudut pandang kafe itu sendiri terutama dalam pengelolaan pesanan dan menu. Oleh karena itu, sistem pengelolaan pesanan dan menu dibangun di atas bahasa pemrograman PHP dan salah satu *framework*-nya, CodeIgniter. PHP merupakan salah satu bahasa *scripting* yang berfungsi untuk membangun aplikasi web dan mengintegrasikannya dengan *server* [16]. Sementara itu, CodeIgniter adalah *framework* PHP *open source* yang menerapkan metode MVC (*Model, View, Controller*) untuk membantu pengembang atau *programmer* dalam membangun aplikasi web secara lebih efisien tanpa harus mulai dari nol [17].

Berkaca dari penelitian-penelitian sebelumnya, sistem yang dibangun dalam penelitian ini mengadopsi konsep pemesanan menu layaknya aplikasi ojek *online*, namun dirancang khusus untuk kebutuhan komersial sebuah kafe. Aplikasi ini menampilkan navigasi menu, fitur pencarian, keranjang pesanan, serta opsi pembayaran COD dan *virtual payment*. Selain itu, aplikasi ini dikembangkan menggunakan React Native dan NodeJS dengan basis JavaScript, sementara pengelolaan data oleh staf dilakukan melalui sistem *website* berbasis CodeIgniter dan PHP, untuk memberikan tampilan *dashboard* yang lebih akurat dan mudah diakses. Aplikasi pemesanan dirancang khusus untuk Kopi Gambus, dengan kategori menu yang meliputi kopi, non-kopi, dan menu pendamping yang dikelola oleh staf sehingga memudahkan dalam memilih menu, mengurangi kesalahan dalam perhitungan pembayaran, dan memungkinkan kustomisasi pesanan tanpa catatan tambahan. Sementara itu, Sistem pengelolaan pesanan dan menu berbasis web untuk memudahkan staf dalam mengakses dan mengelola data serta mencatat data pesanan untuk keperluan analisis bisnis pada masa depan.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Dalam mencapai tujuan penelitian, langkah awalnya adalah menyusun tahapan penelitian yang sederhana. Hal ini diperlukan agar langkah-langkah berikutnya dapat mengacu pada sebuah referensi sehingga tidak akan melenceng dari konsep awal. Pemetaan sumber data dilakukan untuk menata sistem yang mudah dipahami oleh penggunanya. Setelah itu, sistem disajikan secara lugas melalui teknologi *mobile* yang mudah diakses. Untuk lebih jelasnya, penelitian ini akan memiliki beberapa tahapan seperti yang tampak pada gambar di bawah ini.



Gambar 1. Tahapan Penelitian.

Tahapan Penelitian pada Gambar 1 di atas mempunyai beberapa langkah, diantaranya ialah identifikasi masalah yang menganalisa kelemahan dan kesenjangan pada objek yang diteliti sehingga dapat dilakukan tahapan kedua yakni pengumpulan data. Pengumpulan data dilakukan bertujuan untuk menggali informasi secara lebih mendalam termasuk alur kerja, proses bisnis, dan teknologi yang sudah digunakan. Langkah berikutnya merupakan eksekusi yang meliputi perencanaan dan pemodelan, konstruksi atau implementasi, serta pengujian.

2.2. Metode Pengumpulan Data

Penelitian ini menggunakan beberapa metode pengumpulan data, yaitu studi literatur dan wawancara. Metode-metode tersebut berperan sebagai acuan dalam pengembangan aplikasi, sehingga diperlukan informasi dan data yang relevan untuk mendukung proses tersebut.

2.2.1. Studi Literatur

Studi Literatur dilakukan untuk membangun fondasi yang kokoh dengan cara mencari, membaca, dan menganalisis berbagai sumber tepercaya seperti jurnal, buku, dan artikel di internet, termasuk Google Scholar. Tujuannya adalah untuk memahami landasan teori, kerangka kerja, temuan sebelumnya, dan trend terbaru dalam agroindustri kopi.

2.2.2. Wawancara

Wawancara bertujuan untuk memahami proses pemesanan menu yang ada saat ini, mengidentifikasi permasalahan dalam sistem pemesanan manual, serta mendapatkan umpan balik dari pengguna. Narasumber dalam proses wawancara ini meliputi pemilik kafe dan beberapa pelanggan kafe. Dari proses wawancara tersebut, diperoleh informasi-informasi penting yang berkaitan pada daftar menu, harga menu, data pelanggan, sampai dengan data staf karyawan. Informasi yang diperoleh dari wawancara ini dapat menjadi bahan berharga dalam merancang dan mengembangkan aplikasi pemesanan menu di Kopi Gambus Krapyak Yog Jakarta.

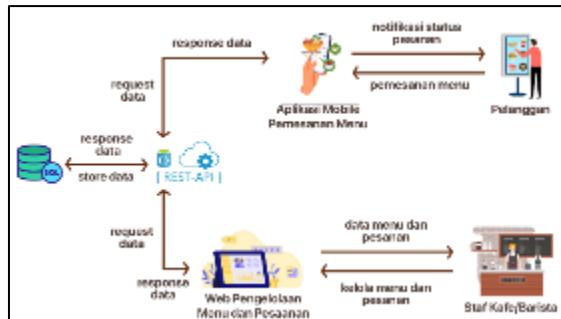
2.3. Perancangan dan Pemodelan

Berdasarkan hasil analisis sistem yang berjalan saat ini, diperlukan rancangan usulan sistem baru yang lebih baik. Rancangan ini harus memperhatikan dan mengatasi kekurangan-kekurangan yang ada pada sistem lama. Alur sistem baru yang dirancang menggambarkan proses pemesanan menu dan pengambilan pesanan berbasis *mobile* di Kopi Gambus. Pelanggan dapat melakukan pemesanan melalui aplikasi, dengan opsi pembayaran *online* atau menggunakan metode bayar tunai saat mengambil pesanan. Setelah pesanan diterima, kafe memprosesnya sesuai dengan preferensi pelanggan. Setelah pesanan siap, pelanggan akan mendapatkan pemberitahuan dari aplikasi dan dapat mengambil pesanan. Jika pelanggan belum membayar secara *online*, pelanggan dapat membayar di kasir kafe saat mengambil pesanan.

2.3.1 Rancangan Arsitektur

Fokus utama dalam perancangan arsitektur sistem adalah penggunaan *Server REST (Representational State Transfer)* API sebagai basis komunikasi antara aplikasi *mobile* Android dan *server*. Dengan menggunakan API ini, aplikasi *mobile* dapat mengirim permintaan ke *server*

untuk memperoleh data atau melakukan operasi seperti pemesanan makanan atau mengambil menu terbaru. Keuntungannya adalah aplikasi tetap terhubung dan berkomunikasi dengan server secara efisien dan fleksibel melalui internet, memperluas jangkauan dan fungsionalitas dari aplikasi pemesanan menu kafe berbasis *mobile* ini.



Gambar 2. Rancangan Arsitektur Sistem.

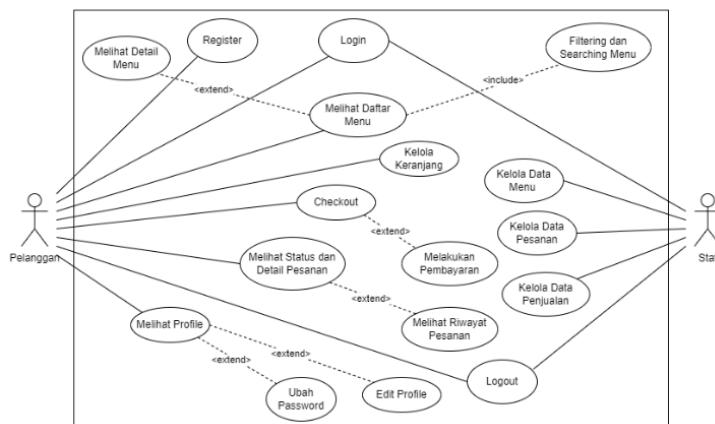
Dari Arsitektur Sistem pada Gambar 2 ini, API mengarah kepada dua aplikasi yang berbeda fungsionalitas dan aktornya, yaitu aplikasi pemesanan menu yang dioperasikan oleh pelanggan serta aplikasi pengelolaan menu dan pesanan yang dioperasikan oleh staf atau karyawan kafe. Masing-masing dari kedua aplikasi ini saling memperbarui data pada API sehingga terjalin relasi antar kedua aplikasi..

2.3.2 Rancangan Konseptual

Tahapan perancangan konseptual berarti proses desain sistem yang melibatkan proses atau aktivitas yang dapat memastikan sistem yang dibangun dapat berjalan dengan sesuai kebutuhan dan tujuan apa yang diharapkan [18]. Pengerjaan langkah ini diawali dengan perancangan *User Experience* menggunakan *tools* yang dapat digunakan untuk membuat UML. Kemudian untuk membuat *User Interface*, dengan membuat *prototype* antarmuka sistem beserta menu-menu yang direncanakan sesuai analisis kebutuhan.

a. Use Case Diagram

Use case mendefinisikan apa yang ditangani sistem dan komponennya dan menggunakan skenario. Hal tersebut merupakan deskripsi proses atau langkah yang menjelaskan apa yang akan dilakukan pengguna terhadap sistem dan sebaliknya. Dalam beberapa *use case* dihubungkan menggunakan notasi garis untuk menunjukkan relasinya dengan elemen lainnya [19].

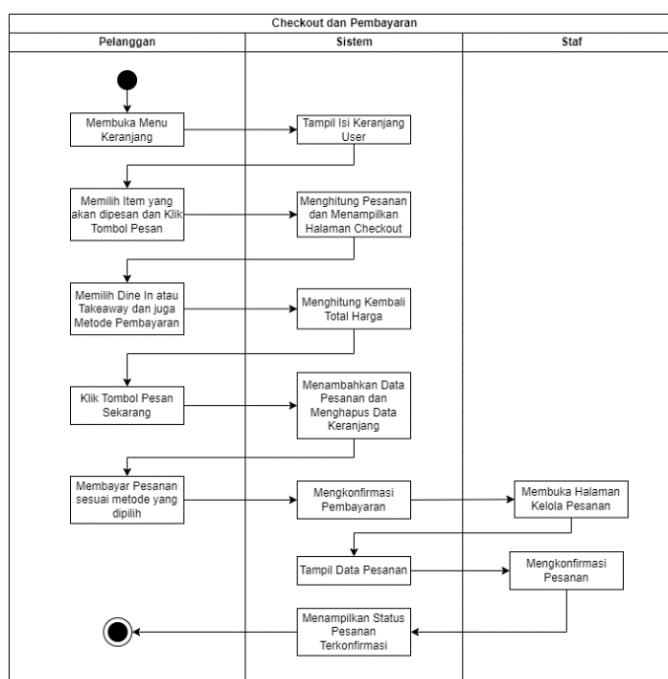


Gambar 3. Use Case Diagram.

Rancangan *Use Case Diagram* pada Gambar 3 di atas menjelaskan bahwa sistem berjalan melibatkan dua aktor yakni pelanggan dan staf karyawan. *Use Case* ini juga memberikan pemahaman bahwa ada lebih banyak aktivitas yang dapat dilakukan oleh aktor dengan *role* pelanggan daripada *role* staf. Selain dari perbedaan tersebut, terdapat juga tindakan yang dapat dilakukan kedua aktor ini, yakni *login* (diperlukan *register* bagi pelanggan), dan juga *logout*.

a. Activity Diagram

Activity diagram merupakan representasi alur kerja atau aktivitas dalam suatu sistem, termasuk proses di dalam program [20]. Dalam definisi yang lebih rinci, *Activity Diagram* mengacu pada diagram yang menggambarkan aktivitas pengguna sistem dari seluruh menu dalam sistem, menunjukkan berbagai aktivitas dalam sistem, cara kerja setiap fitur, dan bagaimana fitur berakhir [21].

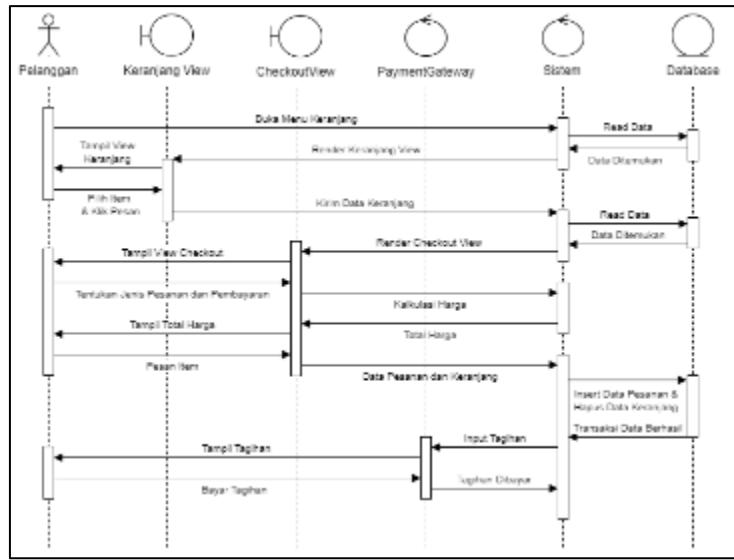


Gambar 4. Activity Diagram pada Proses Checkout dan Pembayaran.

Pada Gambar 4 menerangkan aktivitas *checkout* dan pembayaran. Fitur ini dapat diakses setelah pelanggan menambahkan sebuah item ke dalam keranjangnya. Dalam menu keranjang, tampil terlebih dahulu semua daftar item yang hendak dipesan. Pelanggan diminta untuk menyeleksi item mana saja yang akan di-*checkout* dan klik tombol “*Checkout*”. Pada Halaman *Checkout*, pelanggan diberi pilihan jenis pesanan antara *Dine In* dan *Takeaway*. Jenis Pesanan ini memengaruhi total harga yang harus dibayar pelanggan nantinya. Selain itu, terdapat pilihan lain yang menentukan metode pembayaran yang dapat dipilih pelanggan. Setelah Pelanggan konfirmasi pesanan dalam menu *checkout* ini, pelanggan klik tombol “*Order Now*”. Dalam alur berikutnya juga melibatkan seseorang staf dalam hal konfirmasi pesanan dan pembayaran.

a. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan interaksi antara objek-objek yang bekerja sama di dalam elemen-elemen suatu *class* [4]. Dalam definisi lain, *sequence diagram* menggambarkan objek dalam *use case* dengan menunjukkan alur pesan serta interaksi pengiriman dan penerimaan pesan di antara objek selama masa pakainya [22].

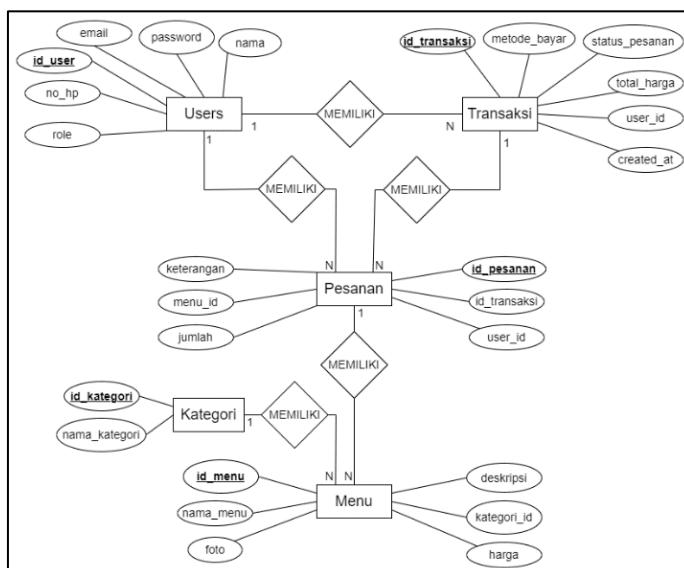


Gambar 5. Sequence Diagram pada Proses Checkout dan Pembayaran.

Gambar 5 yang berupa Sequence Diagram di atas menunjukkan alur pemesanan dan pembayaran dalam sistem aplikasi. Pelanggan memilih item dari keranjang, melakukan checkout, dan sistem menghitung total harga melalui Payment Gateway. Setelah pesanan dikonfirmasi, data disimpan di database dan tagihan ditampilkan untuk pembayaran. Proses berakhir ketika pembayaran berhasil dikonfirmasi.

a. Entity Relationship Diagram

ERD (*Entity Relationship Diagram*) adalah visualisasi atau diagram yang dipakai untuk mengilustrasikan koneksi antara entitas (objek) dalam suatu basis data. Dalam ERD, entitas direpresentasikan sebagai kotak yang memuat atribut-atribut yang berkaitan dengan entitas tersebut. Keterhubungan antara entitas-entitas ini ditampilkan dengan menggunakan garis yang menghubungkannya.



Gambar 6. Entity Relationship Diagram.

Diagram Hubungan Entitas yang telah tersaji pada Gambar 6 di atas menjelaskan bahwa setiap entitas yang ada pada *database* ini pasti memiliki sebuah *primary key* yang sudah ditandai dengan garis bawah. Selain itu, entitas diperbolehkan untuk memiliki satu atau lebih atribut di dalamnya. Beberapa entitas ini dihubungkan dengan relasi yang mengandung kardinalitas masing-masing entitas terhadap relasi. Entitas yang terdapat dalam sistem ini berjumlah lima entitas, diantaranya Data User, Pesanan, Menu, Kategori, dan Varian. Sedangkan hubungan antar entitas telah digambarkan yang meliputi satu pengguna dapat memiliki beberapa pesanan, satu pesanan dapat memiliki beberapa menu, satu menu dapat memiliki beberapa varian, serta satu kategori dapat memiliki beberapa menu.

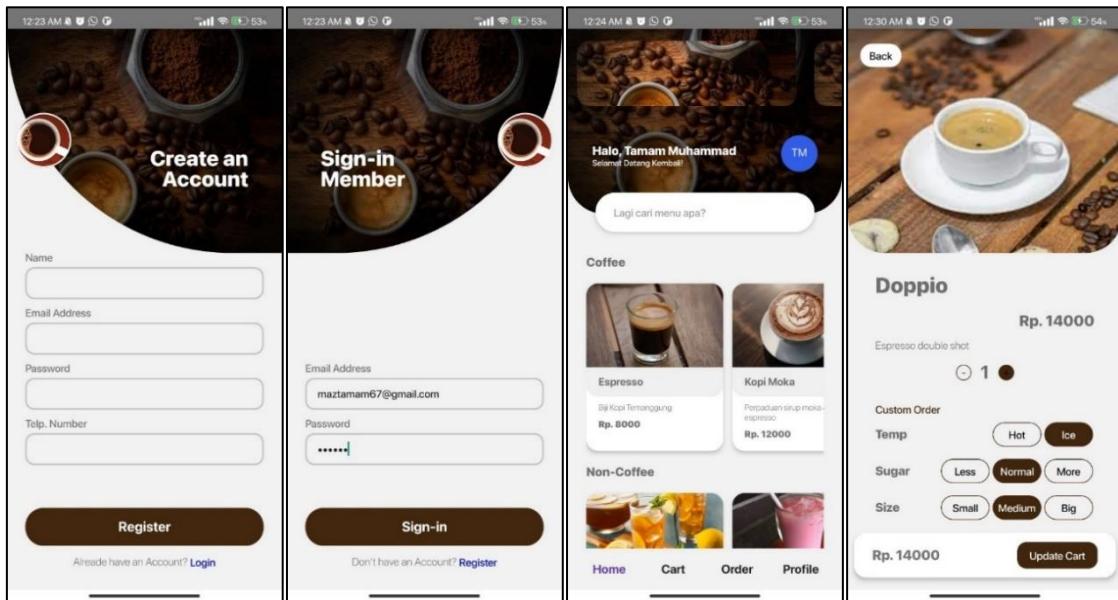
2.4. Pengujian

Pengujian dilakukan dengan tujuan untuk mengidentifikasi kesalahan atau kekeliruan dalam sistem [23]. Pengujian dilakukan menggunakan metode *black box*, yang berfokus pada evaluasi fungsionalitas tanpa melihat struktur internal [24]. Hasil pengujian menunjukkan sistem berfungsi sesuai dengan ekspektasi, berdasarkan kesesuaian antara hasil uji dan tujuan yang telah ditetapkan.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Implementasi program dilakukan dalam dua lingkungan, yaitu *Mobile* dan *Website*. Dalam sistem berbasis *mobile* pengimplementasian dilakukan menggunakan bahasa JavaScript dengan *library* React Native 0.6 yang berjalan di atas NodeJS. Sementara itu, untuk implementasi sistem berbasis *website* dibangun menggunakan bahasa PHP dengan bantuan dari *framework* CodeIgniter 4.

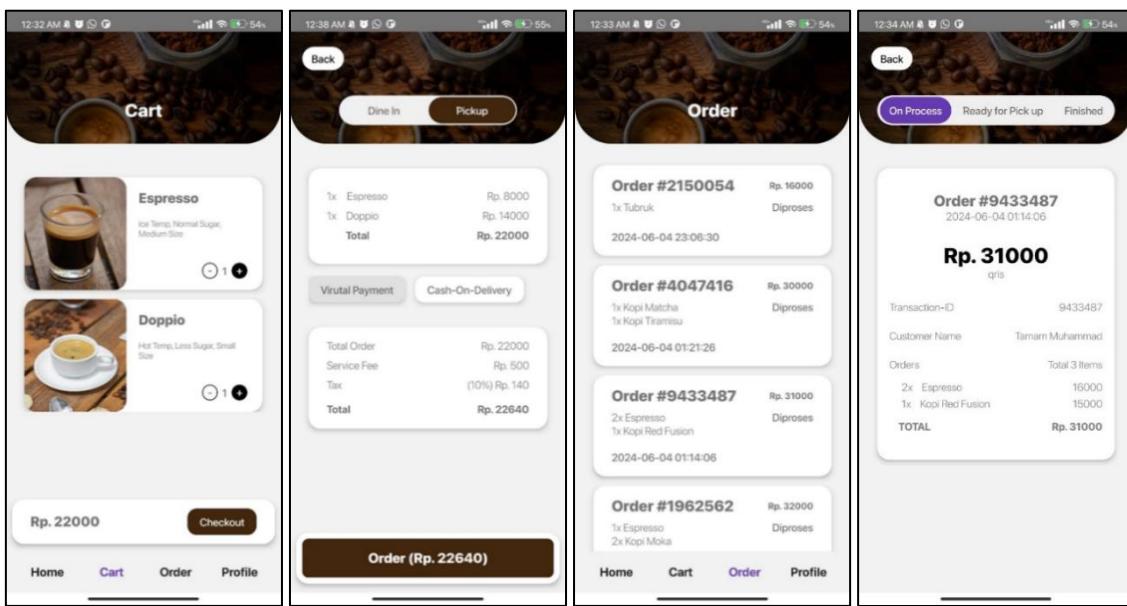


Gambar 7. Implementasi Halaman: (a) *Login*, (b) *Register*, (c) Utama, dan (d) Produk.

Dalam Gambar 7 menjelaskan terkait autentikasi pelanggan, yakni *login* dan *register* serta halaman utama dan halaman produk. *Login* Pelanggan meminta pelanggan untuk menginputkan data diri, mencakup email dan *password* ke dalam dua inputan. Inputan yang sesuai dengan data dalam tabel di *database* akan meloloskan pelanggan ke dalam Halaman Utama.

Kemudian untuk halaman *register* diberi validasi *required* sehingga wajib untuk diinputkan oleh pelanggan. Apabila salah satu saja tidak tervalidasi maka menampilkan pesan *field* yang tidak sesuai. Sementara itu, apabila registrasi berhasil, maka halaman mengarahkan ke antarmuka *login* dengan pesan keberhasilan.

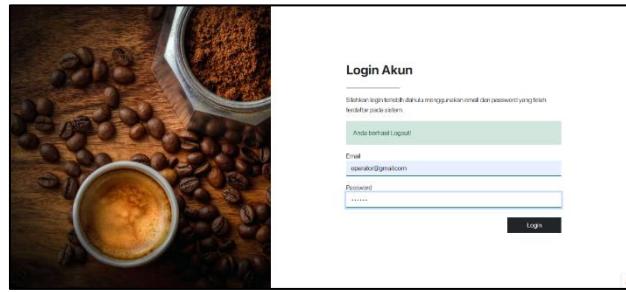
Kemudian Halaman utama memiliki *header* yang berisi pesan ucapan selamat datang dan kolom pencarian. Sementara itu, termasuk halaman utama ini menampilkan tab pada bagian bawah halaman yang berfungsi untuk menavigasi antar 4 halaman. Sedangkan untuk *main content*-nya berisi produk berdasarkan kategori. Menu-menu yang tampil berjejer ke kanan. Produk-produk ini memiliki halamannya sendiri apabila diklik oleh pelanggan. Halaman tersebut menampilkan data-data menu atau item secara lebih mendetail baik foto, nama, kategori, harga, sampai dengan deskripsi. Selain itu, juga menampilkan jumlah dan varian menu dari pesanan apabila telah pelanggan masukkan ke dalam keranjang. Di bagian bawahnya terdapat fungsi untuk menambah atau mengubah jumlah dan varian dari menu di dalam pesanan pelanggan.



Gambar 8. Implementasi Halaman: (a) Keranjang, (b) *Checkout*, (c) Pesanan, dan (d) Detail Pesanan.

Dalam Gambar 8, terdapat halaman keranjang, checkout, pesanan, dan detail pesanan. Di dalam halaman keranjang, konten utama berisi setiap item di dalam keranjang pelanggan. Selain itu, juga tampil jumlah item yang akan dipesan pelanggan dan dapat diatur sesuka hati. Pada bagian atas dari tab navigasi menampilkan total harga setiap pesanan serta tombol untuk melakukan *checkout* pesanan. Berikutnya untuk melakukan pemesanan dapat menggunakan dua metode pembayaran. Dua metode tersebut ialah *Virtual Payment* dan *Cash-On-Delivery* atau COD. Untuk pembayaran COD, transaksi memiliki status pesanan yang masih menunggu dibayar. Sementara untuk metode pembayaran *Virtual Payment* menerapkan fitur *payment gateway* dari Midtrans.

Sementara itu untuk mengetahui pesanan apa saja yang telah dibuat oleh pelanggan dapat beralih ke halaman pesanan Halaman ini berisi *header* dan konten utama yang terdiri dari setiap pesanan yang telah dibuat oleh pelanggan baik yang masih menunggu pembayaran sampai dengan yang selesai. Pada setiap pesanan yang telah tampil pada halaman pesanan memiliki antarmuka sendiri untuk menampilkan data pesanan secara lebih rinci. Dalam halaman ini *header* berisi status pesanan. Sementara konten utama berisi setiap data-data penting dari sebuah pesanan.



Gambar 9. Implementasi Halaman Login Staf.

Selanjutnya, pada Gambar 9 menjelaskan autentikasi pada bagian sistem kelola berbasis website yang ditujukan kepada staf karyawan. Antarmuka dari halaman ini berisi sebuah form *login* dengan dua inputan yaitu *email* dan *password*. Apabila staf berhasil diidentifikasi oleh sistem melalui form *login* tersebut, maka sistem akan menampilkan halaman *dashboard* untuk kelola menu dan kelola pesanan.

| # | Nama Menu | Kategori | Harga | Option |
|---|-----------------|----------|-------|--------|
| 1 | Espresso | Coffee | 8000 | |
| 2 | Kopi Moka | Coffee | 12000 | |
| 3 | Kopi Red Fusion | Coffee | 15000 | |
| 4 | Kopi Matcha | Coffee | 15000 | |
| 5 | Kopi Tiramisu | Coffee | 15000 | |
| 6 | Kopi Gula Aren | Coffee | 15000 | |
| 7 | Kopi Gula Jawa | Coffee | 15000 | |
| 8 | Kopi Sarien | Coffee | 15000 | |

| # | Tanggal | Nama | Pesanan | Metode Bayar | Status Pesanan | Option |
|---|---------------------|----------------|------------------------------------|--------------|------------------|--------|
| 1 | 2024-05-06 00:39:47 | Tanam Muhammad | 1x Espresso | qrIs | Diproses | |
| 2 | 2024-05-05 08:41:17 | Guest | 1x Kopi Red Fusion | COD | Menunggu Dibayar | |
| 3 | 2024-05-05 08:40:02 | Guest | 1x Kopi Red Fusion 2x Espresso | COD | Menunggu Dibayar | |
| 4 | 2024-05-04 23:06:30 | Tanam Muhammad | 1x Tubruk | COD | Diproses | |
| 5 | 2024-05-04 01:21:26 | Tanam Muhammad | 1x Kopi Matcha 1x Kopi Tiramisu | qrIs | Diproses | |
| 6 | 2024-05-04 01:14:08 | Tanam Muhammad | 2x Espresso | qrIs | Diproses | |
| 7 | 2024-05-04 01:14:08 | Tanam Muhammad | 1x Kopi Red Fusion | qrIs | Diproses | |

Gambar 10. Implementasi Halaman Kelola: (a) Menu dan (b) Pesanan.

Kemudian di dalam antarmuka kelola menu dan pesanan pada Gambar 10 ini terbagi menjadi tiga bagian, pertama ialah *sidebar* yang berisi menu apa saja yang dapat diakses oleh staf, kedua ada *header* yang berguna untuk menampilkan tanggal saat ini dan menyembunyikan *sidebar*, dan yang ketiga ialah konten utama yaitu tabel yang berisi data singkat dari setiap menu atau pesanan. Dari tabel ini, setiap data atau barisnya memiliki beberapa tombol, diantaranya: (1) tambah, untuk menampilkan form tambah yang berupa *modal*, (2) *edit*, juga akan menampilkan form berupa *modal* yang memiliki fungsi untuk mengubah data, (3) *delete*, untuk menghapus satu data yang dimaksud, serta (4) sebuah tombol untuk mengelola setiap pesanan yang tercatat sistem.

| # | Nama Menu | Kategori | Harga |
|---|-----------------|----------|-------|
| 1 | Espresso | Coffee | 8000 |
| 2 | Kopi Moka | Coffee | 12000 |
| 3 | Kopi Red Fusion | Coffee | 15000 |
| 4 | Kopi Matcha | Coffee | 15000 |
| 5 | Kopi Tiramisu | Coffee | 15000 |
| 6 | Kopi Gula Aren | Coffee | 15000 |
| 7 | Kopi Gula Jawa | Coffee | 15000 |
| 8 | Kopi Sarien | Coffee | 15000 |

Gambar 11. Rancangan Arsitektur Sistem.

Setiap *modal* yang terdapat dalam sistem ini akan tampak seperti Gambar 11. Pada gambar tersebut, *Modal* merupakan form tambah yang berisi beberapa inputan guna untuk data-data menu yang lebih lengkap. Sementara pada opsi *edit* yang memunculkan *modal* yang berisi form *edit* data yang menyerupai form tambah data.

3.2 Pengujian Sistem

3.2.1 Pengujian Aplikasi Pemesanan

Dalam penelitian ini, sistem diuji menggunakan metode *User Acceptance Test* pada 20 orang responden sebagai pelanggan untuk mengevaluasi sejauh mana hasil perancangan sistem sesuai dengan analisis kebutuhan yang telah disusun sebelumnya. Hasil pengujian ini terdokumentasi dalam Tabel 1 dan berfokus pada aplikasi *mobile* yang digunakan oleh pelanggan.

Tabel 1. Hasil Pengujian Aplikasi Pemesanan Pelanggan

| Kelas Uji | Skenario Uji | Jumlah Keberhasilan | Persentase Keberhasilan |
|---|--|---------------------|-------------------------|
| Autentikasi | Register Pelanggan | 18 | 90% |
| | Login Pelanggan | 19 | 95% |
| Menu Utama | Input Pencarian | 18 | 90% |
| | Lihat Halaman Produk | 20 | 100% |
| | Akses Menu Cart | 20 | 100% |
| | Akses Menu Order | 20 | 100% |
| | Akses Menu Profile | 20. | 100% |
| | Akses Menu Home | 20 | 100% |
| Kelola Keranjang di Halaman Produk | Tambah Produk (+) | 20 | 100% |
| | Kurangi dan Hapus Produk (-) | 20 | 100% |
| | Tambah Produk ke Keranjang | 18 | 90% |
| | Kurangi atau Hapus Produk dari Keranjang | 18 | 90% |
| | Kembali dari Halaman Produk | 20 | 100% |
| Kelola Keranjang di Halaman Keranjang | Lihat Halaman Produk dari Keranjang | 20 | 100% |
| | Tambah Produk (+) | 20 | 100% |
| | Kurangi dan Hapus Produk (-) | 19 | 95% |
| | Klik tombol “checkout” | 20 | 100% |
| Cek Status dan Riwayat Pesanan di Halaman Pesanan | Lihat Detail Pesanan | 20 | 100% |
| | Kembali dari Detail Pesanan | 20 | 100% |
| Data Diri dan Akun | Update Data Profile | 17 | 85% |
| | Logout Akun | 19 | 95% |
| Checkout | Order COD | 20 | 100% |
| | Order via Midtrans | 18 | 90% |
| | Pembayaran via Midtrans | 18 | 90% |
| | Kembali ke Menu Utama | 20 | 100% |

Berdasarkan hasil pengujian UAT terhadap 25 skenario utama pada Tabel 1 di atas, ditemukan bahwa tidak semua skenario berjalan sempurna. Misalnya, pada proses pembayaran dengan metode virtual payment (Midtrans), tingkat keberhasilan hanya mencapai 90% akibat beberapa kendala seperti kegagalan koneksi internet dan timeout API. Adapun rata-rata keberhasilan dari seluruh skenario uji adalah **96,4%**, yang menunjukkan bahwa sistem telah berfungsi dengan baik, meskipun masih terdapat area yang perlu diperbaiki.

3.2.2 Pengujian Sistem Kelola

Selain menguji aplikasi pemesanan yang digunakan oleh pelanggan, pengujian *User Acceptance Test* juga diterapkan pada sistem pengelolaan menu dan pesanan yang digunakan oleh 10 staf karyawan kafe. Hasil pengujian ini terdokumentasi dalam Tabel 2.

Tabel 2. Hasil Pengujian Sistem Pengelolaan Pesanan

| Kelas Uji | Skenario Uji | Jumlah Keberhasilan | Persentase Keberhasilan |
|-------------|--|---------------------|-------------------------|
| Autentikasi | Login Staf | 10 | 100% |
| Menu Utama | Menyembunyikan dan Menampilkan Sidebar | 8 | 80% |

| | | | |
|----------------|--|----|------|
| | Akses Menu Kelola Menu | 10 | 100% |
| | Akses Menu Kelola Pesanan | 10 | 100% |
| | Logout Akun | 10 | 100% |
| Kelola Menu | Akses Form Tambah Menu | 7 | 70% |
| | Tambah Menu | 7 | 70% |
| | Akses Form Edit Menu | 7 | 70% |
| | Edit Menu | 7 | 70% |
| | Hapus Salah Satu Menu | 10 | 100% |
| Kelola Pesanan | Akses Form Tambah Pesanan | 7 | 70% |
| | Tambah Pesanan | 7 | 70% |
| | Proses Pesanan yang "Menunggu Dibayar" | 9 | 90% |
| | Siapkan Pesanan yang "Diproses" | 9 | 90% |
| | Selesaikan Pesanan yang "Siap Diambil" | 9 | 90% |
| | Hapus Pesanan | 10 | 100% |

Dari 15 skenario UAT dalam Tabel 2 tersebut, sebagian besar berjalan baik dengan rata-rata keberhasilan **86,7%**. Login staf, navigasi menu, dan penghapusan data berhasil 100%. Kendala muncul pada form tambah/edit menu dan pesanan (70%) serta sidebar (80%), yang perlu perbaikan. Hal ini dapat disebabkan oleh konflik atau kesalahan yang terjadi pada *front end* sistem sehingga depedensi kadang tidak dimuat dengan benar.

3.2.3 Pengujian Kepuasan Pengguna

Untuk mengukur tingkat kepuasan dan kemudahan penggunaan aplikasi pemesanan dari sisi pelanggan, dilakukan pengujian menggunakan metode *System Usability Scale* (SUS) terhadap 20 pelanggan dan 10 staf karyawan. Setiap responden diminta menilai 7 pernyataan standar dengan skala penilaian 1–5. Berikut merupakan nilai rata-rata setiap pertanyaan yang terdokumentasikan dalam Tabel 3.

Tabel 3. Hasil Pengujian Kepuasan Pengguna

| No | Pernyataan | Jenis Pertanyaan | Skor Rata-rata |
|----|--|------------------|----------------|
| 1 | Saya merasa sistem ini akan sering saya gunakan | Positif | 3.7 |
| 2 | Saya merasa sistem ini terlalu rumit dan membingungkan | Negatif | 1.1 |
| 3 | Saya merasa sistem ini mudah digunakan | Positif | 4.0 |
| 4 | Saya membutuhkan bantuan teknis untuk menggunakan sistem ini | Negatif | 1.0 |
| 5 | Fitur-fitur dalam sistem ini berjalan dengan baik dan terintegrasi | Positif | 3.9 |
| 6 | Saya merasa banyak inkonsistensi dalam sistem ini | Negatif | 1.2 |
| 7 | Saya dapat belajar menggunakan sistem ini dengan cepat | Positif | 4.1 |
| 8 | Saya perlu memahami banyak hal sebelum menggunakan sistem ini | Negatif | 1.4 |
| 9 | Saya merasa nyaman dan tidak ragu saat menggunakan sistem ini | Positif | 3.8 |
| 10 | Sistem ini terasa tidak logis dan tidak terstruktur | Negatif | 0.9 |

Dari Pengujian SUS pada Tabel 3 tersebut, didapatkan nilai **84,75** dengan cara mendapatkan rata-rata dari setiap pernyataan positif dan negatif yang kemudian keduanya dijumlahkan dan dikalikan dengan nilai 2,5. Berdasarkan rentang standar SUS (0–100), skor ini menunjukkan **hasil yang cukup baik**, yang berada di atas rata-rata. Sistem ini diterima dengan baik oleh pengguna dan menunjukkan bahwa sebagian besar pengguna merasa nyaman menggunakan sistem tersebut. Namun, masih ada ruang untuk beberapa perbaikan untuk mencapai skor yang lebih tinggi. Selain dari 7 pernyataan itu, juga terdapat satu pertanyaan yang meminta pendapat pelanggan terhadap waktu tunggu yang mereka butuhkan untuk mendapatkan pesanan mereka. Dan dari satu pertanyaan tersebut, didapatkan waktu tunggu yang dibutuhkan paling lama ialah **5-7 menit** dengan persentase waktu tunggu berkisar **50%**. Hal ini menunjukkan bahwa proses pemesanan dan penyajian menu pada sistem berlangsung dengan relatif cepat.

Meskipun aplikasi pemesanan berbasis mobile dan sistem kelola pesanan berbasis web memiliki kelebihan yang telah disebutkan, terdapat beberapa potensi masalah yang perlu diperhatikan. Ketergantungan terhadap koneksi internet menjadi kendala utama, di mana hal tersebut dapat menyebabkan proses pemesanan dan pembayaran terganggu atau gagal. Selain itu, downtime server dapat menghambat transaksi. Masalah kompatibilitas perangkat juga dapat muncul, terutama pada ponsel dengan spesifikasi rendah atau sistem operasi lama yang tidak mendukung aplikasi. Oleh karena itu, diperlukan strategi mitigasi seperti validasi data, caching, penanganan error secara real-time, serta pengujian kompatibilitas lintas perangkat dan sistem.

4. KESIMPULAN

Hasil penelitian ini menunjukkan bahwa aplikasi telah dirancang dengan dua platform berbeda, yaitu aplikasi *mobile* untuk pelanggan dan *website* untuk staf kafe. Aplikasi *mobile* dikembangkan menggunakan *framework* React Native dengan JavaScript, sedangkan sistem *website* memanfaatkan PHP dengan *framework* CodeIgniter 4. Aplikasi pemesanan menu memungkinkan pelanggan untuk melakukan registrasi, *login*, menampilkan menu, dan memperbarui profil. Selain itu, pelanggan dapat menambahkan pesanan ke keranjang, melakukan *checkout* dengan metode pembayaran beragam, serta melihat total harga dari setiap pesanan yang otomatis tercatat dalam daftar pesanan. Di sisi staf, sistem berbasis *website* menyediakan fitur untuk menampilkan, menambah, mengubah, dan menghapus menu atau pesanan secara manual. Staf juga dapat memperbarui status pesanan, sehingga memudahkan mengelola operasional kafe.

Kemudian berdasarkan pengujian UAT pada aplikasi pemesanan berbasis mobile dan sistem kelola berbasis web, kedua sistem tersebut memperoleh nilai yang cukup baik, sehingga dapat berfungsi sebagaimana mestinya. Selanjutnya, pengujian kepuasan pengguna menggunakan System Usability Scale (SUS) menunjukkan hasil yang lebih tinggi dari rata-rata, yang menandakan bahwa mayoritas pengguna merasa nyaman dan mudah menggunakan aplikasi. Selain itu, sistem berhasil mengurangi waktu tunggu pelanggan menjadi sekitar 5-7 menit, dengan persentase pengurangan waktu tunggu mencapai 50%, yang menunjukkan peningkatan signifikan dalam efisiensi pemrosesan pesanan dan penyajian menu dibandingkan sistem sebelumnya.

DAFTAR PUSTAKA

- [1] A. N. Yusril, I. Larasati, and P. al Zukri, “Systematic Literature Review Analisis Metode Agile dalam Pengembangan Aplikasi Mobile,” *Sistemasi: Jurnal Sistem Informasi*, vol. 10, no. 2, pp. 369–380, 2021, doi: 10.32520/stmsi.v10i2.1237.
- [2] A. Novianti, “Literature Review: Analisis Metodologi Dan Bidang Penerapan Dalam Perancangan Aplikasi Mobile,” 2022.
- [3] H. A. Mumtahana, S. Nita, and A. W. Tito, “Pemanfaatan Web E-Commerce untuk Meningkatkan Strategi Pemasaran,” *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 1, pp. 6–15, Jun. 2017, doi: 10.23917/khif.v3i1.3309.
- [4] T. Arianti, A. Fa’izi, S. Adam, and M. Wulandari, “Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language),” *Jurnal Ilmiah Komputer Terapan dan Informasi*, vol. 1, no. 1, pp. 19–25, 2022.
- [5] P. A. K. Wardani, F. R. Reinaldy, and I. P. Tyas, “Analisis Faktor-Faktor Keputusan Pembelian Makanan Dengan Menggunakan Jasa ShopeeFood,” *Literasi: Jurnal Kajian Keislaman Multi-Perspektif*, vol. 2, no. 1, pp. 163–192, 2021.
- [6] R. B. Hadiprakoso, *Pemrograman Berorientasi Objek: Teori dan Implementasi dengan Java*, 1st ed. Tangerang: RBH, 2021.
- [7] Mardalius, F. Dristyan, and A. Syafnur, “Sistem Informasi Geografis Penyebaran Covid-19 di Kabupaten Asahan menggunakan Framework Codeigniter 4,” *Journal of Science and Social Research*, vol. 4, no. 3, pp. 347–351, Oct. 2021, doi: 10.54314/jssr.v4i3.714.

- [8] Edy, Ferdiansyah, W. Pramusinto, and S. Waluyo, "Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 1, no. 3, pp. 106–112, 2017, doi: 10.29207/resti.v3i2.860.
- [9] S. N. Yanti and E. Rihyanti, "Penerapan Rest API untuk Sistem Informasi Film Secara Daring," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 1, p. 195, Mar. 2021, doi: 10.32493/informatika.v6i1.10033.
- [10] A. M. Kasiba and Sularno, "Perancangan Aplikasi Pemesanan Nomor Antrian di Blackbeard Barbershop Berbasis Android," *Journal of Informatics and Busisnes*, vol. 1, no. 1, pp. 1–5, 2023.
- [11] Durahim, Rusdiyanto, M. Mahfud, and S. M. Prasetyo, "Implementasi Belanja Bahan Bangunan Menggunakan Aplikasi ‘Berbasis Web,’" *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, no. 3, pp. 655–665, Apr. 2023.
- [12] D. Krisbiantoro and Sarmini, "Penerapan Payment Gateway dan Antrian Pada Pemesanan Kuliner di Purwokerto Berbasis Android," *Jusikom: Jurnal Sistem Komputer Musirawas*, vol. 6, no. 1, pp. 73–83, Jun. 2021, doi: 10.32767/jusikom.v6i1.1151.
- [13] Husein, H. Akbar, M. N. Hisyam, and M. B. Legowo, "Payment Gateway on E-Canteen Website Application," *International Journal of Progressive Sciences and Technologies (IJPSAT)*, vol. 12, no. 2, pp. 108–112, 2019.
- [14] T. M. M. Puspitasari and D. Maulina, "Implementasi Payment Gateway Menggunakan Midtrans Pada Marketplace Travnesia.Com," *Journal Mobile and Forensics (MF)*, vol. 1, no. 1, pp. 22–29, Sep. 2019, doi: 10.12928/mf.v1i1.997.
- [15] F. R. Pratama, N. Santoso, and L. Fanani, "Pengembangan Aplikasi E-Commerce Menggunakan Payment Gateway Midtrans," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 4, pp. 1133–1140, Apr. 2020.
- [16] I. Murni, A. Sari, B. R. Lubis, and A. Ikhwan, "Pengamanan Pesan Rahasia dengan Algoritma Vigenere Cipher Menggunakan PHP," *Journal on Education*, vol. 5, no. 2, pp. 3466–3476, 2023, doi: 10.31004/joe.v5i2.1027.
- [17] R. S. Wandela, "Rancang Bangun Sistem Informasi Administrasi Pelanggan Fitness Berbasis Web," 2019.
- [18] D. Purnomo, "Model Prototyping Pada Pengembangan Sistem Informasi," *JIMP: Jurnal Informatika Merdeka Pasuruan*, vol. 2, no. 2, pp. 54–61, Aug. 2017, doi: 10.37438/jimp.v2i2.67.
- [19] L. Setiyani, "Implementasi Cybersecurity pada Operasional Organisasi," in Prosiding Seminar Nasional: Inovasi & Adopsi Teknologi, LPPM STMIK ROSMA, Sep. 2021, pp. 246–260.
- [20] A. Voutama and E. Novalia, "Perancangan Aplikasi M-Magazine Berbasis Android Sebagai Sarana Mading Sekolah Menengah Atas," *Jurnal Tekno Kompak*, vol. 15, no. 1, pp. 104–115, 2021, doi: 10.33365/jtk.v15i1.920.
- [21] R. Sari, F. Hamidy, and Suaidah, "Sistem Informasi Akuntansi Perhitungan Harga Pokok Produksi Pada Konveksi Sjm Bandar Lampung," *Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 1, pp. 65–73, 2021.
- [22] Ismanto and P. T. Pungkasanti, "Aplikasi Perpustakaan Berbasis Unified Modelling Language," *Information Science and Library*, vol. 2, no. 2, pp. 63–72, Dec. 2021, doi: 10.26623/jisl.v2i2.4349.
- [23] T. Armanda and A. D. Putra, "Rancang Bangun Aplikasi E-Commerce Untuk Usaha Penjualan Helm," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 1, no. 1, pp. 17–24, 2020, doi: 10.33365/jatika.v1i1.145.
- [24] Supriadi, T. B. Pandia, K. F. Siahaan, W. Hendratno, and Nurhayati, "Penggunaan React Native dalam Pengembangan Aplikasi Media Pembelajaran Interaktif Perakitan Komputer untuk Meningkatkan Prestasi Belajar Siswa," *JTePen: Jurnal Teknologi Dalam Pendidikan*, no. 1, pp. 4–8, Aug. 2023.