

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi mengalami kemajuan pesat setiap tahunnya. Hal ini dibuktikan dengan jumlah pengguna internet di Indonesia yang meningkat sebesar 202,6 juta pada tahun 2021. Jumlah ini meningkat 15,5 persen atau 27 juta orang dibandingkan Januari 2020. Perkembangan teknologi berdampak pada setiap aspek kehidupan, termasuk dunia kuliner (Vydrio & Wafi, 2021). Salah satu dampak signifikan dari perkembangan ini adalah peningkatan permintaan akan solusi pemesanan makanan yang efisien dan inovatif. Food court, sebagai pusat kuliner yang menawarkan berbagai makanan dan minuman memiliki potensi besar untuk mengadopsi inovasi teknologi tersebut. Dengan mengimplementasikan teknologi informasi, *food court* dapat meningkatkan pengalaman pelanggan, mempercepat proses pemesanan dan pembayaran, serta mengoptimalkan manajemen operasional mereka.

Food court atau pujasera merupakan sebuah area di dalam plaza dengan banyak penjual makanan yang berdekatan serta menyediakan sebuah area untuk umum yang dipergunakan oleh pengunjung sebagai tempat makan (Dhanny & Rama, 2019). Di lingkungan kampus Universitas Hasanuddin, Kuliner Dapur Nusantara (Kudapan) BNI menjadi salah satu pusat destinasi kuliner mahasiswa yang mengadopsi konsep *food court*. Kudapan BNI menyediakan berbagai kios makanan yang menawarkan berbagai pilihan kuliner untuk mahasiswa dan staf universitas. Lokasinya yang strategis di area kelas Perkuliahan Bersama (PB), Perpustakaan Pusat, dan dekat dengan Gedung Rektorat Unhas menjadikannya pilihan utama untuk makan siang dan istirahat. Meskipun Kudapan BNI memiliki area yang cukup luas, terdiri dari dua lantai, hal tersebut tidak cukup untuk menampung padatnya mahasiswa yang memenuhi tempat ini, terutama pada saat jam makan siang. Kepadatan ini sering kali mengakibatkan ketidaknyamanan pelanggan dan pelayanan yang kurang optimal dari penjual.

Pada tahun 2020, Kudapan BNI Universitas Hasanuddin sempat menerapkan sistem pembayaran yang baru. Sistem pembayaran baru Kudapan BNI Universitas Hasanuddin pada saat itu dengan menggunakan satu kasir untuk digunakan bersama seluruh penjual. Sistem satu kasir ini ternyata diterapkan karena adanya kerugian yang dialami oleh pihak unhas yang mencapai 200 juta. Namun kebijakan ini menuai banyak keresahan dari mahasiswa yang kerap berbelanja di Kudapan BNI, mereka menganggap hal ini sangat tidak efisien dan membuang waktu karena antrian yang diakibatkan oleh kebijakan ini cukup panjang. Tetapi, nyatanya hal ini disebabkan oleh mahasiswa sendiri yang menyebabkan kerugian pihak unhas, karena dalam sistem pembayaran dahulu langsung pada penjual, banyak mahasiswa yang lupa membayar atau bahkan sengaja untuk tidak membayar setelah makan, sehingga penjual merasa rugi. (Sain & Akram, 2020). Sayangnya, sistem ini kemudian dihentikan kurang lebih enam bulan diterapkan.

Berdasarkan uraian yang ada, terdapat beberapa masalah krusial yang perlu diatasi di Kudapan BNI Universitas Hasanuddin. Peningkatan jumlah pengunjung dan kompleksitas operasional, diperlukan solusi inovatif yang dapat mengoptimalkan pelayanan, meningkatkan efisiensi transaksi, dan menjaga integritas sistem pembayaran. Rancang bangun aplikasi pemesanan kuliner berbasis *website* diharapkan

akan menjadi solusi tepat untuk meningkatkan efisiensi dan mengatasi masalah yang telah diuraikan. Aplikasi ini dapat memberikan sejumlah fitur yang mencakup, sistem pemesanan, sistem pembayaran, dan manajemen antrian.

Solusi ini diharapkan dapat meningkatkan efisiensi operasional Kudapan BNI, mengurangi antrian, dan menciptakan lingkungan yang lebih nyaman bagi pengguna. Selain itu, dengan pilihan pembayaran yang fleksibel, aplikasi ini dapat membantu mencegah kerugian akibat lupa atau ketidakjujuran dalam pembayaran. Dengan demikian, rancang bangun aplikasi ini diharapkan dapat memberikan solusi untuk meningkatkan kualitas pelayanan di Kudapan BNI Universitas Hasanuddin.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat dirumuskan permasalahan yang terjadi sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi Pemesanan Pada *Food Court* Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin Berbasis Web?
2. Bagaimana efektivitas dari aplikasi Pemesanan Pada *Food Court* Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin Berbasis Web?

1.3 Batasan Penelitian

Berikut ini merupakan beberapa batasan dalam penelitian ini.

1. Aplikasi ini tidak mencakup fitur pembayaran yang terintegrasi oleh pihak ketiga. Jadi, diasumsikan saldo yang terdapat pada *user* sudah ditentukan jumlahnya dan proses *topup* saldo hanyalah simulasi.

1.4 Tujuan Penelitian

Berdasarkan latar belakang yang telah diuraikan diatas, dapat dirumuskan permasalahan yang terjadi sebagai berikut:

1. Merancang dan membangun aplikasi pemesanan pada *food court* Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin berbasis web yang interaktif dan responsif menggunakan teknologi web modern.
2. Untuk mengetahui efektivitas penggunaan aplikasi pemesanan pada *food court* Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin berbasis web.

1.5 Landasan Teori

1.5.1 Kudapan BNI Universitas Hasanuddin

Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin merupakan salah satu tempat makan yang memiliki konsep *Food Court*. Kudapan BNI bertempat di area kelas Perkuliahan Bersama (PB), Perpustakaan Pusat, dan tidak begitu jauh dari Gedung Rektorat Universitas Hasanuddin. Karena lokasinya yang sangat strategis, Kudapan BNI Unhas menjadi salah satu destinasi kuliner favorit mahasiswa Unhas. Sebelum jadi seperti sekarang ini, Kudapan BNI Unhas memiliki Sejarah yang cukup panjang.



Gambar 1. Kudapan BNI Universitas Hasanuddin

Awalnya, tempat makan ini didirikan pada 1987 dengan sebutan Jasa Boga (Jasbog). Sejak awal dibuka, menu yang disajikan begitu beragam. Mulai dari nasi campur, bakso super, ayam goreng, sampai buras dan ketupat. Meskipun memiliki beragam jenis makanan dan letaknya yang sangat strategis, Jasbog dinilai kurang cocok dengan saku rata-rata mahasiswa kala itu. Harganya berkisar pada 150 hingga 400 rupiah pada 1987.

Setelah berdiri bertahun-tahun, Jasbog kemudian direnovasi besar besaran pada 13 Februari 2015. Pembangunan tersebut bekerja sama dengan pihak BNI yang mengusulkan dana hibah sehingga biaya dan fasilitas Pembangunan akan ditanggung penuh oleh BNI. Setelah berbulan-bulan pelaksanaan renovasi, kantin Jasbog Kembali dibuka dengan perubahan nama menjadi Kuliner Dapur Nusantara (Kudapan) BNI. Sistem perdagangan di dalam pun turut berganti. Hanya ada 25 pedagang asli Jasbog yang tetap berjualan, ditambah lima pedagang dharma wanita. Pada tahun 2018, Kudapan BNI Kembali direnovasi dan itu merupakan renovasi terakhir yang dilakukan sampai saat skripsi ini ditulis.

Dinamika yang terjadi pada Kudapan BNI bukan hanya terdapat pada renovasi infrastruktur, akan tetapi Kudapan BNI juga pernah mengalami masalah pada proses transaksi. Pada tahun 2020, Kudapan BNI pernah mengeluarkan kebijakan sistem satu kasir. Sistemnya yaitu melakukan pembayaran dengan menggunakan satu kasir untuk digunakan bersama oleh setiap penjual. Penerapan sistem ini dikarenakan adanya kerugian yang dialami pihak unhas yang mencapai 200 juta rupiah. Kerugian ini ternyata disebabkan oleh mahasiswa itu sendiri yang terkadang lupa atau bahkan dengan sengaja tidak membayar makanan yang telah dimakan. (Sain & Akram, 2020)

Adapun alur transaksinya yaitu dimulai dengan pembeli mengambil lembar menu di kedai dan memilih menu yang ingin dipesan. Lalu pembeli menyerahkan lembaran

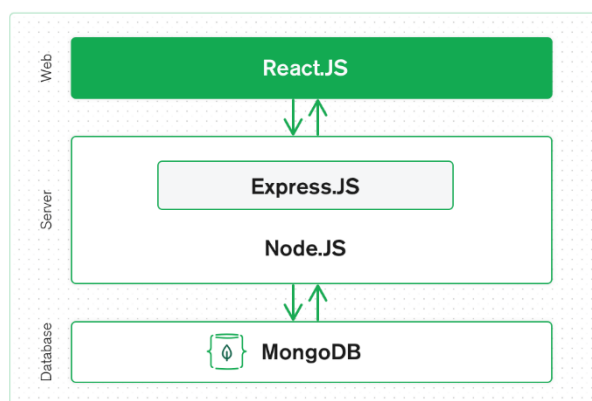
tersebut ke kasir untuk membayar dan mengambil struk pembayaran. Kemudian pembeli kembali lagi ke kedai untuk menyerahkan struk pembayaran. Makanan akan disiapkan dan diantar ke meja pembeli atau dapat diambil sendiri di kedai.

Namun, sistem ini hanya bertahan 6 bulan saja setelah diberlakukan karena keluhan dari mahasiswa yang menganggap sistem ini menyebabkan antrian yang panjang karena pembeli seperti harus antri dua kali untuk membeli makanan. (Vanessa, 2023)

1.5.2 MERN Stack

MERN *stack* adalah *framework* populer yang mengintegrasikan teknologi *javascript* untuk memudahkan pembuatan aplikasi *web full-stack*. MERN adalah singkatan dari kumpulan teknologi yang membentuk *stack* yaitu MongoDB, Express.js, React.js, dan Node.js. Pengembang web menemukan bahwa pendekatan yang berpusat pada *javascript* menyediakan lingkungan pengembangan yang konsisten untuk membangun aplikasi *web* yang dinamis, sehingga mengurangi kebutuhan untuk beralih konteks di antara bahasa pemrograman yang berbeda (Erickson, 2024)

Arsitektur MERN membantu pengembang dengan mudah membangun arsitektur tiga tingkat (*front end*, *back end*, *database*) yang sepenuhnya menggunakan *javascript* dan JSON.



Gambar 2. Arsitektur MERN

Pada bagian *database*, MongoDB yang merupakan *database* dokumen berbasis NoSQL, memungkinkan pengimanan data dalam *format* JSON yang fleksibel dan dapat di skalakan. MongoDB mendukung kinerja yang tinggi dengan berbagai fitur seperti *indexing*, *aggregation*, *load balancing*, menjadikannya pilihan ideal untuk menangani volume data besar dalam aplikasi *modern*. (Silalahi & Wahyudi , 2018)

Untuk lapisan *server*, Express.js yang merupakan sebuah *framework* Node.js yang digunakan untuk membuat aplikasi web yang dinamis. Express.js dibuat oleh TJ Holowaychuk yang pertama kali dirilis pada tahun 2010. Framework ini sangat populer digunakan dalam pengembangan aplikasi *web modern* karena kemampuan yang luar dan kemudahan penggunaannya. (Holowaychuk, 2022)

Express.js dilengkapi dengan berbagai fitur yang dapat memudahkan pengembangan aplikasi web. Beberapa fitur tersebut seperti :

- Routing: Express.js memungkinkan pengembangan untuk membuat routing yang kompleks dan fleksibel dengan menggunakan metode seperti 'app.get()', 'app.post()', dan 'app.put'. (Express.js, 2022)
- Middleware: Express.js memungkinkan pengembangan untuk membuat *middleware* yang dapat digunakan untuk melakukan berbagai tugas seperti autentikasi, validasi data, dan *caching*. (Express.js, 2022)
- Templating Engine: Express.js dapat digunakan dengan berbagai templating engine seperti EJS, Pug, dan Jade untuk membuat aplikasi yang dinamis. (Express.js, 2022)
- Support untuk Database: Express.js dapat digunakan dengan berbagai database seperti MySQL, PostgreSQL, dan MongoDB untuk menyimpan data. (Express.js, 2022)

Node.js sendiri adalah lingkungan *runtime* JavaScript yang dibangun di atas mesin V8 milik Chrome. Ini memungkinkan pengembang untuk menjalankan kode JavaScript di sisi server, menjadikannya alat yang sangat efisien untuk membangun aplikasi web yang cepat dan skalabel. Node.js mendukung model pemrograman non-blokir yang memungkinkan penanganan banyak koneksi secara bersamaan tanpa memerlukan thread tambahan. (GeeksforGeeks, 2024)

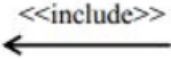
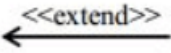
Untuk sisi frontend, React.js atau React yang dikembangkan oleh Meta (*facebook*), merupakan library *javascript* yang sangat populer yang digunakan untuk pengembangan aplikasi *web* dan *mobile*. React.js hadir dan mengubah cara aplikasi web dibuat. Pada react.js, setiap komponen berurusan dengan *state*-nya dan membuatnya menjadi UI. Dengan ide komponen daripada layout dalam *javascript*, banyak informasi yang dapat diteruskan ke aplikasi tanpa banyak kesulitan dan dengan demikian membantu dalam menjaga state dari DOM. (Rawat & Mahajan, 2020)

1.5.3 Use case diagram

Use case diagram adalah *diagram* UML yang menyajikan pandangan Tingkat tinggi bagaimana sistem digunakan seperti yang dilihat dari perspektif orang luar (aktor) selama fase desain. *Use case diagram* digunakan untuk menspesifikasikan perilaku sistem yang diimplementasikan. Jadi *use case diagram* terdiri dari aktor, *use case*, dan relasinya. (R. Ganesh, 2020)

Tabel 1. Komponen *use case diagram*

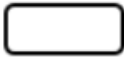
SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berinteraksi dengan <i>use case</i>
	<i>Use case</i>	Abstraksi dan interaksi antara sistem dan aktor

	<i>Association</i>	Abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i>
	<i>Generalization</i>	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>
	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> yang lain
	<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

1.5.4 Activity diagram

Activity diagram merupakan pemodelan yang menggambarkan sebuah sistem kerja dari sebuah objek atau sebuah sistem, sebuah *activity diagram* dengan sebuah alur secara tersruktur proses kerja dari *use case* yang sedang diproses dari titik awal sampai titik akhir, setiap aktivitas digambarkan notasi-notasi sesuai fungsinya (Aliman & Wilianti, 2021). Komponen-komponen pada *use case diagram* di antaranya sebagai berikut:

Tabel 2. Komponen pada *use case diagram*

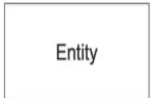
SIMBOL	NAMA	KETERANGAN
	<i>Start Point</i>	Bagaimana objek dibentuk atau diawali
	<i>End Point</i>	Bagaimana objek diakhiri
	<i>Activities</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Decision</i>	Menggambarkan suatu Keputusan atau tindakan yang harus diambil pada kondisi tertentu

1.5.5 Entity Relationship Diagram

Entity Relationship Diagram adalah cara di mana pemodelan *database* yang digunakan adalah skema konseptual yang mewakili semacam model data semantik yang sistematis. Sistem yang digunakan *entity relationship* adalah *database* relational dengan

pendekatan *top-down*. *Diagram* yang digunakan ialah suatu gambaran model entity-relationship yang disebut dengan entity-relationship *diagram*, ER *diagram* atau ERD. (Pulungan et al, 2023)

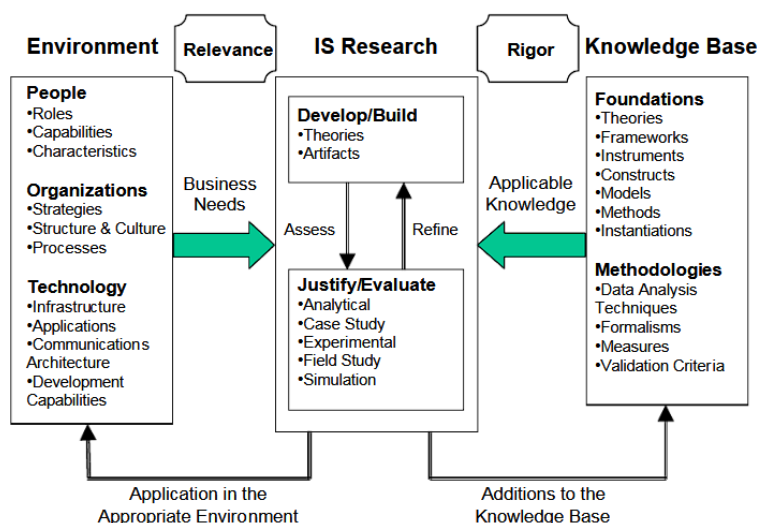
Tabel 3. Komponen pada *activity diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Entity</i>	Merupakan suatu simbol untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.
	<i>Attribute</i>	Merupakan suatu simbol yang menjelaskan karakteristik suatu entitas dan juga relasinya.
	<i>Strong Relationship</i>	Menggambarkan hubungan beberapa entitas berdasarkan fakta pada suatu lingkungan.
	<i>Connection</i>	Menggambarkan keterkaitan antara simbol berupa garis penghubung.

1.5.6 Design Science Research

Design Science Research (DSR) adalah pendekatan penelitian yang berfokus pada pengembangan dan evaluasi artefak untuk menyelesaikan masalah praktis di bidang sistem informasi. DSR bertujuan untuk menghasilkan pengetahuan baru melalui inovasi dalam desain artefak teknologi, yang dapat berupa model, metode, atau sistem

Dalam literatur penelitian, ruang lingkup penelitian sistem informasi dibagi menjadi manusia, organisasi, dan teknologi. Dari ruang lingkup ini dirumuskan secara menyeluruh kaitannya dengan permasalahan dan fenomena yang muncul, kemudian dari ruang lingkup itu dibentuklah sebuah pengembangan baik itu berupa teori ataupun artefak. (Hevner, et al., 2004)



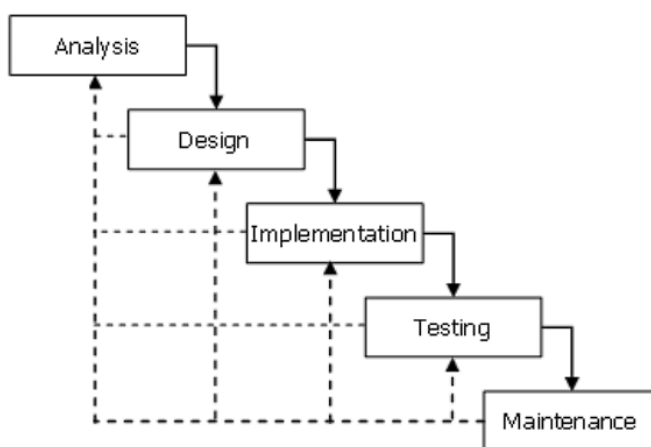
Gambar 3. Information systems research framework

Pada Gambar 3. Dijelaskan bahwa sistem informasi terbagi atas 3 ruang lingkup yang menjadi penyusun dari sebuah sistem informasi. Ruang lingkup tersebut menjadi dasar permasalahan yang menjadi gagasan untuk melakukan sebuah penelitian terhadap sistem informasi dalam pengembangan sebuah teori atau artefak.

1.5.7 Metode Waterfall

Menurut (Wahid, 2020), metode *waterfall* atau metode air terjun sering dinamakan siklus hidup klasik (*classic life cycle*). Nama model sebenarnya adalah “*Linear Sequential Model*” yang menggambarkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna. Metode ini memiliki tahapan-tahapan yang berurutan, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan tahap pendukung.

Tahapan dari metode *waterfall* dapat dilihat pada gambar berikut :



Gambar 4. Tahapan dari metode waterfall

1. *Requirement* (Kebutuhan)
pengumpulan dan analisis persyaratan untuk mengetahui dan memahami komponen perangkat lunak yang memenuhi kebutuhan pengguna.
2. *Design* (Desain)
Proses perancangan pembuatan perangkat lunak, termasuk arsitektur perangkat lunak dan representasi antarmuka perangkat lunak
3. *Implementation* (Implementasi)
Proses implementasi suatu perancangan pada perangkat lunak berupa program komputer yang saling terintegrasi, sesuai dengan perancangan yang dibuat pada tahap sebelumnya.
4. *Testing* (Pengujian)
Proses pengujian bertujuan untuk memverifikasi apakah sistem yang dihasilkan memenuhi kebutuhan pengguna dan meminimalkan kesalahan yang ada.
5. *Maintenance* (Pemeliharaan)
Proses pemeliharaan dapat berupa pembaruan perangkat lunak ketika beberapa komponen baru dirasa perlu, atau proses perbaikan *bug* yang tidak terdeteksi pada proses pengujian sebelumnya.

1.5.8 Blackbox Testing

Blackbox testing merupakan metode pengujian yang berfokus pada spesifikasi fungsional perangkat lunak, penguji dapat menentukan serangkaian kondisi *input* dan melakukan pengujian terhadap spesifikasi fungsional program. Proses pengujian *blackbox* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap fiturnya. Pengujian ini diperlukan untuk mengetahui program tersebut sesuai dengan kebutuhan atau tidak. (Shadiq, et al., 2021)

1.5.9 User Acceptance Test (UAT)

User Acceptance Test (UAT) merupakan metode yang digunakan untuk mengevaluasi *response* pengguna terkait sistem yang telah dikembangkan. Salah satu teknik yang sering digunakan dalam UAT yaitu penggunaan angket skala *likert* dengan lima atau tujuh pilihan. Angket tersebut digunakan untuk mendapatkan masukan dari pengguna terkait pengalaman mereka dalam menggunakan sistem yang telah dikembangkan. (Erlangga & Pramana, 2023). *User Acceptance Test* umumnya dilakukan oleh klien atau pengguna akhir, biasanya tidak fokus pada identifikasi masalah sederhana seperti kesalahan ejaan, maupun cacat pada *software* (Suprpto, 2021).

Indikator keberhasilan yang diukur dalam penelitian *User Acceptance Testing* (UAT) dapat bervariasi tergantung pada konteks dan tujuan pengujian. Salah satu indikator utama adalah kepuasan pengguna, yang mencerminkan seberapa puas pengguna terhadap sistem yang diuji. Tingkat kepuasan ini sering diukur melalui angket atau wawancara, dan dapat menjadi tolok ukur penting untuk menilai penerimaan teknologi informasi oleh pengguna (Supriatna, 2021). Indikator lain yang tidak kalah penting adalah kemudahan penggunaan, yang menilai seberapa mudah pengguna dapat berinteraksi dengan sistem. Penelitian oleh Jogi Abraham (2021) menunjukkan bahwa

hasil UAT pada sistem menunjukkan bahwa penyusunan menu serta isi tiap-tiap menu pada website harus sesuai dengan kebutuhan pengguna. Selain itu, efektivitas sistem juga diukur, yang mencakup seberapa baik sistem memenuhi tujuan yang ditetapkan, termasuk kecepatan dan efisiensi dalam menyelesaikan tugas tertentu.

Pada penelitian ini, penulis akan menggunakan skala *likert* yang terdiri dari empat penilaian yaitu Tidak Setuju (STS), Cukup Setuju (TS), Setuju (S), dan sangat setuju (SS) kepada tiga kategori *user* yaitu admin, vendor, dan customer.

1.5.10 Load testing

Load testing adalah metode pengujian perangkat lunak yang dirancang untuk mengevaluasi kinerja daya tahan suatu sistem atau aplikasi di bawah kondisi beban kerja berat atau kondisi beban maksimum. Tujuan utama dari *load testing* adalah untuk menentukan batas kinerja suatu sistem, menemukan penyebab kegagalan, dan menilai seberapa baik sistem dapat mempertahankan kinerja di bawah beban ekstrim. *Load testing* membantu mengidentifikasi batas kinerja website, mengungkapkan masalah yang mungkin muncul pada tingkat beban yang tinggi, serta memberikan wawasan tentang performa dan keandalan website (Raweyai, 2024). Pada penelitian ini, penulis melakukan *load testing* dengan bantuan *software* yaitu *Apache Jmeter*TM.

1.5.11 Penelitian Terkait

Dalam menyelesaikan tugas akhir ini, penulis menggunakan beberapa jurnal penelitian sebagai referensi. Jurnal-jurnal tersebut penulis jadikan sebagai perbandingan dalam menyusun tugas akhir ini, dengan mengambil jurnal yang memiliki keterkaitan dengan topik yang sama, serta tujuan yang sejalan.

Penelitian yang berjudul “Perancangan Sistem Informasi Pemesanan Makanan dan Minuman Berbasis Web Pada Restoran Lamongan Cahaya” yang dilakukan oleh Wahyu Pudyawardana pada tahun 2023. Penelitian ini bertujuan untuk mengembangkan suatu sistem informasi pemesanan makanan dan minuman berbasis web yang dapat mengatasi permasalahan dalam pencatatan dan pengelolaan data serta transaksi pemesanan pada restoran di Desa Gandasari. Sistem ini diharapkan dapat meningkatkan efisiensi dan akurasi dalam proses pemesanan, pengantaran, dan pengelolaan data restoran. Peneliti menggunakan metode pengumpulan data melalui observasi, studi pustaka, dan wawancara. Peneliti Melakukan observasi langsung di tempat penelitian yaitu Restoran Lamongan Cahaya, untuk memperoleh informasi mengenai proses kegiatan yang ada di restoran. Informasi tersebut kemudian di terapkan pada tahap perancangan sistem, dan juga implementasi sistem. Hasil dari penelitian tersebut adalah sistem pemesanan menu berbasis web dipasang pada komputer server sehingga dapat digunakan dengan baik. Sistem pemesanan menu terdiri dari *back office* yang digunakan oleh staff administrator yang bertanggung jawab untuk mengelola sistem, *front office* yang digunakan oleh pelanggan untuk memesan menu yang ditawarkan oleh Restoran Lamongan Cahaya.

Penelitian selanjutnya yang berjudul “Sistem Informasi Pemesanan menu Makanan Berbasis Web (Studi Kasus: Restoran Bukit Randu Bandara)” oleh Tuti Handayani, Ipung Gunawan, dan Rohmat Taufiq pada tahun 2020. Penelitian ini

bertujuan untuk mengembangkan sistem *informasi* pemesanan makanan dan minuman secara *online* berbasis *web* yang dapat membantu restoran dalam memperluas area pemasaran dan meningkatkan jumlah pelanggan. Metode yang digunakan dalam penelitian ini adalah metode *waterfall*. terdiri dari *analysis* (analisis), *design* (desain), *implementation* (implementasi), *testing* (pengujian), dan *maintenance* (pemeliharaan). Berdasarkan hasil pengujian dengan menggunakan metode *black box testing*, diketahui bahwa seluruh pengujian bisa dilakukan sehingga dikatakan pengujian tersebut valid dan menghasilkan aplikasi web yang sesuai dengan kebutuhan *user*.

Penelitian selanjutnya yang berjudul “Sistem Informasi Aplikasi Pemesanan Makanan Restoran Berbasis Web Menggunakan Metode *Agile Development*” oleh Vendi Blessing Gulo, Agung Triayudi, dan Agus Iskandar. Pada tahun 2023. Tujuan penelitian ini adalah untuk mengembangkan sistem *informasi* pemesanan makanan restoran berbasis *web* yang efisien dan dapat mempermudah administrasi dalam menangani data pemesanan makanan. Metode yang digunakan oleh penelitian ini adalah metode *Agile Development*. Metode ini merupakan konsep pengembangan yang menekankan kecepatan dalam pengerjaan, kemampuan merespons perubahan yang diinginkan oleh klien dengan cepat, dan melibatkan secara aktif setiap pihak yang terlibat dalam proses pengembangan. Hasil dari penelitian tersebut, pengembangan sistem *informasi* pemesanan makanan restoran berbasis web menggunakan metode *Agile Development* dapat memberikan solusi yang efisien dan mempermudah administrasi dalam menangani data pemesanan makanan. Metode *Agile Development* memberikan fleksibilitas dan adaptabilitas dalam menghadapi perubahan kebutuhan pengguna.

BAB II
METODE PENELITIAN

2.1 Waktu dan Tempat Penelitian

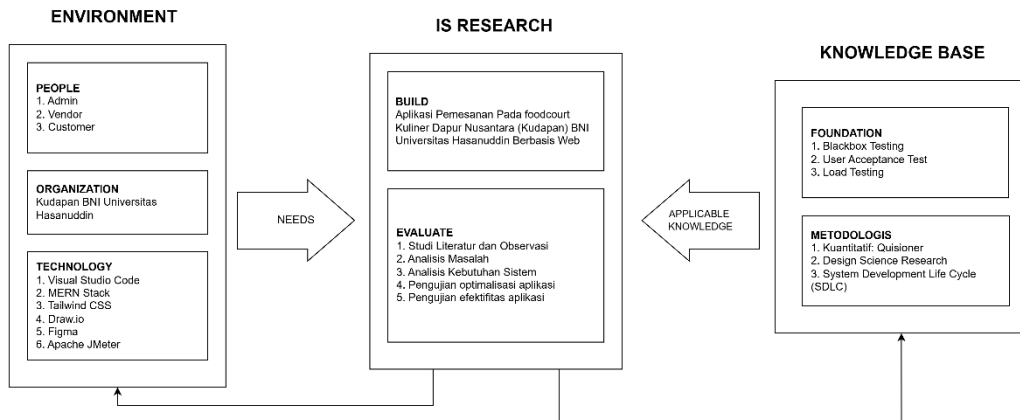
Penelitian ini dilaksanakan pada bulan April sampai dengan September 2024, dan bertempat di Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin.

Tabel 4. Jadwal penelitian

Tahapan Penelitian		2024																											
		April				Mei				Juni				Juli				Agustus				September				Oktober			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Observasi																													
Studi Literatur																													
Pengembangan Sistem	Requirement																												
	Design																												
	Implementation																												
	Testing																												
	Maintenance																												

2.2 Design Science Research

Design Science Research yang diterapkan pada penelitian ini bertujuan untuk menghasilkan artefak untuk memudahkan pemesanan produk dan manajemen antrian pada Kuliner Dapur Nusantara (Kudapan BNI) Universitas Hasanuddin yang didasarkan pada tiga komponen utama, yaitu *Environment* (Lingkungan), *IS Research* (Penelitian Sistem Informasi), dan *Knowledge Base* (Basis pengetahuan).



Gambar 5. Design science research diagram

Pada aspek *environment* terdiri dari *people* (orang), *organization* (organisasi), dan *technology* (teknologi). *People* (orang) mencakup admin, vendor, dan juga customer dari Kudapan BNI Universitas Hasanuddin. *Organization* (organisasi) yang terlibat yaitu Vendor Kudapan BNI Universitas Hasanuddin. Kemudian, *technology* (teknologi) yang digunakan dalam penelitian yaitu *visual studio code*, MERN Stack, Tailwind CSS, Draw.io, Figma, dan Apache Jmeter sebagai *tools* untuk melakukan *load testing*.

Pada aspek *IS Research*, terdiri dari *build* (membangun), dan *Evaluate* (evaluasi). Pada tahap *build*, dilakukan pembuatan artefak Aplikasi Pemesanan Pada Food Court Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin. Pada tahap *evaluate* dilakukan dari studi literatur dan juga observasi untuk mengumpulkan informasi. Setelah itu, dilakukan analisis masalah dan analisis kebutuhan sistem untuk mengetahui masalah berdasarkan informasi yang didapatkan dari tahap sebelumnya dan menentukan kebutuhan sistem yang akan dikembangkan. Setelah artefak berhasil dirancang, tahapan selanjutnya adalah menilai kinerja artefak dengan melakukan pengujian optimalisasi dan efektivitas untuk menilai optimalitas dan keefektifan aplikasi.

Knowledge Base yang mendukung penelitian ini terdiri dari *foundation* (dasar) dan *metodologi*. *Foundation* mencakup *blackbox testing*, *User Acceptance Test* (UAT), dan *load testing*. Kemudian metodologi yang digunakan pada penelitian ini yaitu kuantitatif: kuisioner, *design science research*, dan System Development Life Cycle (SDLC) model *waterfall*.

Penerapan metode *design science research* dalam pengembangan aplikasi pemesanan pada *food court* Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin diharapkan berguna dalam meningkatkan kemudahan dalam pemesanan dan manajemen antrian pada Kudapan BNI Universitas Hasanuddin.

2.3 Metode Pengumpulan Data

Dalam penelitian ini, dilakukan beberapa tahap pengumpulan data untuk menunjang penelitian. Pemilihan metode pengumpulan data sangat penting karena data yang diperoleh akan digunakan sebagai dasar dalam pengembangan sistem. Pada penelitian ini, dilakukan beberapa metode pengumpulan data diantaranya adalah sebagai berikut :

1. Observasi

Observasi adalah metode pengumpulan data dengan cara mengamati langsung objek penelitian di lapangan. Dalam penelitian ini, Observasi dilakukan untuk memahami secara mendalam kondisi lingkungan yang menjadi objek penelitian. Observasi dilakukan pada Kuliner Dapur Nusantara (Kudapan) BNI Universitas Hasanuddin.

2. Studi Literatur

Studi literatur merupakan metode pengumpulan data dengan cara menelaah berbagai sumber tertulis yang relevan dengan topik penelitian. Sumber-sumber tersebut bisa berupa buku, jurnal, artikel ilmiah, laporan penelitian, serta dokumen-dokumen lain yang terkait. Pada penelitian ini, penulis melakukan studi literatur dengan mencari informasi yang tersedia baik itu dalam bentuk jurnal, buku, maupun informasi dari *website* yang relevan dengan penelitian.

2.4 Metode Pengembangan Sistem

Pada penelitian ini, metode pengembangan sistem yang digunakan adalah metode *waterfall*. Metode ini terdiri dari beberapa tahap yang harus dilalui secara berurutan, tahap tersebut adalah sebagai berikut :

1. *Requirements* (Kebutuhan)

Pada tahapan ini, peneliti melakukan analisis kebutuhan apa saja yang akan diperlukan oleh sistem sehingga dipahami apa saja fitur yang diharapkan ada pada sistem.

2. *Design* (Desain)

Pada tahapan ini, peneliti merancang *use case diagram* dan *activity diagram* untuk memberikan gambaran umum bagaimana nantinya sistem yang akan dibuat bekerja. Kemudian peneliti juga membuat rancangan *Entity Relationship Diagram* untuk memvisualisasikan struktur data dan aliran informasi dalam sistem. Lalu selanjutnya peneliti membuat rancangan *user interface* agar memberikan gambaran tampilan visual sistem dengan lebih jelas.

3. *Implementation* (Implementasi)

Pada tahapan ini, peneliti mulai mengimplementasikan rancangan sebelumnya menjadi suatu aplikasi yang berbasis web dengan menggunakan teknologi MERN Stack yang terdiri dari MongoDB sebagai database, Express.js untuk sisi *backend*, React.js untuk sisi *frontend*, dan Node.js sebagai *runtime*.

4. *Testing* (Pengujian)

Pada tahapan ini, peneliti melakukan pengujian dengan metode *black box testing* untuk mengetahui apakah sistem yang telah dibangun sudah berfungsi secara optimal. Setelah itu, peneliti melakukan pengujian *UAT* untuk mengetahui efektivitas penggunaan sistem berdasarkan penilaian *user*. Kemudian, peneliti

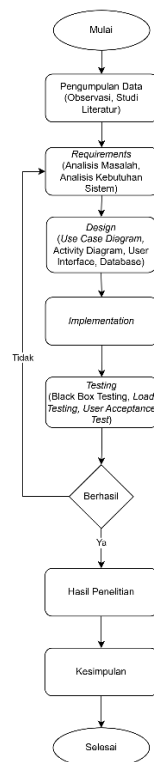
juga melakukan *performance testing* dengan *load testing* menggunakan jmeter untuk mengetahui performa aplikasi jika digunakan oleh banyak *user* dalam waktu yang hampir bersamaan.

5. *Maintenance* (Pemeliharaan)

Pada tahapan ini, peneliti melakukan pemeliharaan sistem secara berkala agar sistem bisa tetap berjalan secara baik dan optimal.

2.5 Tahapan Penelitian

Tahapan penelitian dimulai dengan proses pengumpulan data seperti observasi dan juga studi literatur. Kemudian dilanjut ke tahapan pengembangan sistem dengan menggunakan metode pengembangan *waterfall* yaitu dimulai dari tahap *requirements* atau analisis kebutuhan, kemudian tahapan *design* atau desain seperti proses perancangan *use case diagram*, *activity diagram*, dan juga perancangan *user interface*. Selanjutnya setelah itu, dilanjut ke tahapan *implementation* atau implementasi yaitu mengimplementasikan rancangan yang ada menjadi sebuah aplikasi web. Setelah itu, dilanjut lagi ke tahap *testing* atau pengujian menggunakan metode *blackbox testing* untuk memastikan aplikasi berjalan dengan baik. Jika sistem tidak berjalan sesuai dengan fungsi dan kebutuhan, maka tahapan akan diulang kembali ke tahap *requirements*. Sedangkan apabila sistem sudah berjalan sesuai dengan kebutuhan, maka penelitian sudah bisa dikatakan selesai. Adapun gambaran penelitian dalam bentuk *flow chart* bisa dilihat pada Gambar 6.



Gambar 6. Tahapan penelitian

2.6 Analisis Pengembangan Sistem

2.6.1 Analisis Masalah

Berdasarkan hasil observasi, ditemukan masalah berupa antrian pembeli dari Kudapan BNI Universitas Hasanuddin cukup padat apalagi pada saat jam makan siang. Kepadatan terjadi di area depan beberapa *tenant* di Kudapan BNI sehingga kerap kali menghalangi jalan pengunjung lain yang ingin membeli di *tenant* yang lain atau yang ingin mengelilingi Kudapan untuk mencari produk yang mereka cari. Masalah lain yang ditemukan adalah kerap kali ada pengunjung yang membeli produk akan tetapi lupa atau bahkan sengaja tidak membayar produk yang mereka beli karena proses pembayaran dan pencatatan masih dilakukan secara manual. Oleh karena itu, diharapkan aplikasi *website* ini bisa menjadi solusi yang tepat untuk mengatasi masalah-masalah tersebut.

2.6.2 Analisis Kebutuhan Sistem

Dalam penelitian ini, dibutuhkan perangkat lunak dan perangkat keras sebagai *tools* atau alat yang dapat mendukung penelitian, serta pengguna sistem (*user*) sebagai bahan analisis kebutuhan dalam perancangan sistem. Adapun kebutuhan sistem dalam penelitian ini yaitu :

1. Perangkat Lunak

- a. *Draw.io*
- b. Figma
- c. HTML
- d. CSS
- e. Javascript
- f. React.js
- g. Express.js
- h. Node.js
- i. MongoDB
- j. *Visual Studio Code*
- k. Google Chrome

2. Perangkat Keras

Dalam penelitian ini, peneliti menggunakan Laptop Lenovo Ideapad Slim 3 dengan spesifikasi *processor* AMD Ryzen 3 5300U, RAM 8 GB, dan SSD 512 GB.

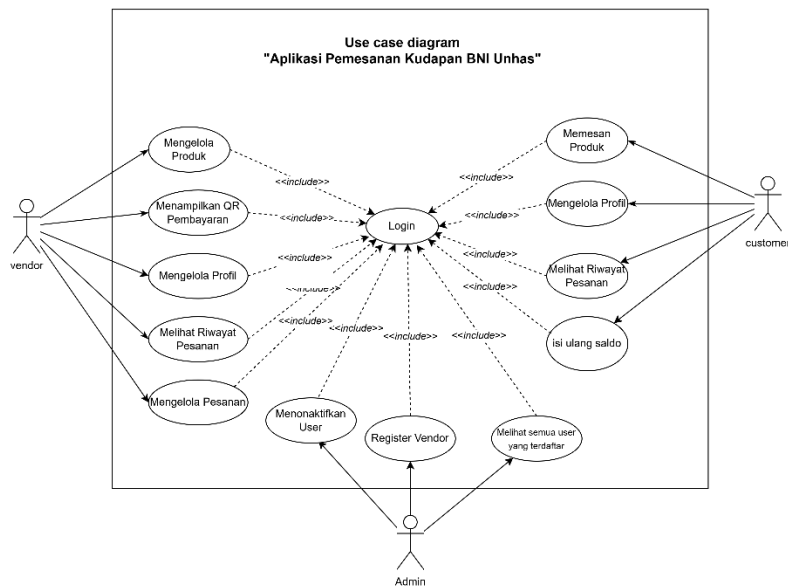
3. Pengguna Sistem

- a. Admin
Admin merupakan *user* yang berperan untuk mengelola data *vendor* dan *customer* yang terdaftar pada aplikasi web. Admin juga dapat melakukan registrasi atau mendaftarkan *user* dengan role *vendor*.
- b. *Vendor*
Vendor merupakan *user* yang berperan sebagai orang yang memiliki toko atau *tenant* di Kudapan BNI. *Vendor* dapat melakukan penambahan produk yang akan dijual dan menentukan harga produk, *vendor* juga dapat melihat Riwayat pesanan and memberikan *QR Code* pembayaran kepada pelanggan yang ingin membayar produk yang dibeli.
- c. *Customer*
Customer berperan sebagai *user* yang membeli produk dari *vendor*. *Customer* dapat memilih produk yang akan dibeli dan memasukkannya ke dalam keranjang

untuk kemudian melakukan *checkout* atau *order* untuk produk yang ada di dalam keranjang.

2.7 Perancangan Sistem

Dalam perancangan sistem, peneliti menggunakan *use case diagram* untuk menggambarkan aktivitas yang dapat dilakukan oleh *user* di dalam sistem. Terdapat tiga aktor yang memiliki masing-masing aktivitas atau perilaku yang berbeda, yaitu , *admin*, *vendor*, dan juga *customer* dapat dilihat pada Gambar 7. yang menunjukkan *use case diagram*.



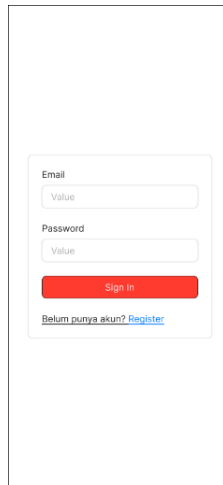
Gambar 7. *Use Case Diagram*

2.8 Rancangan User Interface

Rancangan *User Interface* (UI) adalah bentuk rancangan tampilan visual yang berhubungan langsung dengan para pengguna agar mendapatkan Gambaran mengenai fitur-fitur yang akan dihadirkan pada sistem. Adapun rancangan *user interface* pada penelitian ini yaitu :

1. Halaman Login

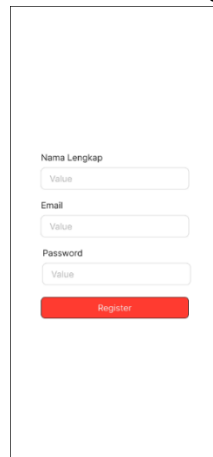
Halaman login adalah halaman yang mengharuskan *user* harus memasukkan *email* dan *password* terlebih dahulu untuk bisa mengakses fitur *website*.

A login form interface with a white background. It contains two input fields: 'Email' and 'Password', each with a 'Value' placeholder. Below the 'Password' field is a red button labeled 'Sign in'. At the bottom, there is a link that says 'Belum punya akun? Register'.

Gambar 8. Halaman login

2. Halaman Register

Halaman Register adalah halaman untuk membuat akun untuk *user* yang belum terdaftar atau belum mempunyai akun untuk mengakses fitur *website*.

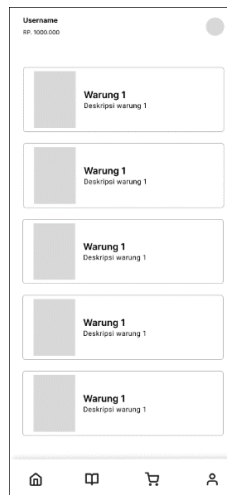
A register form interface with a white background. It contains three input fields: 'Nama Lengkap', 'Email', and 'Password', each with a 'Value' placeholder. Below the 'Password' field is a red button labeled 'Register'.

Gambar 9. Halaman register

3. Halaman Customer

a. Halaman Daftar Toko | *Homepage (Customer)*

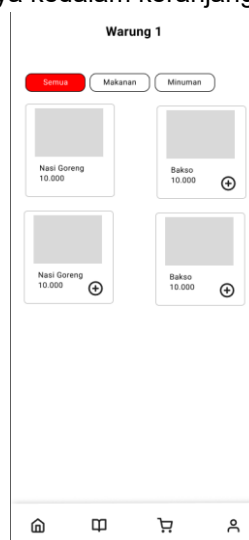
Halaman daftar Toko atau *Homepage* adalah halaman yang pertama kali muncul ketika *user* telah melakukan tahap *login*. Halaman ini berisi list-list dari toko yang terdaftar pada website yang kemudian produknya bisa di *order* oleh *user*.



Gambar 10. Halaman daftar toko

b. Halaman Daftar Produk (*Customer*)

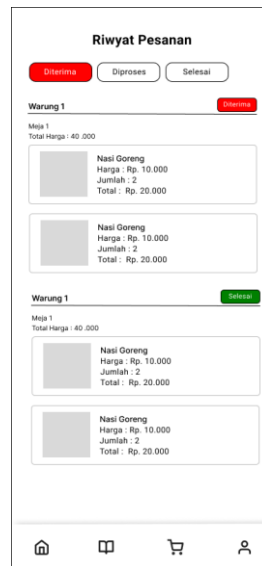
Halaman Daftar Produk merupakan halaman *list* dari produk yang dimiliki oleh sebuah toko yang dipilih *user*. Produk-produk tersebut dapat diorder oleh *user* dengan cara memasukkannya kedalam keranjang terlebih dahulu.



Gambar 11. Halaman daftar produk

c. Halaman Riwayat Pesanan (*Customer*)

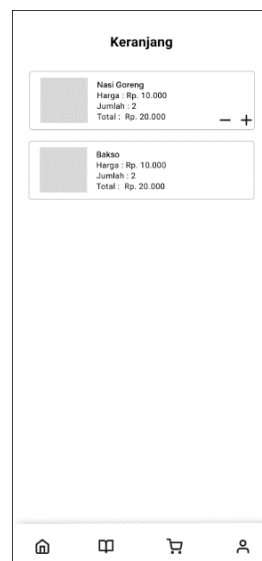
Halaman Riwayat Pesanan merupakan halaman untuk melihat riwayat pemesanan produk dari *user*. *User* dapat melihat riwayat dan juga status dari pesannya apakah baru diterima, diproses, dan selesai.



Gambar 12. Halaman riwayat pesanan

d. Halaman Keranjang (*Customer*)

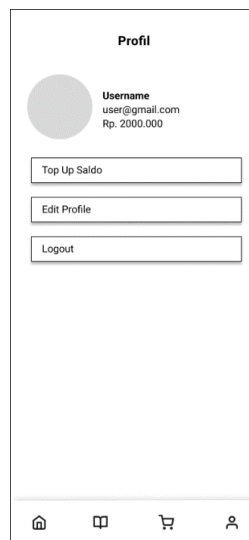
Halaman Keranjang merupakan halaman untuk melihat produk-produk yang telah dimasukkan kedalam keranjang. Halaman ini juga merupakan halaman untuk mengorder atau *checkout* produk yang ada di dalam keranjang.



Gambar 13. Halaman keranjang

e. Halaman Profil (*Customer*)

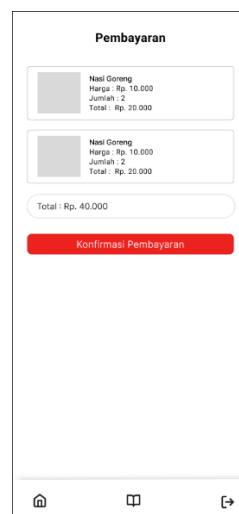
Halaman ini adalah halaman untuk melihat dan mengelola profil dari *user*. Jadi, *user* dapat melakukan *edit* profil, *top up* saldo, dan juga *logout* dari aplikasi.



Gambar 14. Halaman *profile*

f. Halaman Pembayaran (*Customer*)

Halaman Pembayaran merupakan halaman untuk mengkonfirmasi pembayaran pesanan. Jadi *user* memvalidasi pesanannya dan juga harga dari pesanan untuk kemudian dilakukan pembayaran.



Gambar 15. Halaman pembayaran

4. Halaman *Vendor*

a. Halaman Kelola Produk (*Vendor*)

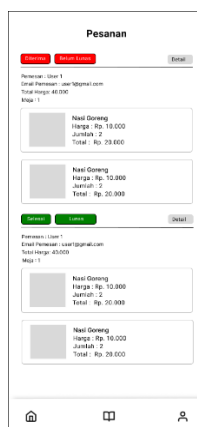
Halaman Kelola Produk merupakan halaman pertama yang akan muncul ketika vendor sudah melalui tahap *login*. Pada halaman ini, vendor dapat mengelola produk yang akan dijual seperti, menambahkan, *edit*, dan juga hapus produk.



Gambar 16. Halaman kelola produk

b. Halaman Riwayat Pesanan (*Vendor*)

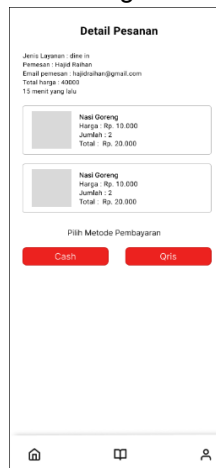
Halaman Riwayat Pesanan pada vendor adalah halaman untuk melihat pesanan-pesanan yang masuk. Jadi vendor bisa melihat pesanan siapa yang lebih dulu masuk dan dapat dikerjakan terlebih dahulu. Vendor juga bisa mengubah status pesanan sesuai dengan apa yang telah dikerjakan seperti pesanan sedang diterima, diproses, selesai. Selain itu, terdapat juga status pembayaran disetiap pesanan, statusnya yaitu lunas dan belum lunas agar vendor bisa mengetahui *customer* mana yang sudah dan belum membayar pesannya. Di Halaman ini juga terdapat button yang dapat menampilkan detail pesanan yang dipilih.



Gambar 17. Halaman riwayat pesanan

c. **Halaman Detail Pesanan (Vendor)**

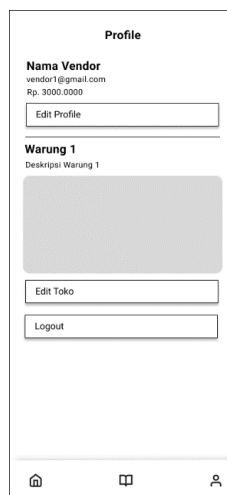
Halaman detail pesanan adalah halaman untuk melihat detail dari pesanan *customer* dan juga memilih metode pembayaran yang ingin digunakan oleh *customer* yaitu lewat QR code atau dengan *cash*.



Gambar 18. Halaman detail pesanan

d. **Halaman Profil (Vendor)**

Halaman Profil *vendor* adalah halaman untuk melihat detail dari profil *user* dan juga detail dari *toko* yang dimiliki oleh *user*.

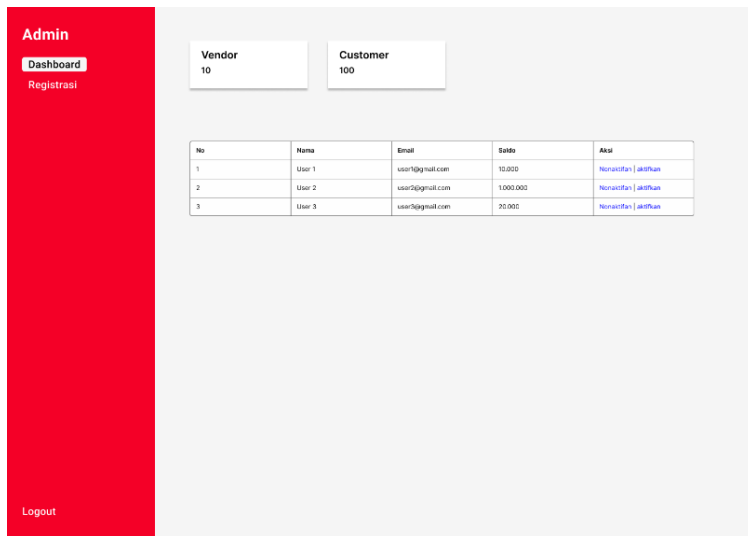


Gambar 19. Halaman profil

5. Halaman Admin

a. Halaman *Dashboard*

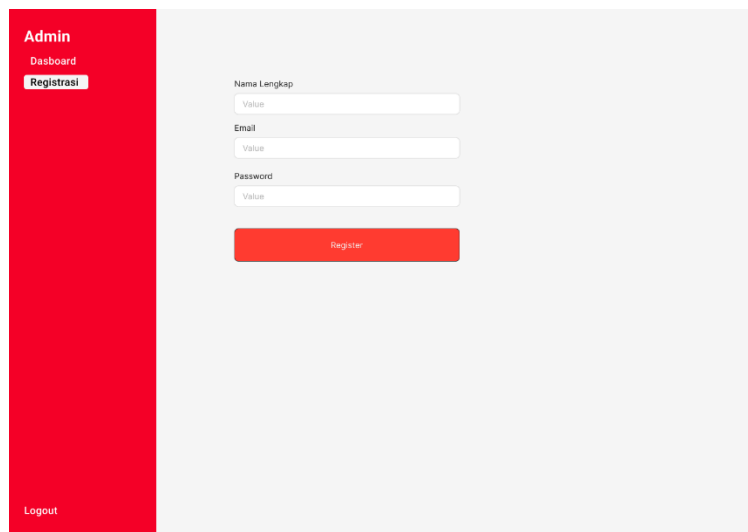
Halaman dashboard adalah halaman untuk melihat berapa *user* yang telah terdaftar yaitu *user* dengan *role customer* dan *vendor*. Pada halaman ini juga admin bisa menonaktifkan *user*.



Gambar 20. Halaman *dashboard*

b. Halaman Registrasi Vendor

Halaman registrasi vendor adalah halaman untuk mendaftarkan atau meregistrasi *user* dengan *role vendor*.



Gambar 21. Halaman registrasi *vendor*