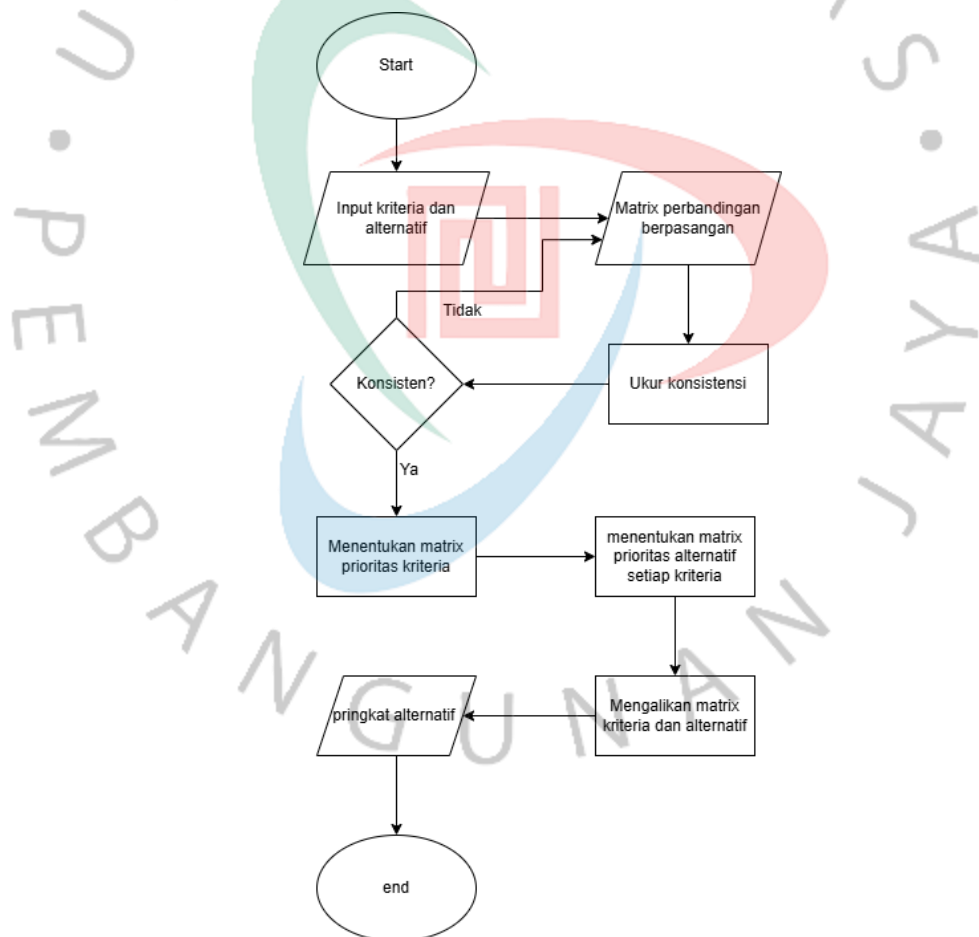


BAB IV PERANCANGAN

Pada bab ini akan menjelaskan perancangan website yang akan dibuat pada tugas akhir.

4.1 Analisis Sistem Terdahulu

Analisis sistem sebelumnya untuk mengetahui bagaimana sistem sebelumnya beroperasi, kelebihan, dan serta kekurangan pada sistem sebelumnya. Pada tugas akhir ini memilih jurnal JUSTIN dengan judul “Rancang Bangun Pencarian Kamar Inap Berbasis Web dengan Menggunakan *Analytical Hierarchy Process* (AHP)” sebagai pembandingan yang disusun oleh (Hengky Anra, Tursina, Ryan Ariessa, 2022)



Gambar 4.1 *Flowchat* Pada Sistem Terdahulu

4.1 Spesifikasi Kebutuhan Sistem Baru

Pada spesifikasi kebutuhan sistem baru adalah menambah fitur seperti rekomendasi rumah kos dan juga pemesanan rumah kos yang menggunakan *first come first served*. Yang dimana user dapat melakukan antrian dan pemesanan melalui *website*.

4.1.1 Spesifikasi Kebutuhan Input

Spesifikasi kebutuhan *input* dibutuhkan data rumah kos berupa harga, fasilitas, luas, dan jarak. Data tersebut digunakan untuk merekomendasikan rumah kos tersebut menggunakan algoritma *AHP*.

4.1.2 Spesifikasi kebutuhan Sistem Output

Hasil dari analisis mengenai kebutuhan *output* yaitu merekomendasikan rumah kos menggunakan algoritma *AHP* serta melakukan antrian pemesanan menggunakan algoritma *FCFS*.

4.2.3 Spesifikasi Kebutuhan Proses

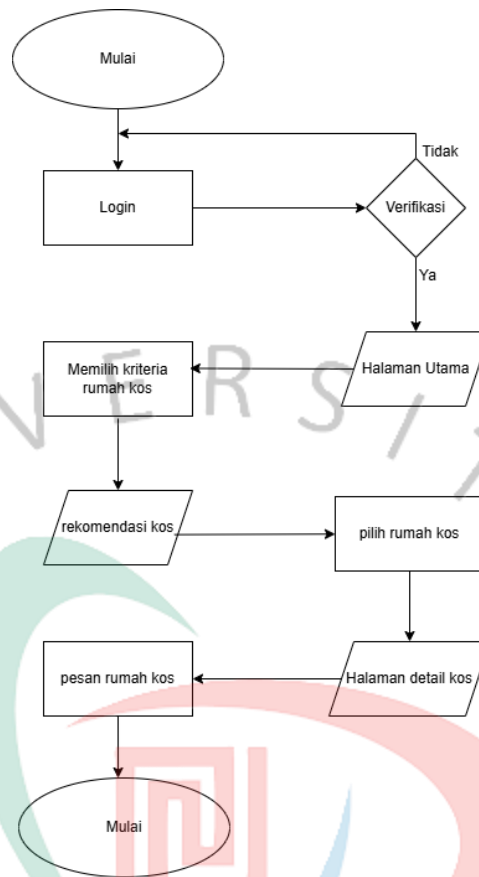
Analisis kebutuhan proses yaitu dengan menerapkan algoritma *AHP* pada rekomendasi rumah kos serta menerapkan algoritma *First Come First Served* pada pemesanan rumah kos.

4.2 Perancangan Sistem

Proses perancangan sistem meliputi tahapan perencanaan dan implementasi sistem serta melibatkan tahapan dalam memproses data dengan dukungan sistem yang telah dirancang. Berikut adalah perancangan sistem *website* menggunakan algoritma *AHP* dan *FCFS* pada rekomendasi rumah kos.

4.2.1 Flowchart

Menurut Zakaria et al. (2023) *Flowchart* adalah alat dasar dalam pengembangan algoritma yang membantu memahami logika program secara lebih jelas. Berikut *flowchart* dari *website* rekomendasi rumah kos.



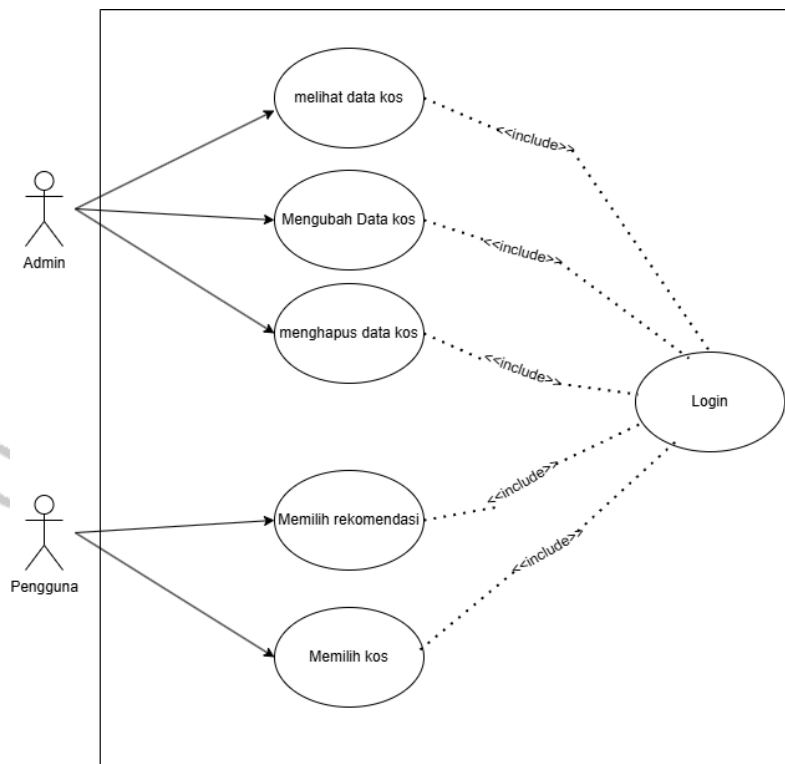
Gambar 4.2 Flowchart Pada Aplikasi Rekomendasi Rumah Kos

Pada gambar 4.1 menggambarkan alur penggunaan *website* oleh pengguna. Tahap pertama dimulai dengan proses login oleh pengguna. Setelah berhasil masuk kedalam sistem, pengguna akan diarahkan ke halaman utama. Pada halaman ini, pengguna dapat memilih kriteria berdasarkan kebutuhan yang diinginkan oleh pengguna. Selanjutnya, sistem menampilkan hasil rekomendasi berdasarkan kriteria yang dipilih oleh pengguna. Pengguna kemudian dapat memilih salah satu rumah kos dari hasil rekomendasi untuk melihat detail dari rumah kos. Di halaman detail ini, tersedia memesan rumah kos.

4.2.2 Usecase

Menurut *Object Management Group* (OMG), *Usecase* Diagram adalah bagian dari *behavioral diagram* dalam UML yang menggambarkan interaksi antara aktor (pengguna atau sistem *eksternal*) dan sistem itu sendiri. Diagram ini menyajikan

fungsionalitas sistem dari sudut pandang luar, tanpa menjelaskan mekanisme internalnya. Berikut *usecase* dari perancangan aplikasi.



Gambar 4.3 *Usecase* Diagram

Pada Gambar 4.2, diperlihatkan aktor admin. Admin merupakan pengguna yang memiliki hak akses untuk mengelola data dalam sistem. Sebelum dapat melakukan pengelolaan, admin diwajibkan untuk melakukan login terlebih dahulu. Setelah berhasil masuk, admin dapat melakukan berbagai aktivitas seperti menambahkan data rumah kos, mengubah informasi rumah kos, serta menghapus data rumah kos dari sistem. Sedangkan pada aktor pengguna, memiliki tujuan untuk mencari rumah kos berdasarkan kriteria tertentu yang diinginkan. Sebelum dapat mengakses fitur dalam aplikasi, pengguna diwajibkan untuk melakukan proses login terlebih dahulu. Setelah berhasil masuk, pengguna dapat memilih rekomendasi rumah kos serta memesan rumah kos.

4.2.3 Skenario *Usecase*

4.1.1.1 Skenario *Usecase* Admin

Table 4.1 Skenario *Usecase* Admin

Nama Use Case	Admin
Aktor	Admin
Deskripsi Use Case	Aktor melakukan manajemen data rumah kos
Precondition	Aktor harus login terlebih dahulu untuk mengakses halaman admin
Tahapan	1. Aktor login 2. Aktor memilih data yang ingin ditambah, diubah, atau dihapus
Postcondition	Jika aktor berhasil login maka aktor dapat mengakses halaman admin

4.1.1.2 Skenario Use Case pengguna

Table 4.2 Skenario *Usecase* Halaman Utama

Nama Use Case	Pengguna
Aktor	pengguna
Deskripsi Use Case	pengguna melakukan login
Precondition	Halaman login akan menampilkan form dan tombol submit untuk masuk ke halaman utama
Tahapan	1. Aktor mengakses website 2. Actor memilih menu login 3. Tampil halaman login berupa form yang muncul 4. Aktor mengisi form dengan akun yang telah didaftarkan sebelumnya 5. Aktor berhasil login dan masuk ke halaman utama

Postcondition	Data yang dimasukan oleh actor akan divalidasi, apabila data tersebut terdapat pada databse maka akan menampilkan halaman utama
---------------	---

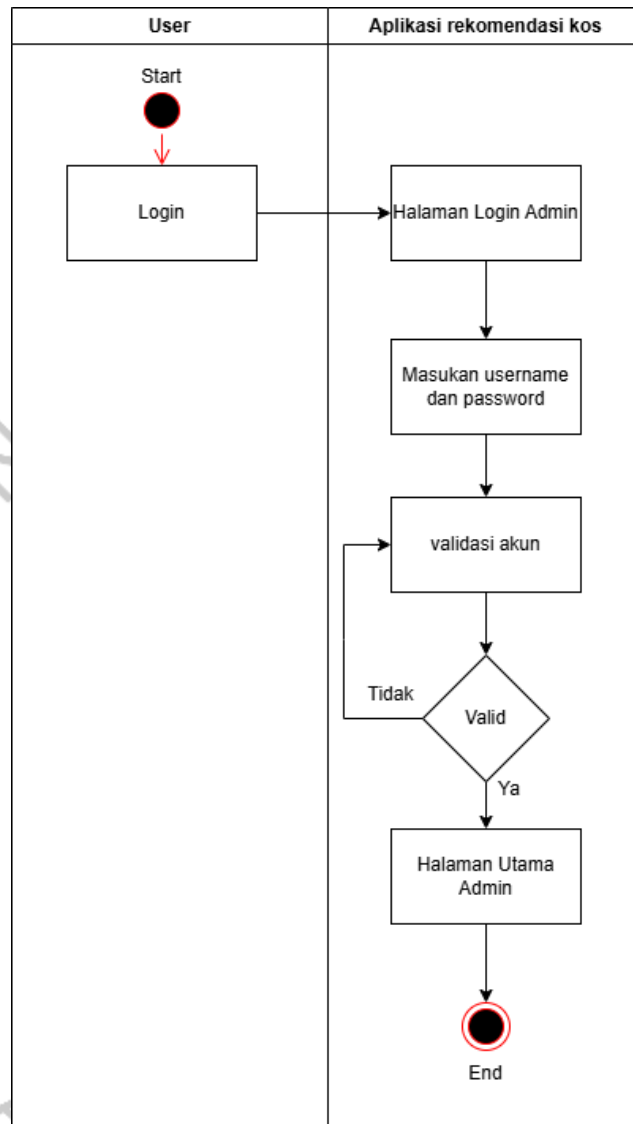
Table 4.3 Skenario *Usecase* Rekomendasi

Nama Use Case	Pengguna
Aktor	Pengguna
Deskripsi Use Case	Pengguna menggunakan rekomendasi pada halaman utama
Precondition	Aktor harus memilih kriteria untuk menampilkan rekomendasi
Tahapan	1. Aktor login 2. Aktor memilih kriteria yang diinginkan 3. Perhitungan algoritma AHP berjalan 4. Tampil hasil rekomendasi
Postcondition	Aktor harus memilih kriteria yang diinginkan terlebih dahulu

4.2.3 Activity Diagram

Menurut Sardar Mudassar Ali Khan (Juni 2023), Menjelaskan bahwa activity diagram adalah jenis *behavioral* diagram dalam *UML* yang digunakan untuk memvisualisasikan alur dan urutan aktivitas atau proses dalam sebuah sistem atau proses bisnis. Diagram ini menunjukkan aktivitas, aksi, dan keputusan yang terjadi, serta bagaimana mereka saling terkait dalam sebuah *workflow*.

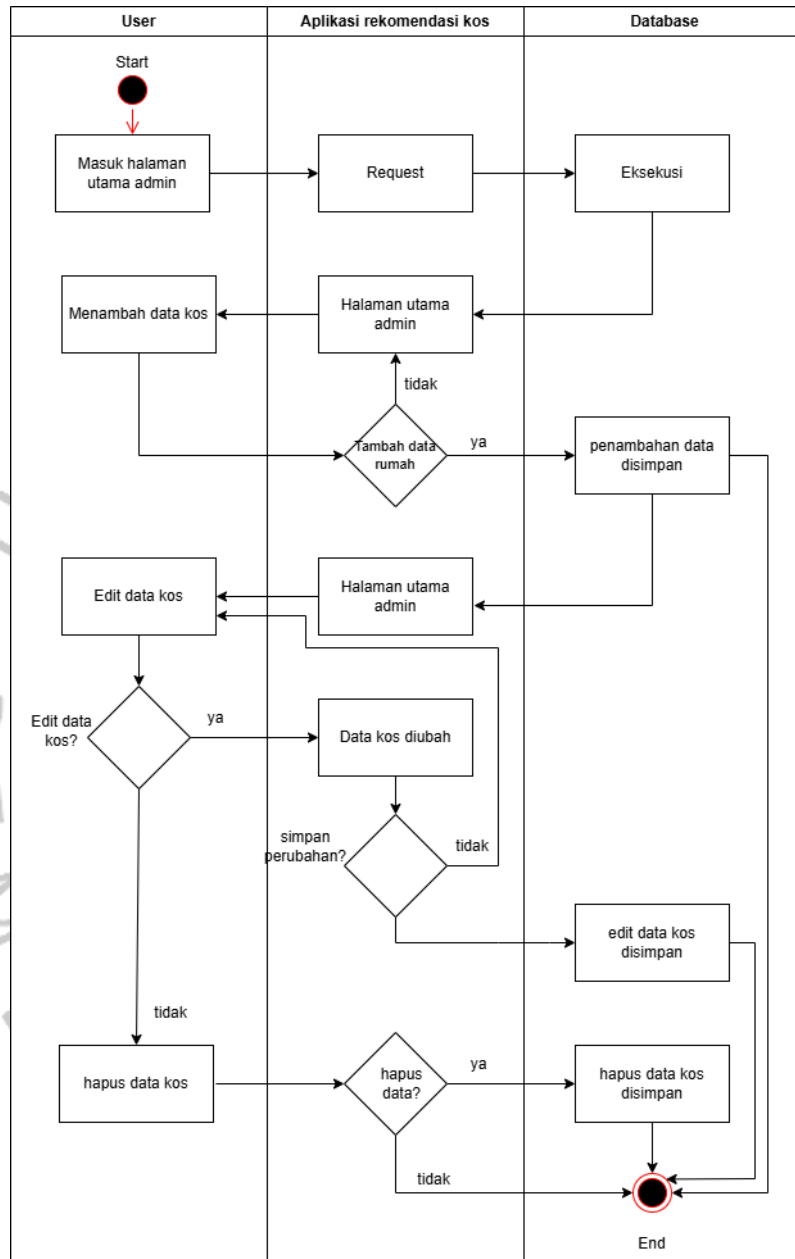
4.3.4.1 Activity Diagram *Login Admin*



Gambar 4.4 Activity Diagram *Login* Pada Halaman Admin

Pada halaman *login*, admin diminta untuk mengisi *username* dan *password* sesuai dengan akun yang telah dibuat sebelumnya. Setelah itu, sistem akan melakukan proses validasi untuk memastikan apakah data yang dimasukkan telah terdaftar. Jika informasi login dinyatakan valid, maka admin akan diarahkan ke halaman utama admin.

4.3.4.2 Activity Diagram Data Kos Admin

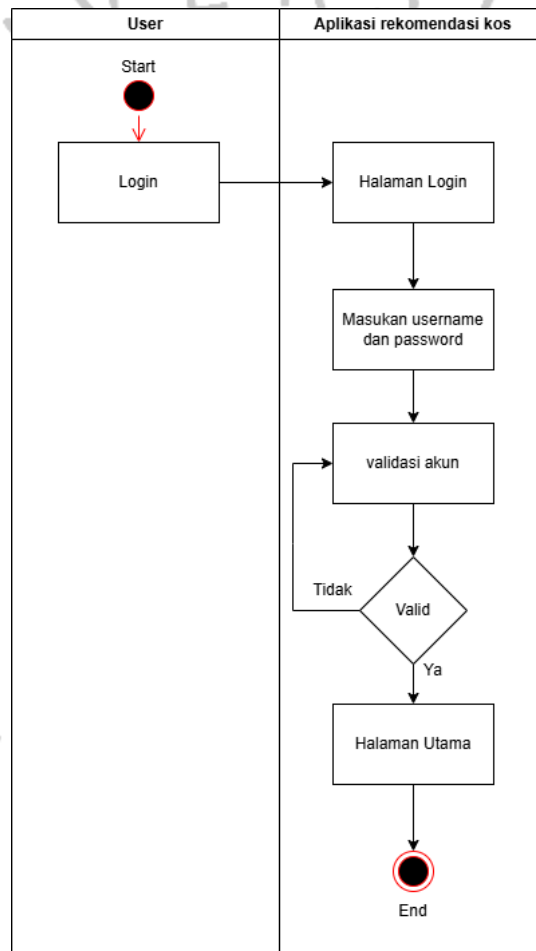


Gambar 4.5 Activity Diagram Halaman Admin

Pada gambar 4.5 sistem mengirim *request* kepada *database*, setelah mengeksekusi *request* sistem menampilkan halaman utama admin. Selanjutnya jika admin ingin menambah data kos maka admin memilih tombol tambah pada halaman utama lalu mengisi form penambahan data jika selesai maka admin akan kembali ke halaman

utama dan data terisi, jika admin membatalkan penambahan data akan kembali ke halaman utama. Untuk mengubah data admin memilih tombol *edit* pada halaman utama admin lalu menampilkan form perubahan data setelah disimpan admin kembali ke halaman utama. Pada penghapusan data admin memilih tombol delete pada halaman utama admin lalu menampilkan *alert box* setelah memilih ya menghapus maka akan kembali ke halaman utama admin dan data terhapus.

4.3.4.3 Activity Diagram Login Pengguna

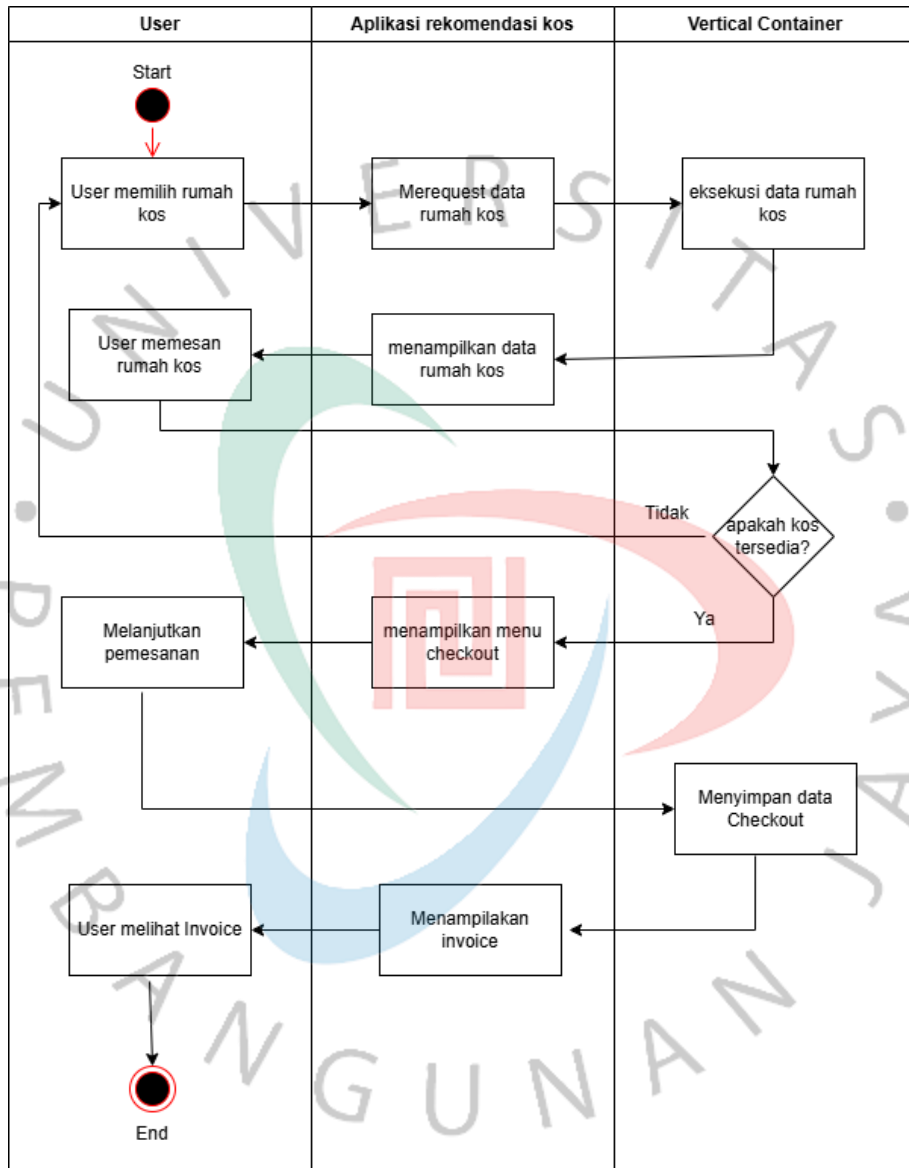


Gambar 4.6 Activity Diagram Login Pengguna

Pada halaman *login*, pengguna diminta untuk mengisi *username* dan *password* sesuai dengan *akun* yang telah dibuat sebelumnya. Setelah itu, sistem akan melakukan proses validasi untuk memastikan apakah data yang dimasukkan telah terdaftar. Jika

informasi login dinyatakan valid, maka pengguna akan diarahkan ke halaman utama aplikasi.

4.3.4.4 Activity Diagram Pemesanan

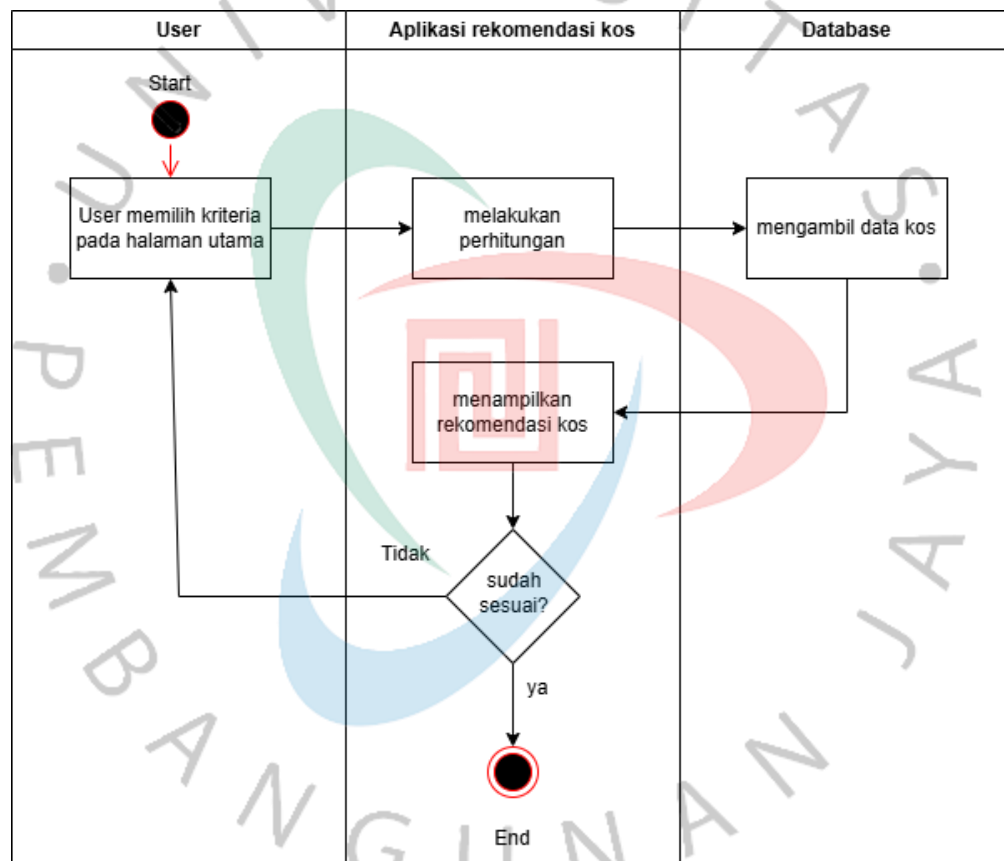


Gambar 4.7 Activity Diagram Pemesanan Rumah Kos

Pada Gambar 4.7, yaitu activity diagram, ditunjukkan alur proses ketika pengguna ingin memesan rumah kos. Setelah berhasil login, pengguna akan diarahkan ke halaman utama yang menampilkan daftar rumah kos yang tersedia untuk dipilih. Selanjutnya, pengguna akan masuk ke halaman detail rumah kos yang dipilih,

kemudian diarahkan ke halaman daftar sewa. Pada halaman daftar sewa inilah algoritma FCFS (First Come First Serve) diterapkan, di mana pengguna diberikan batas waktu tertentu untuk melakukan checkout sewa sebelum pesanan otomatis dibatalkan oleh sistem. Setelah melakukan checkout, pengguna diminta untuk mengisi data-data yang diperlukan. Terakhir, pengguna dapat melihat invoice sebagai bukti pemesanan rumah kos, dan proses pemesanan dinyatakan berhasil.

4.3.4.5 Activity Diagram Rekomendasi



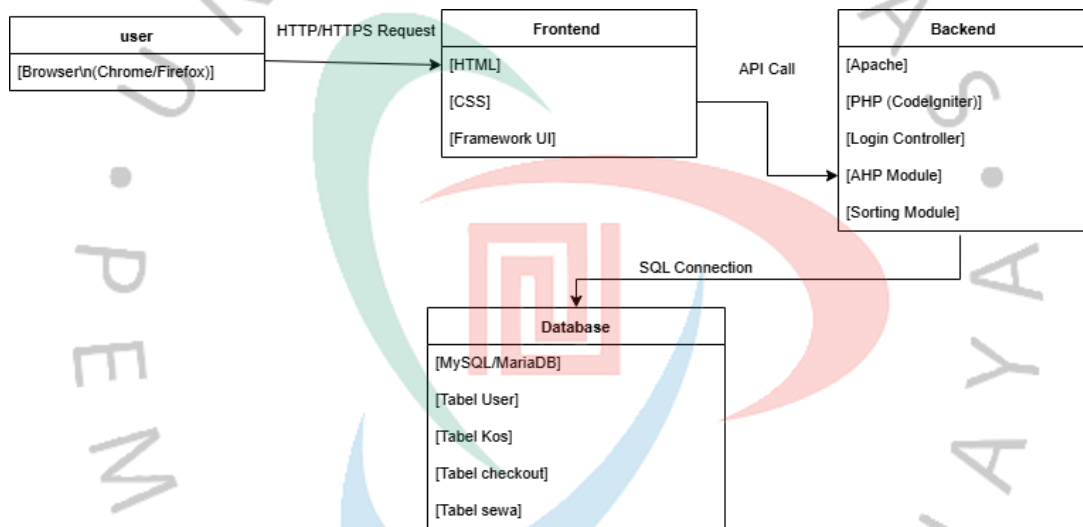
Gambar 4.8 Activity Diagram Rekomendasi Rumah Kos

Pengguna memilih kriteria yang diinginkan seperti harga, fasilitas, luas, jarak melalui halaman utama. Setelah memilih kriteria yang diinginkan *aplikasi* menerima masukan dari pengguna dan melakukan proses perhitungan berdasarkan kriteria yang diberikan. Sistem terhubung dengan database untuk mengambil data kos yang tersedia

berdasarkan kebutuhan perhitungan. Setelah proses perhitungan selesai, aplikasi akan menampilkan hasil rekomendasi berdasarkan kriteria yang dipilih.

4.3.5 Deployment Diagram

Menurut Sari & Gunawan (2023) dalam jurnal International Journal of Computer Applications, Deployment Diagram. Diagram yang digunakan untuk memodelkan arsitektur fisik sistem, menggambarkan bagaimana perangkat keras dan perangkat lunak saling terhubung serta bagaimana komponen perangkat lunak di-deploy pada node fisik. Berikut deployment diagram pada perancangan aplikasi rekomendasi.



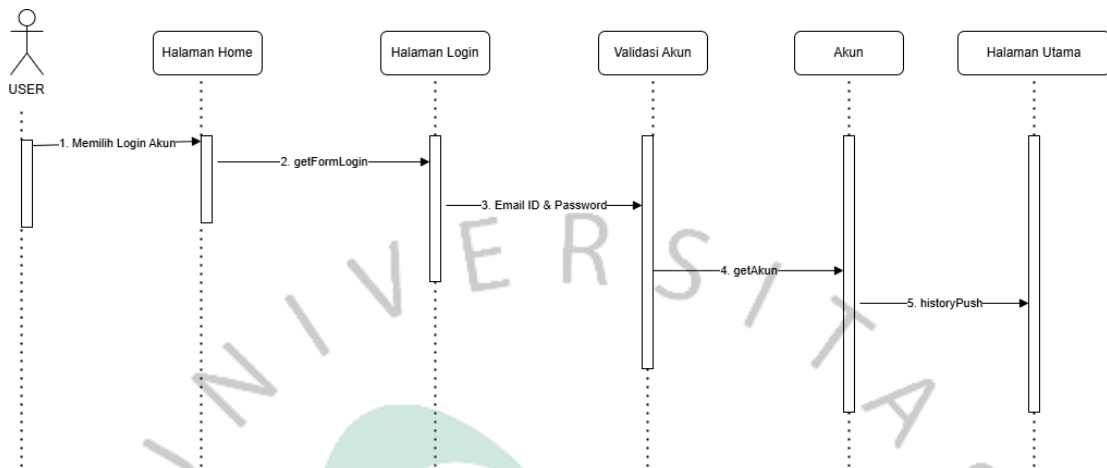
Gambar 4.9 Deployment Diagram

User menggunakan laptop atau Smartphone untuk mengakses aplikasi rumah kos. Selanjutnya mengirim permintaan kepada frontend menggunakan HTTP/HTTPS request. Setelah frontend menerima request dari user maka frontend menampilkan aplikasi rumah kos. Lalu frontend akan merequest kepada backend melalui API untuk mengambil data ke database. Setelah database menerima request dari backend, database memberikan data kepada backend melalui SQL Connection.

4.3.6 Sequence Diagram

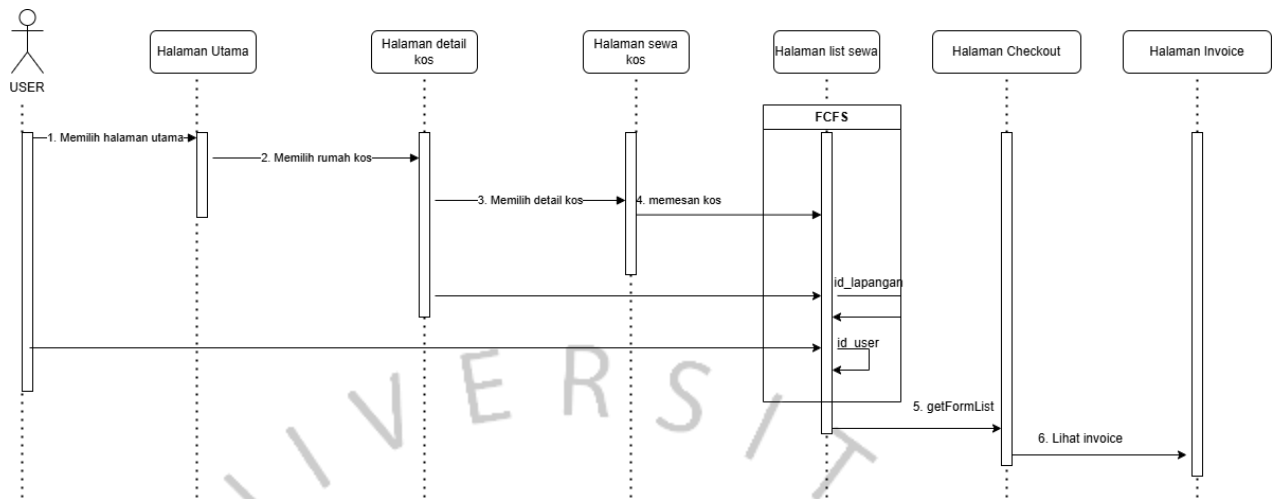
Menurut Nugroho (2023), *Sequence* diagram adalah diagram interaksi yang menggambarkan bagaimana objek berinteraksi satu sama lain melalui pesan-

pesan dalam suatu urutan waktu tertentu untuk menjalankan sebuah skenario proses. Berikut adalah sequence diagram yang telah dirancang.



Gambar 4.10 *Sequence Diagram Login*

Gambar 4.10 memperlihatkan *sequence* diagram yang menggambarkan alur interaksi antara pengguna dan sistem dalam proses *login*. Alur dimulai ketika pengguna berada pada halaman *home* dan memilih opsi *login* menggunakan akun yang telah terdaftar sebelumnya. Selanjutnya, sistem akan menampilkan halaman login yang meminta pengguna untuk mengisi *username* dan kata sandi. Setelah data tersebut diinput, sistem akan melakukan proses validasi terhadap informasi akun. Jika proses validasi berhasil, sistem akan mengeksekusi fungsi *getAkun* untuk mengambil data pengguna yang sesuai, dan selanjutnya pengguna akan diarahkan menuju halaman *home* kategori sebagai indikasi bahwa proses *login* telah berhasil diselesaikan.



Gambar 4.11 *Sequence Diagram* Memesan Rumah Kos

Gambar 4.11 menggambarkan *sequence diagram* yang merepresentasikan interaksi antara pengguna dengan sistem aplikasi dalam melakukan proses pemesanan rumah kos. Proses dimulai ketika pengguna memilih rumah kos melalui halaman utama. Selanjutnya, setelah itu, pengguna memasuki halaman detail lapangan, di mana tersedia informasi mengenai spesifikasi yang dapat dipilih sesuai preferensi. Setelah memilih kos yang sesuai, pengguna melanjutkan proses ke halaman daftar sewa. Pada tahap ini, sistem menerapkan algoritma *First Come First Serve* (FCFS), di mana pengguna diberikan batas waktu tertentu untuk melakukan *checkout*. Apabila pengguna tidak menyelesaikan proses *checkout* dalam batas waktu tersebut, maka sistem akan secara otomatis membatalkan pemesanan. Halaman daftar sewa juga memanfaatkan data dari entitas kos dan pengguna. Apabila proses *checkout* berhasil dilakukan sebelum waktu habis, pengguna akan diarahkan ke halaman *checkout* untuk mengisi formulir pemesanan. Setelah semua data diisi dengan lengkap, sistem akan menampilkan invoice sebagai bukti transaksi pemesanan rumah kos, dan proses pemesanan dianggap selesai secara valid.

4.3.5 Perancangan Database

Perancangan basis data merupakan tahap penting dalam proses pengembangan aplikasi, karena memungkinkan penyimpanan data yang diolah secara terstruktur dalam sebuah sistem database. Berikut ini disajikan rancangan basis data yang digunakan dalam penelitian ini.

Tabel 4.1 Tabel *Database Kos*

No	Field	Tipe data	panjang
1.	Id(pk)	Integer	11
2.	Nama	Varchar	100
3.	Alamat	Text	
4.	Harga	Decimal	10.2
5.	Deskripsi	Text	
6.	Foto	Varchar	255
7.	luas	Varchar	25
8.	Fasilitas	Varchar	255
9.	Jarak	Varchar	25

Tabel 4.2 Tabel *Database Sewa*

No	Field	Tipe data	Panjang
1	Id(pk)	Integer	10
2	User_id(FK)	Integer	11
3	Kos_id(FK)	Varchar	50
4	Durasi	Text	
5	Tgl_sewa	Varchar	50

Tabel 4.3 Tabel *Database Checkout*

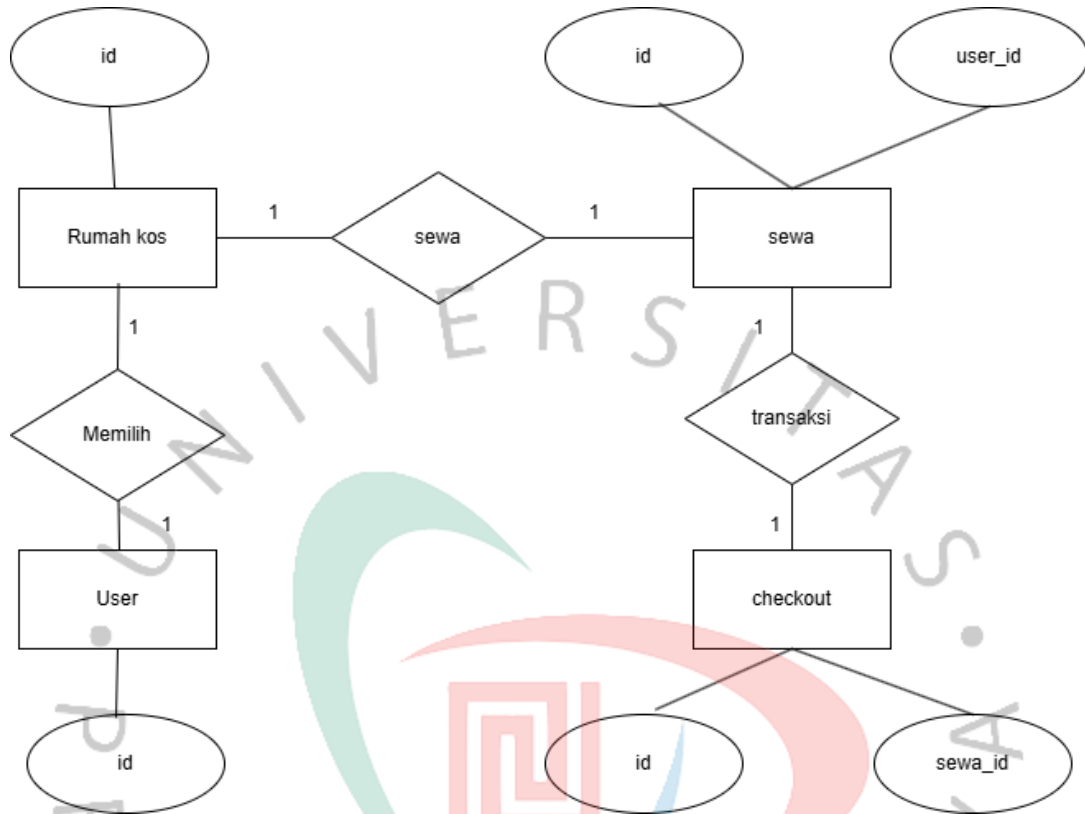
No	Field	Tipe data	Panjang
1	Id(PK)	Integer	10
2	User_id(FK)	Varchar	50
3	Sewa_id(FK)	Varchar	50
4	Namadepan	Varchar	50
5	Namabelakang	Varchar	50
6	Email	Varchar	50
7	Notlp	Varchar	14
8	Alamat	Varchar	70
9	Kode_transaksi	Varchar	50

Tabel 4.4 Tabel *Database Users*

No	Field	Tipe data	Panjang
1	Id(PK)	Integer	10
2	Username	Varchar	50
3	Email	Varchar	50
4	password	Varchar	50

Tabel di atas menunjukkan rancangan basis data yang digunakan dalam pengembangan aplikasi rekomendasi rumah kos. Terdapat empat tabel utama dalam struktur basis data ini, yaitu tabel *users*, *kos*, *sewa*, dan *checkout*. Tabel *users* berfungsi untuk menyimpan informasi pengguna yang terdaftar dalam sistem. Tabel *kos* digunakan untuk menampung data rumah kos yang tersedia untuk disewa. yang kemudian akan dihubungkan secara relasional dengan tabel *kos* melalui atribut tertentu. Tabel *sewa* berperan dalam menyimpan data transaksi penyewaan kos yang masih dalam proses atau belum diselesaikan oleh pengguna, dan berfungsi sebagai tahap awal sebelum pengguna melakukan *checkout*. Terakhir, tabel *checkout* merekam data transaksi penyewaan yang telah diselesaikan oleh pengguna. Beberapa atribut penting dalam tabel *checkout* meliputi *id* sebagai *primary key*, *user_id* yang merupakan foreign key dari tabel *users*, dan *sewa_id* yang juga merupakan foreign key dari tabel *sewa*. Selain itu, tabel ini juga memuat informasi detail terkait data penyewa yang melakukan transaksi. Struktur ini dirancang untuk mendukung proses penyewaan kos secara efisien dan terintegrasi antar entitas.

4.3.6 ERD



Gambar 4.12 ERD

Gambar 4.12 menunjukkan *Entity Relationship Diagram* (ERD) yang terdiri dari empat entitas utama dan tiga jenis relasi antar entitas. Entitas pertama adalah user, yang memiliki atribut id sebagai *primary key* untuk mengidentifikasi setiap pengguna secara unik. Entitas ini memiliki relasi *one-to-one* dengan entitas rumah kos, yang berarti satu pengguna hanya dapat memilih satu rumah kos dalam satu proses transaksi.

Entitas kedua, yaitu rumah kos, juga memiliki atribut id sebagai *primary key*. Entitas rumah kos selanjutnya memiliki hubungan *one-to-one* dengan entitas sewa, yang menunjukkan bahwa satu rumah kos hanya dapat digunakan dalam satu transaksi penyewaan pada satu waktu.

Entitas sewa merupakan entitas ketiga dalam ERD ini, dan memiliki atribut id sebagai *primary key*, serta user_id yang bertindak sebagai *foreign key* yang merujuk ke entitas

user. Entitas ini memiliki hubungan *one-to-one* dengan entitas checkout, yang berarti setiap transaksi penyewaan hanya dapat diselesaikan melalui satu kali proses *checkout*.

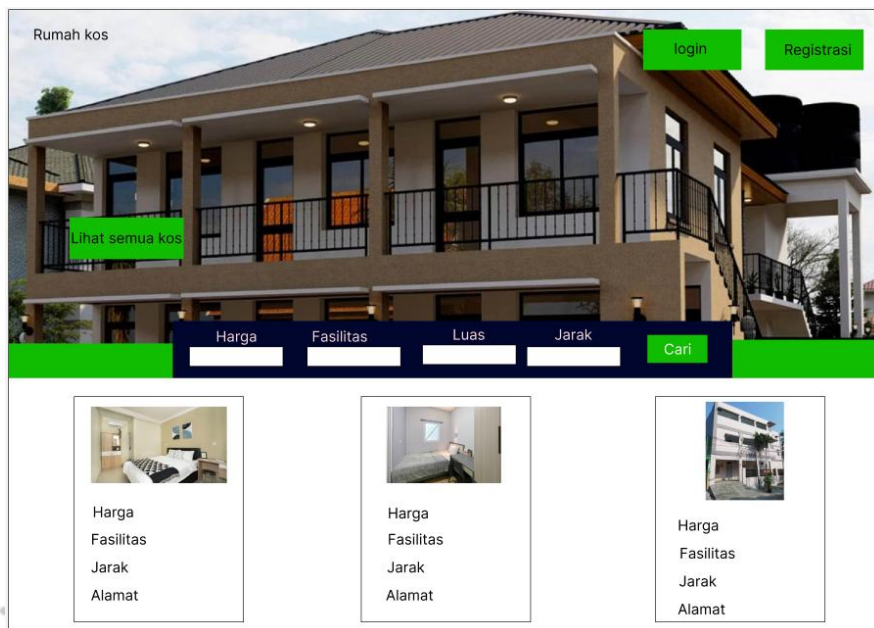
Terakhir, entitas *checkout* memiliki atribut id sebagai *primary key*, dan atribut sewa_id sebagai *foreign key* yang menghubungkannya dengan entitas sewa. Struktur hubungan antar entitas ini dirancang untuk mencerminkan alur logis dan terintegrasi dalam sistem pemesanan, dengan menjaga keterkaitan antar data dan keakuratan transaksi.

3.4.6 Fitur Sorting kos

Fitur ini memungkinkan pengguna untuk mengurutkan daftar kos berdasarkan jarak terdekat ke kampus, harga terendah atau tertinggi, dan kelengkapan fasilitas sesuai kebutuhan. Hitung jarak kos ke kampus menggunakan rumus Haversine atau fungsi bawaan (misal `distance()` pada Google Maps API atau GeoDjango) pada sorting jarak. Pada sorting harga menggunakan Sorting ascending berdasarkan harga terendah ke harga tertinggi dan Sorting descending, harga tertinggi ke terendah. Pada sorting fasilitas, urutkan kos berdasarkan jumlah fasilitas terbanyak ke tersedikit atau berdasarkan fasilitas tertentu yang dipilih pengguna (misal hanya menampilkan kos dengan WiFi dan kamar mandi dalam).

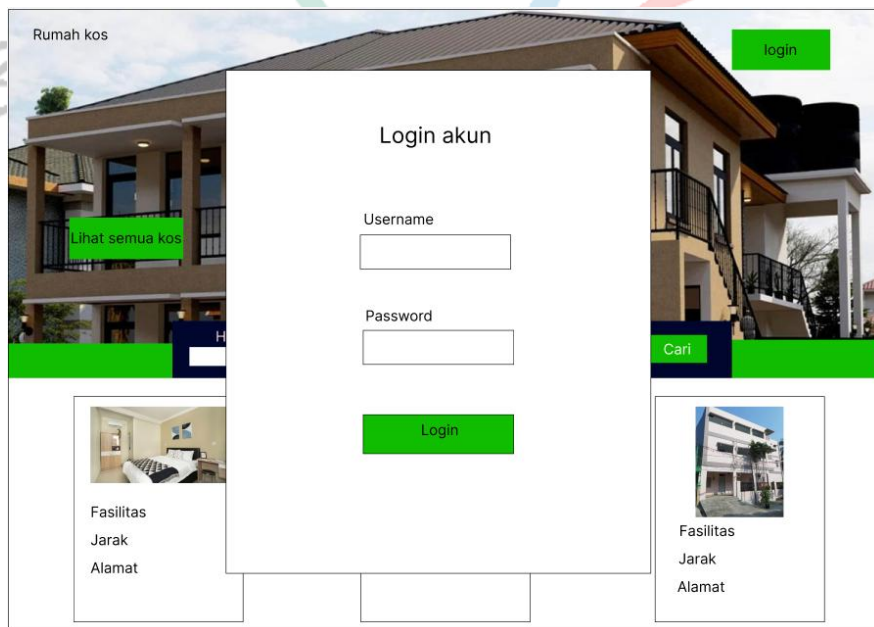
3.4.7 Perancangan Tampilan Antarmuka

Mockup merupakan Gambaran untuk membangun website yang akan di realisasikan. Serta mockup sebagai Gambaran mengenai fitur dan juga tampilan.



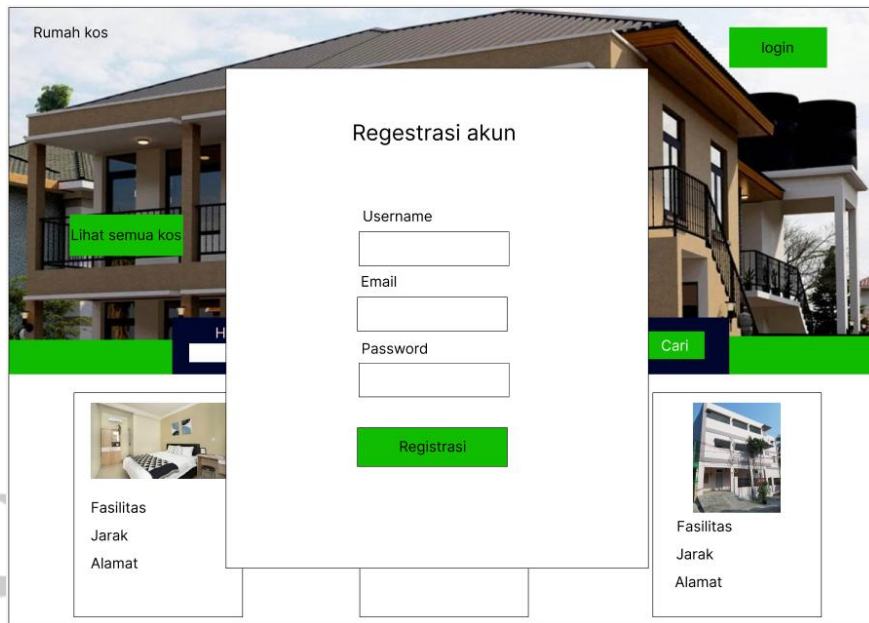
Gambar 4.13 Rancangan Tampilan Halaman Utama

Gambar 4.13 menampilkan rancangan desain halaman utama yang dilengkapi dengan slide gambar serta ringkasan daftar rumah kos dalam bentuk card. Setiap card menampilkan gambar rumah kos, harga kos, dan dapat diklik untuk mengakses detail rumah kos yang dipilih.



Gambar 4.14 Rancangan Tampilan *login*

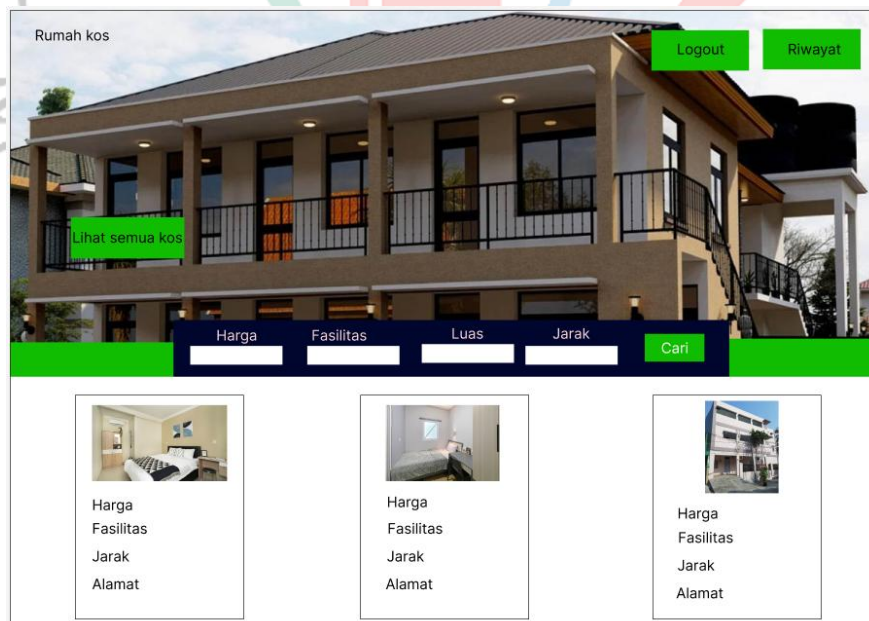
Pada gambar 4.14 Menunjukkan form untuk *login* yang berisikan *username*, *password*, serta tombol *login*.



The image shows a web application interface for finding rental houses. A central white modal box titled "Registrasi akun" (Account Registration) is overlaid on a background image of a modern house. The modal contains input fields for "Username", "Email", and "Password", followed by a green "Registrasi" button. The background interface includes a header with "Rumah kos" and a "login" button, a sidebar with "Lihat semua kos" and a "Cari" button, and a list of house cards with details like "Fasilitas", "Jarak", and "Alamat".

Gambar 4.15 Rancangan Tampilan Registrasi

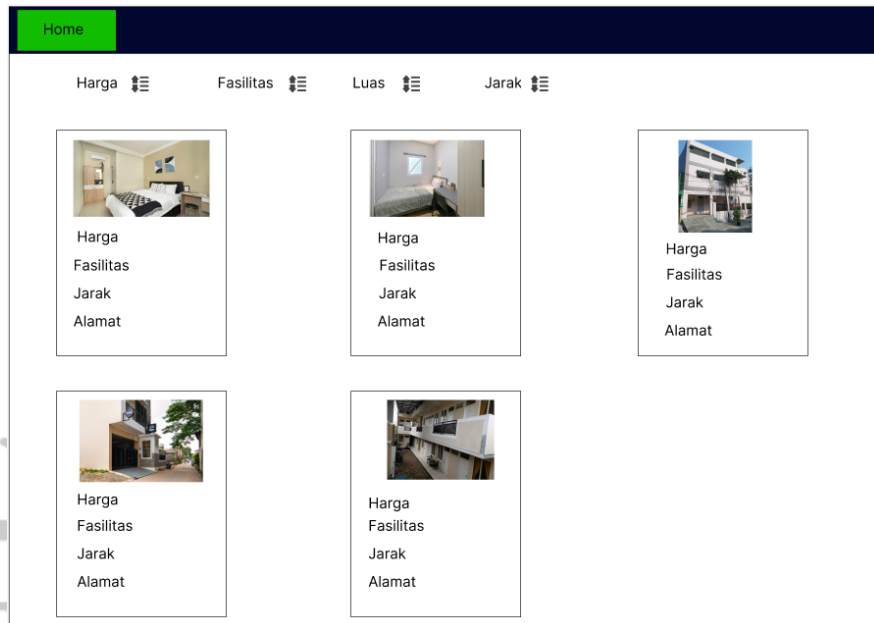
Pada gambar 4.15 merupakan tampilan untuk registrasi akun.



The image shows the main page of the web application after a user has logged in. The background is the same house listing page. The modal box is no longer present. Instead, there is a search bar with fields for "Harga", "Fasilitas", "Luas", and "Jarak", and a green "Cari" button. The sidebar now includes a "Logout" button and a "Riwayat" button. The list of house cards remains at the bottom, each with a thumbnail image and details like "Harga", "Fasilitas", "Jarak", and "Alamat".

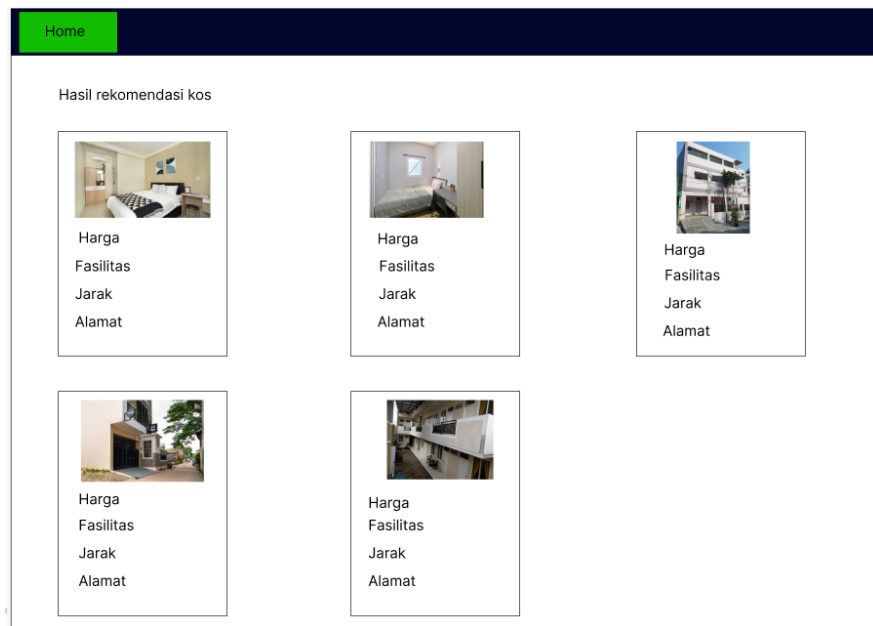
Gambar 4.16 Halaman Utama Setelah *Login*

Gambar 4.16 menunjukkan rancangan tampilan halaman utama setelah melakukan login. Setelah login bertambah satu tombol untuk menuju pada halaman riwayat transaksi.



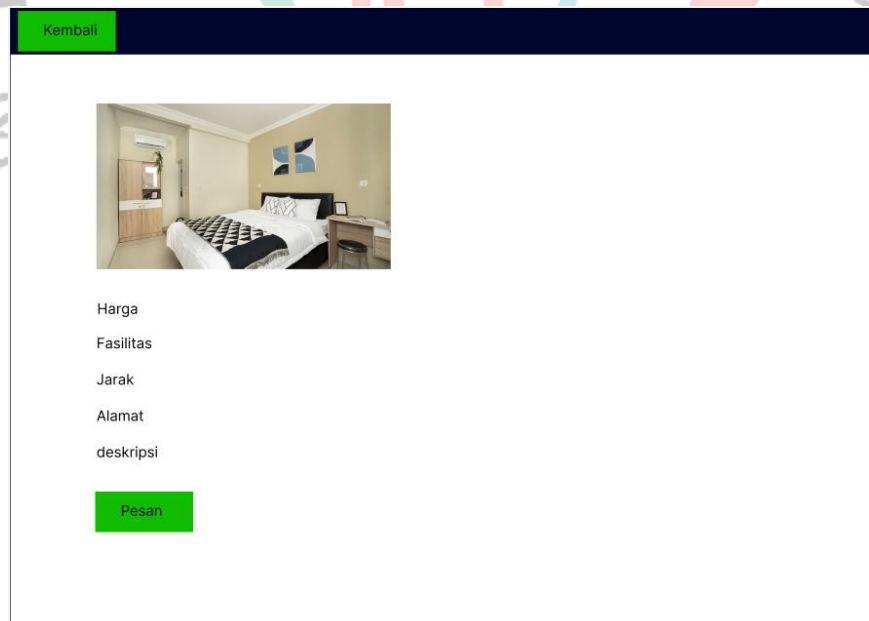
Gambar 4.17 Rancangan Tampilan List Rumah Kos

Pada gambar 4.17 menunjukan halaman semua list rumah kos. Pada halaman ini pengguna dapat melakukan sorting berdasarkan harga, fasilitas, luas, dan jarak.



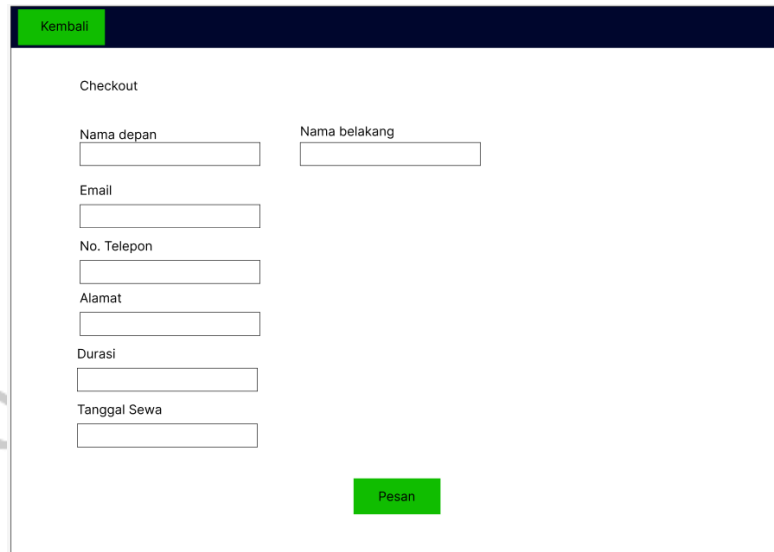
Gambar 4.18 Rancangan Tampilan Hasil Rekomendasi

Pada gambar 4.18 merupakan hasil dari rekomendasi yang sebelumnya diinput berdasarkan kriteria pada halaman utama rumah kos.



Gambar 4.19 Rancangan Tampilan Halaman Detail Rumah Kos

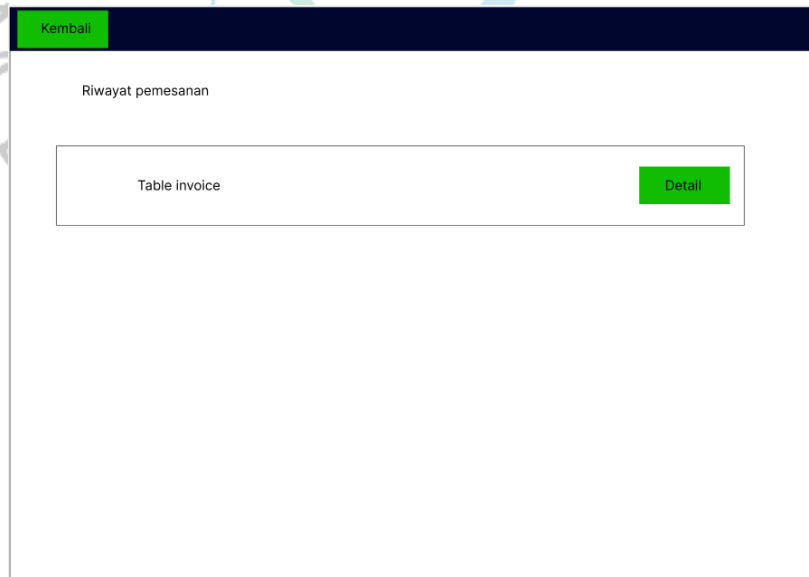
Pada gambar 4.19 merupakan detail rumah kos yaitu harga, fasilitas, jarak, Alamat, luas, deskripsi, serta tombol untuk memesan. Setelah pengguna memesan maka akan menuju halaman check out.



The screenshot shows a web interface for a checkout process. At the top, there is a dark blue header with a green button labeled 'Kembali'. Below the header, the title 'Checkout' is centered. The form contains several input fields: 'Nama depan' and 'Nama belakang' (two columns), 'Email', 'No. Telepon', 'Alamat', 'Durasi', and 'Tanggal Sewa' (one column). A green button labeled 'Pesan' is positioned at the bottom right of the form area.

Gambar 4.20 Rancangan Tampilan Halaman *Checkout*

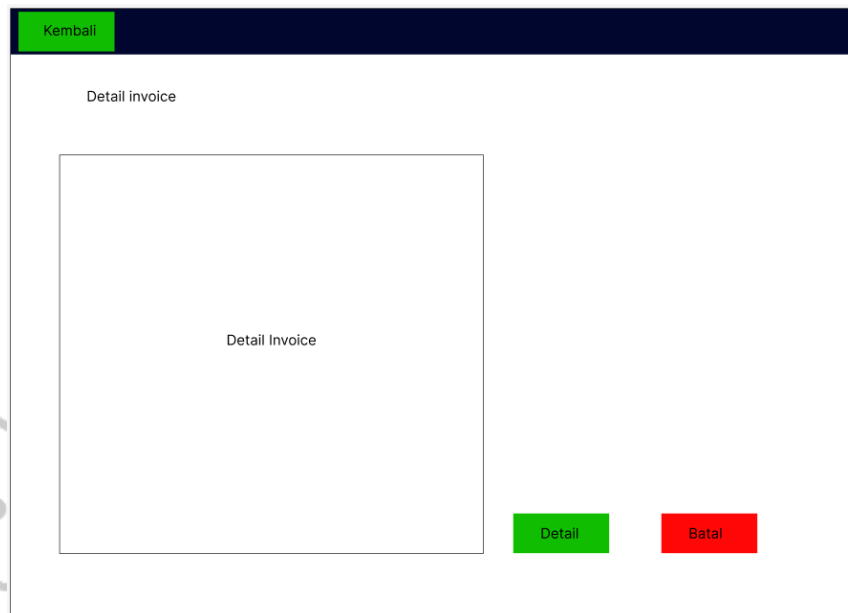
Pada gambar 4.20 setelah pengguna mengisi form pemesanan maka pengguna akan menuju kehalaman riwayat pemesanan.



The screenshot shows a web interface for the order history page. At the top, there is a dark blue header with a green button labeled 'Kembali'. Below the header, the title 'Riwayat pemesanan' is centered. The main content area contains a box with the text 'Table invoice' and a green button labeled 'Detail' to its right.

Gambar 4.21 Rancangan Tampilan Halaman Riwayat Pemesanan

Pada gambar 4.21 merupakan halaman Riwayat pemesanan yang menampilkan invoice pemesanan pengguna. Selain itu terdapat tombol detail untuk melihat detail dari invoice.



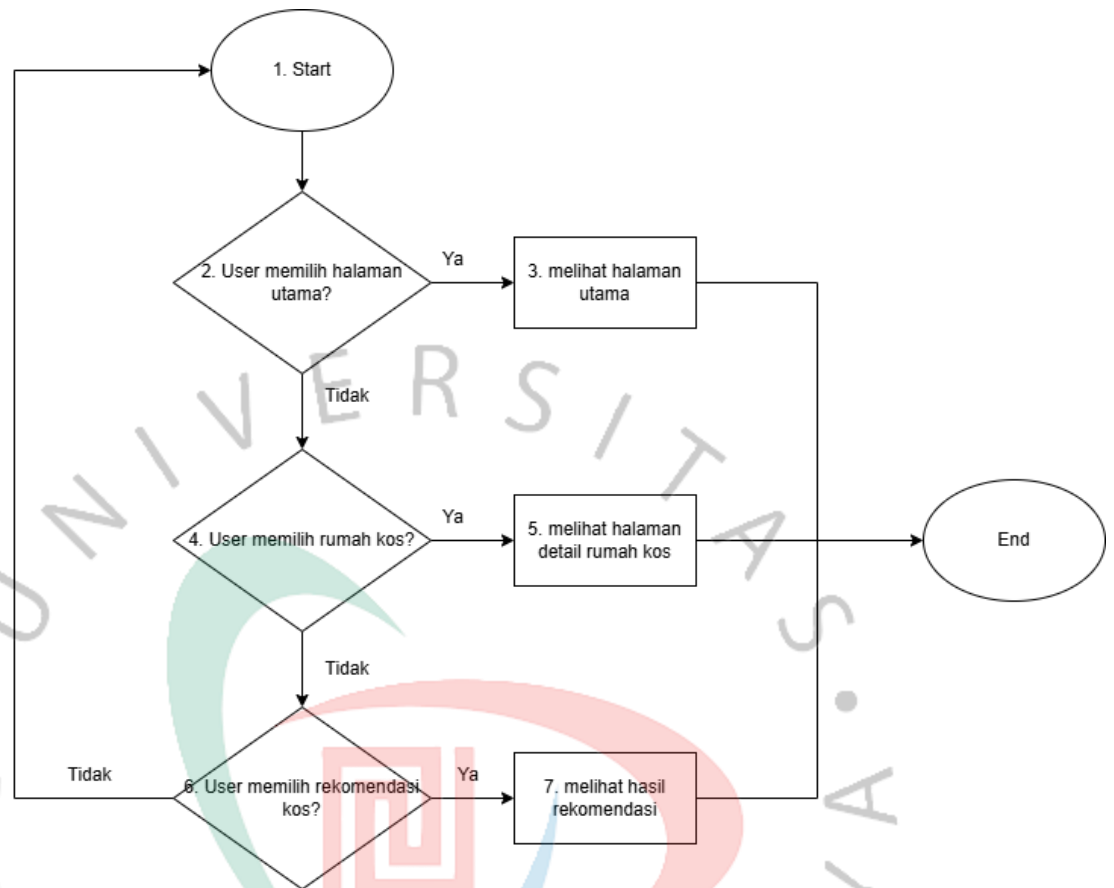
Gambar 4.22 Rancangan Rampilan Halaman Detail Invoice

Pada rancangan tampilan halaman detail invoice terdapat data invoice. Terdapat dua tombol yaitu tombol bayar dan juga batal.

4.3.7 Perancangan Pengujian

4.3.7.1 Rencana Pengujian Non-Fungsional

Saat merancang sebuah aplikasi, tahap pengujian sangat penting untuk memastikan bahwa aplikasi berfungsi sesuai dengan skenario yang direncanakan. Baik pengujian *white-box* maupun *black-box* biasanya didokumentasikan dalam bentuk tabel pengujian agar lebih mudah diperiksa dan dianalisis. Untuk metode *white-box*, diperlukan visualisasi alur pengujian menggunakan *flowgraph*. *Flowgraph* ini berfungsi untuk menggambarkan jalannya eksekusi kode dan struktur kontrol yang diuji. Di bawah ini adalah representasi visual dari alur pengujian berbasis *white-box*.

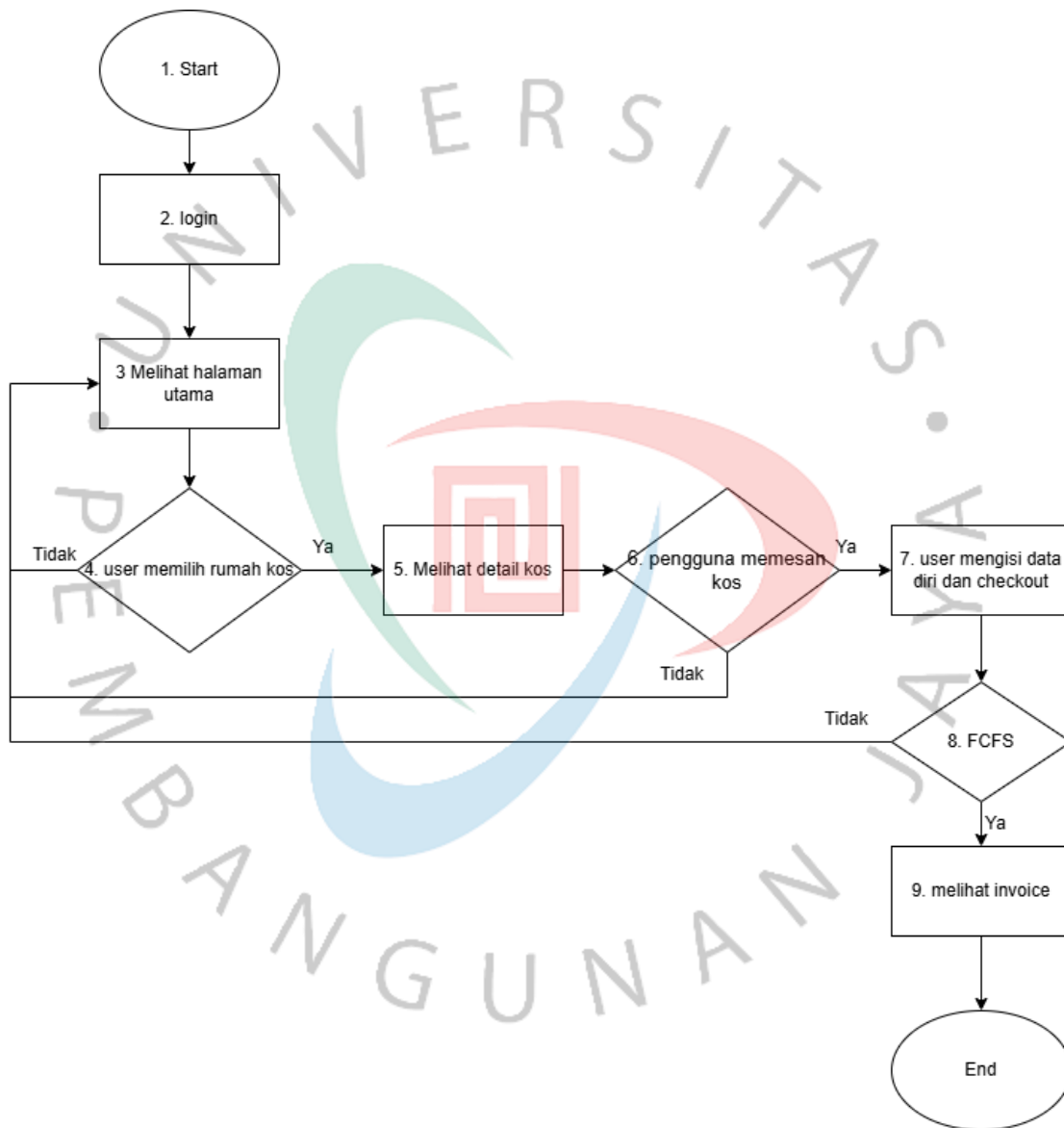


Gambar 4.23 Flowgraph Menu Halaman Utama

Gambar di atas menunjukkan flowgraph navigasi menu yang menggambarkan alur menuju setiap halaman yang terdapat pada website. Alur flowgraph tersebut adalah sebagai berikut:

1. Flowgraph dimulai.
2. Pada kondisi pertama, jika pengguna menekan tombol halaman utama, maka flowgraph akan berlanjut ke langkah nomor 3. Jika tidak, flowgraph akan berlanjut ke langkah nomor 4.
3. Pada kondisi kedua, jika pengguna memilih rumah kos, maka flowgraph akan menuju langkah nomor 5. Jika tidak, flowgraph akan berlanjut ke langkah nomor 6.

4. Pada kondisi ketiga, jika pengguna memilih rekomendasi kos, maka flowgraph akan menuju langkah nomor 7. Jika tidak, pengguna akan kembali ke langkah nomor 1.
5. Flowgraph selesai.



Gambar 4.24 *Flowgraph* Pemesanan

Gambar 4.17 merupakan flowgraph yang menggambarkan alur proses ketika pengguna berada pada halaman sewa kos. Adapun penjelasan flowgraph tersebut adalah sebagai berikut:

1. Flowgraph dimulai.
2. Pengguna melakukan login.
3. Pengguna diarahkan ke halaman utama.
4. Kondisi pertama, apakah pengguna memilih rumah kos? Jika ya, flowgraph akan melanjutkan ke langkah nomor 5. Jika tidak, flowgraph akan kembali ke langkah nomor 3.
5. Pada halaman detail kos, pengguna akan melihat informasi detail rumah kos yang dipilih, kemudian flowgraph berlanjut ke langkah nomor 6.
6. Kondisi kedua, apakah pengguna memesan kos? Jika ya, flowgraph akan melanjutkan ke langkah nomor 7. Jika tidak, flowgraph akan kembali ke langkah nomor 3.
7. Pengguna mengisi data diri dan melakukan checkout, kemudian flowgraph berlanjut ke langkah nomor 8.
8. Kondisi ketiga, sistem akan menerapkan algoritma FCFS dengan batas waktu tertentu. Jika waktu masih berlaku, flowgraph akan melanjutkan ke langkah nomor 9. Jika tidak, flowgraph akan kembali ke langkah nomor 3.
9. Flowgraph selesai.

4.3.7.2 Rencana Pengujian Fungsional

Perancangan pengujian sistem dilakukan dengan menggunakan metode black box testing untuk mengevaluasi fungsi-fungsi dari perangkat lunak yang telah dikembangkan. Penjelasan lebih detail disajikan sebagai berikut:

Tabel 4.5 Pengujian Dengan *Black Box*

No	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
----	--------------------	-----------------------	-----------------

1	Pengguna melakukan login	Berhasil login
2	Pengguna mengakses halaman utama	Menampilkan halaman utama
3	Pengguna mengakses halaman detail kos	Menampilkan detail kos

