

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Pada analisis, terdapat penjelasan terkait identifikasi masalah dan pemecahan masalah dari penelitian berikut.

3.1.1 Identifikasi Masalah

Identifikasi masalah dalam penelitian ini berkaitan dengan manajemen rute kunjungan tim *sales* ke berbagai toko. Permasalahan utama terletak pada ketiadaan sistem terstruktur dalam penentuan rute kunjungan, yang mengakibatkan *inefisiensi* operasional. Tim *sales* sering kali harus melakukan penentuan rute secara manual dan mengalami redundansi perjalanan yang tidak perlu, hal ini berdampak pada pemborosan waktu, tenaga, dan sumber daya perusahaan. Situasi ini mengindikasikan adanya kebutuhan akan implementasi sistem yang dapat mengoptimalkan perencanaan rute kunjungan toko secara sistematis dan efisien.

Untuk mengatasi masalah ini, peneliti menggunakan analisis SWOT. Tabel SWOT ini akan membantu dalam menentukan kekuatan, kelemahan, peluang, dan ancaman saat membangun dan menerapkan solusi rute optimal. Analisis SWOT akan membantu menemukan cara untuk mengatasi kelemahan dan memanfaatkan peluang untuk meningkatkan efisiensi penjualan *sales*.

Tabel 3.1 Framework SWOT

Strengths (Kekuatan)	Opportunities (Peluang)
Memiliki tim <i>sales</i> yang berpengalaman dan memahami kebutuhan pasar lokal.	Kemajuan teknologi yang memungkinkan pengembangan sistem otomatisasi rute kunjungan.
Basis pelanggan yang sudah mapan dan loyal.	Potensi peningkatan efisiensi operasional yang dapat meningkatkan

	kepuasan pelanggan.
Weaknesses (Kelemahan)	Threats (Ancaman)
Proses penentuan rute kunjungan yang masih manual dan tidak efisien.	Perubahan kebijakan atau regulasi yang dapat mempengaruhi operasional perusahaan.

3.1.2 Pemecahan Masalah

Penelitian ini berfokus pada pengembangan sebuah aplikasi android yang mampu merekomendasikan rute atau urutan toko yang optimal bagi tim *sales*. Dengan demikian, diharapkan aplikasi ini tidak hanya menghemat biaya operasional kendaraan yang digunakan, tetapi juga meningkatkan efisiensi waktu dan produktivitas tim *sales* secara keseluruhan.

Sebagai bagian dari metodologi penelitian, aplikasi ini akan memanfaatkan teknologi Mapbox Optimization API v1 sebagai alat utama untuk menghitung rute optimal. Dengan penerapan teknologi ini, diharapkan proses kunjungan toko oleh tim *sales* menjadi lebih efisien dan terstruktur, sehingga dapat memberikan kontribusi signifikan terhadap peningkatan kinerja dan kepuasan pelanggan.

Untuk memastikan solusi yang terarah dan spesifik, peneliti menggunakan framework SMART (*Specific, Measurable, Achievable, Relevant, Time-bound*). Sehingga, setiap aspek masalah diuraikan secara jelas, terukur, relevan dengan kebutuhan *sales*, serta dapat diselesaikan dalam batasan waktu tertentu.

Tabel 3.2 Framework SMART

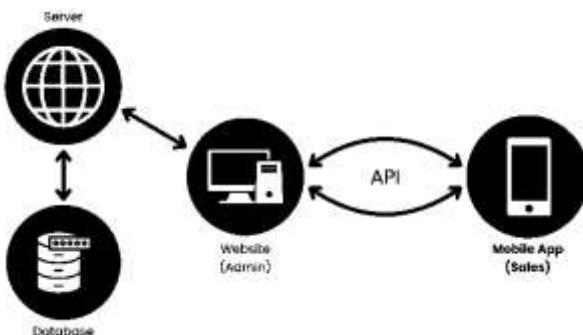
Specific (Spesifik)	Measurable (Terukur)
Mengembangkan aplikasi dengan fitur rekomendasi rute atau urutan	Keberhasilan diukur melalui pengurangan jarak & waktu perjalanan tim <i>sales</i> .

toko yang akan dikunjungi oleh tim <i>sales</i> .	
Achievable (Dapat Dicapai)	Relevant (Relevan)
Memanfaatkan teknologi yang tersedia seperti Mapbox Optimization API v1 untuk pengembangan aplikasi. Implementasi tersebut dapat diselesaikan oleh peneliti dalam waktu 8 bulan hingga Juni 2025.	Solusi ini relevan untuk mengatasi masalah inefisiensi rute kunjungan toko yang dihadapi oleh tim <i>sales</i> .
Time-Bound (Batas Waktu)	
Pengembangan aplikasi dan visualisasi rute diharapkan selesai dalam waktu 8 bulan hingga Juni 2025, dengan maksimal tambahan waktu 1 bulan	

3.2 Perancangan

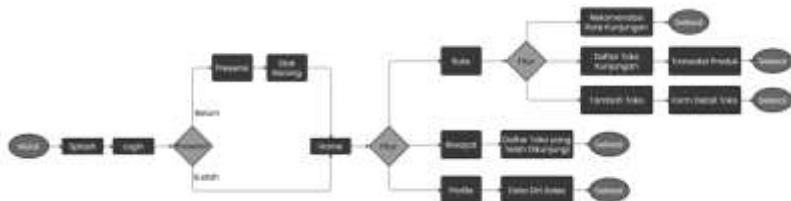
3.2.1 Perancangan Sistem

Penelitian ini berfokus pada pengembangan aplikasi *mobile* yang akan digunakan oleh *sales*, yang akan dihubungkan menggunakan API ke bagian *backend* sistem *website* admin. Dapat dilihat dalam arsitektur diagram pada gambar 3.1.



Gambar 3.1 Arsitektur Diagram

Perancangan sistem pada aplikasi *mobile* dalam penelitian ini digambarkan melalui sebuah flowchart seperti pada gambar 3.2, menguraikan alur kerja pengguna dari awal hingga akhir.



Gambar 3.2 Flowchart

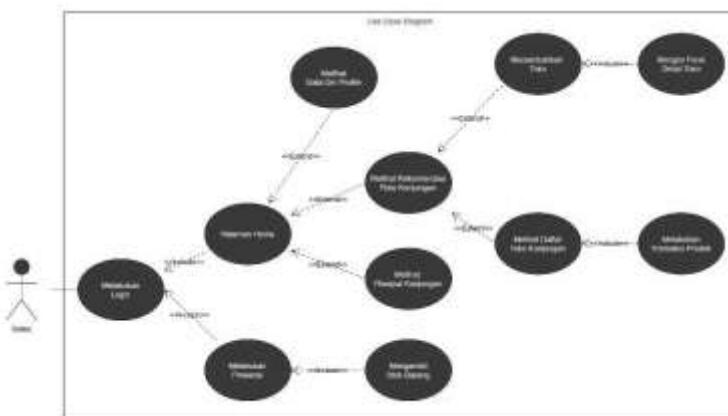
Proses dimulai dengan langkah pertama *Login*, yang merupakan akses pertama pengguna ke dalam sistem. Setelah berhasil masuk, sistem secara otomatis akan melakukan validasi terhadap status presensi pengguna. Jika pengguna belum melakukan presensi, alur sistem akan mengarahkan mereka untuk melakukan presensi terlebih dahulu, diikuti dengan akses ke stok barang, sebelum akhirnya kembali ke halaman *Home*. Namun, apabila pengguna sudah tercatat telah melakukan presensi, mereka akan langsung diarahkan menuju halaman *Home*, hal ini dirancang untuk mempercepat akses pengguna yang sudah memenuhi prasyarat awal. Dari halaman *Home*, pengguna disajikan dengan beragam fitur yang terintegrasi, digambarkan oleh percabangan alur keputusan. Salah satu

fitur utama adalah Rekomendasi Rute, yang jika dipilih akan menampilkan fragmen Rekomendasi Rute untuk memandu aktivitas lapangan pengguna.

Selain itu, dari fitur Rekomendasi Rute ini, pengguna juga dapat mengakses Daftar Toko Kunjungan dan melakukan Transaksi Produk, menandakan fungsi inti dari aplikasi ini dalam mendukung operasional penjualan. Memungkinkan pengguna untuk Tambah Toko baru dan mengisi Form Detail Toko dari alur fragmen rekomendasi rute, mengakomodasi penambahan data toko secara dinamis. Fitur lain yang tersedia adalah Riwayat, yang memungkinkan pengguna untuk meninjau Daftar Toko yang Telah Dikunjungi, memberikan rekam jejak aktivitas mereka. Terakhir, terdapat fitur Profile yang dirancang untuk menampilkan Data Diri *Sales*, memungkinkan pengguna untuk melihat informasi personal mereka. Seluruh alur kerja ini diakhiri dengan titik Selesai, menandakan selesainya suatu proses atau task yang dilakukan pengguna dalam aplikasi. Perancangan ini secara keseluruhan menekankan kemudahan navigasi dan kelengkapan fungsionalitas untuk mendukung tugas harian pengguna aplikasi *mobile* ini. Selain itu, penelitian ini menggunakan *Use case* diagram dan activity diagram untuk mengidentifikasi sistem berikut:

3.2.1.1 Use Case Diagram User

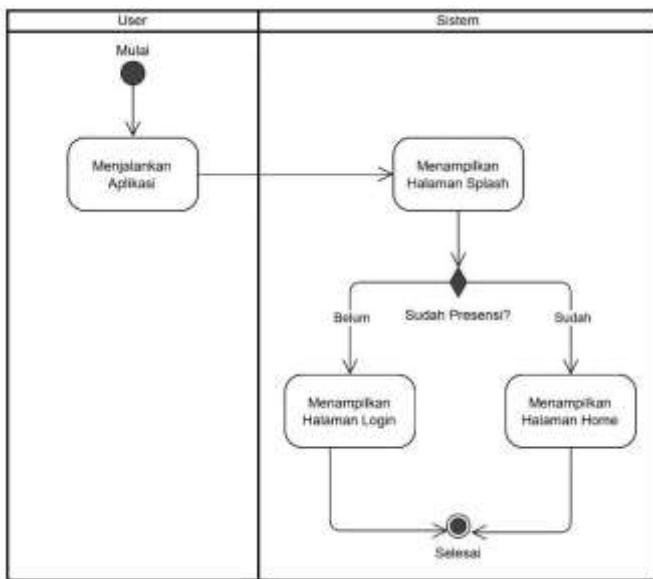
Use case diagram digunakan untuk menggambarkan secara visual hubungan antara aktor dan sistem dalam menjalankan berbagai fungsi. Pada penelitian ini, *Use case* diagram berfungsi untuk memetakan interaksi antara *User Sales* dengan fitur utama dalam aplikasi. Adapun *Use case* diagram dari penelitian ini ditampilkan pada gambar 3.3 berikut:



Gambar 3.3 Use case Diagram

Diagram *Use case* yang telah dirancang menunjukkan hubungan antara aktor (*User Sales*) dengan berbagai fitur yang tersedia dalam aplikasi *mobile* yang sedang dikembangkan. Diagram ini menyajikan gambaran visual dari beberapa skenario penggunaan fitur utama, seperti proses *login*, presensi, pemberian rekomendasi rute, pelaksanaan transaksi, serta fitur-fitur pendukung lainnya. Relasi “*include*” digunakan untuk menunjukkan bahwa suatu *Use case* selalu mencakup langkah-langkah dari *Use case* lain, sedangkan relasi “*extend*” digunakan untuk menggambarkan bahwa suatu aksi tambahan dapat terjadi sebagai perluasan dari skenario utama dalam kondisi tertentu.

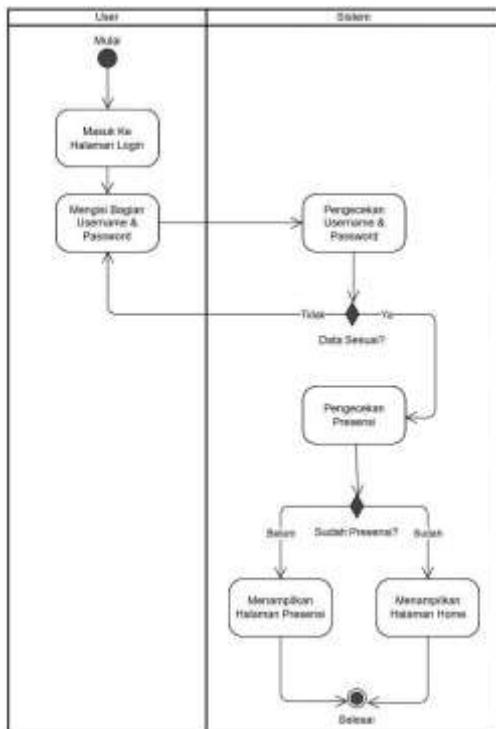
3.2.1.2 Activity Diagram SplashScreen Sales



Gambar 3.4 Activity Diagram SplashScreen

Activity diagram di atas menggambarkan alur awal ketika pengguna menjalankan aplikasi. Setelah aplikasi dijalankan oleh pengguna, sistem akan menampilkan halaman *splash* sebagai tahap awal pemrosesan. Selanjutnya, sistem melakukan pengecekan status presensi dengan mengirimkan permintaan (*request*) ke endpoint *getPresence*. Jika pengguna belum melakukan presensi, maka sistem akan menampilkan halaman *login* untuk proses autentikasi. Sebaliknya, jika presensi telah dilakukan, sistem langsung menampilkan halaman utama. Alur ini memastikan bahwa pengguna hanya dapat mengakses fitur utama aplikasi setelah presensi tercatat.

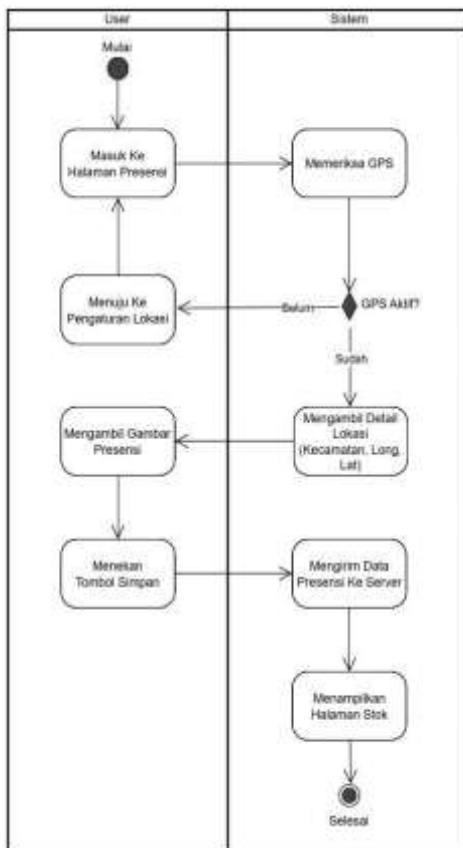
3.2.1.3 Activity Diagram Halaman *Login*



Gambar 3.5 Activity Diagram Halaman *Login*

Activity diagram pada gambar 3.5 menggambarkan alur masuk pengguna ke dalam sistem, dimulai dengan *User* yang memulai aktivitas. *User* kemudian masuk ke halaman *login* dan mengisi bagian *username & password*. Sistem menerima input ini dan melakukan pengecekan *username & password*. Jika data tidak sesuai, alur kembali ke langkah pengisian *username & password*. Jika data sesuai, sistem melanjutkan dengan pengecekan presensi. Jika *User* belum presensi, sistem akan menampilkan halaman presensi. Setelah *User* menyelesaikan presensi, atau jika *User* sudah presensi sejak awal, sistem akan menampilkan halaman *Home*.

3.2.1.4 Activity Diagram Melakukan Presensi

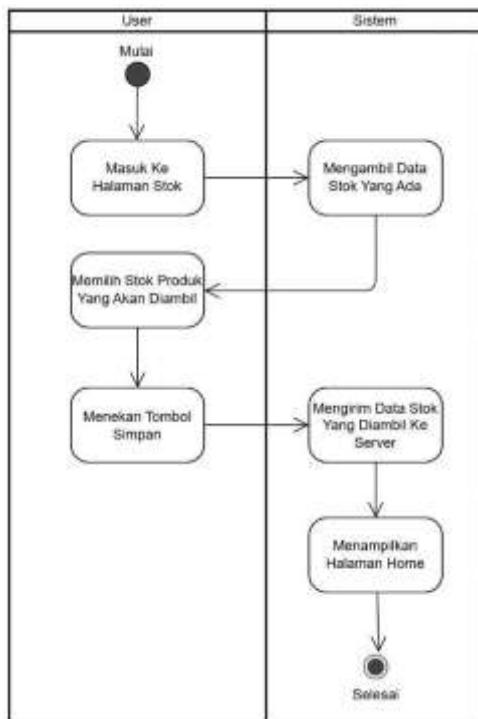


Gambar 3.6 Activity Diagram Splash Melakukan Presensi

Activity diagram pada gambar 3.6 menggambarkan alur presensi pengguna, dimulai ketika *User* memulai proses. *User* kemudian masuk ke halaman presensi, dan pada saat yang sama, sistem akan memeriksa status GPS. Jika GPS belum aktif, sistem akan mengarahkan *User* untuk mengaktifkan pengaturan lokasi. Setelah GPS aktif (atau jika sudah aktif sebelumnya), sistem akan mengambil detail lokasi yang meliputi kecamatan, *longitude*, dan *latitude*. Selanjutnya, *User* mengambil gambar

presensi, lalu menekan tombol simpan. Setelah tombol simpan ditekan, sistem akan mengirimkan data presensi tersebut ke *server*. Terakhir, sistem akan menampilkan halaman stok, yang menandakan berakhirnya alur presensi ini.

3.2.1.5 Pengambilan Stok Produk

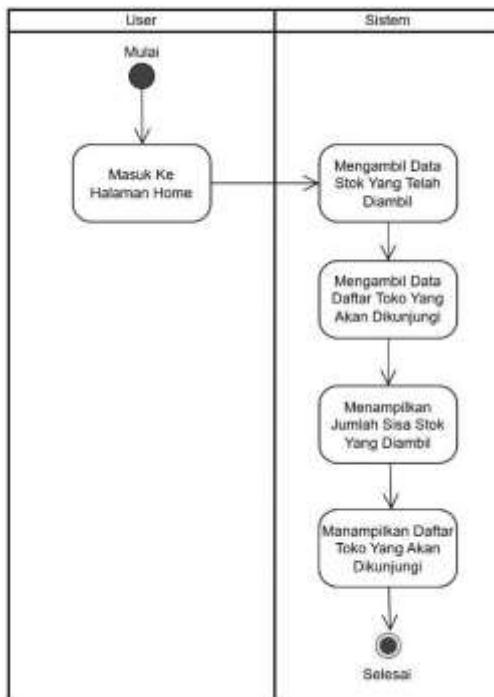


Gambar 3.7 Activity Diagram Pengambilan Produk

Activity diagram gambar 3.7 menjelaskan alur pengambilan stok oleh *User* dalam sistem. Proses dimulai ketika *User* masuk ke halaman stok. Secara bersamaan, sistem akan mengambil data stok yang tersedia. Setelah data stok ditampilkan, *User* kemudian memilih stok produk yang akan diambil. Setelah *User* memilih, ia akan menekan tombol simpan. Kemudian, sistem akan mengirimkan data stok yang telah diambil tersebut

ke *server*. Akhirnya, sistem akan menampilkan halaman *Home*, dan alur pengambilan stok telah selesai.

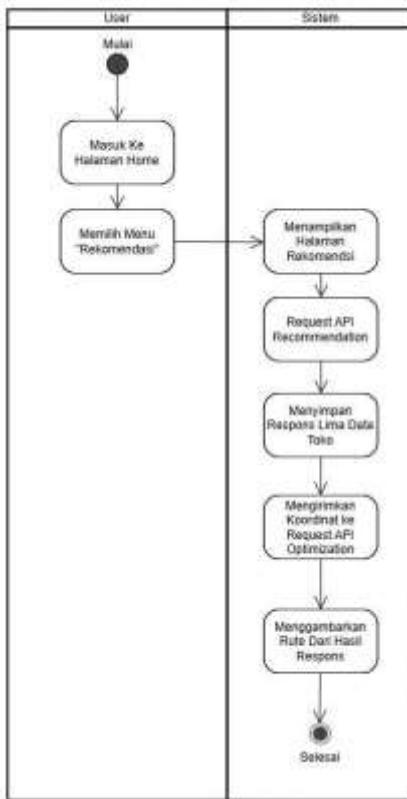
3.2.1.6 Activity Diagram Halaman *Home*



Gambar 3.8 Activity Diagram Halaman *Home*

Activity diagram gambar 3.8 menggambarkan alur *User* setelah berhasil masuk ke halaman *Home* sistem. Begitu *User* berada di halaman *Home*, sistem secara otomatis akan mengambil data stok yang telah diambil sebelumnya dari *server*. Setelah itu, sistem akan melanjutkan dengan mengambil data daftar toko yang akan dikunjungi oleh *sales* dari *server*. Terakhir, sistem akan menampilkan daftar toko yang akan dikunjungi oleh *User*, yang menandai selesainya proses penataan halaman *Home* ini.

3.2.1.7 Activity Diagram Rute Rekomendasi

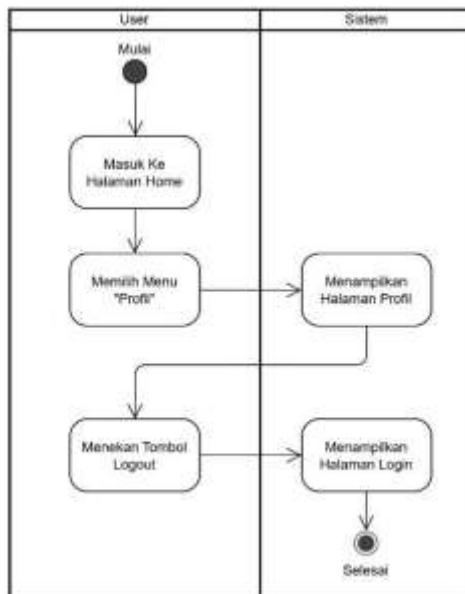


Gambar 3.9 Activity Diagram Rute Rekomendasi

Alur dalam fitur rekomendasi rute pada gambar 3.9 dimulai saat pengguna membuka aplikasi dan masuk ke halaman *Home*. Dari halaman ini, pengguna kemudian memilih menu “Rekomendasi”. Setelah menu dipilih, sistem akan mengambil alih dengan menampilkan halaman rekomendasi dan melakukan *request* ke API rekomendasi untuk mendapatkan daftar toko yang direkomendasikan. Sistem kemudian menyimpan lima data toko hasil dari respon API tersebut. Selanjutnya, sistem mengirimkan koordinat toko-toko tersebut ke API Optimization v1

untuk menghitung rute terbaik. Setelah menerima hasil dari API Optimization v1, sistem akan menggambarkan rute optimal di peta. Proses ini berakhir setelah rute berhasil ditampilkan kepada *user*.

3.2.1.8 Activity Diagram Melakukan Logout

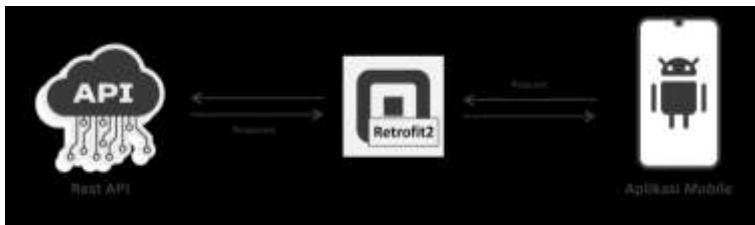


Gambar 3.10 Activity Diagram Melakukan Logout

Activity diagram pada gambar 3.10 menggambarkan alur *User* untuk melakukan *logout* dari sistem. Proses dimulai ketika *User* masuk ke halaman *Home*. Dari halaman *Home*, *User* memilih menu "Profil", yang kemudian akan membuat sistem menampilkan halaman Profil. Di halaman Profil, *User* menekan tombol "Logout". Setelah tombol *Logout* ditekan, sistem akan menampilkan kembali halaman *Login*, yang menandakan selesainya proses *logout* ini.

3.2.2 Perancangan Data

Pada penelitian ini, perancangan data digambarkan pada hubungan aliran data antara Rest API, Retrofit serta Aplikasi *Mobile*.



Gambar 3.11 Perancangan Data

Dalam perancangan data aplikasi *mobile*, REST API berperan sebagai antarmuka yang memfasilitasi komunikasi terstruktur antara aplikasi *mobile* dan server. *Retrofit*, sebuah *library* yang diimplementasikan dalam pengembangan aplikasi *mobile*, digunakan untuk mengirimkan permintaan dan menerima respons dari REST API yang dikembangkan oleh tim *backend*.

Proses pertukaran data dimulai saat aplikasi *mobile* mengirimkan *HTTP request* (seperti *GET*, *POST*, *PUT*, atau *DELETE*) ke REST API menggunakan *Retrofit*. *Retrofit* bertugas mengelola dan memformat permintaan ini agar sesuai dengan spesifikasi REST API. Setelah diterima, REST API akan memproses permintaan berdasarkan *end-point* yang dituju. *End-point* ini merupakan URL yang mengidentifikasi sumber daya atau *function* yang diminta oleh aplikasi *mobile*.

Selanjutnya, REST API mengirimkan respons kembali ke aplikasi *mobile* melalui *Retrofit*. Respons ini dapat berisi data yang diminta, *status code* yang mengindikasikan keberhasilan operasi, atau *error code* jika terjadi kesalahan. Aplikasi *mobile* kemudian menerima respons ini dan melanjutkan proses yang diperlukan, seperti menampilkan data kepada pengguna atau menjalankan *function* lanjutan.

Dengan demikian, REST API, *Retrofit*, dan aplikasi *mobile* saling berinteraksi dalam pertukaran data yang terstruktur. *Retrofit* berfungsi sebagai jembatan yang menghubungkan aplikasi *mobile* dengan REST API, memungkinkan aplikasi untuk mengakses dan berinteraksi dengan sumber daya yang disediakan oleh REST API secara efisien dan terstandardisasi.

3.2.3 Perancangan *User Interface / Mock-up* aplikasi

Perancangan *user interface* digunakan untuk memberikan gambaran secara jelas dan terperinci selama proses penelitian mengenai berbagai aspek yang akan dibuat, dikembangkan, serta disesuaikan oleh peneliti guna memastikan kesesuaian dengan kebutuhan pengguna dan tujuan penelitian.



Gambar 3.12 Rancangan Halaman Splash

Pada aplikasi ini memiliki tampilan splash screen seperti pada gambar 3.12 di atas.



Gambar 3.13 Rancangan Halaman Login

Aplikasi ini memiliki halaman *login* seperti pada gambar 3.13 dengan inputan pengguna seperti *username* dan *password*.



Gambar 3.14 Rancangan Halaman Presensi

Pada gambar 3.14 menggambarkan tampilan halaman presensi dengan masukan berupa gambar sebagai bukti untuk *sales*.



Gambar 3.15 Rancangan Halaman Pengambilan Stok Produk

Setelah *sales* melakukan presensi, selanjutnya akan dilanjutkan dengan menuju form pengambilan produk seperti pada gambar 3.15. Form ini bertujuan untuk mendata berapa banyak produk yang akan dibawa oleh *sales*.



Gambar 3.16 Rancangan Halaman Konfirmasi

Setelah *sales* selesai mendata jumlah produk yang akan dibawa, maka akan muncul halaman konfirmasi seperti pada gambar 3.16 sebagai tanda form berhasil diisi.



Gambar 3.17 Rancangan Halaman Dashboard

Dashboard ini berisi beberapa informasi data seperti nama *sales*, status produk yang berisi total penjualan produk dan toko yang telah dikunjungi, progress penjualan selama satu bulan, jumlah stock, dan rekomendasi toko yang dikunjungi.



Gambar 3.18 Rancangan Halaman Rekomendasi

Fitur rekomendasi rute dapat dilihat pada menu rekomendasi dan akan tampil seperti pada gambar 3.18. Gambar tersebut berisi adanya

visualisasi rute yang harus dikunjungi *sales* dan dibawahnya adalah daftar dari toko yang akan dikunjungi.



Gambar 3.19 Rancangan Halaman Detail Transaksi

Halaman transaksi pada gambar 3.19 berisi nama toko dan juga detail data jumlah produk yang diberikan ke toko tersebut. Jika tidak ada transaksi, maka *sales* bisa memilih checkbox yang bertuliskan tidak ada transaksi untuk memberikan tanda bahwa sudah mengunjungi toko.



Gambar 3.20 Rancangan Halaman Tambah Toko

Gambar 3.20 memperlihatkan halaman tambah toko dimana didalamnya berisi beberapa inputan yang harus diisi oleh *sales* agar dapat mendata detail toko tersebut. Mulai dari detail lokasi, data pedagang tersebut apakah berada didalam atau diluar pasar, nama toko, nama pemilik, no hp dan foto dari toko tersebut.



Gambar 3.21 Rancangan Halaman Profil

Pada menu profil, terdapat halaman yang menunjukkan data diri dari *sales* seperti nama, nomor KTP, nomor Hp, dan email, serta tombol logout.

3.1 Rancangan Pengujian

Rancangan pengujian aplikasi ini akan menggunakan pendekatan Black Box Testing untuk menguji fungsi utama tanpa melihat kode sumber. Pengujian dilakukan dengan memverifikasi apakah aplikasi dapat menentukan rute optimal dari satu toko hingga beberapa toko sesuai dengan skenario pengguna yang bervariasi. Selain itu, uji fungsional akan dilakukan untuk memastikan aplikasi dapat memberikan rekomendasi rute secara akurat dan sesuai dengan data lokasi yang diterima. Pengujian juga mencakup analisis efisiensi jarak yang dibutuhkan tim *sales* sebelum dan sesudah penerapan fitur rekomendasi rute, guna mengukur sejauh mana

fitur ini membantu meningkatkan produktivitas dan efektivitas operasional.

$$Efisiensi = \left(\frac{(Jarak Awal - Jarak Rekomendasi)}{Jarak Awal} \right) \times 100\%$$

Tabel 3.3 Rancangan Pengujian Blackbox

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	Splash Screen	Menjalankan halaman splash screen	User melakukan <i>install</i> program dan menjalankan program	Melanjutkan ke halaman <i>login</i>	-
2	<i>Login Sales</i>	Verifikasi fungsi <i>login</i>	Buka aplikasi HangSambal	Muncul halaman <i>login</i>	-
			Isi formulir <i>login</i> dengan data akun yang telah terdaftar (email dan password yang	Dapat mengisi <i>username</i> dan <i>password</i>	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
sesuai)					
3	<i>Login Sales</i>	Verifikasi data tidak valid saat Log In	Klik tombol "Masuk" pada aplikasi HangSambal	Berhasil masuk kedalam aplikasi	-
			Isi formulir <i>login</i> dengan data akun yang tidak terdaftar (email atau password salah)	Dapat mengisi <i>username</i> dan <i>password</i>	-
4	Presensi Sales	<i>Sales</i> melakukan presensi dengan lokasi yang	Klik tombol "Masuk" pada aplikasi HangSambal	Sistem memberikan pesan error terkait data yang tidak valid	-
			User berhasil melakukan presensi	Menuju halaman presensi	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		sesuai	<i>login</i>		
			Mengambil posisi <i>user</i> berada (Kecamatan)	Sistem mengubah textView sesuai dengan kecamatan <i>user</i> berada	-
			<i>User</i> mengambil gambar untuk bukti presensi	Gambar dapat diambil dan ditampilkan pada halaman presensi	-
			Klik tombol "Simpan"	Data presensi dapat terkirim ke <i>Backend</i>	-
5	Presensi <i>Sales</i>	<i>Sales</i> melakukan presensi dari lokasi yang tidak sesuai dengan lokasi penempatan yang telah	<i>User</i> berhasil melakukan <i>login</i>	Menuju halaman presensi	-
			Mengambil posisi <i>user</i> berada	Sistem mengubah textView	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		ditentukan	(Kecamatan)	sesuai dengan kecamatan <i>user</i> berada	
		<i>User</i> mengambil gambar untuk bukti presensi	Gambar dapat diambil dan ditampilkan pada halaman presensi	-	
		Klik tombol "Simpan"	Sistem memberikan pesan "Kecamatan Tidak Terdaftar" terkait data yang tidak valid	-	
6	Form Pengambilan Produk	<i>Sales</i> melakukan pengambilan produk dari gudang	<i>User</i> mengisi jumlah produk yang akan dibawa	Jumlah produk bisa disesuaikan	-
		Klik tombol	Data stok terkirim ke		

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
			"Simpan"	<i>Backend</i>	
7	<i>Dashboard</i>	<i>Sales</i> membuka halaman utama	<i>User</i> berhasil melakukan <i>login</i> dan presensi	Sistem menampilkan data stok produk dan daftar toko pada halaman utama	-
8	Rekomendasi Rute	Menampilkan hasil rute rekomendasi toko	Menekan tombol menu Rekomendasi	Animasi jalur optimal ke toko ditampilkan pada peta	-
9	Tambah Toko	<i>Sales</i> menambahkan toko baru	Menekan tombol menu Rekomendasi	Halaman rekomendasi ditampilkan	-
			Menekan tombol "Tambah Toko"	Halaman detail toko ditampilkan	-
			Mengisi form detail toko	Form detail toko terisi	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
			Menekan tombol “Simpan”	Menyimpan data ke sever dan menampilkan notifikasi berhasil	-
10	Transaksi Toko	<i>Sales</i> melakukan transaksi	Menekan tombol menu Rekomendasi	Sistem menampilkan halaman rekomendasi	-
			Menekan tombol transaksi yang ada pada setiap toko	Berpindah ke halaman form produk yang dijual	-
			Mengisi form produk yang dijual	Dapat memberikan jumlah produk yang dijual	-
			Menekan tombol “Selanjutnya”	Berpindah ke halaman dokumentasi	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
			Mengambil gambar dokumentasi bukti produk	Dapat mengambil gambar dan ditampilkan pada halaman dokumentasi	-
			Menekan tombol “Kirim”	Menyimpan data ke sever dan menampilkan notifikasi berhasil	-
11	Riwayat Kunjungan	Menampilkan halaman toko yang telah dikunjungi	Menekan tombol menu “Riwayat”	Sistem menampilkan halaman riwayat	-
				Memilih toko yang akan diperiksa	Sistem menampilkan halaman detail toko kunjungan
12	Profil <i>User</i>	Menampilkan data <i>user</i>	Menekan tombol menu “Profil”	Halaman profil ditampilkan	-
13	Logout	<i>Sales</i> menekan	Menekan	Halaman	-

No	Fitur yang Diuji	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
		tombol logout	tombol menu “Profil”	profil ditampilkan	
			Menekan tombol logout	User berhasil keluar dan sistem menampilkan halaman <i>login</i>	-