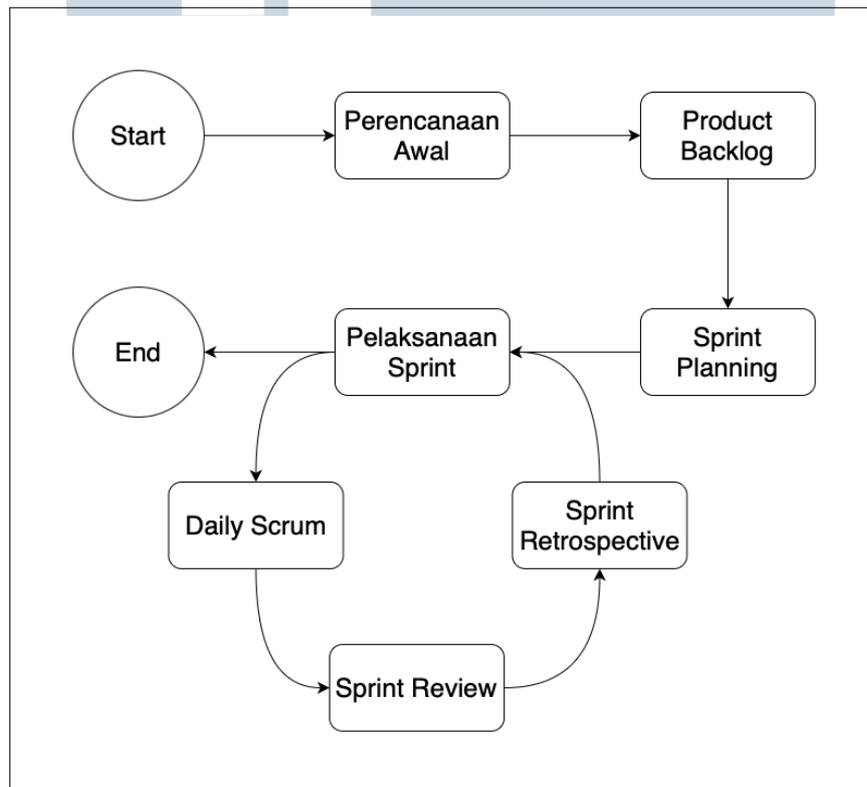


BAB 3 METODOLOGI PENELITIAN

Pengembangan aplikasi dalam penelitian ini dilaksanakan dengan menggunakan metodologi *Scrum* dengan kerangka waktu selama enam minggu yang dibagi menjadi enam *sprint*, di mana setiap *sprint* memiliki durasi satu minggu. Proses pengembangan diawali dengan diskusi yang dilakukan bersama *Product Owner* untuk memahami kebutuhan aplikasi secara menyeluruh. Tahapan kegiatan dalam penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

Penjabaran secara lengkap dari seluruh proses pengembangan akan dicantumkan dalam bagian-bagian selanjutnya, yang mencakup penyusunan *product backlog*, hasil implementasi pada setiap fase *sprint* yang telah dilaksanakan selama proses pengembangan aplikasi, serta dokumentasi teknis berupa *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* yang mendukung desain dan implementasi sistem.

3.1 Product Backlog

Product backlog untuk aplikasi AlfaResto telah disusun berdasarkan hasil analisis kebutuhan yang dilakukan melalui serangkaian diskusi bersama *Product Owner*. Proses ini telah diarahkan untuk memastikan pemahaman menyeluruh terhadap kebutuhan bisnis, ekspektasi pengguna, dan ruang lingkup pengembangan sistem. Kebutuhan-kebutuhan fungsional utama telah diidentifikasi dari sudut pandang customer sebagai aktor tunggal dalam sistem, kemudian telah diformulasikan menjadi *user story* yang merepresentasikan interaksi pengguna dengan aplikasi. Selain itu, aktivitas non-fungsional yang bersifat teknis dan berskala proyek, seperti tahap preparation berupa penyusunan dokumentasi serta tahap finalization berupa proses pengujian dan finalisasi aplikasi, telah dimasukkan ke dalam *product backlog* untuk memastikan bahwa seluruh ruang lingkup pengembangan telah tercakup secara menyeluruh. Daftar lengkap *product backlog* dapat dilihat pada Tabel 3.1.

Tabel 3.1. Daftar Product Backlog

Product Backlog
Pembuatan <i>use case diagram</i> sistem
Pembuatan <i>activity diagram</i> dan <i>sequence diagram</i> aplikasi
Evaluasi dan penentuan teknologi pengembangan
Perancangan <i>mockup</i> antarmuka pengguna
Perancangan struktur dan skema <i>database</i>
Customer dapat mendaftar sebagai member baru
Customer dapat melakukan <i>login</i> menggunakan data kredensial yang dimiliki
Setelah berhasil <i>login</i> , customer dapat melihat tampilan halaman aplikasi dengan dua tab, yaitu Resto dan Order History , dengan tab Resto ditampilkan pertama kali
Customer dapat melihat profil restoran dan daftar menu
Customer dapat melihat nama, deskripsi, dan harga dari setiap menu
Customer dapat mencari menu berdasarkan nama
Customer dapat menambahkan menu ke keranjang dan menyesuaikan jumlahnya
Customer dapat melihat ikon keranjang jika terdapat item di dalam keranjang
Customer dapat mengklik ikon keranjang untuk melihat halaman <i>order summary</i>
Customer dapat menginput atau memilih alamat pengiriman menggunakan fitur penunjukan lokasi
Lanjut pada halaman berikutnya

Tabel 3.1 Daftar Product Backlog (lanjutan)

Product Backlog
Customer dapat memilih metode pembayaran (misalnya: E-Money, Transfer) yang selalu berhasil
Customer dapat menekan tombol "Pesan Sekarang" untuk melakukan pemesanan
Customer dapat melihat daftar <i>order history</i> yang diurutkan berdasarkan tanggal terbaru
Customer dapat mengklik salah satu <i>order history</i> untuk melihat detail pesanan tersebut
Customer dapat melihat pelacakan pengiriman secara <i>real-time</i>
Customer dapat melakukan <i>chat</i> dengan restoran dari halaman detail <i>order history</i>
Customer hanya dapat mengirimkan pesan berupa teks di fitur <i>chat</i>
Customer akan menerima notifikasi ketika pesanan akan diantar oleh kurir
Customer akan menerima notifikasi saat ada pesan <i>chat</i> baru yang masuk
Finalisasi aplikasi dan perbaikan bug
Pengujian seluruh fungsionalitas aplikasi

3.2 Sprint Planning

Berdasarkan daftar *product backlog* yang telah disusun, proses perencanaan *sprint* disepakati melalui diskusi kolaboratif antara tim pengembang dan *Product Owner*. Perencanaan ini mengacu pada tenggat waktu yang ditetapkan serta mempertimbangkan kompleksitas fitur dan kapasitas tim. Setiap *sprint* dirancang berdurasi satu minggu dengan total enam *sprint* yang dilaksanakan secara bertahap. *User stories* didistribusikan sesuai prioritas dan kompleksitas, di mana *sprint* pertama difokuskan pada pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, dan penentuan *tech stack*; *sprint* kedua untuk perancangan antarmuka pengguna dan desain *database*; *sprint* ketiga hingga kelima untuk fase pengkodean dan implementasi fitur utama; serta *sprint* keenam untuk finalisasi dan pengujian aplikasi.

Dalam pelaksanaan setiap *sprint*, *Daily Scrum* dilaksanakan harian untuk memantau kemajuan dan koordinasi tim. *Sprint Review* dan *Sprint Retrospective* dilakukan di akhir setiap *sprint* untuk evaluasi pencapaian dan identifikasi area perbaikan guna memastikan kualitas hasil akhir pengembangan. Untuk mendukung perencanaan tersebut, rangkuman perencanaan aktivitas pengembangan dalam enam *sprint* ditampilkan secara sistematis pada Tabel 3.2 sebagai acuan dalam

penjadwalan pengerjaan fitur serta distribusi beban kerja mingguan.

Tabel 3.2. Sprint Planning Overview

Sprint Planning Overview
Sprint 1
Pembuatan <i>use case diagram</i> untuk dokumentasi kebutuhan fungsional sistem
Pembuatan <i>activity diagram</i> dan <i>sequence diagram</i> untuk visualisasi alur proses bisnis dan teknis aplikasi
Evaluasi dan penentuan teknologi yang akan digunakan dalam pengembangan
Sprint 2
Perancangan tata letak dan <i>mockup</i> antarmuka pengguna aplikasi
Penyusunan struktur koleksi dan dokumen pada <i>database Firestore</i>
Perancangan format dan akses data antar dokumen dalam <i>Firestore</i>
Penyiapan struktur awal UI dan <i>ViewModel</i> untuk halaman utama
Sprint 3
Customer dapat mendaftar sebagai member baru
Customer dapat melakukan <i>login</i> menggunakan data kredensial yang dimiliki
Setelah berhasil <i>login</i> , customer dapat melihat tampilan halaman aplikasi dengan dua tab: Resto dan Order History
Customer dapat melihat profil restoran dan daftar menu
Customer dapat melihat nama, deskripsi, dan harga dari setiap menu
Customer dapat mencari menu berdasarkan nama
Customer dapat menambahkan menu ke keranjang dan menyesuaikan jumlahnya
Sprint 4
Customer dapat melihat ikon keranjang jika terdapat item di dalam keranjang
Customer dapat mengklik ikon keranjang untuk melihat halaman <i>order summary</i>
Customer dapat menginput atau memilih alamat pengiriman menggunakan fitur penunjukan lokasi
Customer dapat memilih metode pembayaran (misalnya: E-Money, Transfer) yang selalu berhasil
Customer dapat menekan tombol "Pesan Sekarang" untuk melakukan pemesanan
Sprint 5
Customer dapat melihat daftar <i>order history</i> yang diurutkan berdasarkan tanggal terbaru
Customer dapat mengklik salah satu <i>order history</i> untuk melihat detail pesanan tersebut
Lanjut pada halaman berikutnya

Tabel 3.2 Sprint Planning Overview (lanjutan)

Sprint Planning Overview
Customer dapat melihat pelacakan pengiriman secara <i>real-time</i>
Customer akan menerima notifikasi ketika pesanan akan diantar oleh kurir
Customer akan menerima notifikasi saat ada pesan <i>chat</i> baru yang masuk
Customer dapat melakukan <i>chat</i> dengan restoran dari halaman detail <i>order history</i>
Customer hanya dapat mengirimkan pesan berupa teks di fitur <i>chat</i>
Sprint 6
Pengujian unit dan integrasi untuk seluruh halaman dan fungsionalitas utama
Simulasi <i>end-to-end</i> dari proses registrasi hingga pemesanan selesai
Pengujian pada skenario <i>edge case</i> dan validasi penanganan error
Perbaikan bug yang ditemukan selama proses pengujian

Berdasarkan pembagian aktivitas yang telah ditetapkan dalam Tabel 3.2, rincian pekerjaan teknis untuk masing-masing *sprint* kemudian diformulasikan. Setiap *user story* dari *product backlog* telah diuraikan ke dalam bentuk tugas-tugas yang lebih terperinci, yang kemudian dituangkan dalam bentuk *sprint backlog*.

Untuk memfasilitasi proses penyusunan *sprint backlog*, setiap *task* dari *user story* yang telah diuraikan kemudian diberi estimasi effort menggunakan pendekatan *story point*. Pendekatan ini merupakan adaptasi dari praktik *Scrum* konvensional, di mana *story point* umumnya diberikan pada level *user stories*. Dalam penelitian ini, estimasi pada tingkat *task* dipilih karena *task* yang ada pada *product backlog* telah memiliki tingkat granularitas yang cukup tinggi. Estimasi effort tersebut dilakukan melalui pendekatan *t-shirt sizing* (S, M, L), yang merepresentasikan tingkat kompleksitas, ketidakpastian, serta effort relatif antar tugas.

Dalam pendekatan ini, ukuran S menunjukkan tugas dengan kompleksitas rendah dan kepastian tinggi. Ukuran M dan L digunakan untuk tugas-tugas yang memiliki tingkat kesulitan yang lebih tinggi, kemungkinan adanya ketergantungan teknis, atau risiko pengembangan. Proses estimasi dilakukan secara kolaboratif oleh seluruh anggota tim pengembang, dengan diskusi terbuka untuk menyepakati ukuran setiap tugas. Masing-masing size tersebut kemudian dikonversikan ke dalam *story point* dengan skala Fibonacci sederhana: S = 1, M = 3, dan L = 5. Penetapan standar ini bertujuan untuk mendukung perencanaan kapasitas *sprint* secara lebih akurat dan realistis.

Standar penentuan size dan konversi *story point* yang digunakan dalam penelitian ini disajikan pada Tabel 3.3.

Tabel 3.3. Kriteria Penilaian Story Point berdasarkan T-Shirt Size

Size	Story Point	Effort	Waktu yang Diperlukan	Kompleksitas dan Risiko
S	1	Minimum effort	Sekitar 0.5 hari kerja	Kompleksitas rendah, tanpa risiko
M	3	Effort ringan–sedang	Sekitar 1–2 hari kerja	Kompleksitas rendah hingga sedang, risiko rendah
L	5	Effort sedang–tinggi	Sekitar 3–5 hari kerja	Kompleksitas sedang, risiko sedang

Dengan mengacu pada standar tersebut, estimasi effort untuk setiap tugas dalam *sprint* telah ditentukan sebelum pelaksanaan, guna memastikan perencanaan kapasitas tim yang lebih terukur.

Fokus Sprint 1 diarahkan pada pengaturan infrastruktur awal seperti repositori proyek dan dokumentasi kebutuhan sistem. Rincian tugas-tugas tersebut dapat dilihat pada Tabel 3.4.

Tabel 3.4. Sprint Backlog 1

Tugas	Ukuran	Story Points
Penyusunan <i>use case diagram</i>	L	5
Pembuatan <i>activity diagram</i> dan <i>sequence diagram</i>	L	5
Evaluasi dan pemilihan <i>tech stack</i>	L	5

Total estimasi effort pada Sprint 1 adalah **15 story point**, terdiri dari 3 tugas berukuran L yang berfokus pada persiapan analisis dan dokumentasi sistem. Setelah tahap persiapan awal diselesaikan, Sprint 2 dirancang untuk memuat aktivitas perancangan struktur *database* dan desain antarmuka pengguna. Rinciannya ditampilkan pada Tabel 3.5.

Tabel 3.5. Sprint Backlog 2

Tugas	Ukuran	Story Points
Desain <i>mockup</i> UI untuk halaman utama aplikasi	L	5
Penyusunan struktur koleksi dan dokumen <i>Firestore</i>	L	5
Perancangan format data dan akses antar dokumen	M	3
Penyiapan struktur awal UI dan ViewModel halaman utama	M	3

Total estimasi effort pada Sprint 2 adalah **16 story point**, dengan kombinasi ukuran M dan L untuk pekerjaan desain UI dan perancangan struktur data. Dengan fondasi desain dan arsitektur yang telah disusun, Sprint 3 difokuskan pada implementasi fitur utama seperti pendaftaran, *login*, navigasi tab, dan penambahan item ke keranjang. Tugas-tugasnya disajikan pada Tabel 3.6.

Tabel 3.6. Sprint Backlog 3

Tugas	Ukuran	Story Points
Implementasi fitur registrasi user	M	3
Implementasi proses <i>login</i> user	M	3
Pembuatan tab navigasi utama aplikasi	S	1
Menampilkan daftar menu restoran	M	3
Menampilkan informasi detail menu (nama, harga, deskripsi)	S	1
Pencarian menu berdasarkan nama	M	3
Penambahan item ke <i>cart</i> dan pengaturan jumlahnya	M	3

Total estimasi effort pada Sprint 3 adalah **17 story point**, dengan dominasi tugas berukuran M yang mencerminkan intensitas pengembangan awal fitur utama aplikasi. Pada Sprint 4, fokus pekerjaan diarahkan pada penyempurnaan alur pemesanan, mulai dari tampilan ikon keranjang hingga pemilihan metode pembayaran. Selain itu, implementasi awal fitur *chat* antara customer dan restoran turut dikembangkan sebagai bentuk interaksi lanjutan yang mendukung kenyamanan pengguna dalam berkomunikasi. Rincian tugas-tugasnya ditampilkan pada Tabel 3.7.

Tabel 3.7. Sprint Backlog 4

Tugas	Ukuran	Story Points
Menampilkan ikon <i>cart</i> secara dinamis	S	1
Menampilkan ringkasan order sebelum pemesanan	S	1
Input alamat pengiriman menggunakan Maps	M	3
Pemilihan metode pembayaran	S	1
Submit order dan update status ke <i>database</i>	M	3
Implementasi fitur <i>chat</i> dengan restoran	M	3
Validasi agar <i>chat</i> hanya mendukung input teks	M	3

Total estimasi effort pada Sprint 4 adalah **15 story point**, didominasi oleh implementasi fitur alur pemesanan dan *chat* yang bersifat interaktif dan fungsional. Fitur pelacakan dan *order history* direncanakan untuk dikembangkan dalam Sprint 5, dengan penekanan pada sinkronisasi data *real-time*, notifikasi berbasis event, serta implementasi lanjutan fitur *chat*. Daftar tugasnya ditampilkan pada Tabel 3.8.

Tabel 3.8. Sprint Backlog 5

Tugas	Ukuran	Story Points
Menampilkan daftar <i>order history</i> user	S	1
Menampilkan detail dari order tertentu	S	1
Pelacakan pengiriman secara <i>real-time</i>	L	5
Notifikasi saat status order menjadi <i>on delivery</i>	L	5
Notifikasi pesan <i>chat</i> masuk	L	5

Total estimasi effort pada Sprint 5 adalah **17 story point**, terdiri dari fitur *real-time* seperti pelacakan pengiriman dan notifikasi, yang membutuhkan integrasi dan sinkronisasi data. Sebagai penutup rangkaian *sprint*, Sprint 6 dialokasikan untuk pelaksanaan pengujian menyeluruh terhadap seluruh fitur yang telah dikembangkan, termasuk pengujian unit dan integrasi, simulasi alur penggunaan aplikasi secara *end-to-end*, serta pengujian terhadap skenario ekstrem. Selain itu, *sprint* ini juga digunakan untuk memperbaiki bug dan menyempurnakan pengalaman pengguna (*UX*) menjelang rilis final aplikasi. Rincian tugas-tugas teknis tersebut dapat ditinjau pada Tabel 3.9.

Tabel 3.9. Sprint Backlog 6

Tugas	Ukuran	Story Points
Pengujian unit dan integrasi seluruh fitur utama	L	5
Simulasi alur lengkap dari <i>register</i> hingga order selesai	M	3
Pengujian <i>edge case</i> dan penanganan error	M	3
Perbaiki bug dan penyempurnaan <i>UX</i>	L	5

Total estimasi effort pada Sprint 6 adalah **16 story point**, mencakup rangkaian aktivitas pengujian dan perbaikan untuk memastikan kualitas dan stabilitas aplikasi sebelum rilis. Dengan demikian, rangkaian *sprint backlog* yang telah disusun menggambarkan proses pengembangan aplikasi secara bertahap, terstruktur, dan sesuai prinsip kerja metodologi *Scrum*.

3.3 Proses Pelaksanaan Sprint

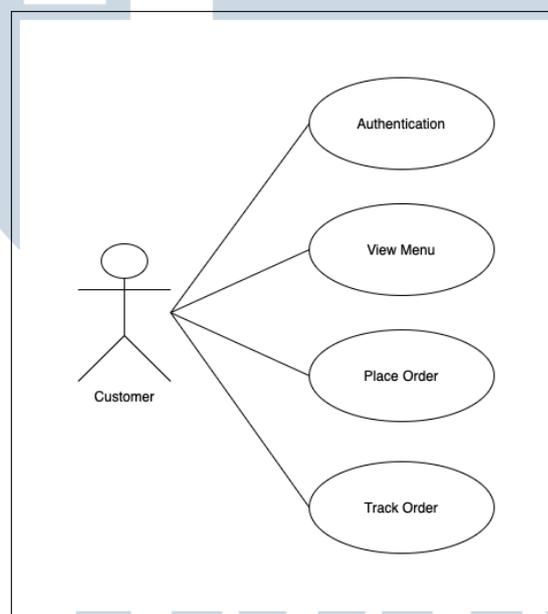
Pelaksanaan enam *sprint* pengembangan aplikasi AlfaResto telah dilaksanakan sesuai dengan perencanaan, dimulai dari fase analisis hingga finalisasi aplikasi. Sprint pertama dan kedua difokuskan pada pembangunan fondasi teknis melalui penyusunan dokumentasi analisis sistem, perancangan desain aplikasi, serta persiapan infrastruktur pengembangan. Sprint ketiga hingga kelima dialokasikan untuk implementasi fitur-fitur utama aplikasi secara bertahap, mulai dari autentikasi hingga pelacakan pesanan dan sistem notifikasi. Sprint terakhir dikhususkan untuk pengujian menyeluruh dan penyempurnaan akhir.

Setiap *sprint* dilengkapi dengan praktik harian dan evaluasi seperti *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* yang merupakan bagian integral dari kerangka kerja *Scrum* [6]. Penjelasan detail mengenai proses pelaksanaan, hambatan yang dihadapi, serta pencapaian yang diperoleh dalam setiap *sprint* diuraikan dalam subbab-subbab berikut.

3.3.1 Sprint 1: System Analysis and Technical Infrastructure Setup

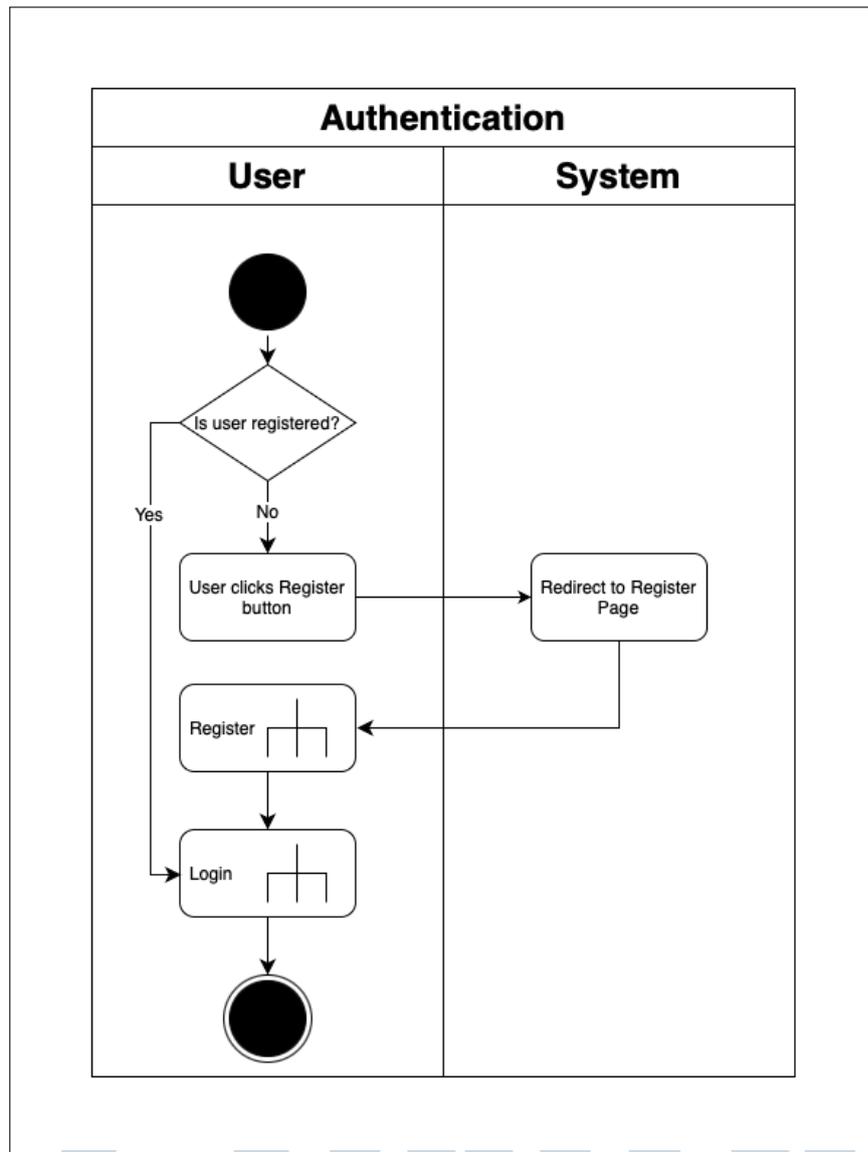
Sprint pertama difokuskan pada analisis sistem, persiapan fondasi teknis pengembangan aplikasi, dan pembangunan infrastruktur pengembangan. *Use case diagram* dibuat untuk mengidentifikasi aktor dan interaksi dalam sistem, *activity diagram* dirancang untuk menggambarkan alur proses bisnis, dan *sequence diagram*

disusun untuk merepresentasikan komunikasi antar objek dalam merealisasikan fungsionalitas sistem yang telah dirancang. Diskusi mendalam dilakukan mengenai teknologi yang akan digunakan dalam pengembangan aplikasi, serta persiapan repositori Git dan struktur proyek Android yang akan menjadi fondasi pengembangan. Keempat *deliverable* utama ini menjadi dasar acuan untuk proses pengembangan pada *sprint-sprint* selanjutnya, dengan tambahan persiapan teknis yang memastikan kesiapan lingkungan pengembangan. Hasil *use case diagram* dapat dilihat pada Gambar 3.2, hasil *activity diagram* ditampilkan pada Gambar 3.3 hingga Gambar 3.9, dan hasil *sequence diagram* disajikan pada Gambar 3.10 hingga Gambar 3.16.



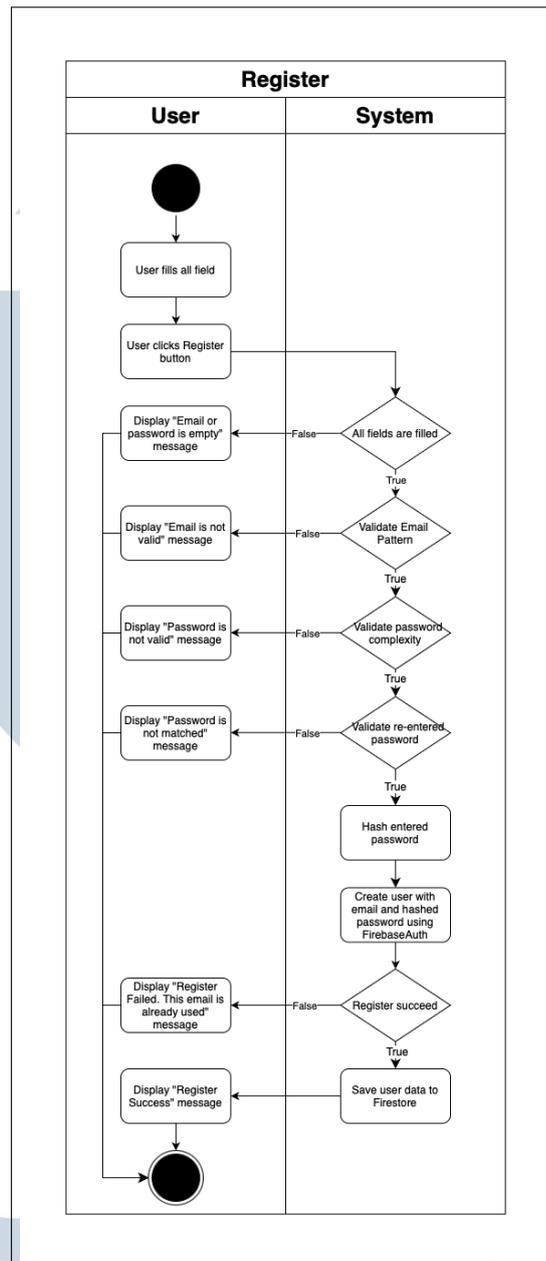
Gambar 3.2. *Use Case Diagram* aplikasi AlfaResto

Use Case Diagram pada Gambar 3.2 menggambarkan interaksi utama antara customer sebagai aktor tunggal dengan sistem AlfaResto yang mencakup empat fungsi utama yaitu Authentication untuk proses masuk sistem, View Menu untuk melihat daftar makanan dan minuman yang tersedia, Place Order untuk melakukan pemesanan, dan Track Order untuk melacak status pesanan yang telah dibuat.



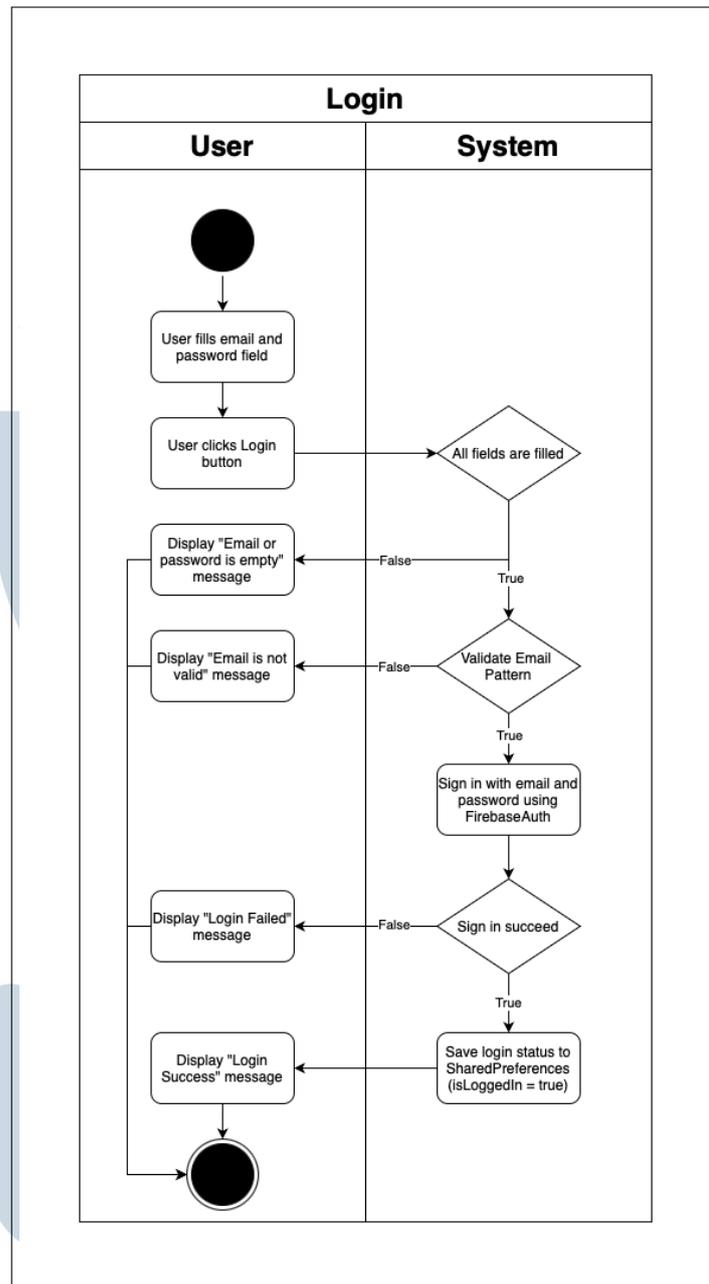
Gambar 3.3. Activity Diagram Authentication

Activity Diagram yang disajikan pada Gambar 3.3 menunjukkan alur proses autentikasi pengguna yang dimulai dengan pengecekan status registrasi pengguna. Jika pengguna belum terdaftar, sistem akan mengarahkannya ke halaman registrasi. Sebaliknya, jika pengguna sudah terdaftar, sistem akan langsung mengarahkannya ke proses masuk untuk mengakses aplikasi.



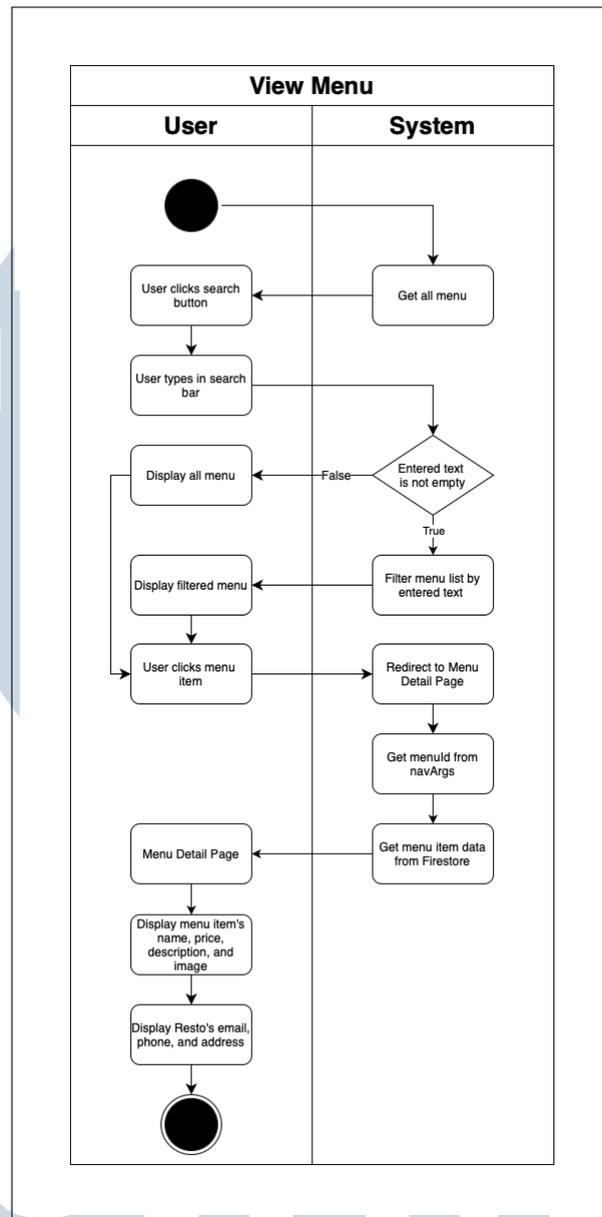
Gambar 3.4. Activity Diagram Register

Berdasarkan Gambar 3.4, proses pendaftaran pengguna baru digambarkan melalui serangkaian tahapan yang meliputi pengisian semua bidang registrasi, validasi kelengkapan data, validasi format surel, validasi kompleksitas kata sandi, validasi konfirmasi kata sandi, proses enkripsi kata sandi, pembuatan akun menggunakan FirebaseAuth, dan penyimpanan data pengguna ke *Firestore*. Setiap tahap validasi disertai dengan pengecekan dan pesan kesalahan yang sesuai untuk memastikan integritas data registrasi.



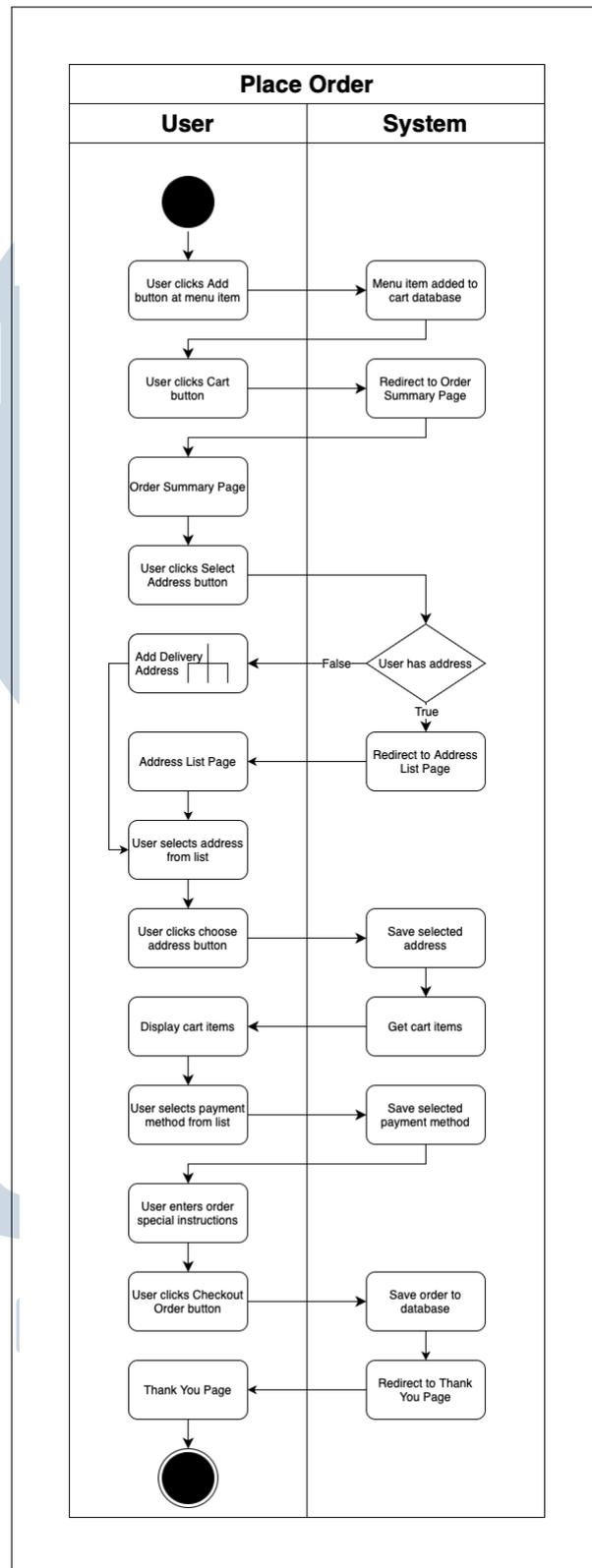
Gambar 3.5. Activity Diagram Login

Alur kerja proses masuk pengguna dijelaskan secara detail dalam Gambar 3.5, yang terdiri dari pengisian surel dan kata sandi, validasi kelengkapan bidang, validasi format surel, proses masuk menggunakan FirebaseAuth, dan penyimpanan status masuk ke *SharedPreferences*. Pesan kesalahan yang sesuai akan ditampilkan apabila terjadi kegagalan pada setiap tahap validasi atau proses autentikasi untuk memberikan panduan yang jelas kepada pengguna.



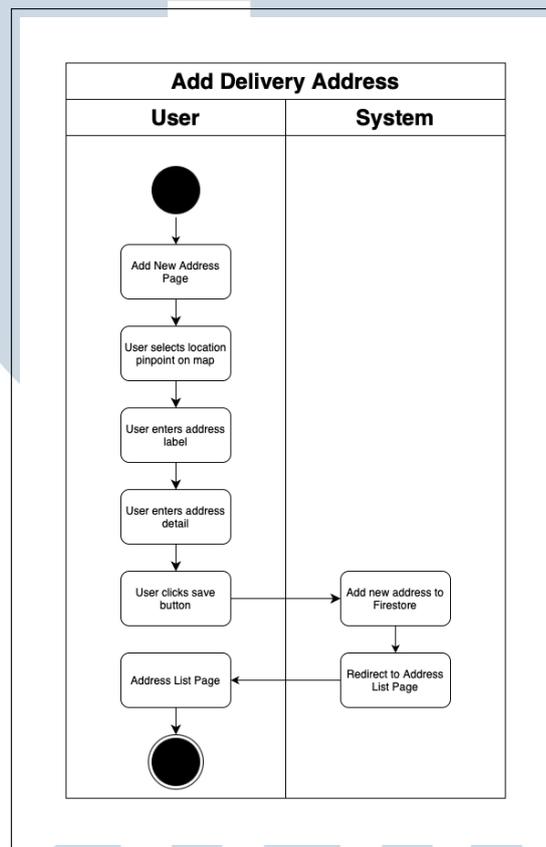
Gambar 3.6. Activity Diagram View Menu

Gambar 3.6 mengilustrasikan proses penampilan menu makanan yang dimulai dengan pengambilan semua data menu dari sistem. Pengguna dapat melakukan pencarian menu melalui bilah pencarian, kemudian sistem akan menyaring menu berdasarkan teks pencarian yang dimasukkan. Ketika pengguna memilih item menu tertentu, sistem akan mengarahkannya ke halaman detail menu yang menampilkan informasi lengkap item tersebut beserta informasi kontak restoran.



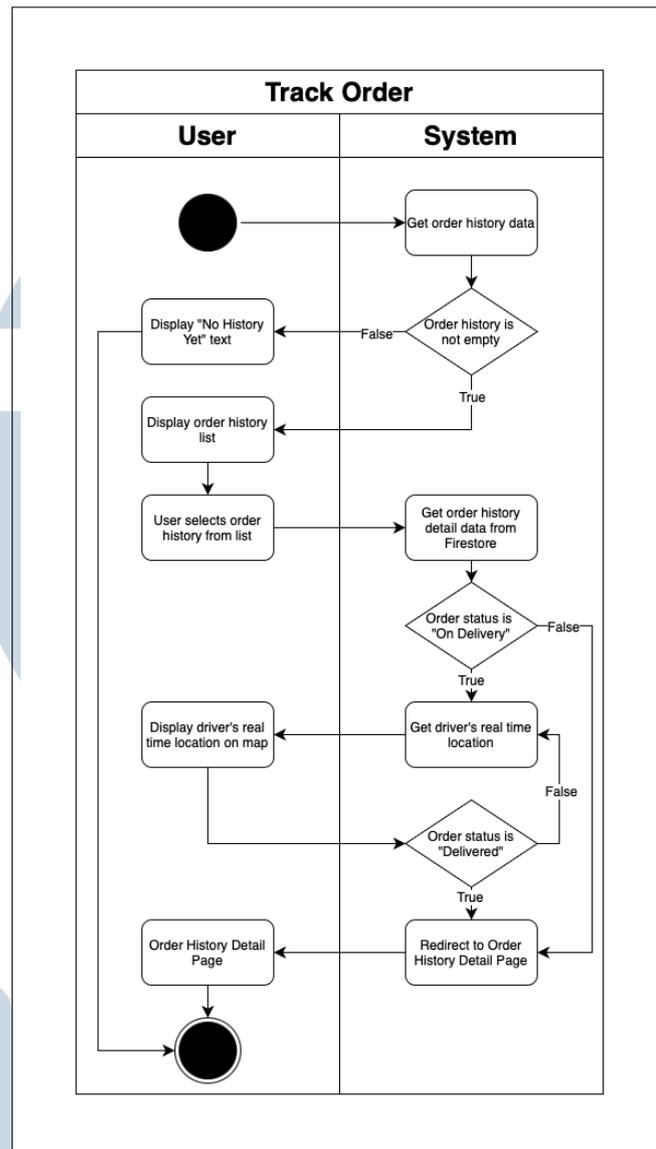
Gambar 3.7. Activity Diagram Place Order

Proses pemesanan pada aplikasi AlfaResto direpresentasikan melalui Gambar 3.7, dimulai dari penambahan item ke keranjang, navigasi ke halaman *order summary*, pengecekan dan pemilihan alamat pengiriman, penampilan item keranjang, pemilihan metode pembayaran, pengisian instruksi khusus, proses checkout yang menyimpan pesanan ke *database*, dan pengalihan ke halaman terima kasih sebagai konfirmasi keberhasilan pemesanan. Setiap langkah dirancang untuk memastikan kelancaran transaksi dan kepuasan pelanggan.



Gambar 3.8. Activity Diagram Add Delivery Address

Mekanisme penambahan alamat pengiriman baru diuraikan dalam Gambar 3.8, yang meliputi navigasi ke halaman tambah alamat, pemilihan lokasi melalui peta interaktif, pengisian label alamat, pengisian detail alamat lengkap, penyimpanan alamat baru ke *Firestore*, dan pengalihan kembali ke halaman daftar alamat untuk menampilkan alamat yang baru ditambahkan. Proses ini dirancang untuk memberikan fleksibilitas kepada pengguna dalam mengelola berbagai alamat pengiriman.

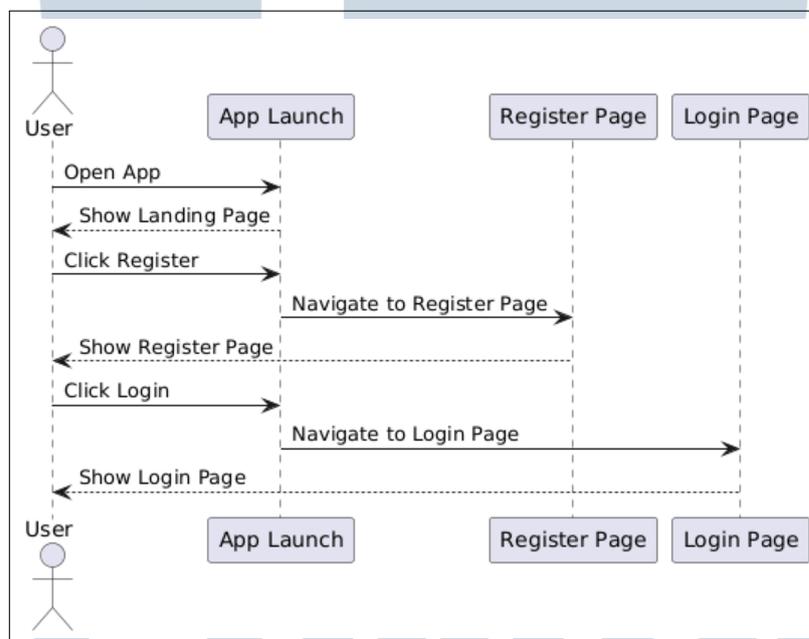


Gambar 3.9. Activity Diagram Track Order

Sistem pelacakan pesanan yang ditampilkan dalam Gambar 3.9 mendeskripsikan proses yang dimulai dengan pengambilan data *order history* dan penampilan daftar pesanan jika tersedia. Setelah pengguna memilih pesanan tertentu untuk melihat detail, sistem akan mengecek status pesanan. Jika pesanan sedang dalam pengiriman, sistem akan menampilkan lokasi pengantar secara *real-time* pada peta. Sebaliknya, jika pesanan sudah selesai, sistem akan mengarahkan pengguna ke halaman detail *order history* untuk memberikan informasi lengkap mengenai transaksi yang telah diselesaikan.

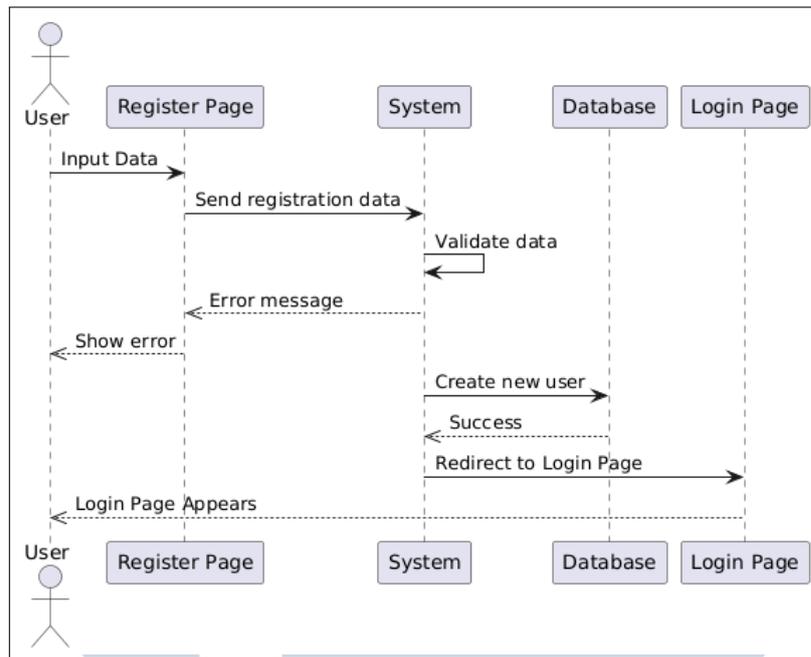
Setelah seluruh alur aktivitas utama pengguna dianalisis dan

direpresentasikan dalam bentuk *activity diagram*, langkah berikutnya adalah mengkaji bagaimana proses-proses tersebut direalisasikan secara teknis melalui interaksi antar objek sistem. Untuk itu, digunakan *sequence diagram* sebagai alat bantu visualisasi yang memperlihatkan urutan komunikasi antar komponen dalam sistem, termasuk antarmuka pengguna, logika aplikasi, layanan *authentication*, dan *database*. Diagram ini menggambarkan secara rinci bagaimana setiap aksi pengguna diterjemahkan ke dalam rangkaian pesan dan pemanggilan fungsi di antara berbagai objek sistem, sehingga dapat memberikan pemahaman yang lebih konkret terhadap implementasi *business logic* dari masing-masing skenario penggunaan yang telah dirancang.



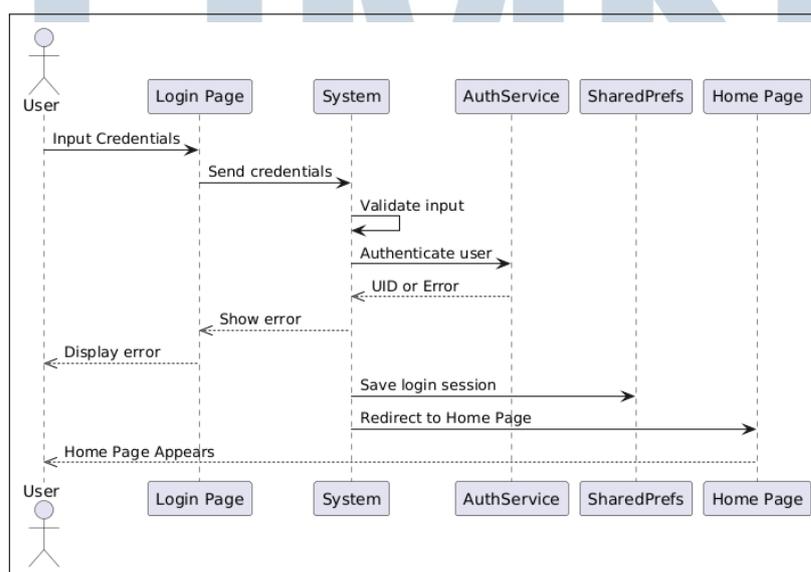
Gambar 3.10. *Sequence Diagram Authentication*

Interaksi awal yang terjadi saat pengguna pertama kali membuka aplikasi AlfaResto tergambar pada Gambar 3.10. Diagram ini memvisualisasikan bagaimana sistem menampilkan *Landing Page* serta memberikan opsi navigasi menuju halaman *Login* atau *Register*, bergantung pada keputusan pengguna. Proses ini sepenuhnya bersifat antarmuka, tanpa melibatkan pemrosesan backend, dan mencerminkan alur kontrol yang bersifat statis namun penting sebagai pintu masuk ke dalam sistem.



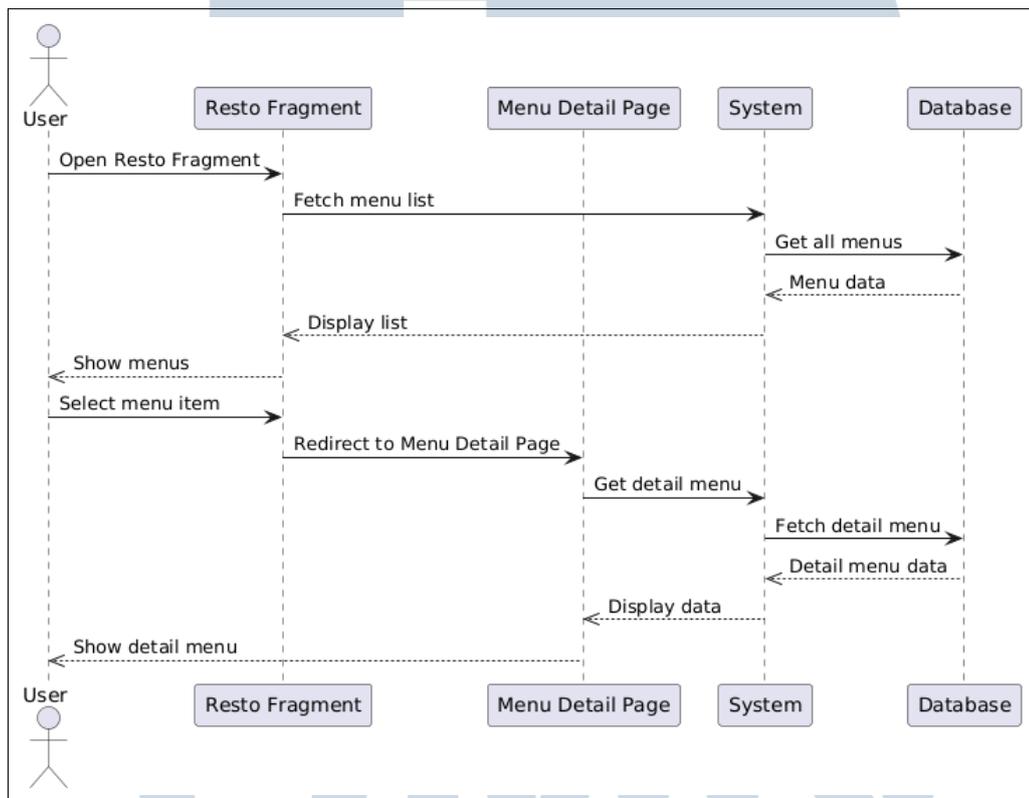
Gambar 3.11. *Sequence Diagram Register*

Selanjutnya, Gambar 3.11 mendokumentasikan alur komunikasi selama proses registrasi akun. *Sequence diagram* ini menyoroti validasi terhadap data masukan pengguna, pemanggilan layanan autentikasi untuk pembuatan akun baru, serta penyimpanan data profil pengguna ke dalam basis data. Interaksi antara *Halaman Registrasi*, *Sistem*, dan *Basis Data* menekankan peran logika aplikasi dalam mengatur jalur respons sukses maupun gagal.



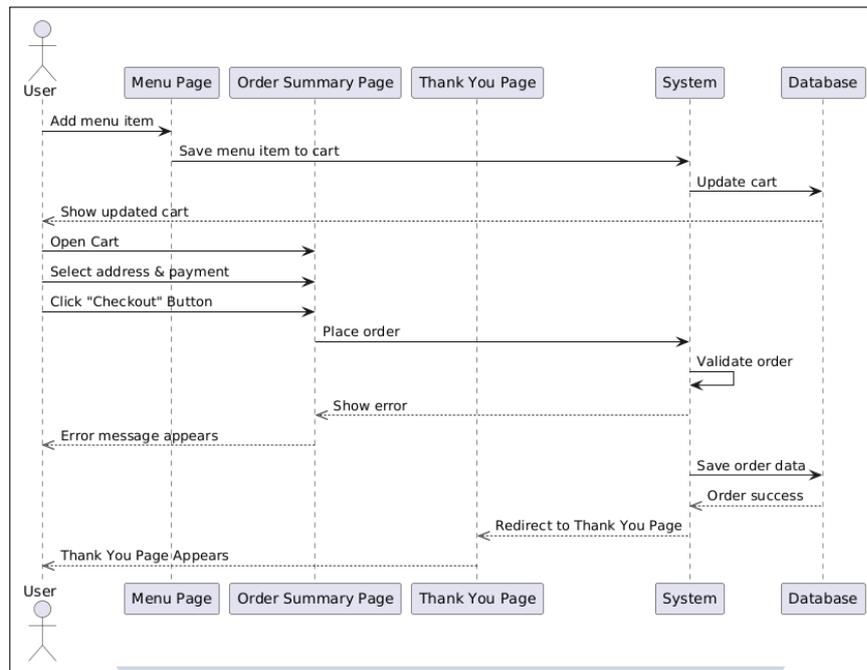
Gambar 3.12. *Sequence Diagram Login*

Sementara itu, proses autentikasi pengguna melalui *login* dijelaskan dalam Gambar 3.12. Pengguna memberikan input kredensial melalui *Halaman Masuk*, yang kemudian diproses oleh *Sistem* untuk validasi dan dikirimkan ke *AuthService*. Jika berhasil, sesi disimpan menggunakan *SharedPreferences*; jika tidak, sistem memberikan umpan balik berupa pesan kesalahan. Diagram ini memperlihatkan bagaimana lapisan logika, antarmuka, dan penyimpanan lokal saling terintegrasi dalam proses otorisasi.



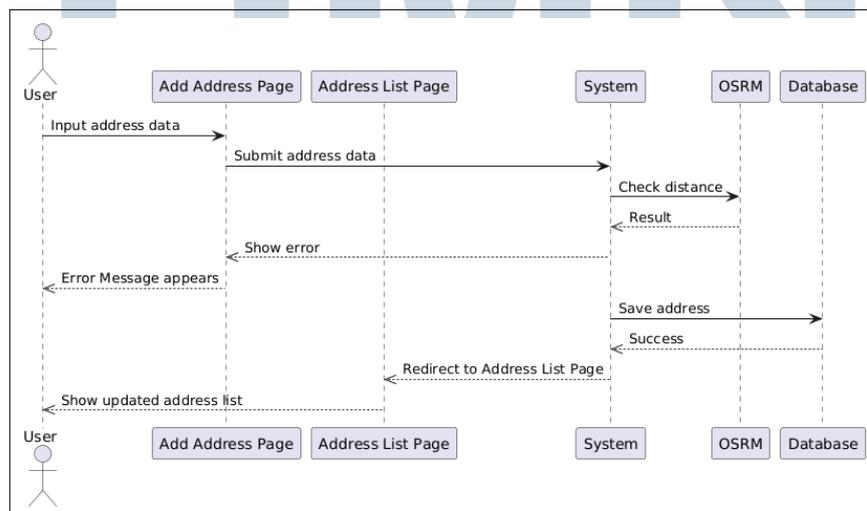
Gambar 3.13. *Sequence Diagram View Menu*

Adapun tahapan penelusuran menu makanan oleh pengguna ditunjukkan melalui Gambar 3.13. Permintaan data dilakukan oleh *Resto Fragment* kepada *Sistem*, yang kemudian mengambil seluruh daftar menu dari *Basis Data*. Ketika pengguna memilih item tertentu, sistem memproses permintaan lanjutan untuk menampilkan detail menu tersebut. Alur ini mencerminkan struktur komunikasi client-server yang responsif dan modular.



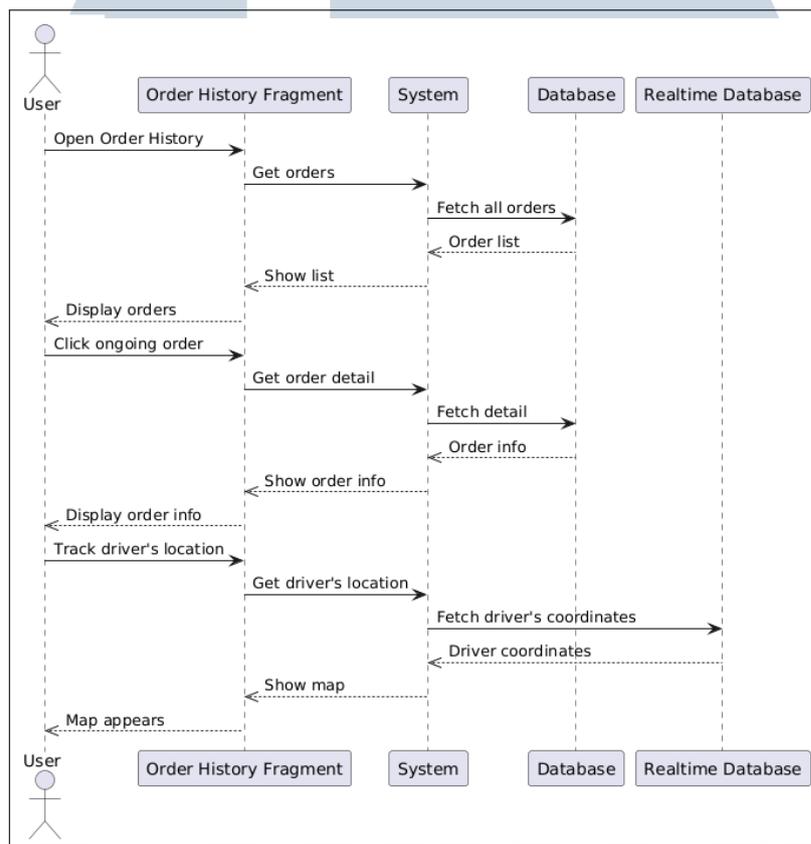
Gambar 3.14. *Sequence Diagram Place Order*

Komunikasi yang lebih kompleks terekam dalam Gambar 3.14, yang menggambarkan urutan aktivitas sistem saat pengguna melakukan pemesanan. Pengguna menambahkan item ke keranjang, sistem memvalidasi kelengkapan data seperti alamat dan metode pembayaran, lalu melanjutkan proses checkout. Setelah pesanan divalidasi dan disimpan ke *Basis Data*, pengguna diarahkan ke halaman konfirmasi. Diagram ini merepresentasikan konsolidasi alur transaksi antara logika bisnis dan data *persistence*.



Gambar 3.15. *Sequence Diagram Add Delivery Address*

Penambahan alamat pengiriman baru oleh pengguna dicerminkan pada Gambar 3.15. Di dalamnya, pengguna mengisi formulir alamat dan memilih lokasi melalui peta. Selanjutnya, *Sistem* melakukan perhitungan jarak menggunakan layanan eksternal, dan jika valid, data disimpan ke dalam *Basis Data*. *Sequence* ini menggambarkan orkestrasi antara sistem utama dan komponen pihak ketiga yang berbasis pengambilan keputusan dinamis.



Gambar 3.16. *Sequence Diagram* Track Order

Terakhir, Gambar 3.16 mengilustrasikan bagaimana sistem menangani pelacakan status pesanan. Dimulai dari pengambilan riwayat pesanan, pengguna memilih pesanan berjalan untuk melihat detail, kemudian sistem menampilkan lokasi kurir secara *real-time* dengan mengambil data dari Realtime Database. Diagram ini memperlihatkan kemampuan sistem untuk menanggapi perubahan status secara terus-menerus dan menyajikan informasi dinamis kepada pengguna.

A Daily Scrum

Daily Scrum Sprint 1 dilaksanakan selama 15 menit untuk sinkronisasi kemajuan dan identifikasi hambatan. Kemajuan pembuatan *use case diagram* dan *activity diagram* dilaporkan secara konsisten oleh setiap anggota tim, dengan rencana kerja harian yang difokuskan pada penyelesaian dokumentasi analisis dan pengaturan infrastruktur pengembangan. Hambatan utama yang diidentifikasi melalui *Daily Scrum* meliputi penentuan cakupan yang tepat untuk use case, kompleksitas dalam visualisasi *activity diagram* yang komprehensif, serta koordinasi dalam pembagian tanggung jawab pengembangan antar anggota tim, yang kemudian ditangani melalui diskusi lanjutan setelah *Daily Scrum*.

B Sprint Review

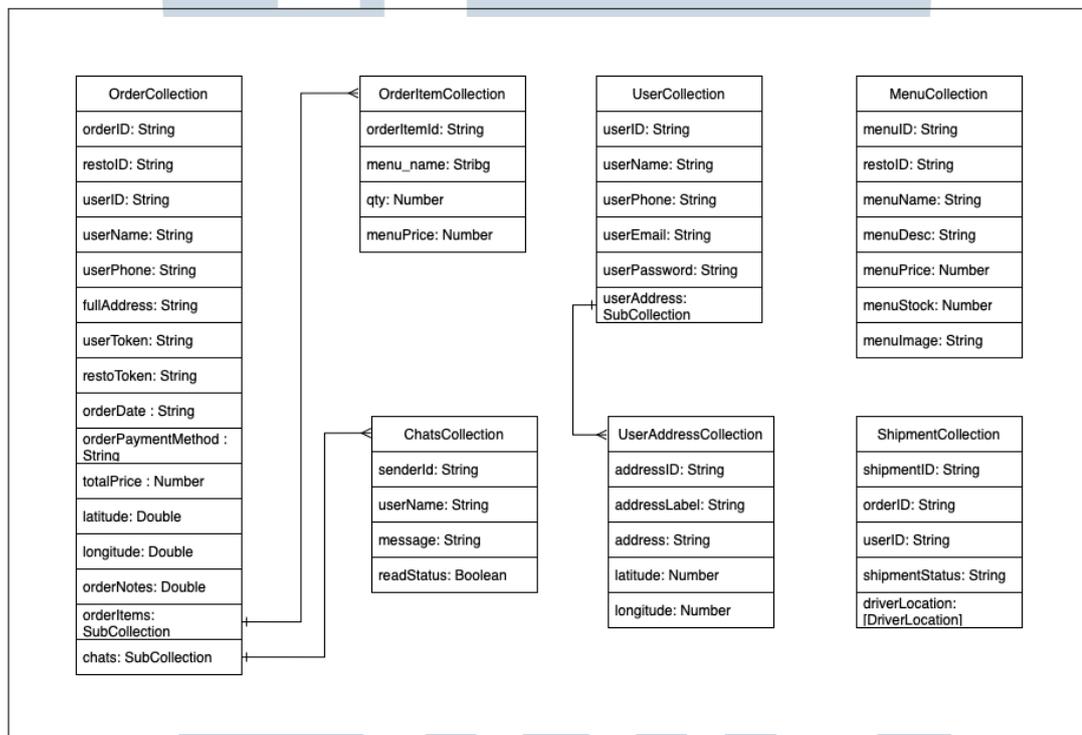
Sprint Review 1 dilaksanakan untuk mendemonstrasikan hasil kerja yang telah diselesaikan kepada *Product Owner*. *Use case diagram* dan *activity diagram* dipresentasikan beserta penjelasan alur proses bisnis yang terdokumentasi, teknologi yang dipilih (*Room, Firestore, Hilt*) didemonstrasikan melalui pengaturan infrastruktur awal, dan struktur proyek Android yang telah disiapkan ditunjukkan sebagai fondasi pengembangan. *Feedback* positif diberikan oleh *Product Owner* terhadap kelengkapan dokumentasi analisis dan pemilihan teknologi yang telah ditetapkan disetujui, serta konfirmasi kesiapan untuk melanjutkan ke fase perancangan pada *sprint* berikutnya dengan fokus pada desain antarmuka pengguna dan arsitektur *database* diberikan.

C Sprint Retrospective

Pada *sprint* pertama, upaya untuk memahami ruang lingkup proyek dan metode kerja *Scrum* dilakukan secara kolaboratif, sehingga proses penyusunan analisis sistem dan persiapan infrastruktur dapat diselesaikan sesuai rencana. Kerja sama tim dalam berbagi informasi dan menyelesaikan dokumentasi berjalan baik. Koordinasi dalam penyesuaian ritme kerja dan penyampaian progres diidentifikasi sebagai aspek yang masih dapat ditingkatkan agar seluruh anggota tim tetap selaras dalam mencapai tujuan *sprint*.

3.3.2 Sprint 2: Design and Database Architecture Planning

Sprint kedua diarahkan pada fase perancangan yang mencakup desain antarmuka pengguna, perancangan struktur *database*, dan konfigurasi arsitektur aplikasi. *Mockup* desain di Figma dikembangkan untuk memberikan gambaran visual aplikasi yang akan dibangun dengan detail yang komprehensif, skema *database* yang mengintegrasikan *Room* dan *Firestore* dirancang untuk mendukung seluruh fungsionalitas sistem, serta *dependency injection* melalui *Hilt* dan *navigation graph* dikonfigurasi untuk memastikan arsitektur yang terstruktur dan mudah dipelihara. Hasil dari *class diagram* dapat dilihat pada Gambar 3.17.



Gambar 3.17. *Class Diagram* aplikasi AlfaResto

Class Diagram pada Gambar 3.17 memvisualisasikan struktur data NoSQL dengan tujuh kelas utama yang dirancang untuk mendukung fungsionalitas sistem secara terintegrasi. *OrderCollection* berfungsi sebagai entitas sentral yang mengelola informasi pesanan dan terhubung dengan koleksi-koleksi lainnya melalui relasi yang telah ditetapkan. Struktur *database* dirancang dengan mempertimbangkan efisiensi query dan skalabilitas sistem untuk mendukung operasional aplikasi food delivery yang responsif dan dapat diandalkan.

A Daily Scrum

Daily Scrum Sprint 2 memfokuskan sinkronisasi kemajuan pembuatan *mockup* antarmuka pengguna di Figma dan perancangan arsitektur *database*. Kemajuan harian dilaporkan oleh tim dalam bentuk penyelesaian *mockup* untuk halaman-halaman utama aplikasi, kemajuan perancangan skema *database Room* dan *Firestore*, serta konfigurasi *navigation graph* dan *dependency injection*. Hambatan yang diidentifikasi meliputi optimisasi struktur *database* untuk kinerja query yang efisien dan koordinasi antara desain visual dengan kebutuhan teknis arsitektur, yang diselesaikan melalui diskusi kolaboratif setelah *Daily Scrum*.

B Sprint Review

Sprint Review 2 dilaksanakan untuk mendemonstrasikan *mockup* lengkap aplikasi di Figma kepada *Product Owner*, menampilkan alur navigasi antar halaman, dan mempresentasikan arsitektur *database* yang telah dirancang. Class diagram dan skema integrasi *Room-Firestore* dijelaskan beserta justifikasi desain untuk mendukung fungsionalitas aplikasi, sementara konfigurasi *dependency injection* dan *navigation graph* didemonstrasikan melalui struktur proyek yang telah disiapkan. Persetujuan diberikan oleh *Product Owner* terhadap desain antarmuka dengan beberapa saran minor untuk penyesuaian tata letak, konfirmasi arsitektur *database* yang dirancang dinyatakan, dan persetujuan transisi ke fase implementasi fitur pada *sprint* berikutnya diberikan.

C Sprint Retrospective

Selama *sprint* kedua, kolaborasi tim semakin terbangun terutama dalam proses perancangan desain antarmuka dan penyusunan arsitektur basis data, sehingga seluruh keluaran yang ditargetkan dapat dicapai secara tepat waktu. Partisipasi aktif anggota dalam diskusi turut mendorong pengambilan keputusan bersama. Diperlukan upaya lanjutan agar keterlibatan seluruh anggota dalam memberikan masukan dapat terwujud secara merata sehingga kualitas rancangan semakin optimal.

3.3.3 Sprint 3: User Authentication and Basic Menu Features

Sprint ketiga menandai dimulainya fase implementasi dengan pengembangan fitur- fitur inti aplikasi yang mencakup sistem autentikasi pengguna komprehensif dan fungsionalitas dasar manajemen menu. Fungsionalitas pendaftaran dengan validasi yang kuat diimplementasikan, sistem *log in* dengan pengelolaan *session* dikembangkan, *splash screen* dengan pemeriksaan *log in* otomatis dibangun, layar utama dengan navigasi tab yang intuitif dikembangkan, fitur pencarian menu berdasarkan nama diintegrasikan untuk membantu pengguna menemukan item secara cepat, tampilan daftar dan informasi detail menu yang mencakup nama, deskripsi, dan harga disajikan secara informatif, logika penambahan item ke keranjang belanja diimplementasikan, serta tampilan ikon keranjang yang responsif dibangun untuk menampilkan status keranjang secara dinamis.

A Daily Scrum

Daily Scrum Sprint 3 berfokus pada sinkronisasi kemajuan implementasi fitur autentikasi dan menu dasar. Kemajuan harian dilaporkan oleh tim dalam bentuk penyelesaian tata letak halaman pendaftaran dan masuk, implementasi logika validasi dan pengelolaan sesi, pengembangan *splash screen* dengan pemeriksaan masuk otomatis, serta integrasi fitur penambahan ke keranjang dengan *database*. Fitur pencarian menu dan tampilan detail informasi menu turut diselesaikan, dengan sinkronisasi antar komponen UI dan query data yang relevan. Hambatan yang diidentifikasi meliputi kompleksitas implementasi validasi regex untuk email dan kata sandi, penanganan manajemen state pada keranjang belanja, serta pengaturan logika tampilan dinamis dalam pencarian dan navigasi detail menu, yang ditangani melalui diskusi teknis setelah *Daily Scrum*.

B Sprint Review

Sprint Review 3 dilaksanakan untuk mendemonstrasikan fungsionalitas sistem autentikasi yang telah diimplementasikan kepada *Product Owner*, termasuk proses pendaftaran dengan validasi lengkap, *log in* dengan *session*, *splash screen* dengan pemeriksaan masuk otomatis, serta antarmuka utama dengan navigasi tab. Fitur pencarian menu dan tampilan informasi detail menu turut didemonstrasikan sebagai bagian dari alur eksplorasi restoran, disertai penambahan item ke

keranjang dan tampilan ikon keranjang yang responsif terhadap perubahan keadaan. *Feedback* positif diberikan terhadap pengalaman pengguna yang informatif dan alur penggunaan yang intuitif, serta konfirmasi kesiapan untuk melanjutkan ke pengembangan fitur pengelolaan pesanan pada *sprint* berikutnya.

C Sprint Retrospective

Pada *sprint* ketiga, tim menunjukkan koordinasi yang lebih baik dalam implementasi fitur autentikasi dan menu, di mana tugas dapat didistribusikan secara jelas antar anggota sehingga pengembangan berjalan efisien. Diskusi harian menjadi media pemantauan kemajuan dan penyelesaian masalah secara kolektif. Standar dokumentasi dan kualitas kode dibahas lebih lanjut agar dapat dipahami dan diterapkan secara konsisten oleh seluruh anggota.

3.3.4 Sprint 4: Order Management and Address Integration

Sprint keempat difokuskan pada pengembangan sistem manajemen pesanan komprehensif dan integrasi alamat pengiriman dengan *Google Maps*. Halaman *order summary* dengan tampilan yang informatif diimplementasikan, sistem penyimpanan dan pengambilan item keranjang menggunakan *database Room* dikembangkan, tampilan daftar keranjang dengan kuantitas dan subtotal yang dinamis dibangun, sistem manajemen alamat dengan logika seleksi diintegrasikan, serta formulir penambahan alamat dengan integrasi *Google Maps* untuk penunjukan lokasi yang akurat dikembangkan; selain itu, fitur *chat* berbasis teks antara customer dan restoran juga diimplementasikan dan divalidasi agar hanya menerima input berupa teks, untuk menjaga kesederhanaan dan keandalan komunikasi dalam konteks pesanan.

A Daily Scrum

Daily Scrum Sprint 4 memfokuskan sinkronisasi kemajuan implementasi fitur pengelolaan pesanan dan integrasi *Google Maps*. Kemajuan harian dilaporkan oleh tim dalam bentuk pengembangan halaman *order summary*, implementasi *database Room* untuk persistensi keranjang, integrasi *API Google Maps* untuk pemilihan alamat, pengembangan formulir input alamat dengan validasi, serta implementasi fitur *chat* antara customer dan restoran. Fitur *chat* diuji agar hanya menerima input teks dan disinkronkan melalui koleksi pesan per pesanan.

Hambatan yang diidentifikasi meliputi pembaruan *real-time* pada keranjang, integrasi Maps dengan penanganan izin lokasi, dan validasi input terbatas pada teks untuk fitur *chat*, yang diselesaikan melalui eksperimen dan eksplorasi dokumentasi eksternal.

B Sprint Review

Sprint Review 4 dilaksanakan untuk mendemonstrasikan alur pesanan lengkap dari pengelolaan keranjang hingga pemilihan alamat dan komunikasi dengan restoran. Halaman *order summary* ditampilkan dengan persistensi keranjang menggunakan *Room*, sistem manajemen alamat didemonstrasikan melalui integrasi *Google Maps*, serta fitur *chat* diuji dengan alur komunikasi berbasis teks antar pengguna dan restoran. Validasi input *chat* berhasil ditunjukkan untuk membatasi pesan hanya berupa teks. *Feedback* positif diberikan terhadap integrasi multi-komponen yang fungsional dan alur pemesanan yang lengkap, serta konfirmasi kesiapan untuk pengembangan fitur pelacakan dan riwayat dinyatakan.

C Sprint Retrospective

Sprint keempat menitikberatkan pada pengembangan fitur manajemen pesanan dan integrasi alamat, dengan kolaborasi tim yang semakin solid saat menghadapi kompleksitas teknis. Kemampuan anggota untuk bekerja sama dalam mengatasi hambatan telah mempercepat penyelesaian pengembangan. Pembagian waktu pengerjaan tugas dan pelaksanaan evaluasi bersama perlu terus dioptimalisasi agar distribusi beban kerja berjalan seimbang.

3.3.5 Sprint 5: Order History and Realtime Tracking

Sprint kelima diarahkan pada pengembangan fitur-fitur pelacakan pesanan *real-time* dan manajemen *order history*. Halaman *order history* dengan tampilan yang terstruktur diimplementasikan, sistem pengambilan dan penampilan data *order history* dari *database* dengan efisien dikembangkan, halaman detail *order history* yang informatif dibangun, tata letak untuk halaman *track order* diimplementasikan, serta sistem pelacakan *real-time* menggunakan *Firebase* dan *OSRM* untuk menampilkan rute pengiriman dan pembaruan lokasi pengantar secara langsung diintegrasikan; di samping itu, sistem notifikasi dirancang untuk memberi peringatan saat status pesanan berubah dan saat ada pesan baru masuk, serta fitur

chat lanjutan dalam halaman *order history* disempurnakan agar pengguna dapat terus berinteraksi dengan restoran secara kontekstual.

A Daily Scrum

Daily Scrum Sprint 5 berfokus pada sinkronisasi kemajuan implementasi fitur pelacakan *real-time* dan pengelolaan *order history*. Kemajuan harian dilaporkan oleh tim dalam bentuk pengembangan halaman *order history*, implementasi *real-time database Firebase* untuk pelacakan, integrasi *API OSRM* untuk perhitungan rute, pengembangan UI tracking dengan pembaruan lokasi, serta integrasi sistem notifikasi untuk status “*on delivery*” dan pesan *chat* masuk. Fitur *chat* dari halaman *order history* juga diaktifkan. Hambatan yang dihadapi mencakup sinkronisasi data *real-time*, pengujian notifikasi antar platform, dan validasi *chat* dengan kondisi *edge case*, yang diatasi melalui eksperimen intensif dan debugging lintas perangkat.

B Sprint Review

Sprint Review 5 dilaksanakan untuk mendemonstrasikan sistem *track order*, *order history*, serta fitur interaktif lanjutan kepada *Product Owner*. Simulasi pelacakan pengiriman dengan pembaruan *real-time* ditampilkan pada antarmuka peta, halaman *order history* disertai detail yang terstruktur diperlihatkan, fitur *chat* dari halaman *order history* diuji, dan sistem notifikasi ditunjukkan berhasil memberikan peringatan pembaruan status pesanan maupun pesan masuk. Integrasi *Firebase* dan *OSRM* dijelaskan sebagai landasan pembaruan *real-time*. *Feedback* sangat positif diberikan terhadap fungsionalitas, stabilitas, dan nilai tambah notifikasi serta *chat* interaktif, dan proyek dinyatakan siap untuk tahap akhir pengujian.

C Sprint Retrospective

Pada *sprint* kelima, keberhasilan dalam implementasi fitur pelacakan pesanan secara *real-time* dan riwayat pesanan didorong oleh kekompakan tim dalam berbagi peran dan mengomunikasikan hambatan yang dihadapi. Kolaborasi aktif dalam pembagian tugas maupun penyelesaian masalah memungkinkan target *sprint* tercapai sesuai harapan. Dukungan tim terhadap anggota yang mengalami kendala menjadi faktor penting dalam menjaga kontribusi optimal hingga akhir *sprint*.

3.3.6 Sprint 6: Comprehensive Testing and Application Finalization

Sprint terakhir difokuskan pada finalisasi aplikasi dan pelaksanaan pengujian menyeluruh untuk memastikan kualitas dan kesiapan aplikasi untuk penyebaran. Pengujian komprehensif yang mencakup penelusuran menyeluruh terhadap seluruh alur aplikasi dilakukan, validasi sinkronisasi data antara *Room* dan *Firestore* dilaksanakan, perbaikan bug kasus tepi dan finalisasi alur yang belum sempurna ditangani, pembersihan aset yang tidak terpakai dan optimisasi sumber daya dilakukan, perbaikan spasi dan tipografi untuk konsistensi visual dilaksanakan, serta pembersihan arsitektur aplikasi untuk kemudahan pemeliharaan yang optimal diterapkan.

A Daily Scrum

Daily Scrum Sprint 6 berfokus pada sinkronisasi kemajuan pengujian komprehensif dan penyelesaian bug. Kemajuan harian dilaporkan oleh tim dalam bentuk penyelesaian pengujian untuk setiap modul aplikasi, identifikasi dan penyelesaian bug yang ditemukan selama fase pengujian, serta optimisasi kinerja untuk pengalaman pengguna yang optimal. Implementasi butir tindakan dari Sprint 5 dipantau, di mana panduan peningkatan pengalaman pengguna menghasilkan antarmuka pelacakan yang lebih menarik dengan peningkatan kepuasan pengguna. Hambatan yang diidentifikasi meliputi kompleksitas dalam reproduksi beberapa kasus tepi yang spesifik, koordinasi jadwal pengujian dengan ketersediaan pemangku kepentingan, dan penyeimbangan antara perbaikan bug dengan optimisasi kinerja dalam jadwal waktu yang terbatas, yang ditangani melalui matriks prioritas berdasarkan tingkat keparahan dan dampak terhadap pengalaman pengguna.

B Sprint Review

Sprint Review 6 dilaksanakan untuk mendemonstrasikan aplikasi AlfaResto yang telah final dan siap untuk penyebaran kepada *Product Owner*. Pengujian ujung ke ujung lengkap didemonstrasikan mulai dari pendaftaran pengguna hingga pelacakan pesanan, optimisasi kinerja ditunjukkan melalui antarmuka pengguna yang responsif dan pemuatan data yang efisien, serta sinkronisasi data antara *Room* dan *Firestore* divalidasi untuk memastikan konsistensi dan keandalan. Pengujian penerimaan komprehensif dilakukan oleh *Product Owner* terhadap

seluruh fungsionalitas aplikasi, persetujuan final diberikan untuk rilis aplikasi berdasarkan hasil pengujian yang memuaskan dan demonstrasi yang komprehensif, serta konfirmasi bahwa aplikasi telah memenuhi seluruh kriteria penerimaan yang ditetapkan di awal proyek dinyatakan.

C Sprint Retrospective

Pada *sprint* terakhir, integrasi hasil kerja seluruh tim dalam fase pengujian dan finalisasi aplikasi menunjukkan efektivitas kerja sama yang telah terbangun sejak awal proyek. Komunikasi yang lebih terbuka serta praktik saling membantu berperan penting dalam penyelesaian perbaikan akhir. Pengetahuan yang diperoleh selama proses *scrum* telah mulai didokumentasikan untuk mendukung keberlanjutan kualitas kerja sama pada proyek-proyek berikutnya.

