

BAB IV HASIL DAN PEMBAHASAN

4.1 Analisis Perancangan Sistem

Analisis perancangan sistem adalah proses yang dilakukan untuk memeriksa dan mengevaluasi desain suatu sistem. Tujuannya adalah untuk memastikan bahwa desain sistem yang telah dibuat secara konseptual dapat memenuhi kebutuhan dan persyaratan yang telah ditetapkan sebelumnya. Setelah mengidentifikasi permasalahan yang terjadi dalam sistem yang sedang berjalan dan melakukan finalisasi terkait kebutuhan pengguna untuk sistem usulan pada bab sebelumnya, langkah selanjutnya adalah melakukan perancangan sistem dengan mempertimbangkan kebutuhan pengguna terhadap sistem usulan tersebut. Pada pelaksanaan analisis dan perancangan sistem untuk sistem informasi pemesanan, digunakan pendekatan metode *agile* dengan model *scrum*. Metode ini dipilih karena dianggap cocok untuk menghadapi perubahan dalam proses bisnis di masa depan. Dengan menggunakan pendekatan ini, diharapkan sistem dapat dengan mudah beradaptasi dengan perubahan yang mungkin terjadi. Berikut ini adalah *product backlog* yang terdapat pada sistem informasi pemesanan makanan yang ditampilkan pada Tabel 4.1 di bawah ini.

Tabel 4.1 *Backlog* Sistem Informasi Pemesanan Makanan

No	<i>Backlog</i>	<i>Item</i>
1	Register	<ul style="list-style-type: none">- Desain tampilan halaman pendaftaran- Implementasi <i>form</i> pendaftaran pengguna baru- Validasi input data pendaftaran- Penyimpanan data pendaftaran ke <i>database</i>
2	Login	<ul style="list-style-type: none">- Desain tampilan halaman <i>login</i>- Implementasi <i>form login</i> pengguna- Validasi data <i>login</i> pengguna- Otentikasi pengguna
3	Halaman <i>Dashboard</i>	<ul style="list-style-type: none">- Desain tampilan halaman <i>dashboard</i>- Implementasi tampilan informasi pengguna- Tampilan menu navigasi dan fitur terkait
4	Daftar Menu	<ul style="list-style-type: none">- Desain tampilan halaman daftar menu- Implementasi tampilan daftar menu restoran

5	Proses Pemesanan	<ul style="list-style-type: none"> - Desain tampilan halaman pemesanan - Implementasi pemilihan makanan - Penambahan item makanan ke keranjang belanja - Validasi dan perhitungan total pesanan
6	Konfirmasi Pesanan	<ul style="list-style-type: none"> - Desain tampilan halaman konfirmasi pesanan - Implementasi penampilan rincian pesanan
7	Pembayaran	<ul style="list-style-type: none"> - Desain tampilan halaman pembayaran - Implementasi pilihan metode pembayaran - Validasi data pembayaran - Konfirmasi pembayaran dan notifikasi
8	Riwayat Pemesanan	<ul style="list-style-type: none"> - Desain tampilan halaman riwayat pemesanan - Implementasi penampilan daftar pemesanan sebelumnya - Tampilan detail pemesanan
9	<i>Enhancement</i> Daftar Menu	<ul style="list-style-type: none"> - Menghilangkan <i>cart icon</i> pada <i>header</i> di Halaman Daftar Menu - Menampilkan <i>hanging cart button</i>
10	<i>Enhancement</i> Proses Pemesanan	<ul style="list-style-type: none"> - Menambahkan fitur tambah catatan - Menambahkan fitur ubah pesanan

Dari tabel yang tertera pada Tabel 4.1 di atas, terlihat bahwa untuk menyelesaikan semua *backlog* yang telah ditentukan, diperlukan sebanyak 3 *product backlog*. Tahap berikutnya adalah merencanakan *sprint*. *Sprint* yang dibentuk akan disusun seperti yang terlihat pada Tabel 4.2 di bawah ini.

Tabel 4.2 *Sprint* Sistem Informasi Pemesanan Makanan

<i>Sprint</i>	<i>Product Backlog</i>	Estimasi Waktu (hari)
1	Register	10
2	Login	10
3	<ul style="list-style-type: none"> - Halaman <i>Dashboard</i> - Daftar Menu 	10
4	<ul style="list-style-type: none"> - Proses Pemesanan - Konfirmasi Pesanan 	10
5	Pembayaran	10
6	Riwayat Pemesanan	10
7	<ul style="list-style-type: none"> - <i>Enhancement</i> Daftar Menu - <i>Enhancement</i> Proses Pemesanan 	10
Total Waktu		70

Dari Tabel 4.2 yang tertera di atas, dapat kita lihat bahwa untuk menyelesaikan menyelesaikan seluruh *Sprint* membutuhkan waktu selama 70 hari.

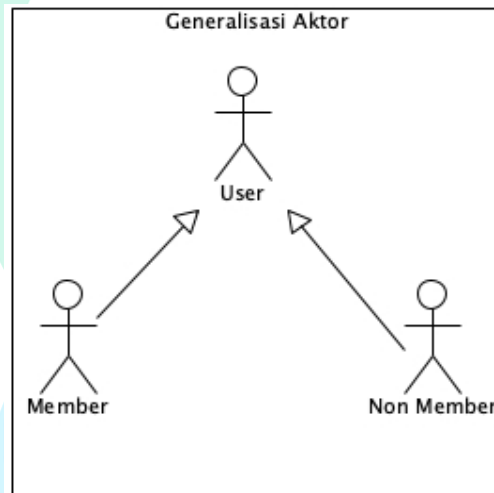
4.2 Perancangan Diagram Sistem Usulan

Dalam pengembangan dan pembuatan sistem informasi pemesanan makanan, digunakan pendekatan metode *Object Oriented Analysis Design* (OOAD)

untuk perancangan diagram sistem usulan. Metode ini mencakup beberapa jenis diagram, termasuk *use case*, *use case specification*, *activity diagram*, *sequence diagram*, *class diagram*, dan *table specification*. Berikut adalah desain diagram usulan untuk sistem informasi pemesanan makanan di restoran Ismaya Group:

4.2.1 Generalisasi Actor

Generalisasi *actor* merujuk pada ide bahwa hubungan antara aktor yang berperan dalam suatu sistem dapat diabstraksikan menjadi jenis aktor yang lebih umum. Tujuan dari generalisasi ini adalah untuk menyederhanakan proses desain. Dalam hal ini, *actor* yang memiliki tugas yang serupa dalam sistem akan digeneralisasikan dan dinyatakan secara umum, seperti yang ditunjukkan dalam Gambar 4.1 ini.



Gambar 4.1 Generalisasi Actor

Berikut penjelasan identifikasi *actor* pada Gambar 4.1 di atas yang dapat dilihat pada Tabel 4.3 di bawah ini.

Tabel 4.3 Pendefinisian Actor

No	Actor	Deskripsi
1	Non Member	Actor yang belum mempunyai akun dan melakukan registrasi pada sistem informasi pemesanan makanan
2	Member	Actor yang sudah mempunyai akun dan melakukan pemesanan
3	ESB Service	Actor yang menampilkan menu yang ada di restoran, menerima pesanan, dan juga

4.2.2 Use Case Diagram

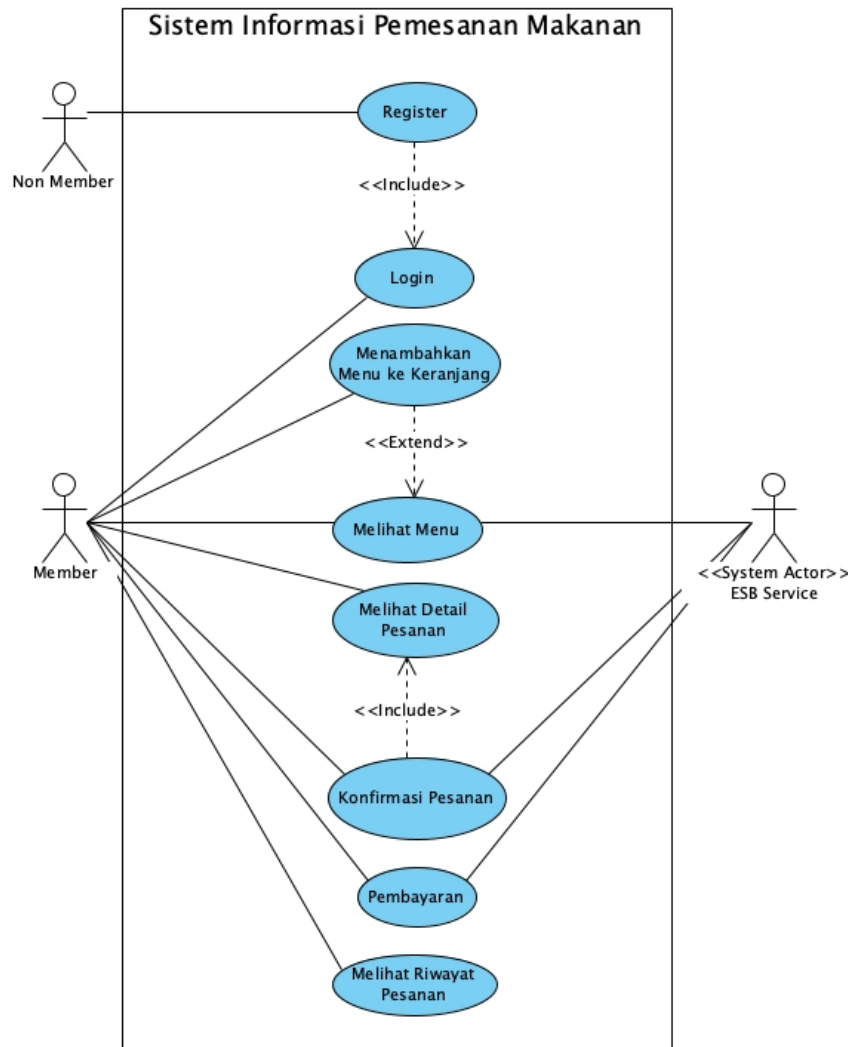
Diagram *use case* digunakan untuk menggambarkan interaksi antara aktor-aktor dengan sistem informasi di restoran Ismaya Group. Langkah

selanjutnya setelah mengidentifikasi aktor adalah pendefinisian *use case* yang akan dibahas pada Tabel 4.4 di bawah ini

Tabel 4.4 Pendefinisian *Use Case*

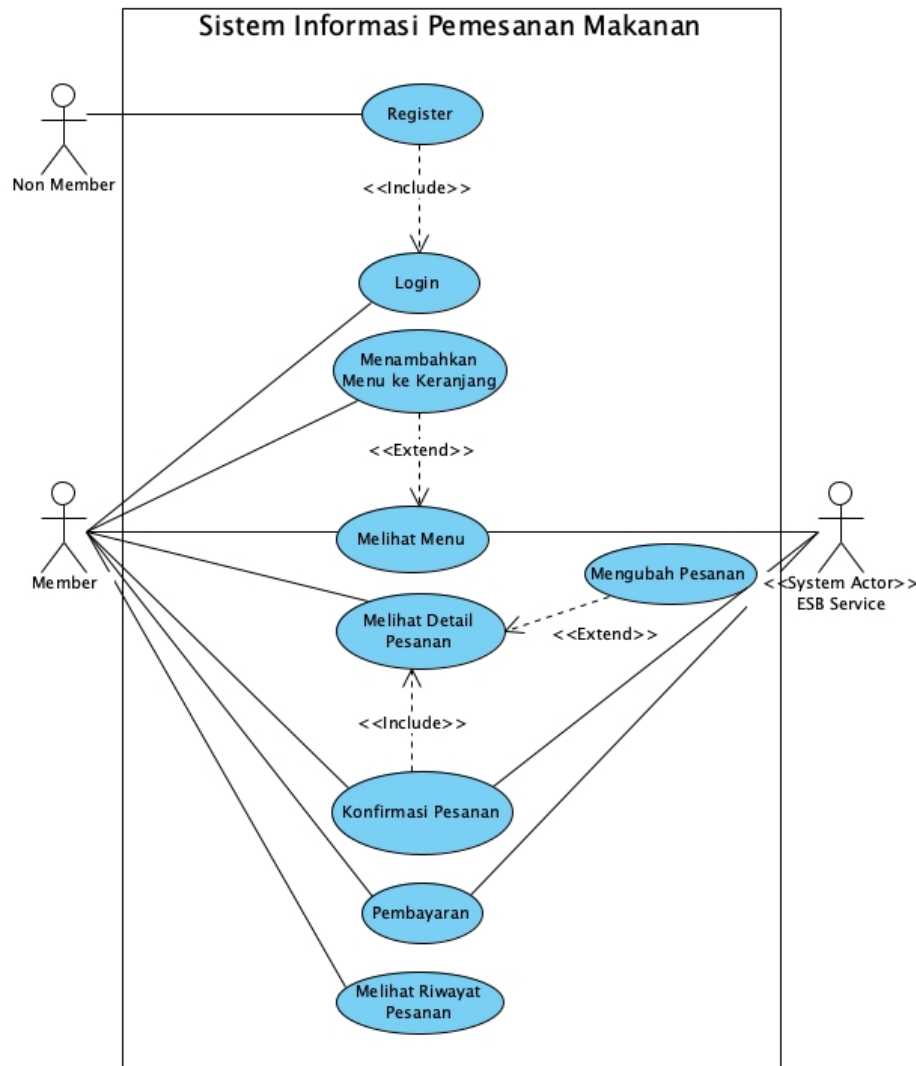
No	Use Case	Deskripsi
1	Login	Aktivitas masuk kedalam sistem yang dilakukan oleh member yang sudah terdaftar
2	Register	Aktivitas pendaftaran ke sistem informasi pemesanan makanan
3	Melihat Menu	Aktivitas melihat daftar menu yang tersedia di restoran
4	Menambahkan Menu ke Keranjang	Aktivitas menambahkan pesanan ke dalam keranjang
5	Melihat Detail Pesanan	Aktivitas melihat detail pesanan di dalam keranjang sebelum dikonfirmasi
6	Mengubah Pesanan	Aktivitas mengubah / kostumisasi pesanan yang belum dikonfirmasi
7	Konfirmasi Pesanan	Aktivitas melakukan konfirmasi pesanan yang akan diproses oleh restoran
8	Pembayaran	Aktivitas melakukan pembayaran untuk pesanan yang sudah dikonfirmasi
9	Melihat Riwayat Pemesanan	Aktivitas melihat Riwayat pemesanan yang sudah dibuat sebelumnya

Dari pendefinisian *Use Case* pada Tabel 4.4 di atas maka *Use Case Diagram* Fase 1 yang penulis usulkan dalam pembuatan aplikasi pemesanan makanan di restoran Ismaya Group dapat dilihat pada Gambar 4.2 di bawah ini.



Gambar 4.2 *Use Case Diagram* Pemesanan Makanan Fase 1

Setelah melakukan evaluasi pada tahap *review* maka *Use Case Diagram* Fase 2 dapat dilihat pada Gambar 4.3 di bawah ini.



Gambar 4.3 Use Case Diagram Pemesanan Makanan Fase 2

Berikut adalah penjelasan aktivitas yang dilakukan oleh aktor pada Gambar 4.2 dan Gambar 4.3 di atas.

1. Register

Aktor *Non Member* yang belum mempunyai akun dan ingin memesan makanan, terlebih dahulu harus melakukan registrasi.

2. Login

Aktor *Member* yang sudah mempunyai akun diharuskan *login* terlebih dahulu untuk menjalankan aplikasi. Aktor dapat melakukan *login* dengan memasukkan nomor telepon yang sudah terdaftar.

3. Melihat Menu

Ketika *Member* pertama kali datang ke restoran, mereka bisa melihat menu-menu yang ada di restoran tersebut dengan melakukan *scan QR* yang diberikan oleh pelayan.

4. Menambahkan Menu ke Keranjang

Setelah melihat menu, *Member* bisa menambahkan pesanan ke keranjang dengan mengatur variant dan qty.

5. Melihat Detail Pesanan

Setelah menambahkan pesanan ke keranjang, *Member* juga bisa melihat detail pesanan yang sudah mereka tambahkan sebelumnya.

6. Mengubah Pesanan

Sebelum melakukan konfirmasi pesanan, *Member* bisa mengubah pesanan mereka pada saat melihat detail pesanan.

7. Konfirmasi Pesanan

Jika *Member* sudah yakin dengan pesannya, mereka bisa konfirmasi pesanan, lalu data pesanan tersebut akan masuk ke sistem ESB yang nantinya akan diproses oleh kasir, lalu pada aplikasi muncul tagihan yang harus dibayar.

8. Pembayaran

Setelah *Member* selesai menikmati pesannya, mereka bisa melakukan pembayaran tagihan langsung dari aplikasi, dan opsi pembayaran bisa menggunakan metode online langsung dari aplikasi ataupun bayar di kasir.

9. Melihat Riwayat Pesanan

Member yang telah menyelesaikan pesanan bisa melihat Riwayat pesanan yang mereka buat.

4.2.3 Spesifikasi *Use Case Diagram*

Spesifikasi *use case* merujuk pada penjelasan mengenai alur atau langkah dari setiap use case yang terdapat dalam skenario. Dalam aplikasi pemesanan makanan ini, penulis telah mengembangkan spesifikasi *use case* yang terperinci, yang dapat ditemukan pada Tabel 4.5 hingga Tabel 4.13 di bawah ini.

Tabel 4.5 Spesifikasi *Use Case Register*

Name	<i>Register</i>	
Actor	Non Member	
Trigger	Aktor ingin melakukan registrasi	
Pre-Condition	Aktor belum mempunyai akun	
Post-Condition	Sistem menampilkan halaman utama	
Success Scenario		
	<i>Actor</i>	<i>System</i>
	1. <i>User</i> memasukkan nama, nomor telepon yang aktif, dan kode <i>referral</i> jika ada	
	2. <i>User</i> klik <i>Get OTP</i>	3. Muncul pilihan OTP dikirim melalui SMS/ <i>WhatsApp</i>
	4. <i>User</i> memilih salah satu opsi pengiriman OTP	5. Sistem mengirimkan OTP
	6. <i>User</i> memasukkan OTP	7. OTP valid, sistem menampilkan halaman utama
Alternative Scenario		
	1. <i>User</i> memasukkan OTP yang salah	2. OTP tidak valid, user gagal masuk ke dalam sistem

Tabel 4.6 Spesifikasi *Use Case Login*

Name	<i>Login</i>	
Actor	Member	
Trigger	Aktor ingin melakukan login	
Pre-Condition	Aktor sudah mempunyai akun	
Post-Condition	Sistem menampilkan halaman utama	
Success Scenario		
	<i>Actor</i>	<i>System</i>
	1. <i>User</i> memasukkan nomor telepon yang sudah terdaftar	
	2. <i>User</i> klik <i>Get OTP</i>	3. Muncul pilihan OTP dikirim melalui SMS/ <i>WhatsApp</i>
	4. <i>User</i> memilih salah satu opsi pengiriman OTP	5. Sistem mengirimkan OTP
	6. <i>User</i> memasukkan OTP	7. OTP valid, sistem menampilkan halaman utama
Alternative Scenario		
	1. <i>User</i> memasukkan nomor telepon yang belum terdaftar	2. Muncul pesan error bahwa nomor belum terdaftar
	3. <i>User</i> memasukkan OTP yang salah	4. OTP tidak valid, user gagal masuk ke dalam sistem

Tabel 4.7 Spesifikasi *Use Case Melihat Menu*

Name	Melihat Menu
Actor	Member dan ESB <i>System</i>
Trigger	Aktor ingin melihat menu yang tersedia

Pre-Condition	Aktor sudah masuk ke halaman utama
Post-Condition	Sistem menampilkan halaman daftar menu
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik menu <i>Scan QR</i>	2. Sistem menampilkan halaman <i>Scan QR</i>
3. <i>User scan QR code</i> yang diberikan oleh kasir	4. Hasil QR valid, sistem menampilkan daftar menu yang tersedia
Alternative Scenario	
1. <i>User</i> mencoba <i>scan</i> sembarang <i>QR code</i>	2. Hasil QR tidak valid, sistem memunculkan pesan error

Tabel 4.8 Spesifikasi *Use Case* Menambahkan Menu ke Keranjang

Name	Menambahkan Menu ke Keranjang
Actor	Member
Trigger	Aktor ingin menambahkan pesanan baru ke keranjang
Pre-Condition	Aktor sudah masuk ke halaman daftar menu
Post-Condition	Sistem menampilkan halaman detail pesanan
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> memilih menu yang akan dipesan	2. Sistem menampilkan detail menu
3. <i>User</i> memilih variant yang ada	
4. <i>User</i> menambahkan catatan	
5. <i>User</i> mengatur qty	
6. <i>User</i> klik <i>Add to cart</i>	7. Sistem menampilkan halaman detail pesanan
Alternative Scenario	
-	-

Tabel 4.9 Spesifikasi *Use Case* Melihat Detail Pesanan

Name	Melihat Detail Pesanan
Actor	Member
Trigger	Aktor ingin melihat detail pesanan sebelum dikonfirmasi
Pre-Condition	Aktor sudah masuk ke halaman daftar menu
Post-Condition	Sistem menampilkan halaman detail pesanan
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik <i>Order Now</i> pada <i>hanging button</i> yang ada di halaman daftar menu	2. Sistem menampilkan halaman detail pesanan yang belum dikonfirmasi
Alternative Scenario	
-	-

Tabel 4.10 Spesifikasi *Use Case* Mengubah Pesanan

Name	Mengubah Pesanan
-------------	------------------

Actor	Member
Trigger	Aktor ingin mengubah pesanan di halaman detail pesanan yang belum dikonfirmasi
Pre-Condition	Aktor sudah masuk ke halaman detail pesanan setelah membuat pesanan
Post-Condition	Sistem menampilkan halaman pesanan yang sudah diperbarui
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik tombol edit pada pesanan yang ingin diubah	2. Sistem menampilkan halaman detail menu sesuai dengan pesanan yang sudah dibuat sebelumnya
3. <i>User</i> mengubah varian	
4. <i>User</i> menambahkan catatan	
5. <i>User</i> mengatur qty	
6. <i>User</i> klik <i>Update cart</i>	7. Sistem menampilkan halaman detail pesanan yang sudah diperbarui
Alternative Scenario	
-	-

Tabel 4.11 Spesifikasi *Use Case* Konfirmasi Pesanan

Name	Konfirmasi Pesanan
Actor	Member dan ESB <i>Service</i>
Trigger	Aktor ingin konfirmasi pesanan agar bisa diproses oleh restoran
Pre-Condition	Aktor sudah masuk ke halaman detail pesanan yang belum dikonfirmasi
Post-Condition	Sistem menampilkan halaman pesanan diproses
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik tombol <i>Order Now</i> di halaman detail pesanan	2. Sistem menampilkan halaman konfirmasi pesanan
3. Pada halaman konfirmasi, <i>user</i> klik <i>Proceed</i>	4. Sistem akan memproses pesanan ke ESB <i>Service</i>
	5. Sistem menampilkan halaman sukses pesanan terkirim
Alternative Scenario	
-	-

Tabel 4.12 Spesifikasi *Use Case* Pembayaran

Name	Pembayaran
Actor	Member dan ESB <i>Service</i>
Trigger	Aktor ingin membayar tagihan
Pre-Condition	Aktor melakukan konfirmasi pesanan
Post-Condition	Sistem menampilkan halaman sukses pembayaran

Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik tombol <i>On-going Order</i> di halaman utama	2. Sistem menampilkan list order yang dibuat oleh <i>user</i>
3. <i>User</i> klik <i>order</i> sesuai dengan restoran tersebut	4. Sistem menampilkan halaman detail tagihan
5. <i>User</i> memilih metode pembayaran	6. Sistem menampilkan daftar metode pembayaran yang ada
7. <i>User</i> memilih salah satu metode pembayaran, lalu klik <i>Pay Now</i>	8. Sistem menampilkan halaman proses pembayaran
	9. Sistem menampilkan pesan sukses
Alternative Scenario	
1. <i>User</i> memilih metode pembayaran online (OVO, GoPay) namun saldo tidak mencukupi	2. Sistem menampilkan pesan error

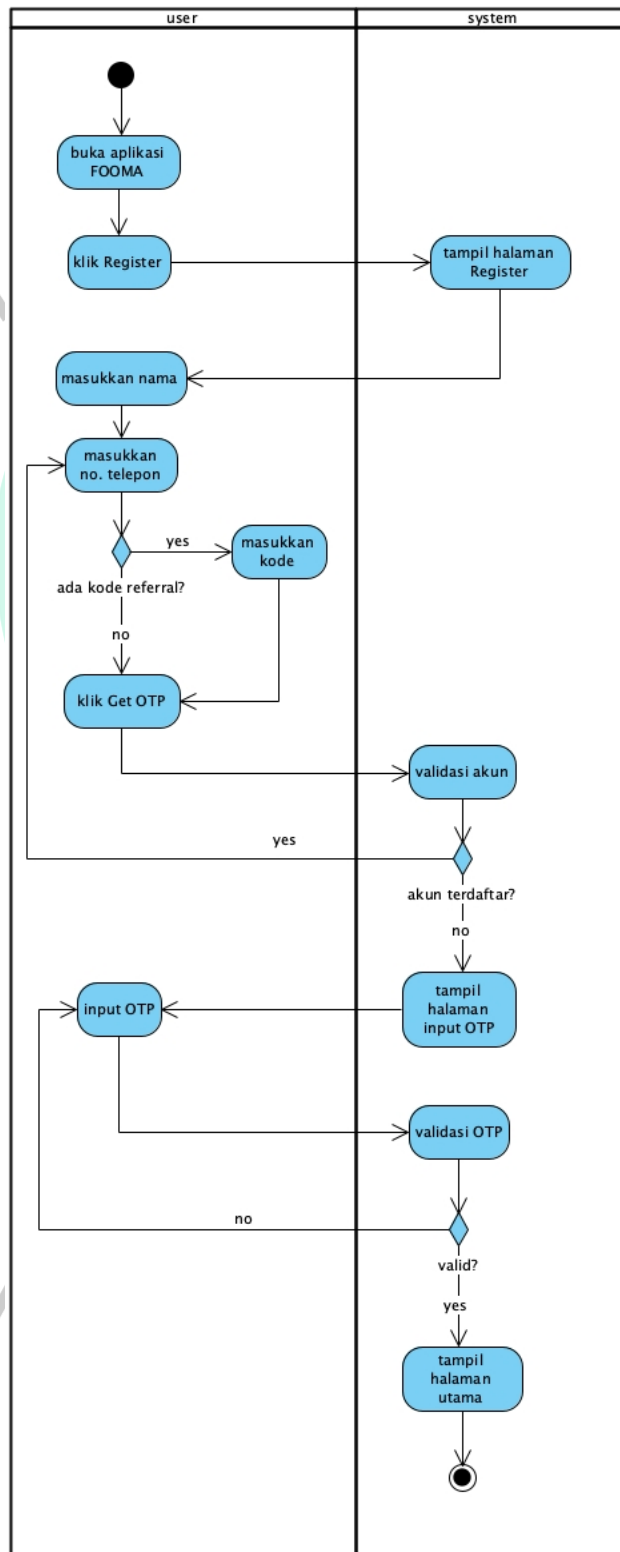
Tabel 4.13 Spesifikasi *Use Case* Melihat Riwayat Pesanan

Name	Melihat Riwayat Pesanan
Actor	Member
Trigger	Aktor ingin melihat Riwayat pesanan yang pernah dibuat
Pre-Condition	Aktor sudah melakukan <i>login</i> dan masuk ke halaman utama
Post-Condition	Sistem menampilkan daftar Riwayat pesanan
Success Scenario	
<i>Actor</i>	<i>System</i>
1. <i>User</i> klik tombol <i>Profile</i> di navigator bawah	2. Sistem menampilkan halaman <i>Profile</i>
3. <i>User</i> klik <i>My Order</i>	4. Sistem menampilkan halaman daftar <i>order</i>
Alternative Scenario	
-	-

4.2.4 Activity Diagram

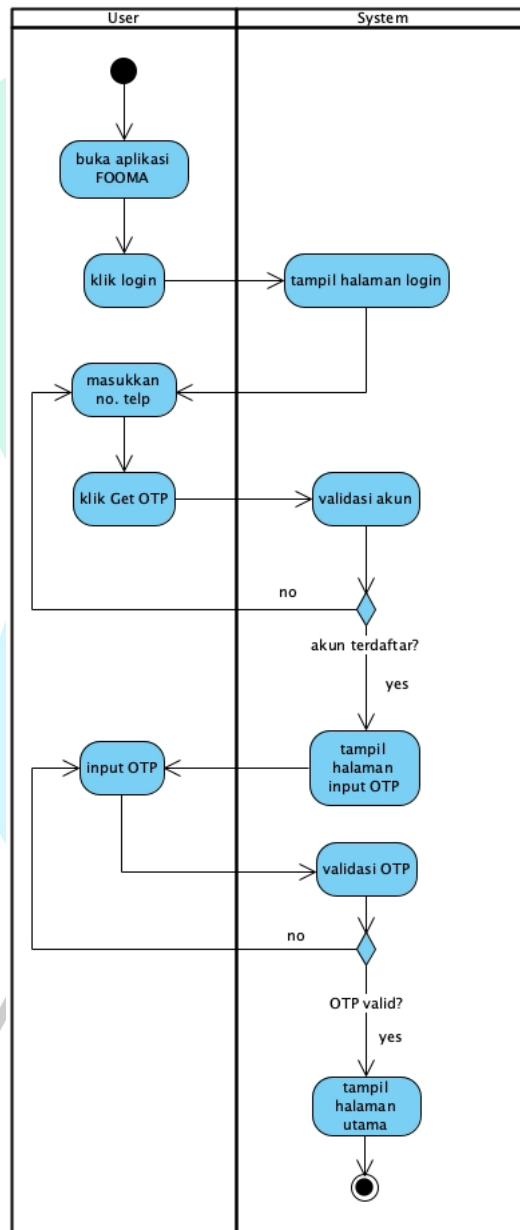
Activity Diagram adalah suatu bentuk perancangan yang digunakan untuk menggambarkan aliran aktivitas secara umum dalam suatu sistem. Tujuannya adalah untuk memberikan gambaran visual tentang bagaimana aktivitas-aktivitas dalam sistem berinteraksi dan mengalir satu sama lain. Berikut adalah contoh perancangan *activity diagram* pada sistem informasi

pemesanan makanan yang menggambarkan alur aktivitas dalam sistem tersebut.



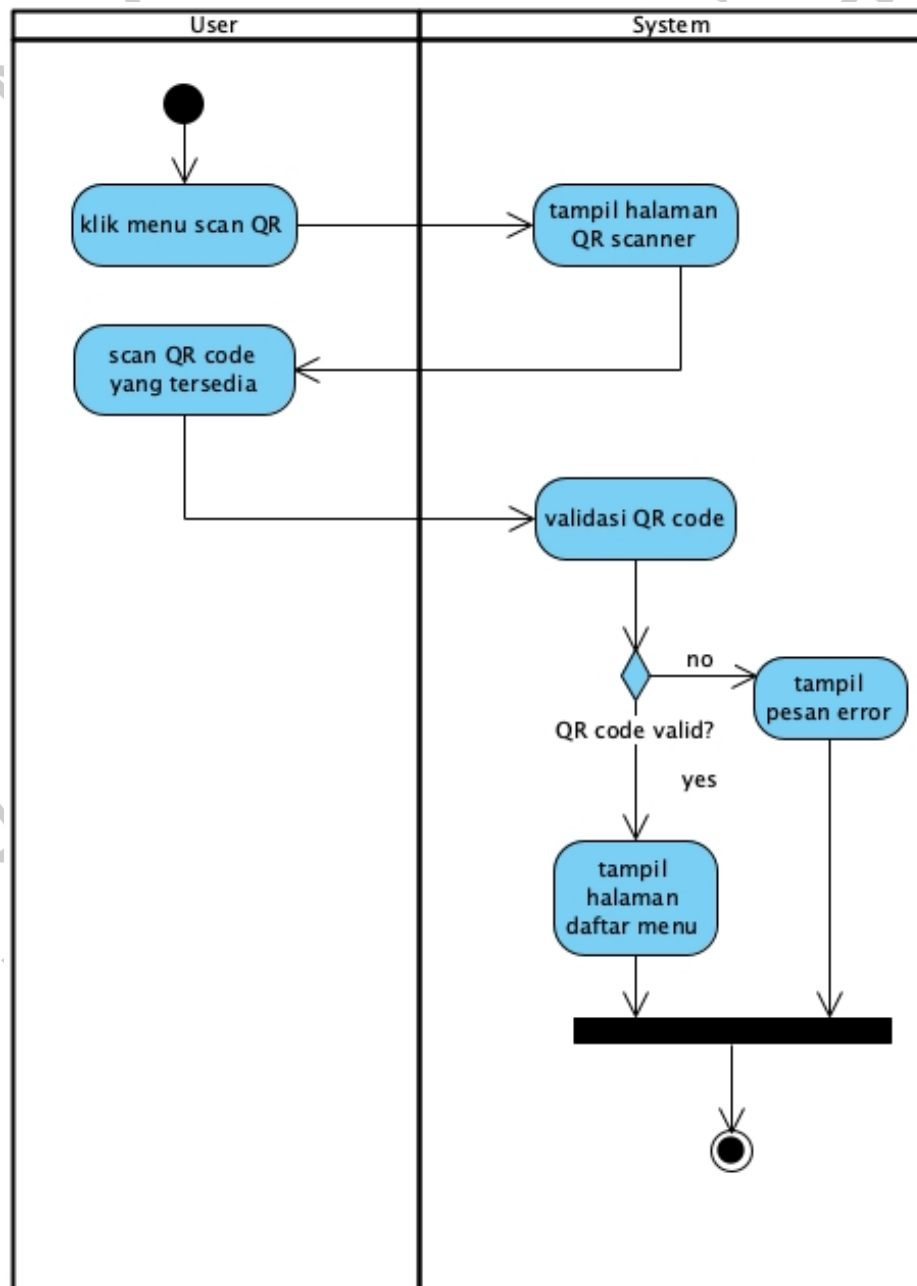
Gambar 4.4 Activity Diagram Registrasi

Pada Gambar 4.4 di atas menunjukkan aktifitas *registrasi* diawali dengan *user* yang belum mempunyai akun harus melakukan *registrasi* di aplikasi dengan klik *registrasi*. Lalu *user* memasukkan nama dan nomor telepon, jika *user* mempunyai kode referral bisa dimasukkan pada field kode referral. Lalu *user* klik tombol *Get OTP*, dan sistem akan mengirimkan kode OTP. Setelah itu *user* memasukkan OTP yang telah diterima tadi, sistem akan melakukan validasi OTP, jika valid maka sistem akan menampilkan halaman utama.



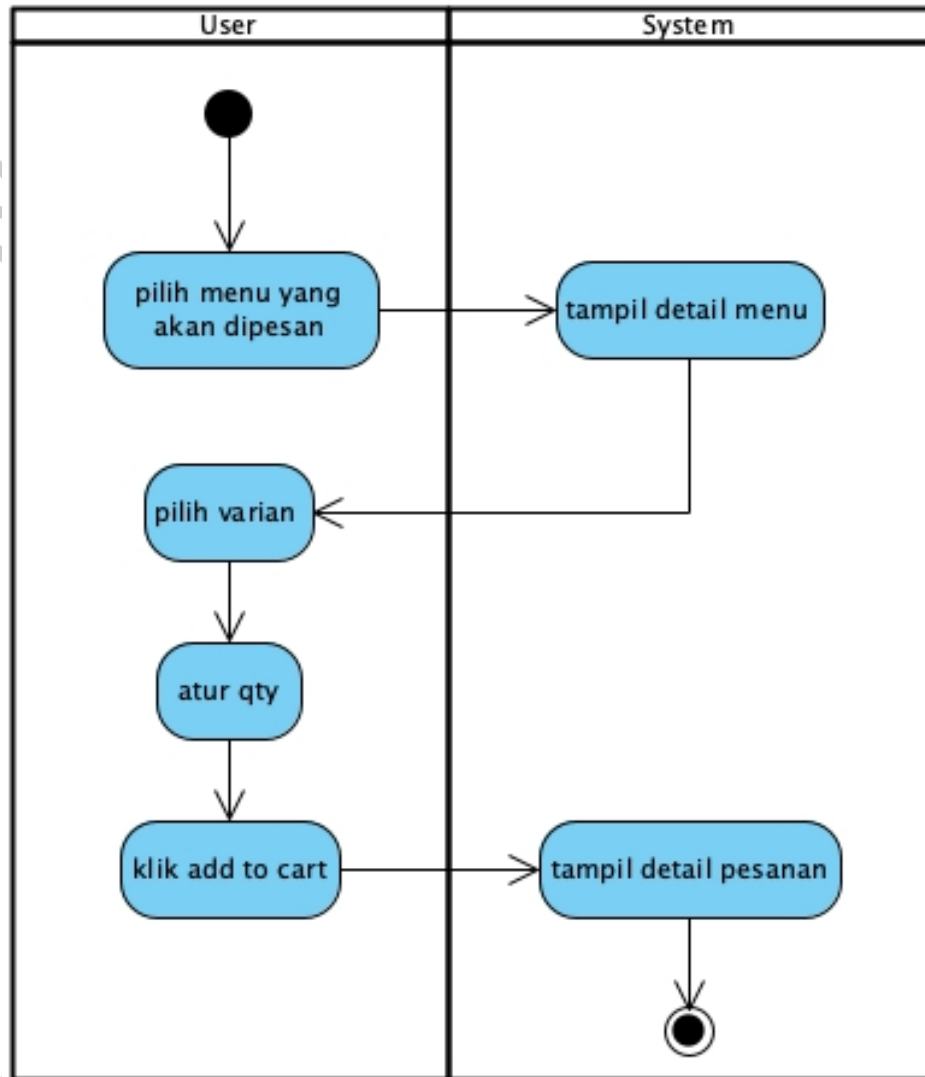
Gambar 4.5 Activity Diagram Login

Pada Gambar 4.5 di atas menunjukkan aktifitas *login* diawali dengan user yang sudah mempunyai akun klik *login*. Lalu *user* memasukkan nomor telepon yang sudah terdaftar lalu *user* klik tombol *Get OTP*, dan sistem akan melakukan validasi nomor telepon. Jika nomor telepon sudah terdaftar maka sistem akan mengirimkan OTP. Setelah itu user memasukkan OTP yang telah diterima, sistem akan melakukan validasi OTP, jika valid maka sistem akan menampilkan halaman utama.



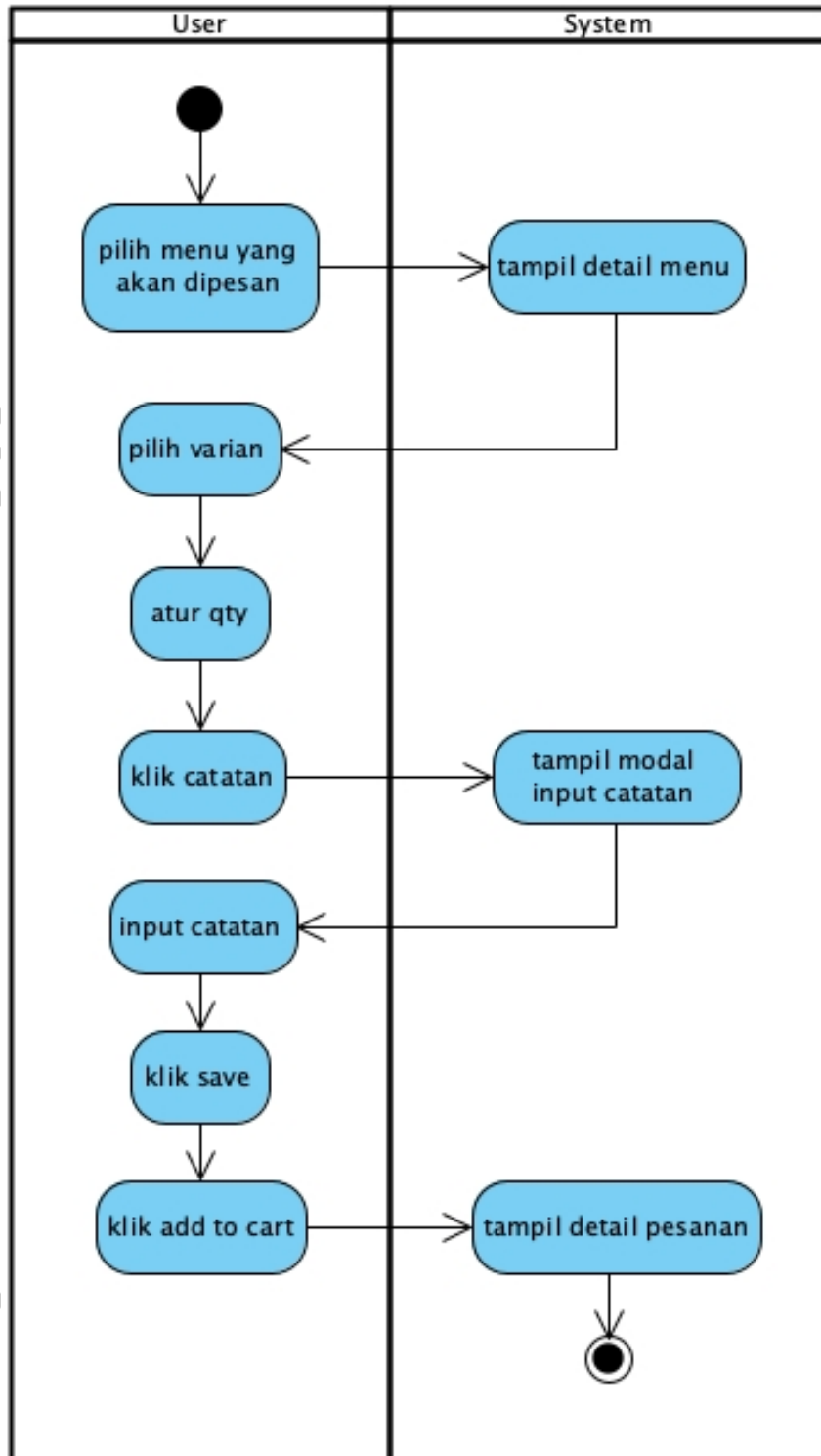
Gambar 4.6 Activity Diagram Melihat Menu

Pada Gambar 4.6 di atas menunjukkan aktifitas melihat menu diawali dengan *user* klik menu *scan QR*, lalu sistem menampilkan halaman *QR scanner*. *User* melakukan *scan QR*, lalu sistem melakukan validasi QR, jika *QR code* valid maka sistem menampilkan halaman daftar menu.



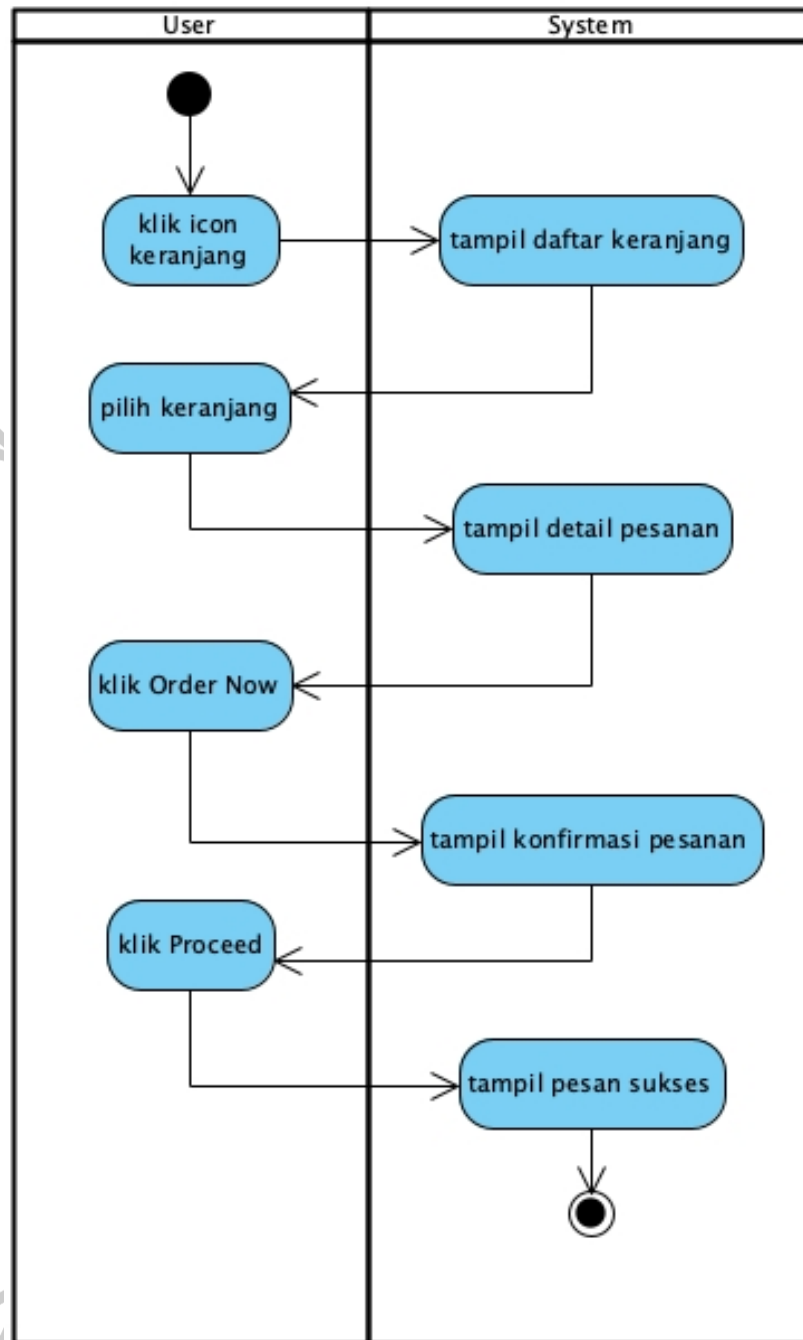
Gambar 4.7 Activity Diagram Menambahkan Menu ke Keranjang Fase 1

Pada Gambar 4.7 di atas menunjukkan Activity Diagram Menambahkan Menu ke Keranjang Fase 1. Aktifitas ini dimulai saat *user* memilih menu yang ingin dipesan, kemudian sistem menampilkan halaman detail menu. Lalu *user* memilih varian dan mengatur qty. Untuk menambahkan menu ke dalam keranjang, *user* harus klik *add to cart*, lalu sistem menampilkan detail pesanan.



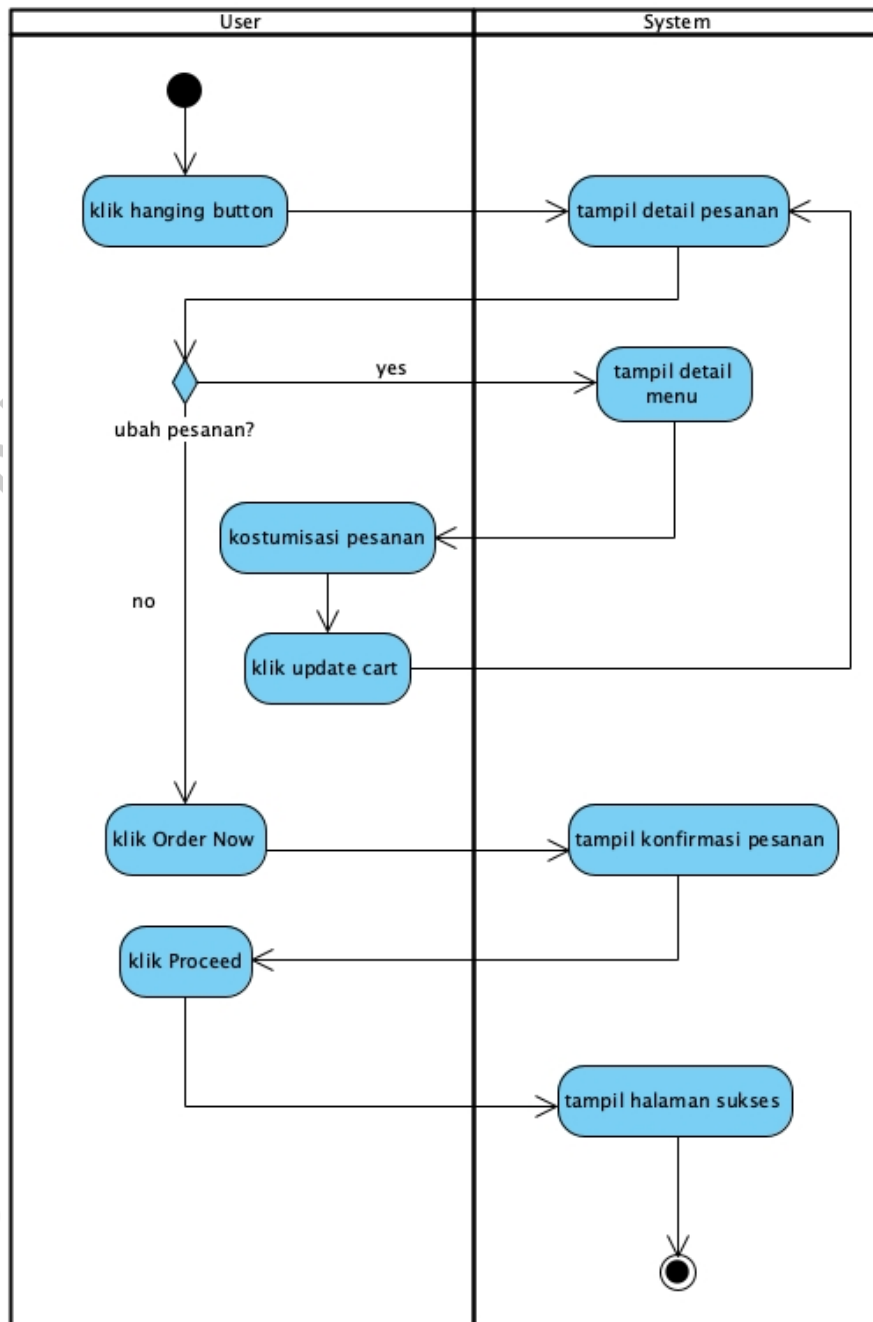
Gambar 4.8 Activity Diagram Menambahkan Menu ke Keranjang Fase 2

Pada Gambar 4.8 di atas menunjukkan *Activity Diagram* Menambahkan Menu ke Keranjang Fase 2 setelah melalui proses evaluasi pada tahap *review*. Aktifitas ini sama seperti pada Fase 1, hanya saja ada penambahan input catatan saat mengatur pesanan.



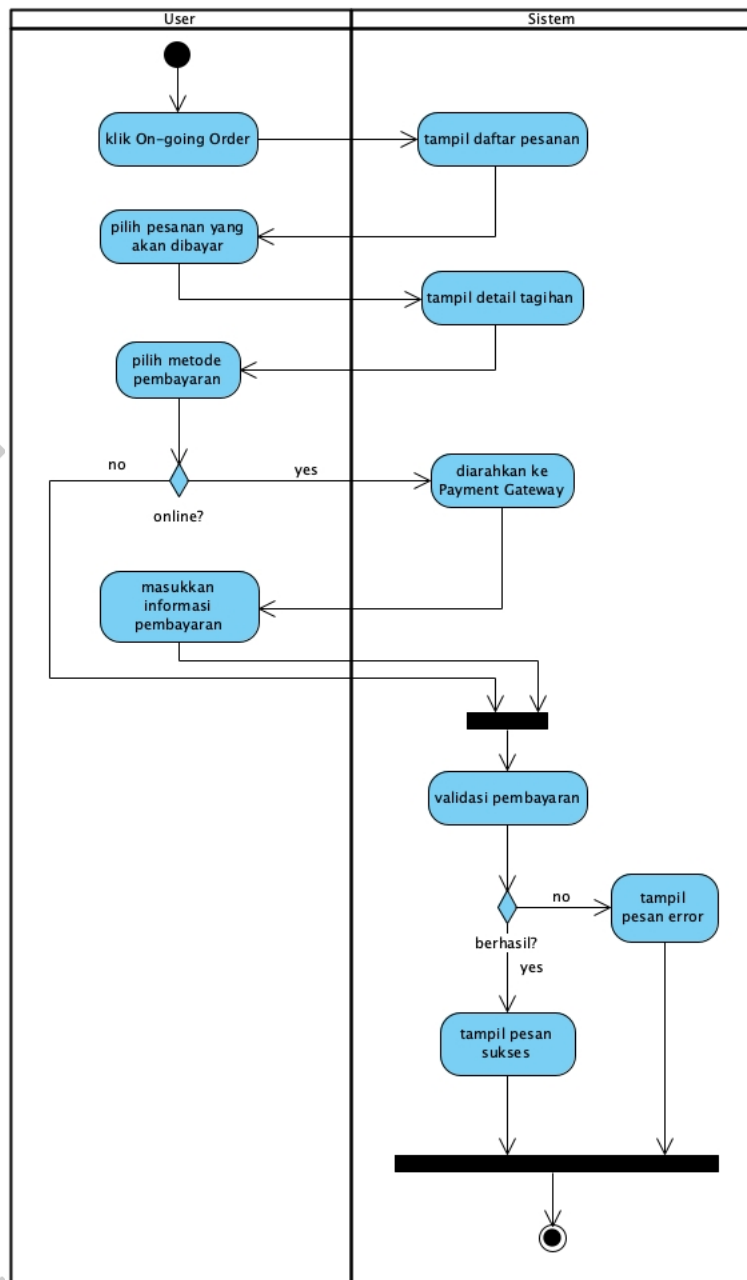
Gambar 4.9 Activity Diagram Konfirmasi Pesanan Fase 1

Pada Gambar 4.9 di atas menunjukkan Activity Diagram Konfirmasi Pesanan Fase 1. Aktifitas ini dimulai saat user memilih klik menu keranjang, lalu sistem menampilkan daftar keranjang. User memilih keranjang yang aktif, lalu sistem menampilkan detail pesanan. Kemudian user klik Order Now, lalu muncul konfirmasi pesanan sampai user klik Proceed kemudian tampil pesan sukses.



Gambar 4.10 Activity Diagram Konfirmasi Pesanan Fase 2

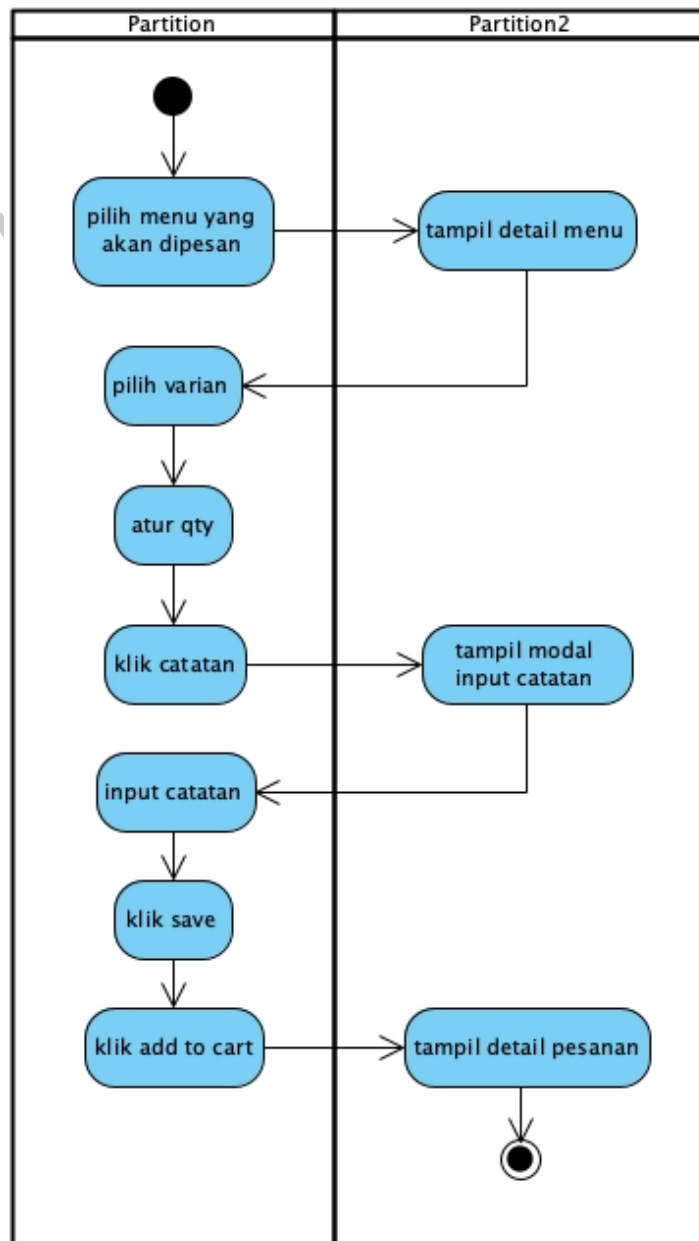
Pada Gambar 4.10 di atas menunjukkan Activity Diagram Konfirmasi Pesanan Fase 2 setelah melalui proses evaluasi pada tahap review. Aktifitas ini sama seperti pada Activity Diagram Konfirmasi Pesanan Fase 1, hanya saja ada penambahan fitur ubah pesanan. Serta untuk akses ke detail pesanan, user tidak perlu ke halaman daftar keranjang, melainkan bisa langsung klik *hanging button*.



Gambar 4.11 Activity Diagram Pembayaran

Pada Gambar 4.11 di atas menunjukkan aktifitas pembayaran, diawali dengan *user* klik menu *On-going Order* untuk melihat pesanan yang belum terbayar. Lalu sistem menampilkan detail tagihan, kemudian *user* memilih metode pembayaran. Jika *user* memilih metode *offline* maka *user* bisa langsung membayar di kasir, namun jika *user* memilih pembayaran online maka sistem akan mengarahkan ke *payment gateway*. Kemudian *user*

masukkan informasi pembayaran, jika sudah selesai maka sistem akan mengecek pembayaran. Jika berhasil maka akan tampil pesan sukses.

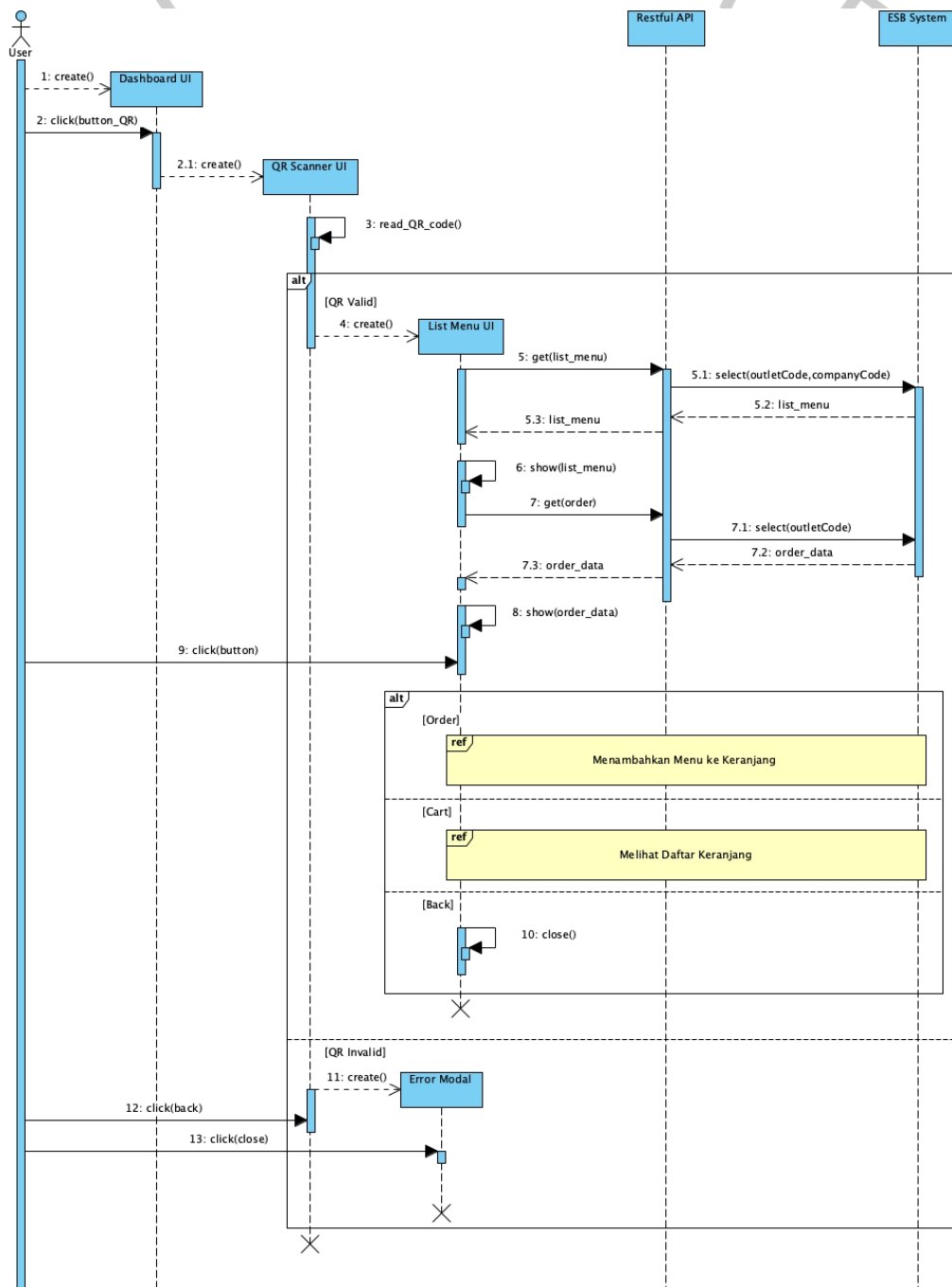


Gambar 4.12 Activity Diagram Melihat Riwayat Pesanan

Pada Gambar 4.12 di atas menunjukkan aktifitas melihat Riwayat pesanan, dimulai dari *user* pilih menu *profile*, lalu sistem menampilkan halaman *profile*. Lalu *user* memilih menu *My Order*, maka sistem akan menampilkan Riwayat pesanan *user* yang pernah dibuat.

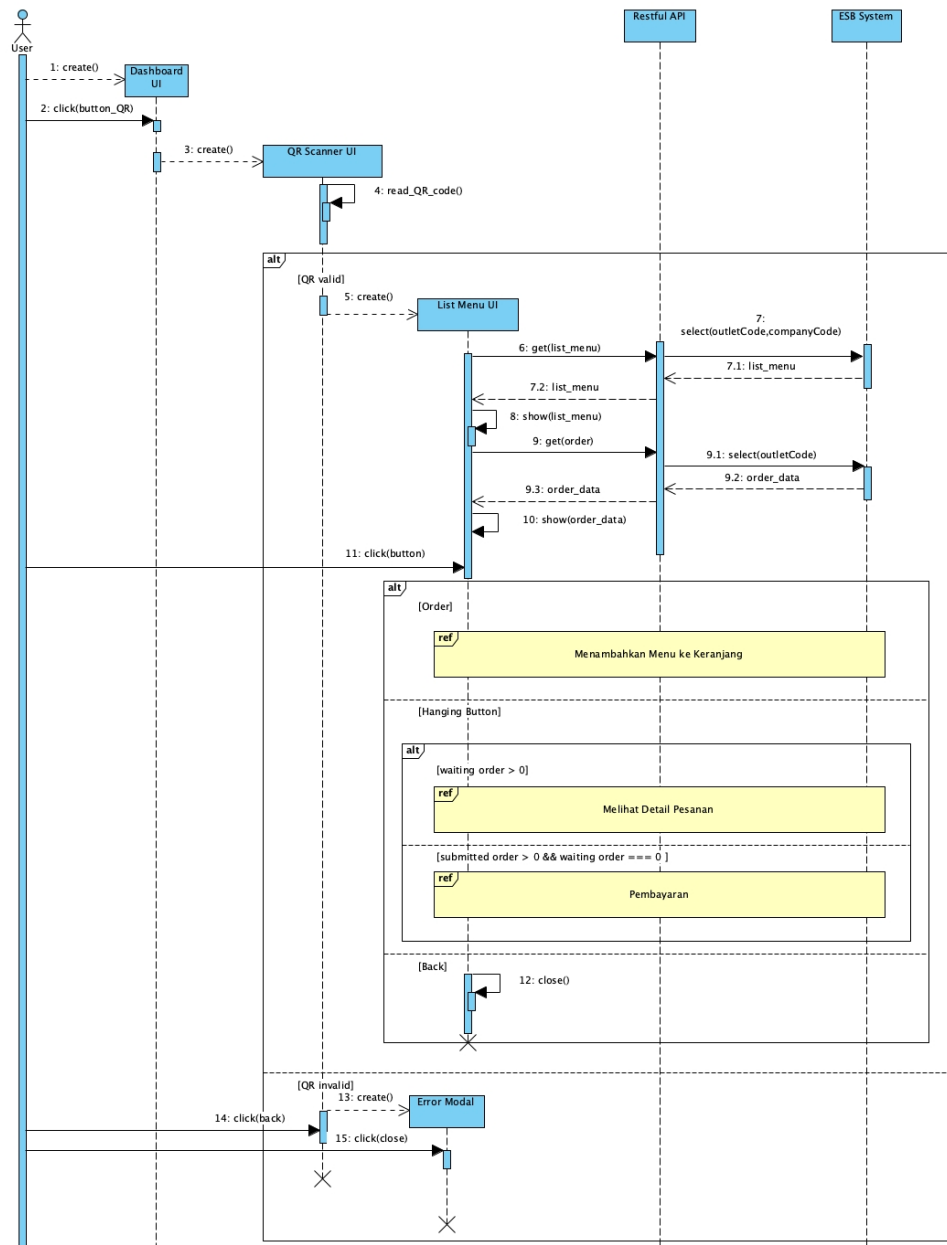
4.2.5 Sequence Diagram

Sequence Diagram adalah suatu bentuk diagram interaksi yang mendetailkan bagaimana suatu tindakan dilakukan. Diagram ini menggambarkan interaksi antara objek-objek dalam suatu konteks kolaboratif. *Sequence Diagram* berfokus pada aspek waktu dengan menggunakan sumbu vertikal dalam diagram untuk secara visual menggambarkan interaksi dan menunjukkan kapan pesan harus dikirim.



Gambar 4.13 Sequence Diagram Melihat Menu Fase 1

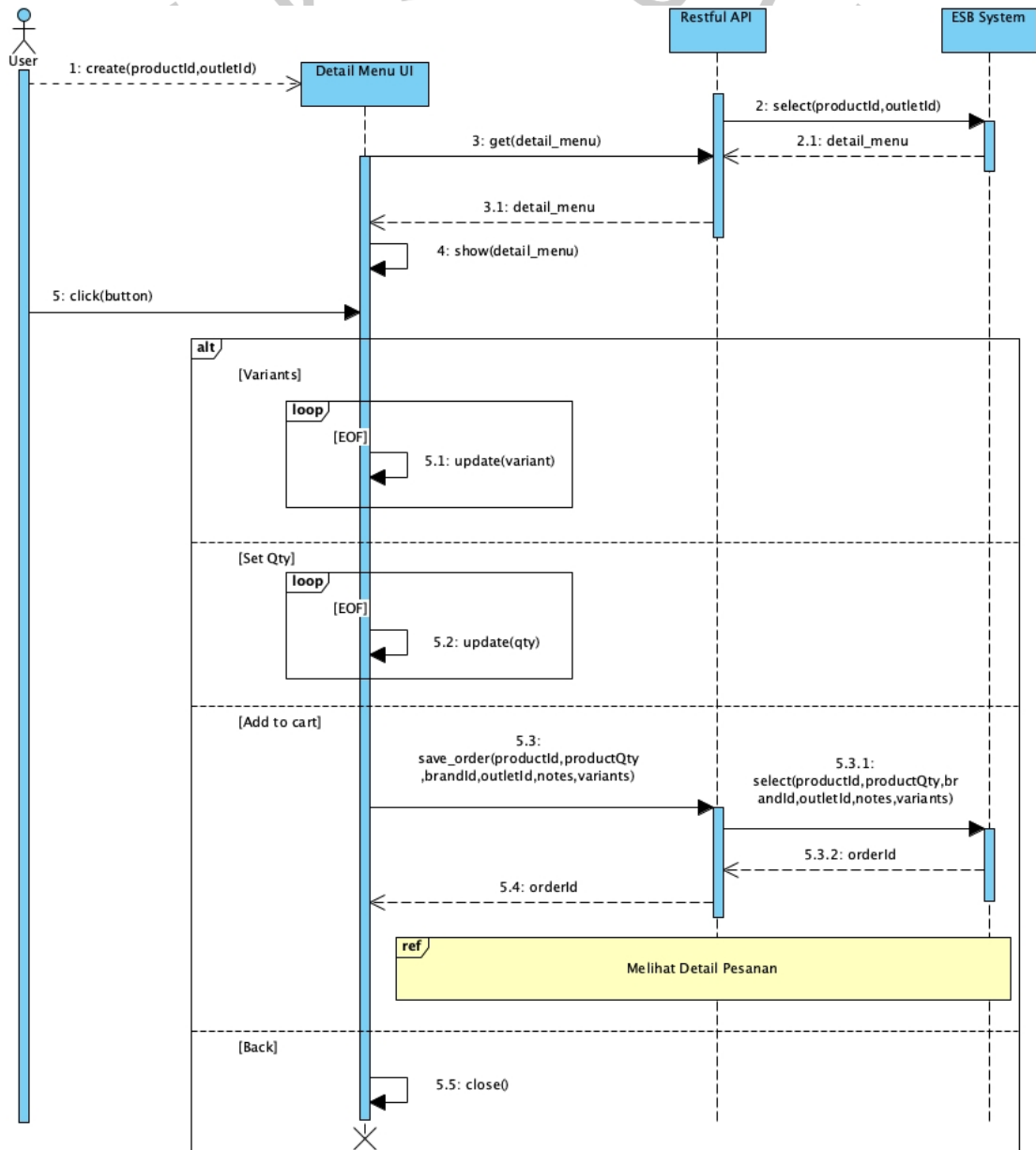
Pada Gambar 4.13 di atas menunjukkan *Sequence Diagram* Melihat Menu Fase 1. Pada fase 1 ketika *user* berhasil melakukan *scan* QR, maka sistem akan menampilkan halaman Daftar Menu. Pada halaman tersebut *user* bisa membuat pesanan dengan klik *Order* pada menu dipilih, selain itu pada halaman tersebut *user* juga bisa melihat daftar keranjang jika mempunyai pesanan yang belum dikonfirmasi sebelumnya.



Gambar 4.14 *Sequence Diagram* Melihat Menu Fase 2

Pada Gambar 4.14 di atas menunjukkan *Sequence Diagram* Melihat Menu Fase 2 setelah melalui proses evaluasi. Ada beberapa penyesuaian

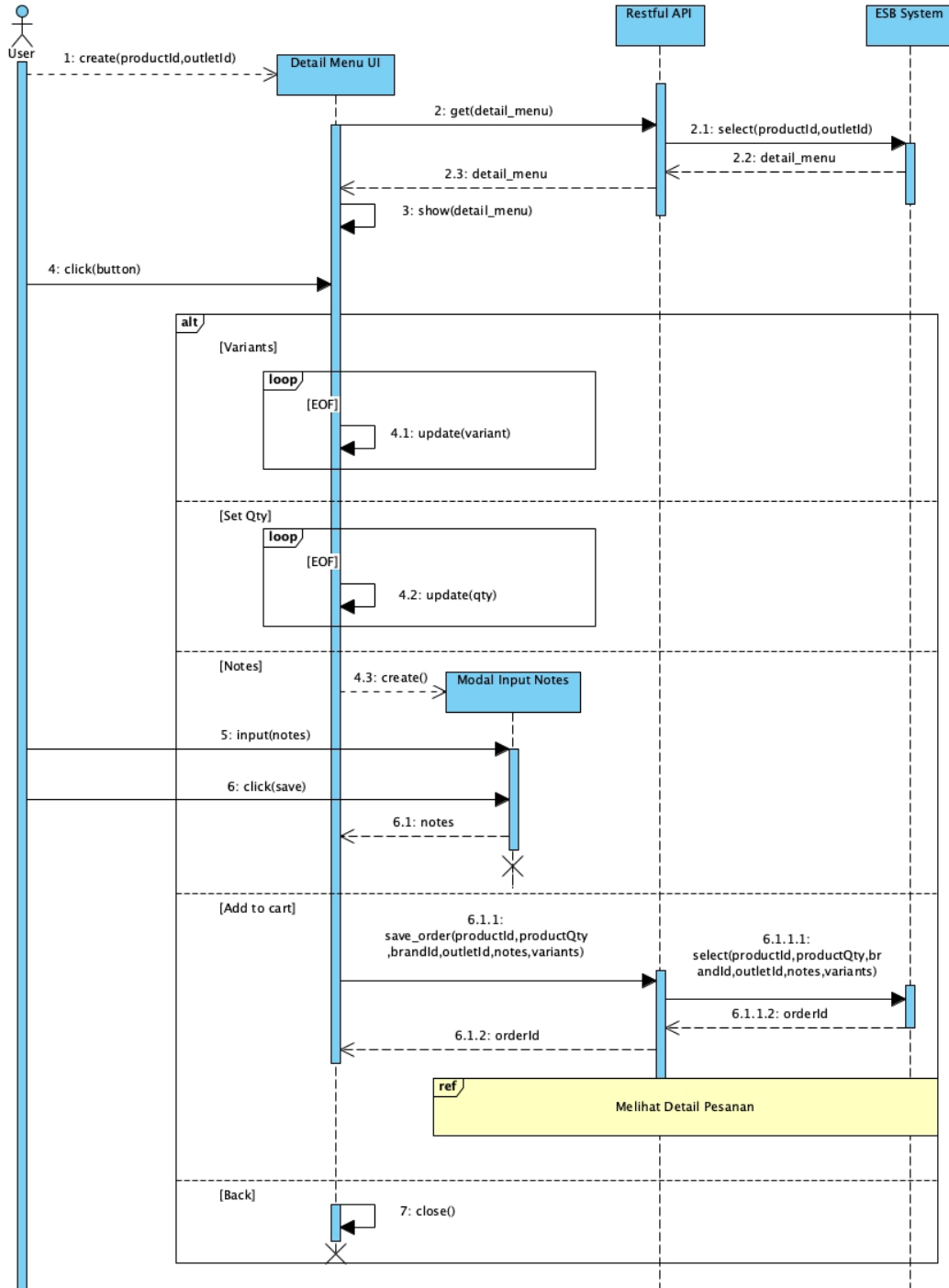
yaitu *takeout* menu keranjang, dan jika ada pesanan yang sedang berlangsung akan muncul *hanging button* yang bisa di klik sesuai dengan kondisi jika jumlah *waiting order* lebih dari 0 maka sistem akan menampilkan detail pesanan, tapi jika jumlah order yang telah di konfirmasi lebih dari 0 dan jumlah *waiting order* sama dengan 0 maka sistem akan menampilkan detail tagihan.



Gambar 4.15 *Sequence Diagram* Menambahkan Menu ke Keranjang Fase 1

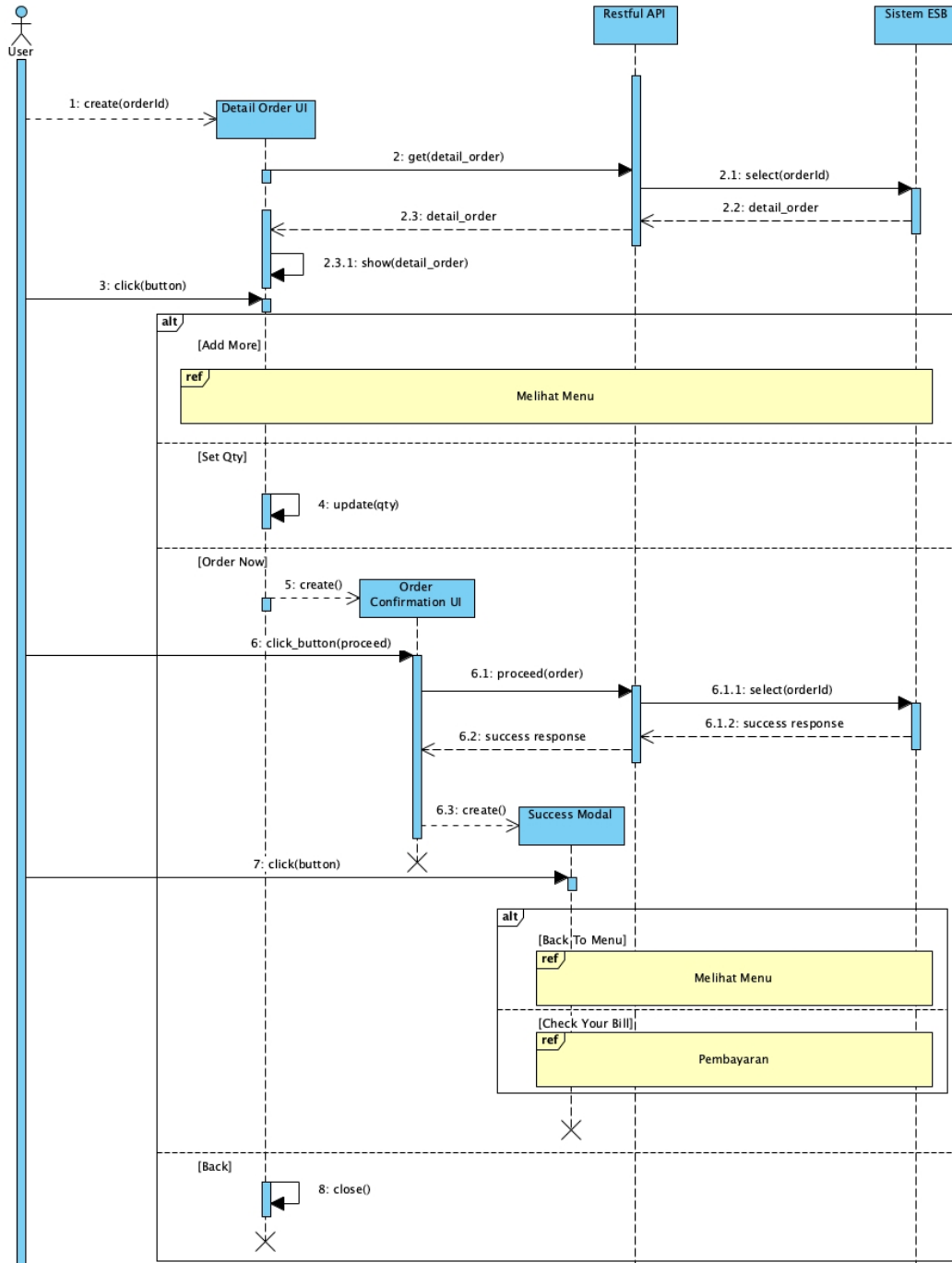
Pada Gambar 4.15 di atas menunjukkan *Sequence Diagram* Menambahkan Menu ke Keranjang Fase 1. Saat *user* klik *order* pada salah satu menu yang ingin dipesan maka sistem akan menampilkan detail menu.

Di halaman tersebut *user* bisa kostumisasi pesanan seperti memilih *variant*, dan mengatur *qty*. Lalu Ketika, *user* klik tombol *add to cart* maka pesanan akan masuk ke sistem.



Gambar 4.16 Sequence Diagram Menambahkan Menu ke Keranjang Fase 2

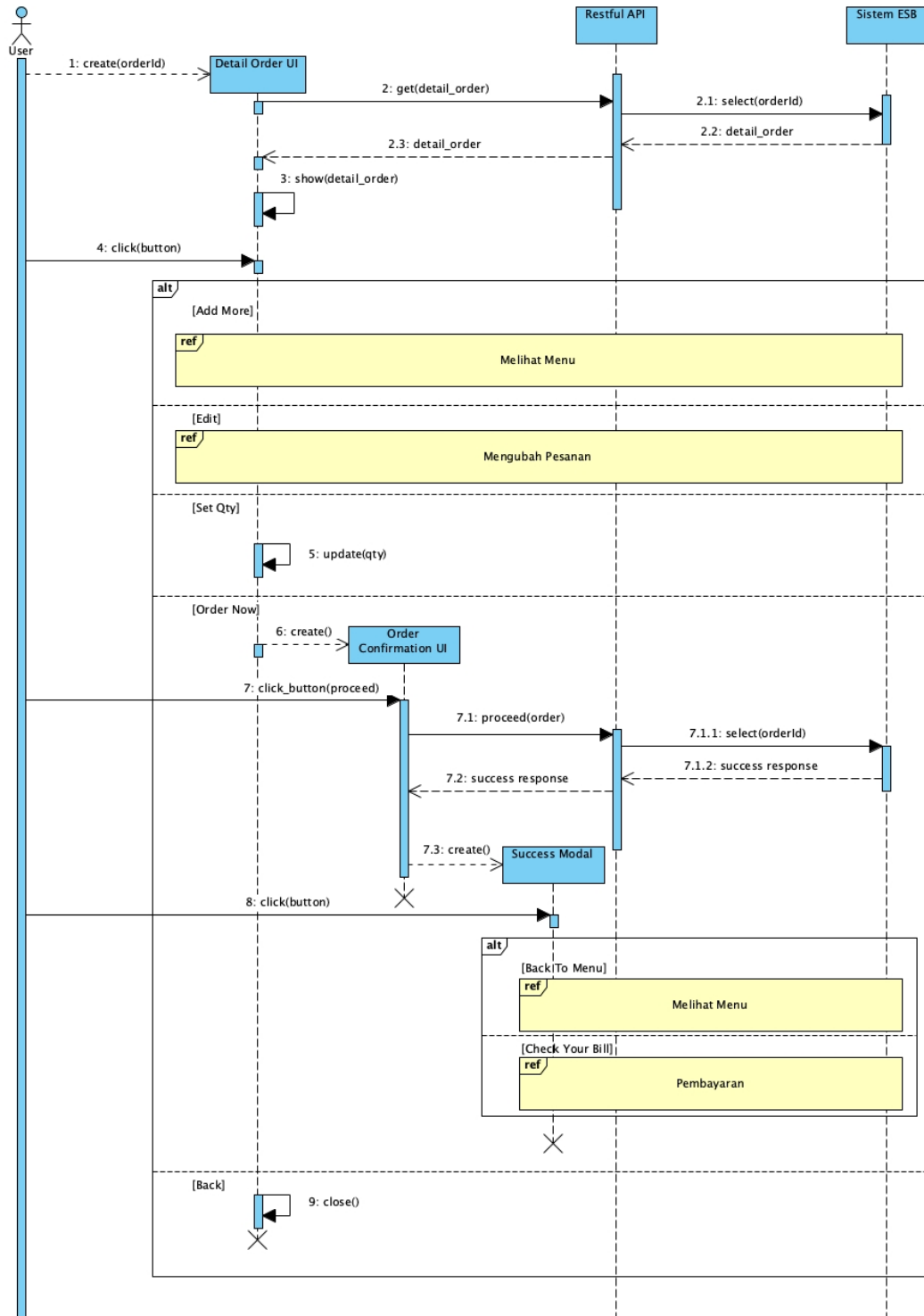
Pada Gambar 4.16 di atas menunjukkan *Sequence Diagram* Menambahkan Menu ke Keranjang Fase 2. Pada fase 2 ini penambahan fitur menambahkan catatan pada pesanan.



Gambar 4.17 *Sequence Diagram* Melihat Detail Pesanan Fase 1

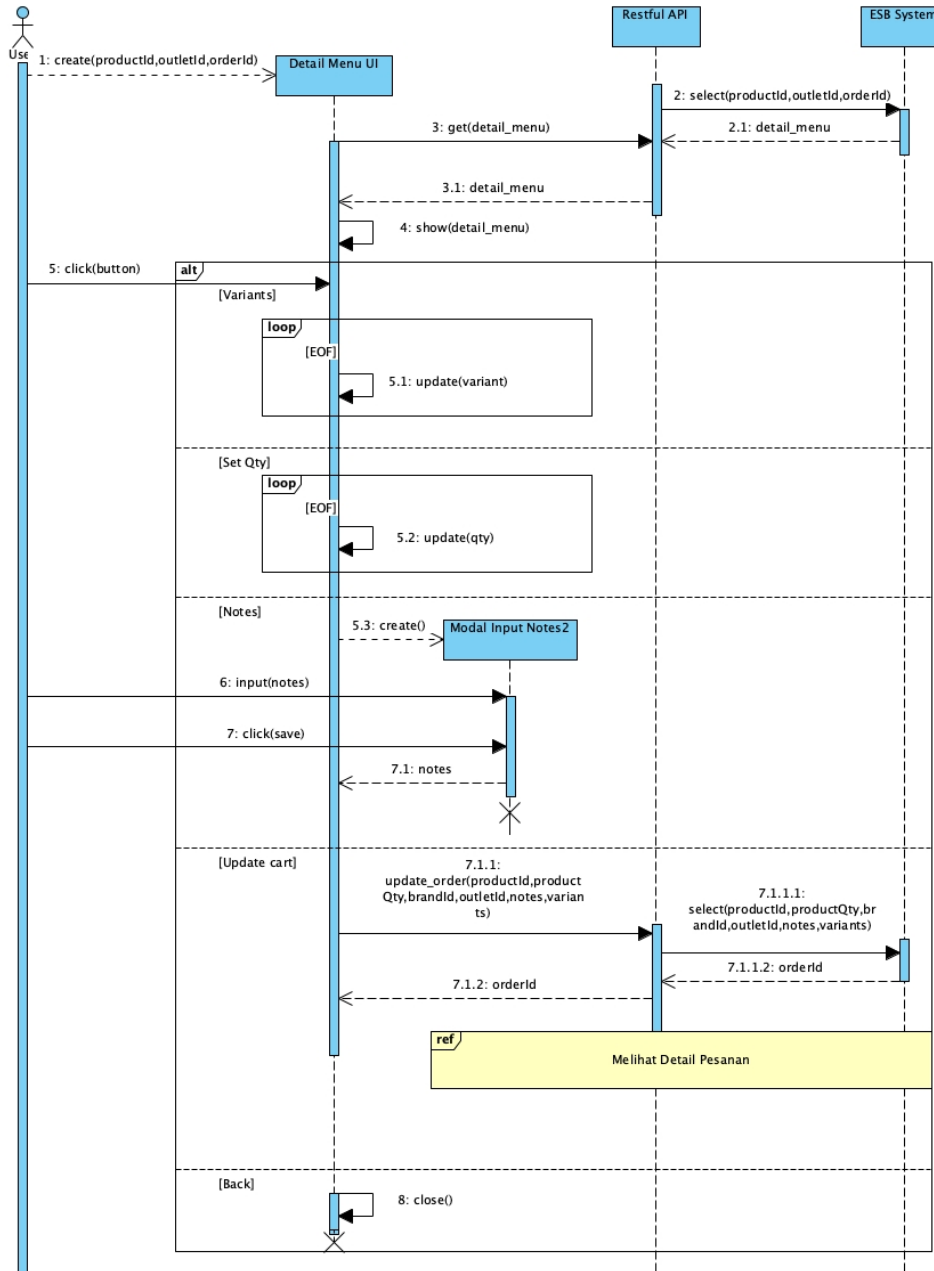
Pada Gambar 4.17 di atas menunjukkan *Sequence Diagram* Detail Pesanan Fase 1. Saat user klik *add to cart* pada menu yang dipilih ataupun,

maka sistem akan menampilkan halaman *detail* pesanan. Di halaman ini pesanan belum dikonfirmasi, dan *user* bisa menambah pesanan.



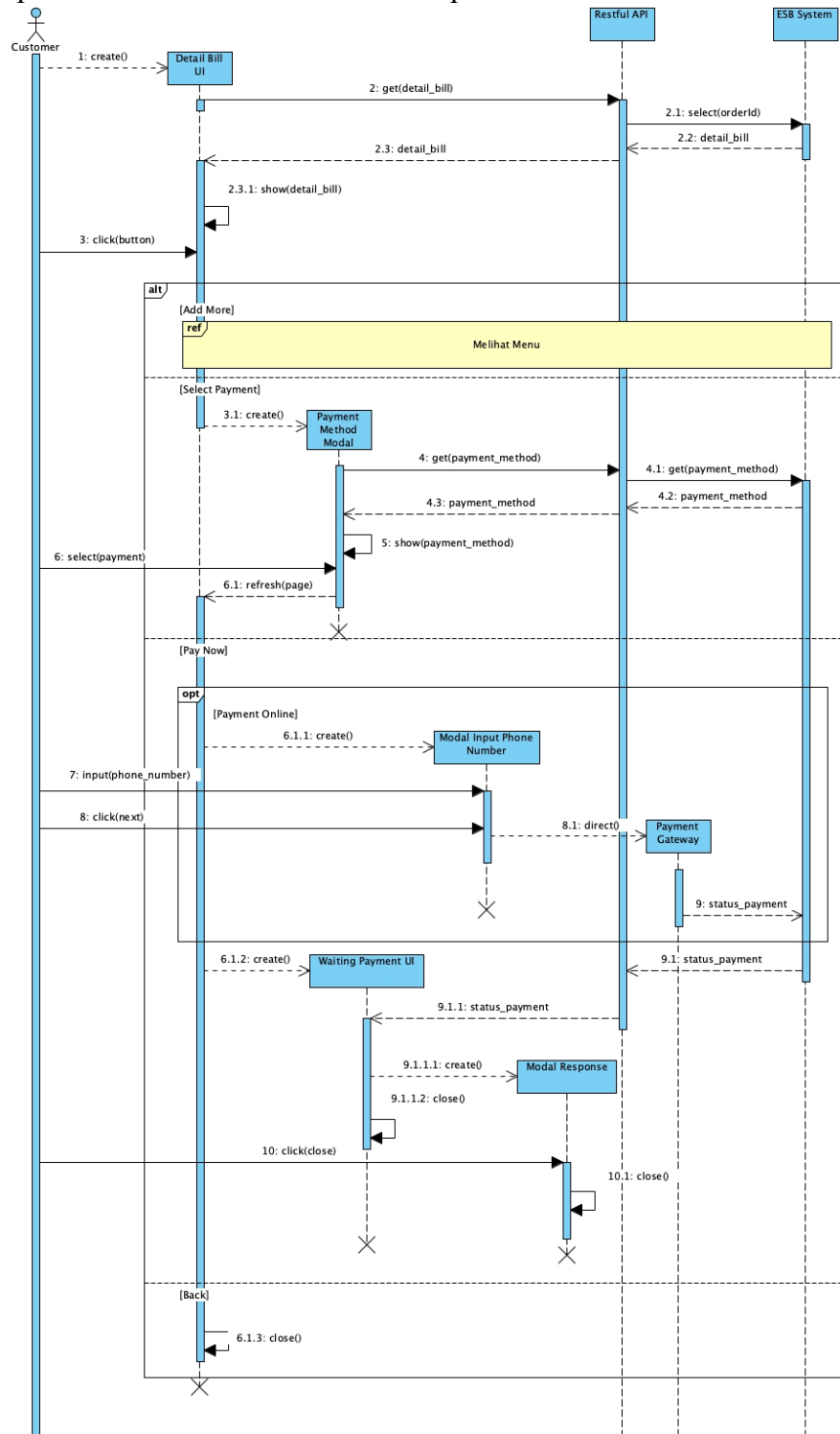
Gambar 4.18 Sequence Diagram Melihat Detail Pesanan Fase 2

Pada Gambar 4.18 di atas menunjukkan *Sequence Diagram* Detail Pesanan Fase 2. Pada fase 2 ini penambahan fitur ubah pesanan sebelum pesanan dikonfirmasi.



Gambar 4.19 *Sequence Diagram* Mengubah Pesanan

Pada Gambar 4.19 di atas menunjukkan *Sequence Diagram* Mengubah Pesanan. Sebelum pesanan dikonfirmasi, pada halaman detail pesanan *user* bisa melakukan ubah pesanan.

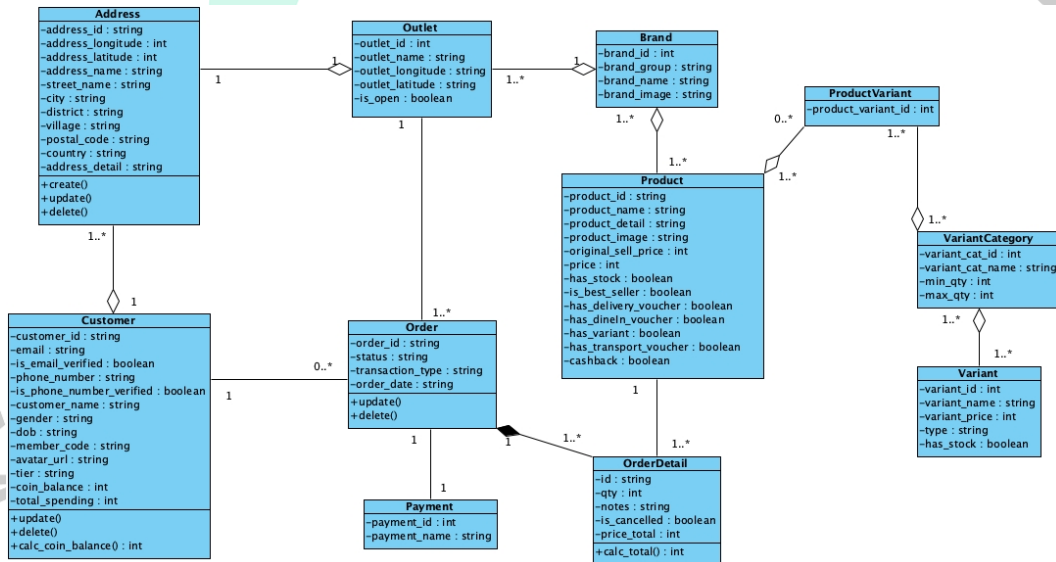


Gambar 4.20 *Sequence Diagram* Pembayaran

Pada Gambar 4.20 di atas menunjukkan *Sequence Diagram* Pembayaran. Pembayaran bisa dilakukan jika pesanan sudah dikonfirmasi. Pembayaran bisa dilakukan melalui *offline* atau *online*. Pembayaran *online*, user tidak perlu ke kasir cukup bayar melalui *smartphone* dengan pilihan *e-wallet* seperti OVO atau GoPay.

4.2.6 Class Diagram

Pada Gambar 4.21 di bawah ini menampilkan dan menjelaskan rancangan class diagram dalam sistem informasi pemesanan makanan. Class diagram ini menggambarkan struktur sistem dan hubungan antar objek-objek pada database sistem informasi pemesanan yang sedang dibangun.



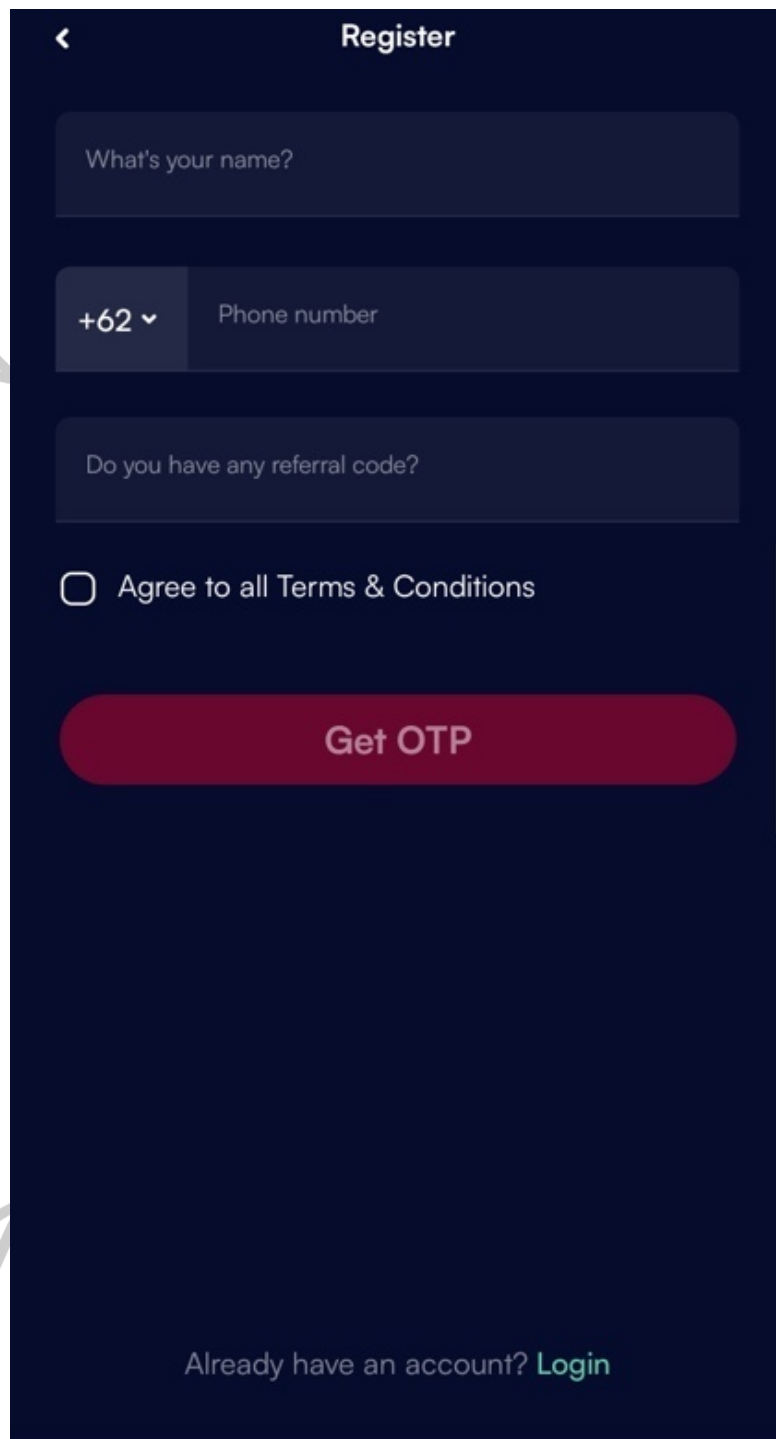
Gambar 4.21 Class Diagram Sistem Informasi Pemesanan Makanan

4.3 Perancangan Antar Muka Pengguna

4.3.1 Rancangan Tampilan

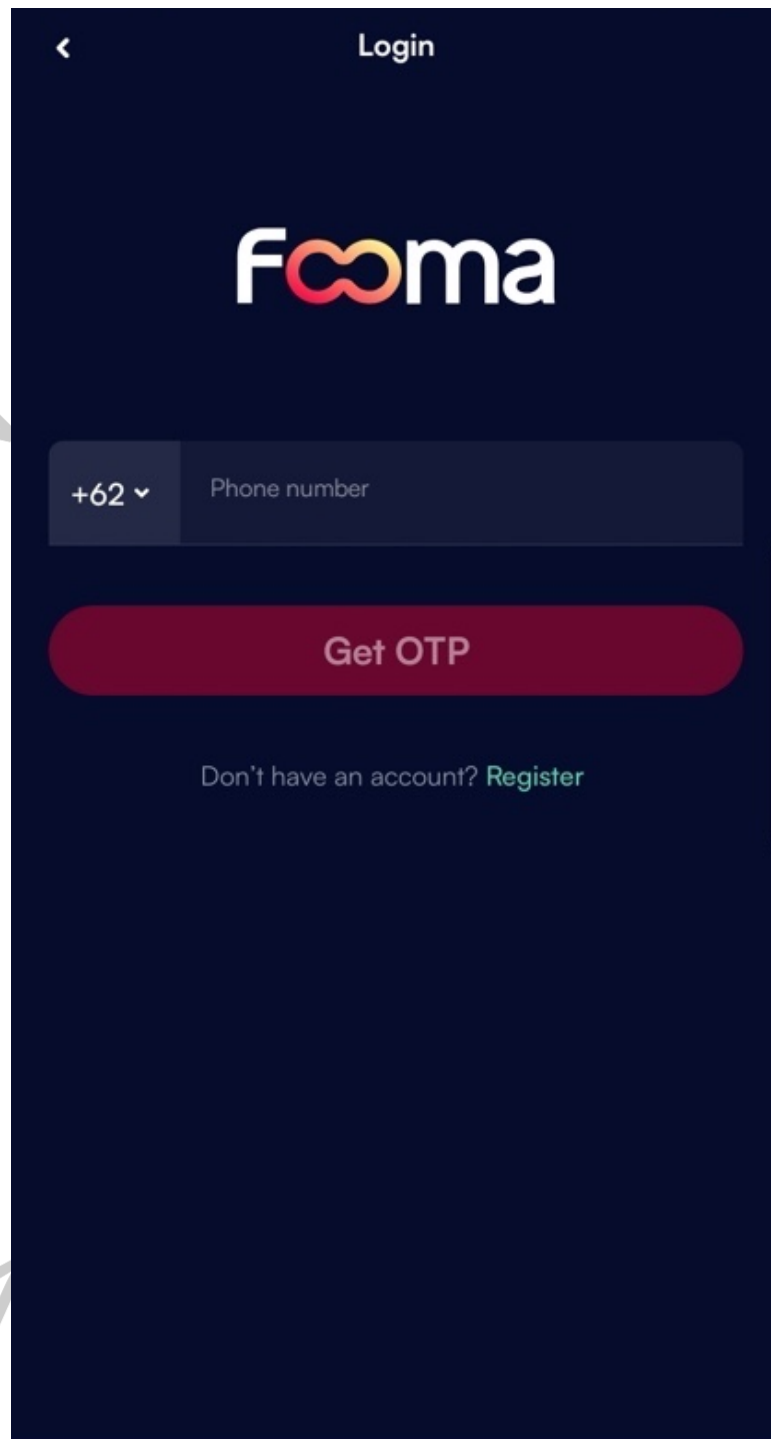
Rancangan tampilan layar atau susunan visual memiliki peran dalam menggambarkan penampilan yang telah direncanakan pada sistem informasi. Dalam konteks sistem informasi pemesanan makanan ini,

terdapat beragam layout tampilan untuk pengguna yang dijelaskan pada Gambar 4.22 sampai Gambar 4.32 di bawah ini.



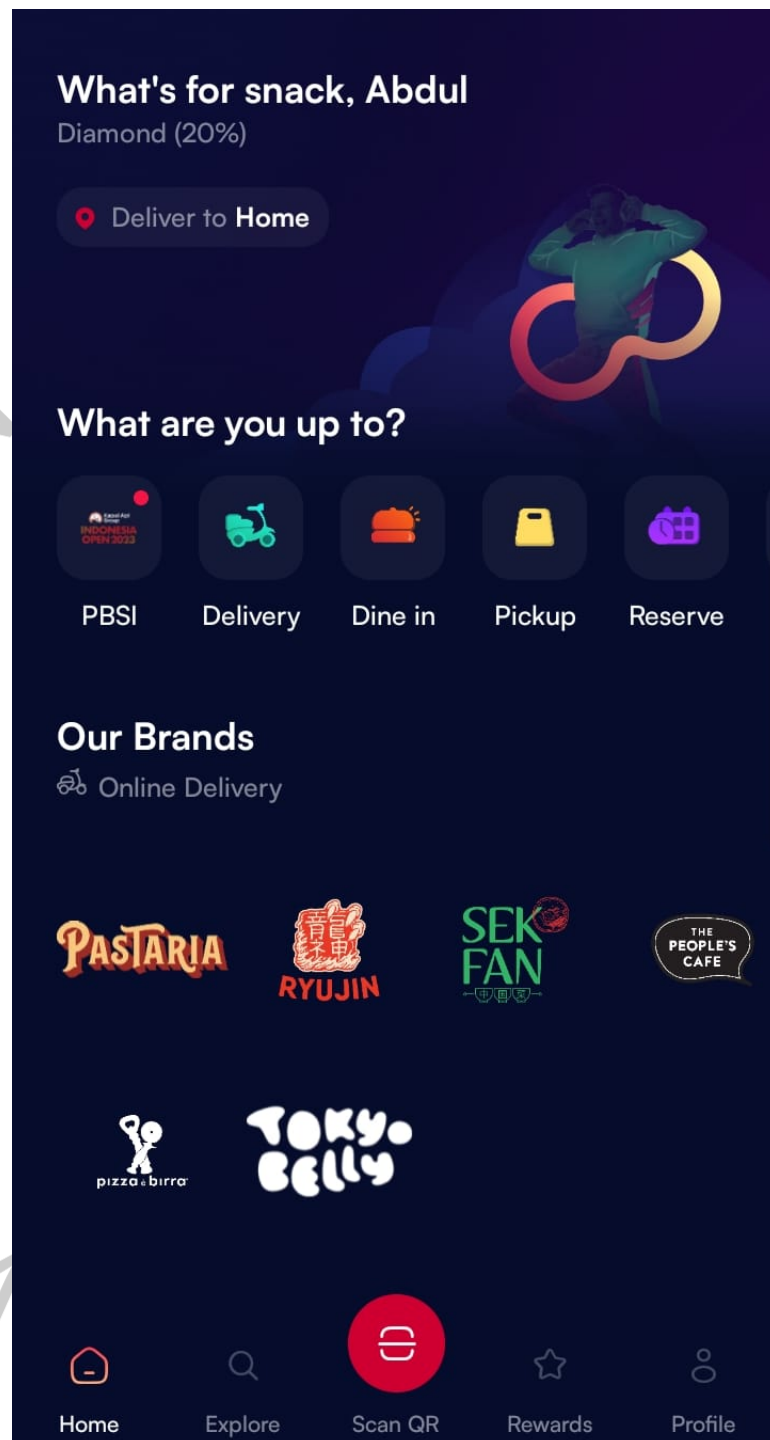
Gambar 4.22 Halaman *Register*

Pada Gambar 4.22 di atas menunjukkan halaman *Register* saat *user* pertama kali membuka aplikasi. *User* tersebut belum mempunyai akun dan ingin membuat akun baru.



Gambar 4.23 Halaman *Login*

Pada Gambar 4.23 di atas menunjukkan halaman *Login* saat *user* sudah mempunyai akun dan ingin akses halaman utama.



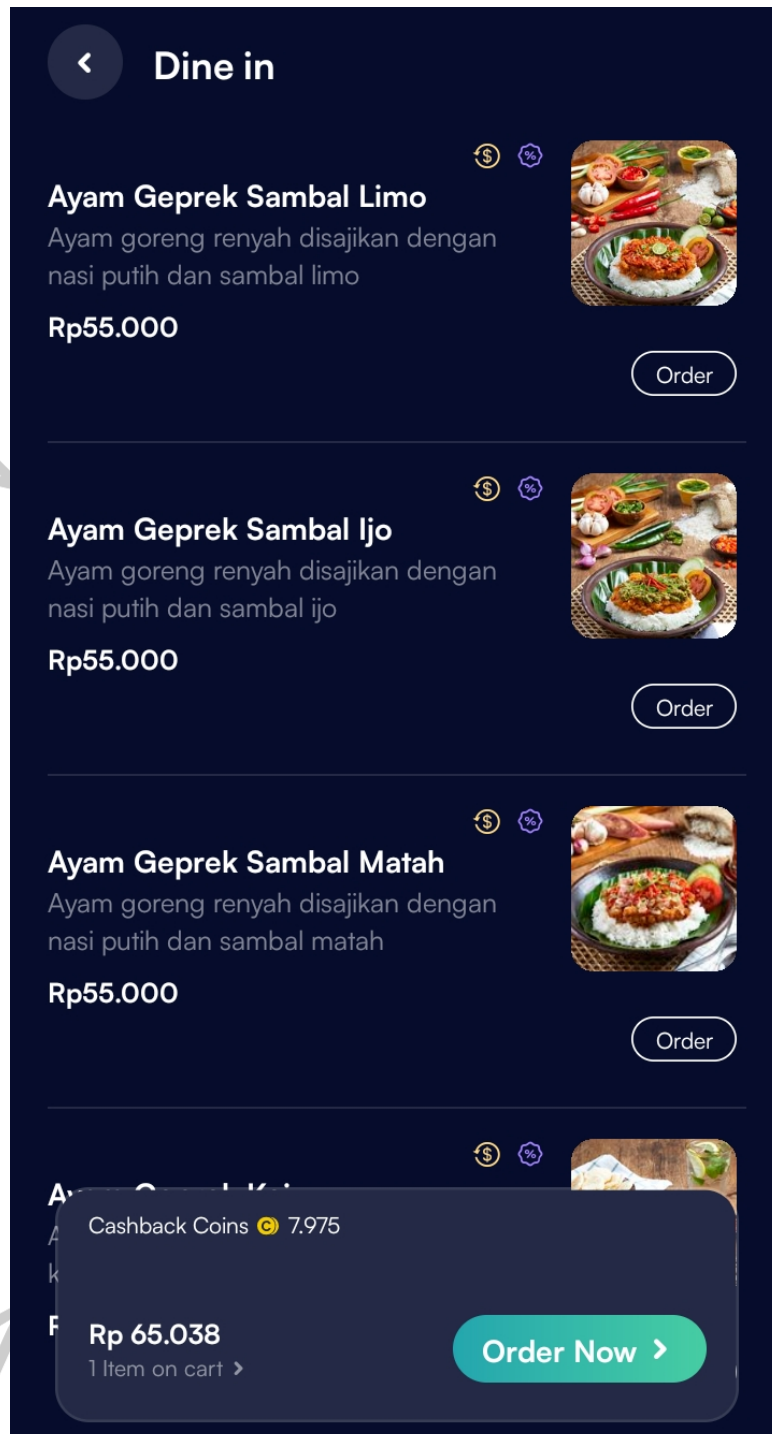
Gambar 4.24 Halaman Utama

Pada Gambar 4.24 di atas menunjukkan halaman utama saat *user* telah berhasil melakukan *register* atau *login*.



Gambar 4.25 Halaman Daftar Menu Fase 1

Pada Gambar 4.25 di atas menunjukkan halaman daftar menu fase 1 saat *user* telah berhasil melakukan *scan* QR menu. Di halaman ini terdapat menu keranjang untuk melihat detail pesanan dari daftar keranjang yang masih aktif.



Gambar 4.26 Halaman Daftar Menu Fase 2

Pada Gambar 4.26 di atas menunjukkan halaman daftar menu fase 2 setelah mengalami perubahan pada fase 1 saat tahap *review*. Pada perubahan ini menu keranjang dihilangkan, dan diganti dengan *hanging button* yang akan muncul ketika ada pesanan yang masih menunggu untuk dikonfirmasi, sehingga *user* tidak perlu ke halaman daftar keranjang.



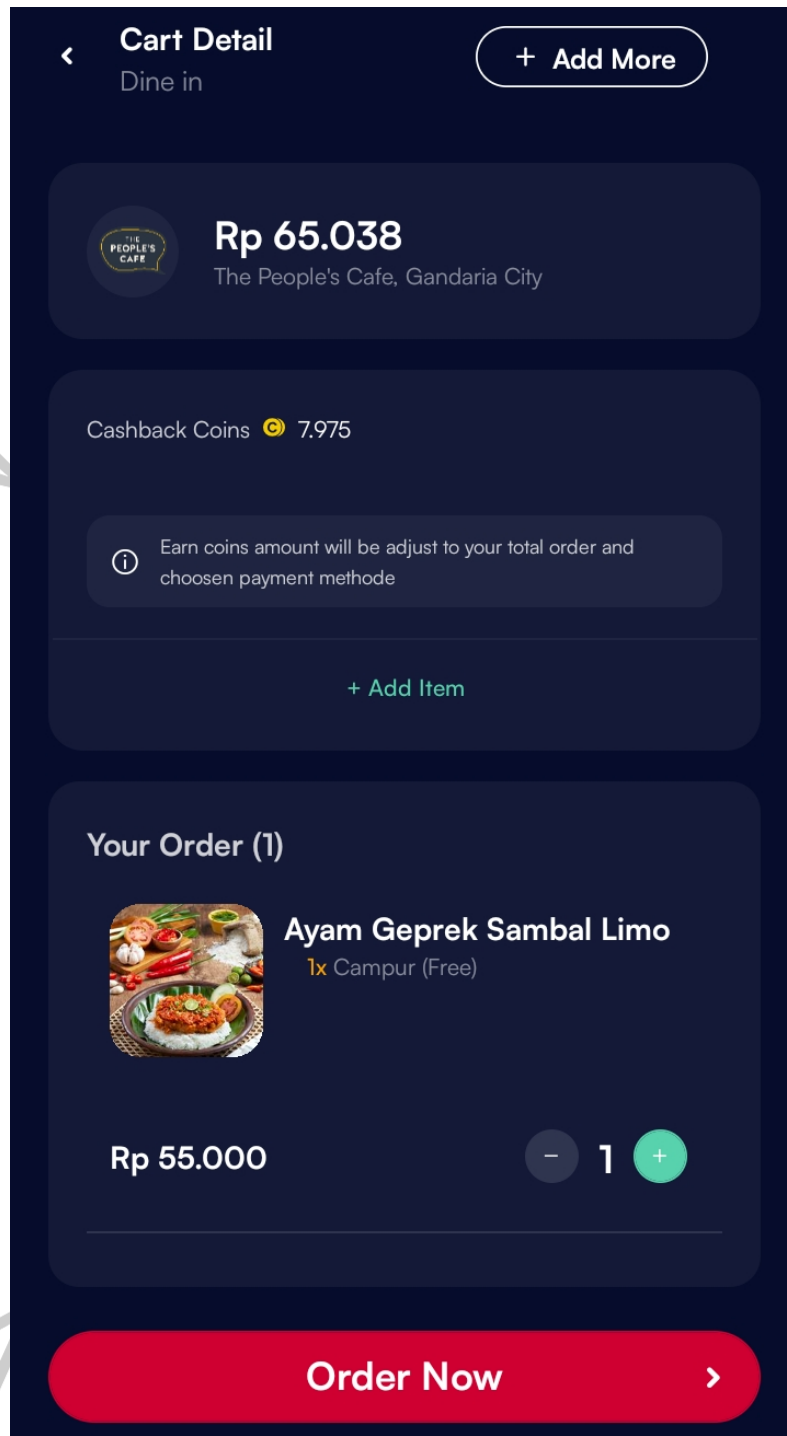
Gambar 4.27 Halaman Detail Menu Fase 1

Pada Gambar 4.27 di atas menunjukkan halaman detail menu fase 1 saat *user* pilih salah satu menu pada daftar menu. Di halaman ini *user* bisa melihat informasi menu, kostumisasi menu dan menambahkannya ke keranjang.



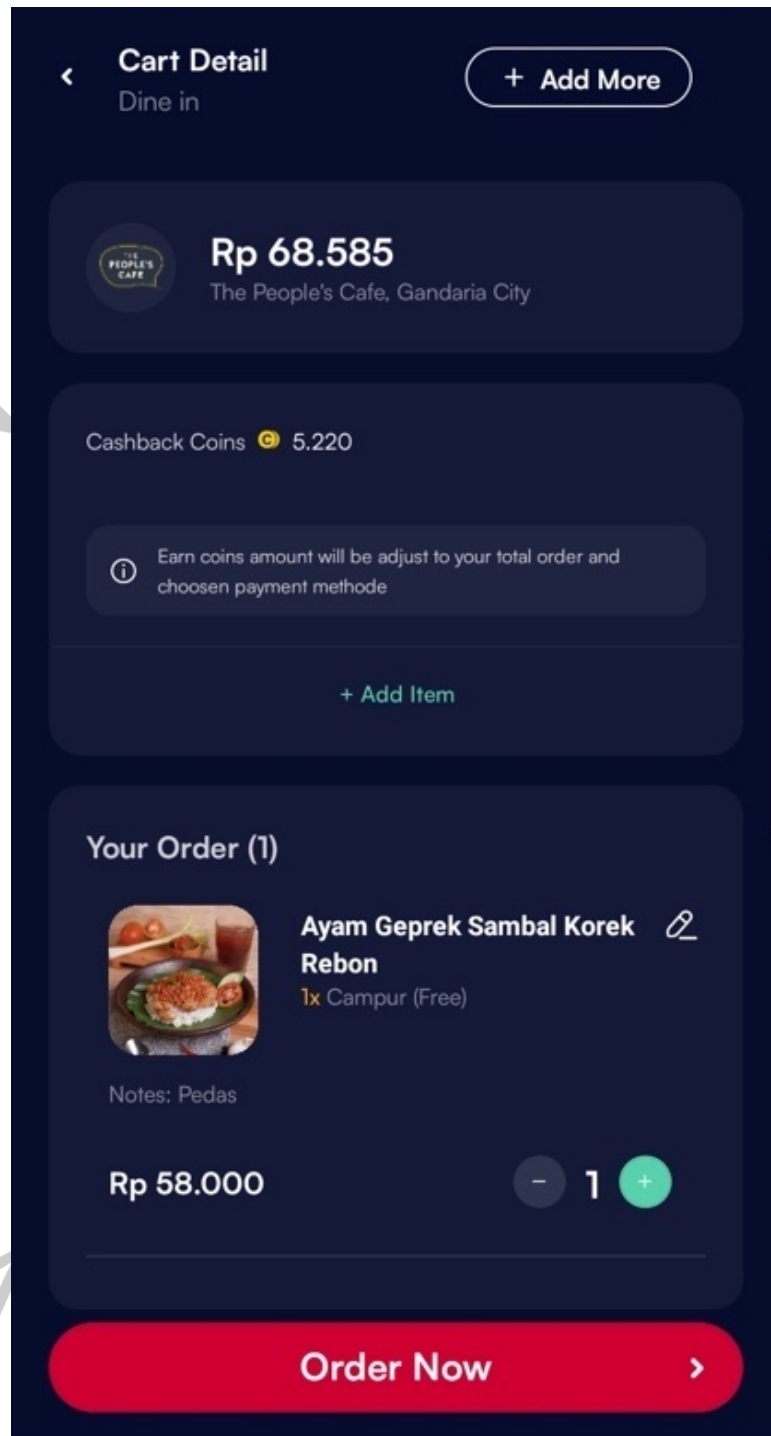
Gambar 4.28 Halaman Detail Menu Fase 2

Pada Gambar 4.28 di atas menunjukkan halaman detail menu fase 2 setelah mengalami perubahan pada tahap *review*. Perubahan ini berdasarkan kebutuhan pengguna yang ingin menambahkan catatan pada pesannya, sehingga pada fase 2 ini ditambahkan fitur tambah catatan.



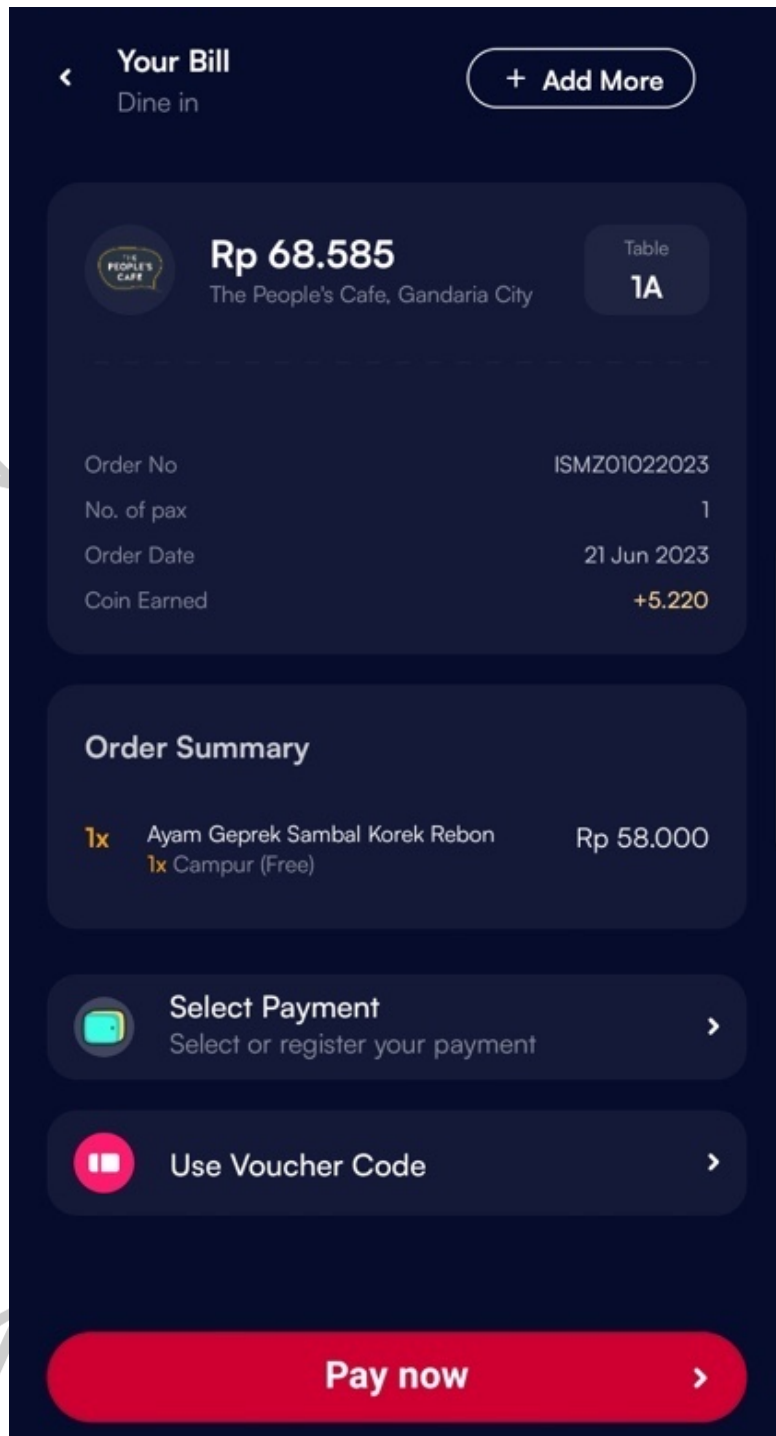
Gambar 4.29 Halaman Detail Pesanan Fase 1

Pada Gambar 4.29 di atas menunjukkan halaman detail pesanan fase 1 setelah *user* menambahkan pesanan. Di halaman ini *user* bisa langsung melakukan konfirmasi pesanan ataupun menambah pesanan.



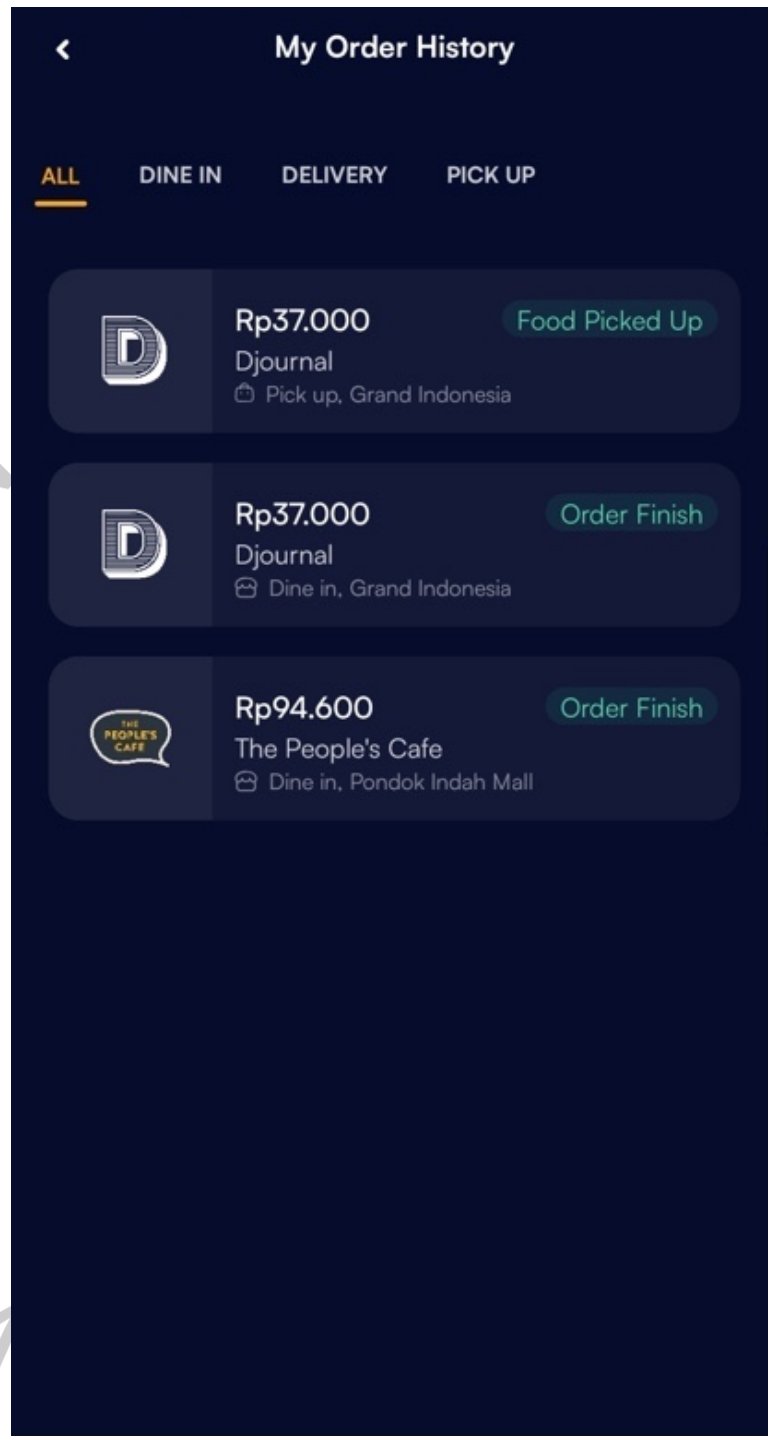
Gambar 4.30 Halaman Detail Pesanan Fase 2

Pada Gambar 4.30 di atas menunjukkan halaman detail pesanan fase 2 setelah mengalami perubahan pada tahap *review*. Perubahan ini berdasarkan kebutuhan pengguna yang ingin mengubah pesanan sebelum melakukan konfirmasi dengan klik *icon* edit pada pesanan.



Gambar 4.31 Halaman Detail Tagihan

Pada Gambar 4.31 di atas menunjukkan halaman detail tagihan setelah *user* berhasil melakukan konfirmasi pesanan. Di halaman ini *user* bisa membayar pesannya dengan memilih metode pembayarannya terlebih dahulu.



Gambar 4.32 Halaman Riwayat Pesanan

Pada Gambar 4.32 di atas menunjukkan halaman riwayat pesanan yang telah dibuat. Halaman ini bisa di akses melalui menu *Profile*.

4.4 Perancangan Implementasi

Tahap implementasi merupakan tahap di mana sistem diuji untuk mengetahui kesiapannya dalam digunakan. Pada tahap ini, juga dilakukan

serangkaian pengujian untuk memastikan bahwa fitur-fitur dalam sistem berjalan sesuai dengan desain yang telah dibuat, serta untuk menemukan bug atau kesalahan. Pengujian dilakukan dengan melakukan proses input pada sistem dan memantau keluaran yang dihasilkan, untuk memverifikasi kesesuaian dengan harapan yang telah ditetapkan. Berikut ini adalah rencana pengujian yang dapat ditemukan dalam Tabel 4.14 di bawah ini

Tabel 4.14 Rancangan Testing

No.	Test Name	Test Steps	Expected Result
1.	Melihat daftar menu	<ol style="list-style-type: none"> 1. Klik tombol <i>Scan QR</i> 2. <i>Scan QR</i> 	Sistem menampilkan halaman daftar menu
2.	Membuat pesanan	<ol style="list-style-type: none"> 1. Pilih menu 2. Pilih variant 3. Atur qty 4. Klik <i>add to cart</i> 	Sistem menampilkan halaman detail pesanan
3.	Membuat pesanan tanpa memilih <i>mandatory variant</i>	<ol style="list-style-type: none"> 1. Kilik menu 2. Klik <i>add to cart</i> 	Tombol <i>add to cart</i> tidak berfungsi / <i>disabled</i>
4.	Konfirmasi pesanan	<ol style="list-style-type: none"> 1. Pilih menu 2. Pilih variant 3. Atur qty 4. Klik <i>add to cart</i> 1. Klik <i>order now</i> 	Muncul halaman konfirmasi pesanan
5.	Bayar tagihan pesanan dengan pembayaran online	<ol style="list-style-type: none"> 1. Kilik <i>on-going order</i> 2. Klik <i>payment method</i> 3. Pilih <i>e-wallet</i> (OVO, Gopay) 4. Masukkan nomor telpon 	Langsung <i>direct</i> ke halaman <i>payment gateway</i>

6.	Melihat Riwayat pesanan	<ol style="list-style-type: none"> 1. Kilik <i>on-going order</i> 2. Klik <i>payment method</i> 3. Pilih <i>e-wallet</i> (OVO, Gopay) 4. Masukkan nomor telpon 5. Setelah pembayaran berhasil, masuk ke menu <i>My Order</i> 	Pesanan yang telah selesai masuk ke daftar riwayat pemesanan
----	-------------------------	---	--

4.5 Analisis Efisiensi

Pada tahap analisis efisiensi ini penulis ingin membuktikan bahwa penggunaan sistem informasi pemesanan makanan ini mampu meningkatkan efisiensi waktu dalam pemesanan makanan. Analisis efisiensi ini dilakukan dengan cara observasi pada lokasi objek penelitian, yaitu mencatat lama waktu yang dibutuhkan pelanggan dalam melakukan pemesanan secara konvensional dan membandingkan dengan pelanggan yang menggunakan sistem informasi pemesanan makanan yang telah dibuat.

4.5.1 Dokumentasi Kegiatan



Gambar 4.33 Kegiatan Pencatatan Pesanan Pelanggan

Pada Gambar 4.33 di atas menunjukkan dokumentasi kegiatan pelayan sedang mencatat pesanan pelanggan di selembar kertas.



Gambar 4.34 Kegiatan Pelayan Input Pesanan pada Sistem POS

Pada Gambar 4.34 di atas menunjukkan dokumentasi kegiatan pelayan sedang input pesanan yang sudah dicatat pada selembar kertas tadi ke sistem POS (ESB).

4.5.2 Hasil Analisis

Hasil analisis yang penulis peroleh berupa data lama waktu saat pelanggan melakukan pemesanan makanan baik secara konvensional

maupun menggunakan sistem informasi pemesanan makanan. Data yang diperoleh sebanyak 50 data yang terbagi antara pemesanan makanan secara konvensional dan digital. Pemesanan makanan secara konvensional dimulai dari pelanggan memanggil pelayan, melihat buku menu, lalu pelayan mencatat pesanan di selembar kertas, kemudian pelayan memasukkan data pesanan ke sistem POS, serta waktu pelanggan melakukan pembayaran di kasir. Sedangkan pemesanan melalui aplikasi dimulai saat pelanggan *scan* QR code, melihat menu, memilih pesanan, hingga pembayaran pada *smartphone* mereka. Hasil analisis akan dapat dilihat pada Tabel 4.15 dan Tabel 4.16 di bawah ini.

Tabel 4.15 Hasil Catatan Waktu Pemesanan Makanan Secara Konvensional

Pelanggan	Waktu Pemesanan Makanan (hh:mm:ss)	Waktu Pembayaran (hh:mm:ss)
Pelanggan 1	00.05.55	00.05.50
Pelanggan 2	00.05.25	00.04.05
Pelanggan 3	00.06.10	00.04.20
Pelanggan 4	00.06.05	00.03.55
Pelanggan 5	00.07.25	00.02.10
Pelanggan 6	00.08.00	00.02.30
Pelanggan 7	00.06.20	00.02.15
Pelanggan 8	00.07.25	00.02.40
Pelanggan 9	00.07.00	00.02.25
Pelanggan 10	00.09.00	00.01.50
Pelanggan 11	00.06.05	00.02.10
Pelanggan 12	00.07.45	00.02.35
Pelanggan 13	00.07.15	00.02.20
Pelanggan 14	00.07.30	00.02.40
Pelanggan 15	00.07.00	00.02.15
Pelanggan 16	00.06.30	00.01.55
Pelanggan 17	00.08.15	00.02.30
Pelanggan 18	00.08.45	00.02.55
Pelanggan 19	00.07.55	00.02.20
Pelanggan 20	00.06.00	00.02.40
Pelanggan 21	00.07.10	00.02.05
Pelanggan 22	00.08.15	00.02.30
Pelanggan 23	00.06.40	00.02.55
Pelanggan 24	00.07.15	00.02.15
Pelanggan 25	00.07.45	00.04.50

Total	02.58.50	01.10.55
--------------	-----------------	-----------------

Dari hasil catatan pada Tabel 4.15 di atas diperoleh rata-rata waktu yang dibutuhkan pelanggan dalam melakukan pemesanan dan pembayaran yaitu:

$$\begin{aligned} \text{Rata - rata Waktu} &= \frac{\text{Total Waktu Pemesanan (detik)} + \text{Total Waktu Pembayaran (detik)}}{\text{Jumlah Pelanggan}} \\ &= \frac{10730 + 4255}{25} = \frac{14985}{25} = 599 \text{ detik} = 9 \text{ menit } 59 \text{ detik} \end{aligned}$$

Dari perhitungan di atas maka diperoleh waktu rata-rata yang dibutuhkan pelanggan dalam melakukan pemesanan makanan dan pembayaran secara konvensional yaitu 9 menit 59 detik.

Tabel 4.16 Hasil Catatan Waktu Pemesanan Makanan Secara Digital

Pelanggan	Waktu Pemesanan Makanan (hh:mm:ss)	Waktu Pembayaran (hh:mm:ss)
Pelanggan 1	00.01.20	00.01.50
Pelanggan 2	00.01.35	00.01.36
Pelanggan 3	00.01.10	00.01.56
Pelanggan 4	00.01.25	00.01.58
Pelanggan 5	00.01.30	00.01.15
Pelanggan 6	00.01.15	00.01.20
Pelanggan 7	00.01.30	00.01.35
Pelanggan 8	00.01.40	00.01.45
Pelanggan 9	00.01.20	00.01.25
Pelanggan 10	00.01.15	00.01.20
Pelanggan 11	00.01.35	00.01.40
Pelanggan 12	00.01.30	00.01.35
Pelanggan 13	00.01.20	00.01.25
Pelanggan 14	00.01.25	00.01.30
Pelanggan 15	00.01.10	00.01.15
Pelanggan 16	00.01.30	00.01.35
Pelanggan 17	00.01.40	00.01.45
Pelanggan 18	00.01.15	00.01.20
Pelanggan 19	00.01.20	00.01.25
Pelanggan 20	00.01.35	00.01.40
Pelanggan 21	00.01.30	00.01.35
Pelanggan 22	00.01.25	00.01.30
Pelanggan 23	00.01.10	00.01.15
Pelanggan 24	00.01.40	00.01.45
Pelanggan 25	00.01.15	00.01.20
Total	00.35.20	00.38.35

Dari hasil catatan pada Tabel 4.16 di atas diperoleh rata-rata waktu yang dibutuhkan pelanggan dalam melakukan pemesanan dan pembayaran yaitu:

$$\begin{aligned} \text{Rata - rata Waktu} &= \frac{\text{Total Waktu Pemesanan (detik)} + \text{Total Waktu Pembayaran (detik)}}{\text{Jumlah Pelanggan}} \\ &= \frac{2120 + 2315}{25} = \frac{4435}{25} = 177 \text{ detik} = 2 \text{ menit } 57 \text{ detik} \end{aligned}$$

Dari perhitungan di atas maka diperoleh waktu rata-rata yang dibutuhkan pelanggan dalam melakukan pemesanan makanan dan pembayaran secara digital yaitu 2 menit 57 detik.

Maka untuk menghitung seberapa efisien waktu antara pemesanan makanan yang dilakukan pelanggan secara konvensional dan menggunakan aplikasi, kita perlu membandingkan rata-rata waktu yang dibutuhkan dalam kedua metode tersebut. Berdasarkan data yang telah diberikan sebelumnya:

1. Rata-rata waktu pemesanan dan pembayaran secara konvensional adalah sekitar 9 menit 59 detik (599 detik).
2. Rata-rata waktu pemesanan dan pembayaran melalui aplikasi adalah sekitar 2 menit 57 detik (177 detik).

Selanjutnya, kita dapat membandingkan kedua rata-rata waktu tersebut:

$$\begin{aligned} \text{Efisiensi Waktu} &= \frac{\text{Rerata waktu konvensional} - \text{Rerata waktu aplikasi}}{\text{Rerata waktu konvensional}} \times 100\% \\ &= \frac{599 - 177}{599} \times 100\% = \frac{422}{599} \times 100\% = 70,45\% \end{aligned}$$

Dengan menggunakan aplikasi, terdapat efisiensi waktu sekitar 70.45% dibandingkan dengan metode konvensional. Hal ini menunjukkan bahwa penggunaan sistem informasi pemesanan makanan melalui aplikasi dapat memberikan efisiensi waktu yang signifikan bagi pelanggan dalam proses pemesanan dan pembayaran.

