

BAB 4

HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan Sistem

Berdasarkan hasil analisa, *report* merupakan salah satu hal yang penting buat *seller*. Hal ini bertujuan agar *seller* dapat memantau performa hasil penjualan mereka di Ralali.com (*seller* dapat mengatur tahun dan bulan berapa yang ingin dilihat. Selain itu, hal tersebut juga akan memicu semangat *seller* untuk dapat meningkatkan hasil penjualan dengan baik. Atas dasar ini, maka akan dibuat fitur *Dashboard Report*. Tujuan dari Fitur *Chat* adalah untuk mempermudah transaksi antara *seller* dengan *buyer*, di mana *seller* dapat menjawab pertanyaan-pertanyaan yang diajukan oleh *buyer* lebih cepat lagi. Fitur *Order Management* bertujuan untuk memudahkan *seller* dalam memproses suatu orderan, dalam fitur ini *seller* dapat melihat daftar orderan yang belum diproses, menerima atau membatalkan orderan serta mereka juga dapat memberikan alasan mengapa mereka harus menolak orderan tersebut, setelah itu *seller* dapat mengirimkan nomor resi kepada *buyer*, *seller* juga dapat melihat daftar total pendapatan yang telah dikirimkan ke Ralali.com ke mereka. Fitur *Profile seller* agar *seller* dapat melihat data *profile* tokonya.

Berdasarkan analisis diatas dan wawancara dengan CTO Ralali.com. Maka, di dapatkanlah *User Requirement* seperti seller dapat melihat laporan penjualan, seller dapat berinteraksi dengan buyer, seller dapat menerima informasi order dengan cepat. Sehingga seller dapat melakukan aktifitas Order Management dengan mudah.

Dalam perancangan aplikasi “Ralali Seller”, metode perancangan yang digunakan adalah metode *Scrum*. *Scrum* merupakan kerangka kerja untuk mengatur dan mengelola pekerjaan yang sifatnya *incremental*, untuk meningkatkan produktivitas dan pengendalian resiko. Berikut ini merupakan aktivitas *scrum* yang telah dilaksanakan.

Tabel 4.1 Product Backlog

Backlog Item	Estimasi (Jam)
Login, Logout	39
Dashboard Report	18

Filter Report	8
Feature Chat	35
Order Processing List	30
View Order Detail	21
Processing Order (Accept/ Reject)	15
Processing Order Request Pickup Order (Ralali Kargo)	21
Submit Airway Bill (Nomor Resi)	16
View Profile	20
Notification System	16
Event Tracking Firebase	14
Total	253

Tabel 4.2 Sprint Backlog 1

<i>Backlog Item</i>	<i>Task</i>	<i>Estimasi (Jam)</i>
Login, Logout	<i>Setup API environment</i> menggunakan Laravel Framework	10
	<i>Setup application environment</i> menggunakan React Native Framework	10
	Integrasi antara aplikasi dengan Ralali Oauth	4
	Membuat <i>Splash Screen</i>	4
	Membuat Halaman Login dan validasinya.	6
	Membuat Tab navigasi menu	3

	Membuat tombol logout pada tab profile	2
Feature Chat	Setup & integrasi Firebase	8
	Integrasi Sendbird & Firebase	5
	Membuat tampilan <i>chat list</i>	8
	Membuat tampilan <i>history chat</i>	8
	Mengirim <i>chat</i>	6

Tabel 4.3 Sprint Backlog 2

<i>Backlog Item</i>	<i>Task</i>	<i>Estimasi (Jam)</i>
Order Processing List	Membuat API order list (Belum diproses, sedang diproses, pengiriman, Diterima dan Settlement)	8
	Membuat tab untuk order list	3
	Membuat halaman Order belum diproses	3
	Membuat halaman Order Sedang diproses	3
	Membuat halaman Order Pengiriman	3
	Membuat halaman Order Diterima	3
	Membuat halaman Order Settlement	3
	Integrasi aplikasi dengan API order list	4
View Order Detail	Membuat API order detail (Order detail & order item)	8
	Membuat halaman Order detail (Menampilkan informasi order dan order item berdasarkan status order)	8
	Integrasi aplikasi dengan API order detail dan order item	5

Tabel 4.4 Sprint Backlog 3

Backlog Item	Task	Estimasi (Jam)
Processing Order (Accept/Reject)	Membuat API <i>process order</i>	8
	Membuat validasi pada order detail (menampilkan button proses pesanan dan item order yang apabila ditekan menampilkan popup modal pilihan terima atau tolak)	
	Membuat halaman tolak pesanan	3
	Integrasi <i>post</i> API di button proses pesanan pada aplikasi	4
Submit Airway Bill (Nomor Resi)	Membuat API <i>submit</i> nomor resi	8
	Membuat validasi pada order detail dengan status sedang diproses yang menampilkan tombol “Masukan Nomor Resi” yang apabila ditekan muncul popup modal yang berisi form input nomor resi dan tombol “Submit”	5
	Integrasi aplikasi dengan API submit nomor resi dengan tombol “Submit” yang ada di popup modal input nomor resi	3
Dashboard Report	Membuat API dashboard dan <i>order count</i>	6
	Membuat halaman dashboard yang menampilkan grafik data	8
	Integrasi aplikasi dengan API dashboard dan order count	4

Tabel 4.5 Sprint Backlog 4

Backlog Item	Task	Estimasi (Jam)
Filter Report	Membuat tombol filter berdasarkan bulan dan menampilkan popup modal yang berisi bulan dan tahun sebagai filter	4
	Integrasi aplikasi dengan API dashboard filter	4
Processing Order Request Pickup Order (Ralali Kargo)	Membuat API <i>request pickup order</i>	5
	Membuat validasi pada order detail dengan status sedang diproses yang menampilkan tombol “Siap dipickup” yang apabila ditekan muncul popup modal yang berisi alamat pengiriman dan tombol “Lanjutkan”	8
	Integrasi aplikasi dengan API <i>request pickup order</i> dengan tombol “Lanjutkan” yang ada di popup modal siap dipickup	8
View Profile	Membuat API <i>profile</i>	8
	Membuat halaman profile & profile detail yang berisikan 3 tab informasi	7
	Integrasi aplikasi dengan API <i>profile</i>	5

Tabel 4.6 Sprint Backlog 5

Backlog Item	Task	Estimasi (Jam)
Notification System	Membuat API <i>notification order</i>	4
	Setup Firebase Notification & integrasi dengan API <i>notification</i>	8

	Menampilkan notifikasi pada saat aplikasi terbuka maupun tertutup	4
Event Tracking	Setup Firebase Event Tracking	6
	Memasang event tracking pada sedia <i>screen</i> dan <i>action</i> pada aplikasi	8

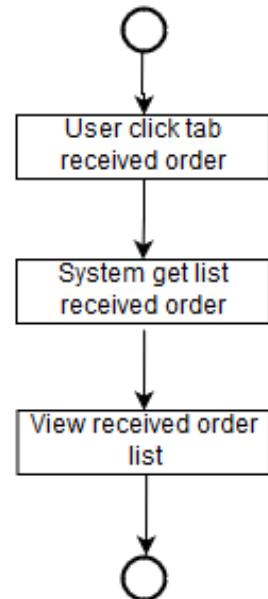
Tabel 4.7 Sprint Review

<i>Sprint</i>	Hasil
1	Berhasil Setup Environment pada Laravel dan React Native, Aplikasi sudah bisa dijalankan dengan menampilkan halaman login yang sudah bisa berfungsi dan masuk ke halaman utama tab navigasi dan pada tab profile sudah ada button logout, tetapi untuk fitur chat masih dalam pengembangan dan dilanjutkan ke <i>sprint</i> berikut nya
2	Fitur Chat sudah bisa berfungsi dengan baik, ditambah lagi dengan aplikasi sudah bisa menampilkan order list pada bagian tab order dan ketika ditekan salah satu dari order list akan diarahkan ke halaman order detail .
3	Aplikasi sudah bisa melakukan aksi terima/tolak order pada order detail dengan status order “Belum diproses”, serta sudah dapat submit nomor resi pada halaman order detail dengan status order “Sedang diproses”. Tetapi untuk fitur ”Dashboard Report” masih dalam pengembangan.
4	Aplikasi sudah bisa menampilkan Dashboard Report yang sebelumnya tidak selesai di sprint sebelum nya serta ada fitur “Filter Report” pada halaman dashboard. Dan juga sudah bisa menampilkan validasi order detail dengan status “Sedang diproses” jika order pengiriman menggunakan “Ralali kargo” dan juga bisa mengupdate status order menjadi “Siap di pickup”. Task API profile sudah selesai namun penggerjaan halaman profile nya masih dalam pengembangan.
5	Sudah bisa menampilkan informasi pada tab profile. Dan juga sudah meng-aplikasi-kan fitur notifikasi untuk order baru . selain itu. Sudah

	memasang event tracking pada setiap screen dan action pada aplikasi ini.
--	--

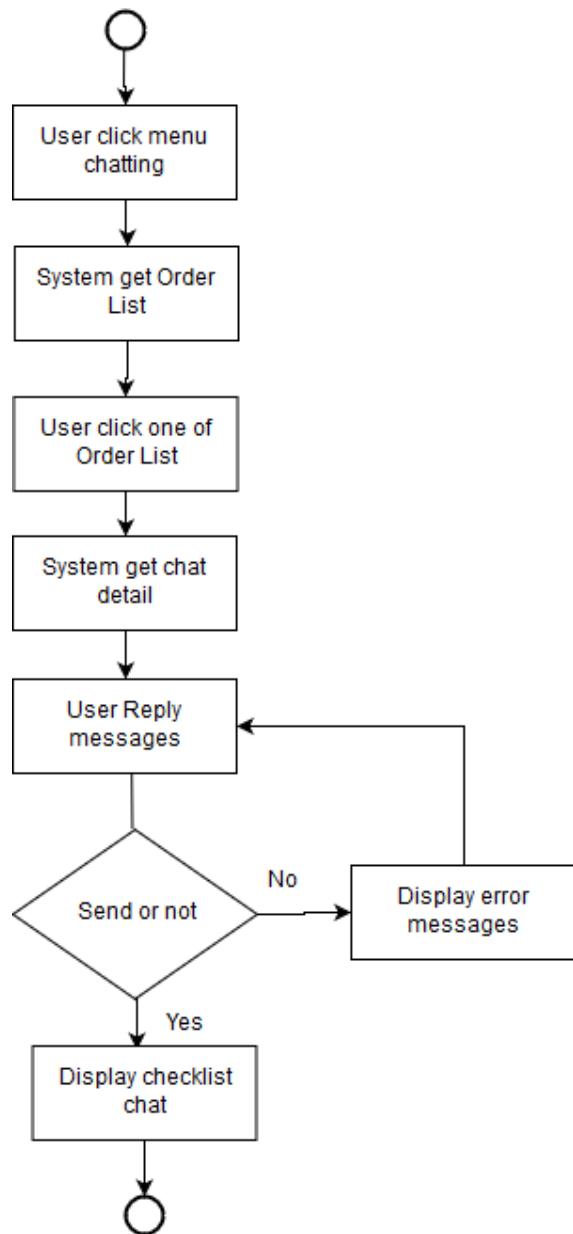
Dibawah ini terdapat diagram *flowchart* yang dirancang sebagai alat bantu komunikasi mengenai alur aplikasi Ralali Seller sebagai kepentingan dokumentasi :

Flow chart Buyer menerima produk

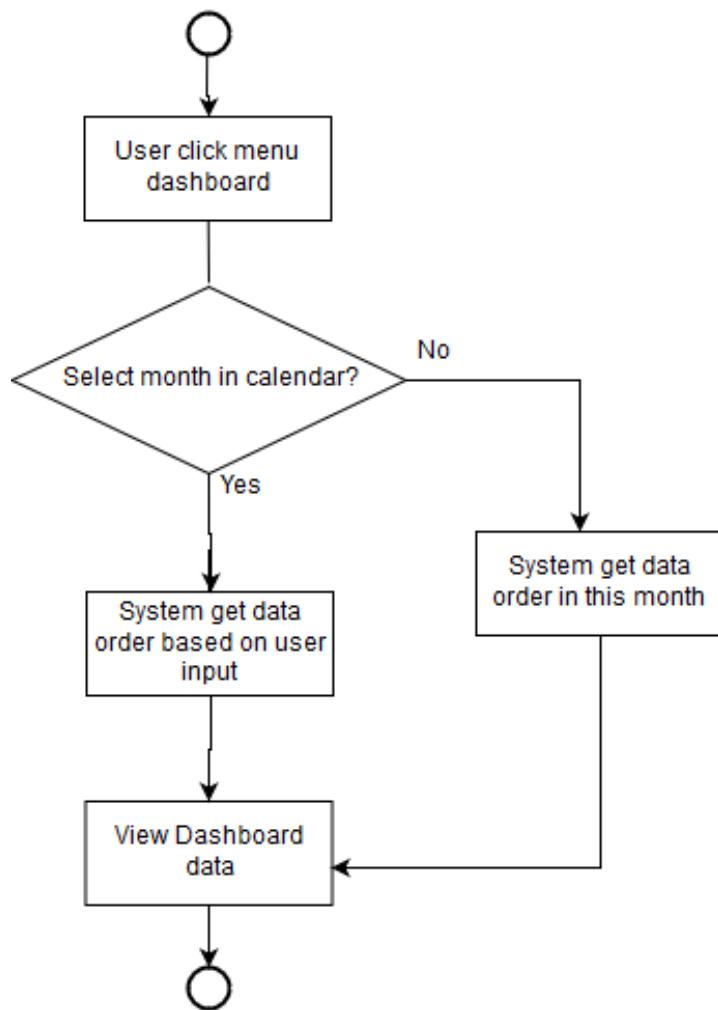


Gambar 4.1 *Buyer Menerima Produk*

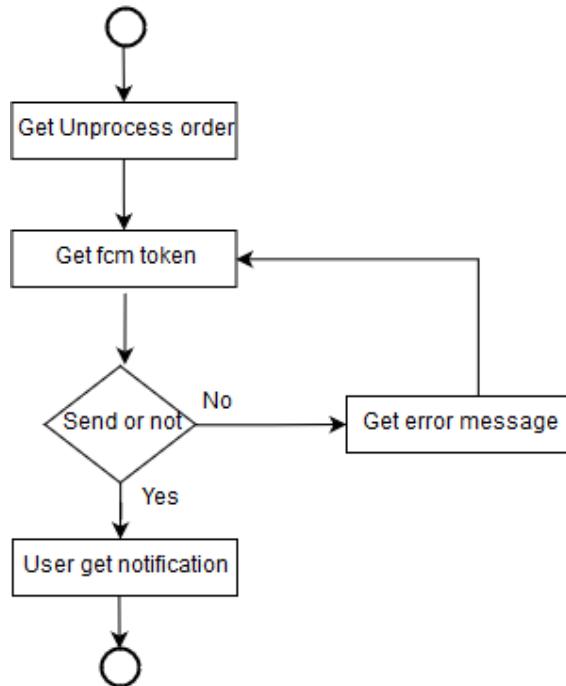
Flow chart Chatting

**Gambar 4.2 Chatting**

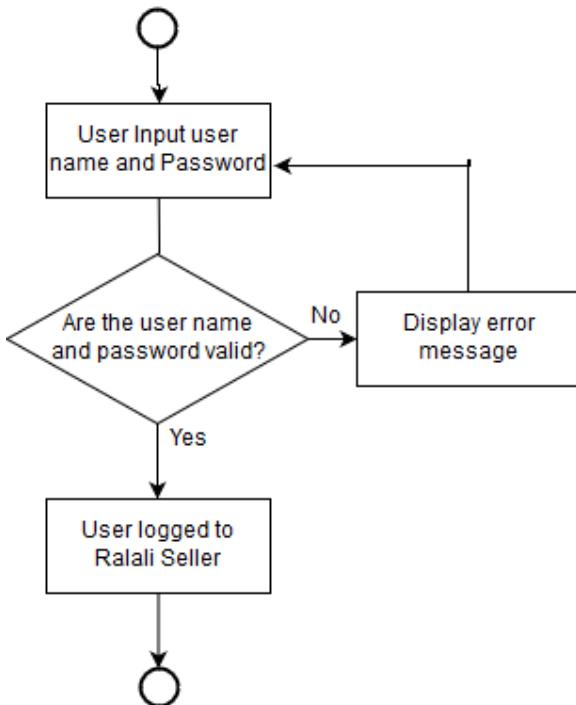
Flow chart View Dashboard Report

Gambar 4.3 *View Dashboard Report*

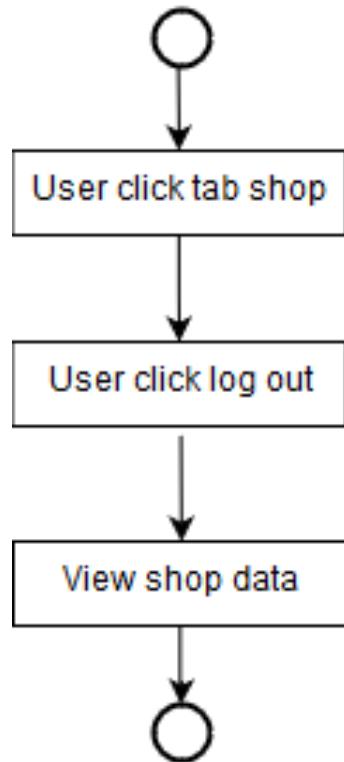
Flow chart get Notification

**Gambar 4.4 Get Notification**

Flow chart Log In

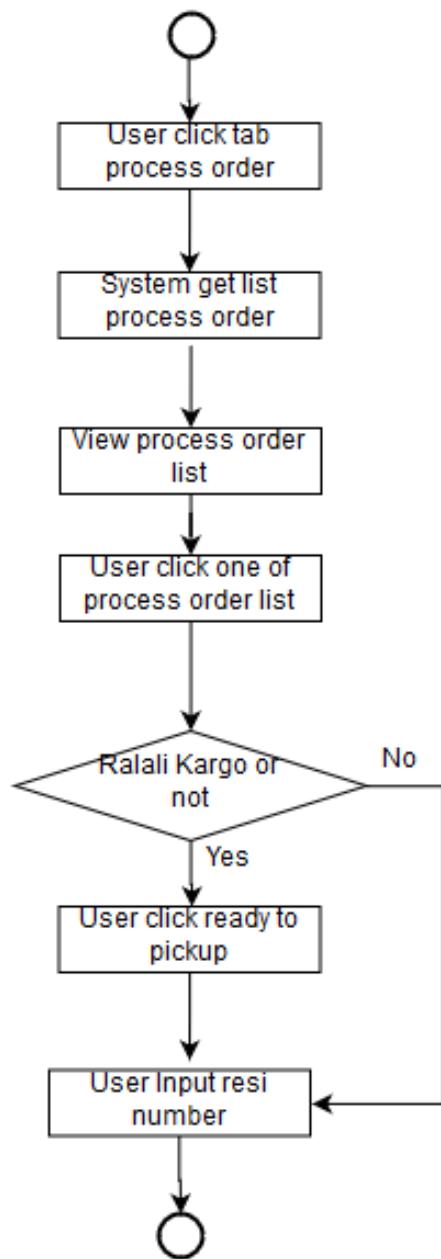
**Gambar 4.5 Login**

Flow chart Log Out

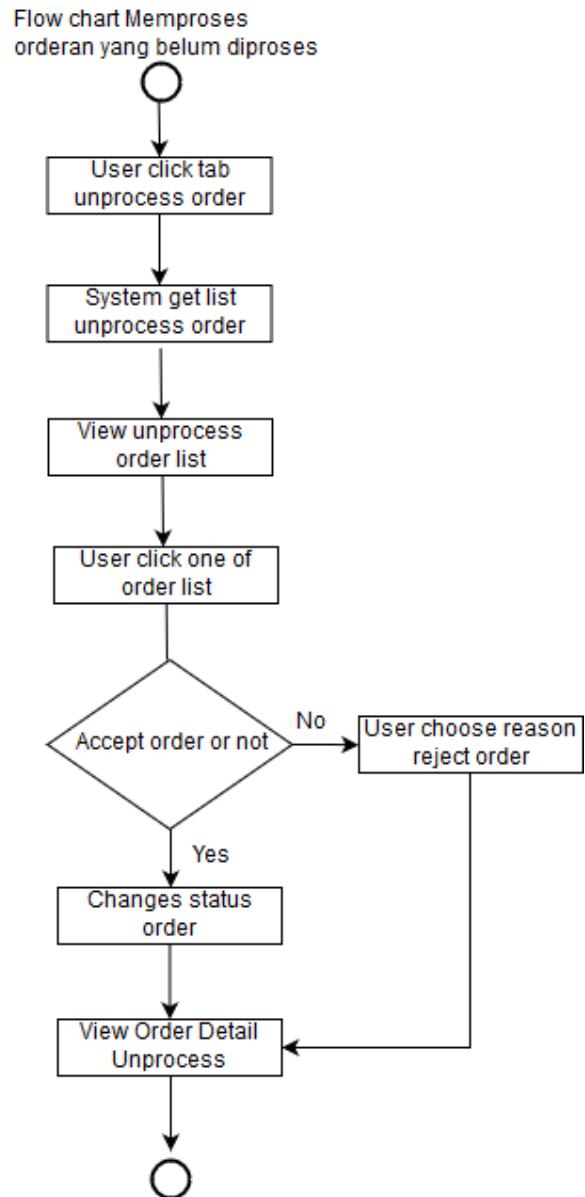


Gambar 4.5 *Logout*

Flow chart Melihat sedang diproses

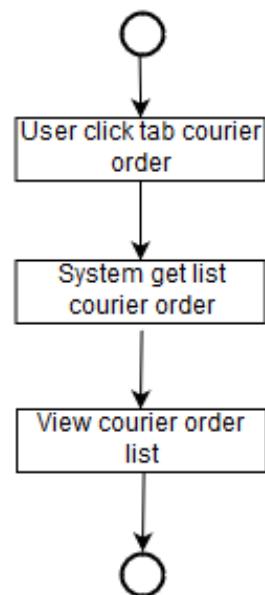


Gambar 4.6 Sedang Diproses



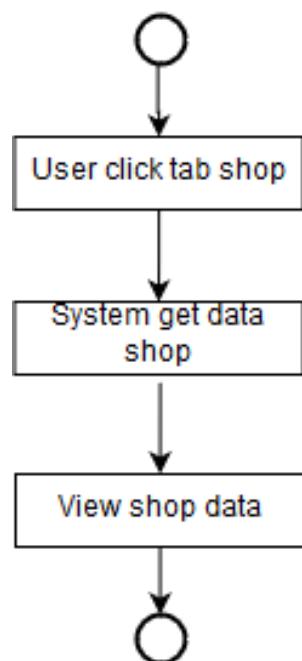
Gambar 4.7 Memproses Orderan Yang Belum Diproses

Flow chart Mengirimkan produk



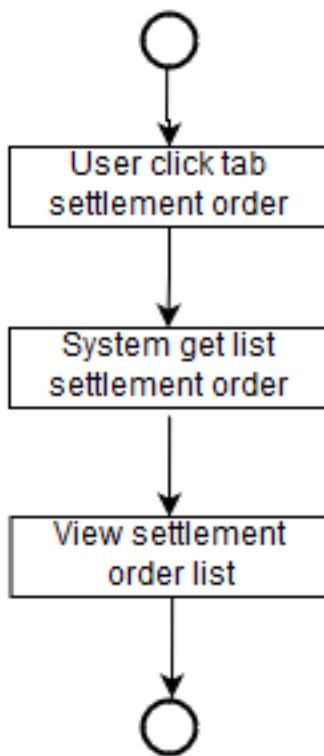
Gambar 4.8 Mengirimkan Produk

Flow chart view toko



Gambar 4.9 View Toko

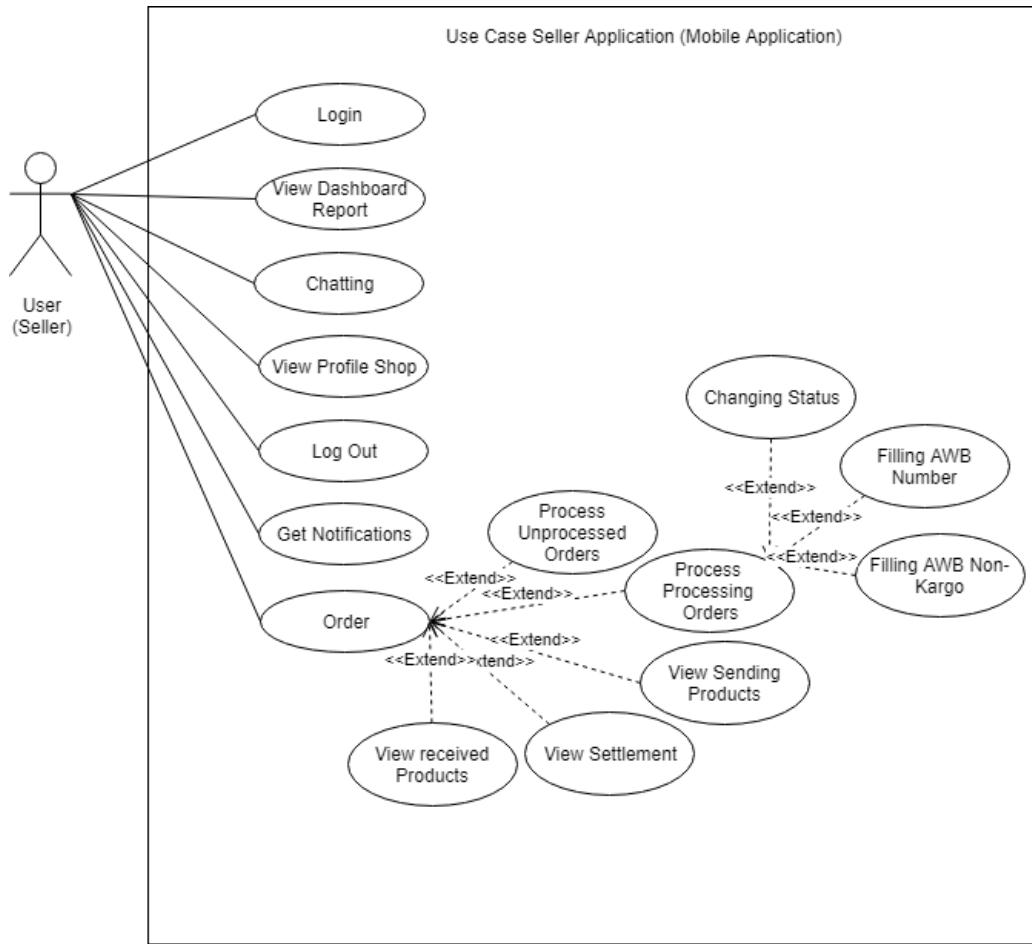
Flow chart Melihat Settlement

**Gambar 4.10 Melihat Settlement**

4.2 Rancangan Sistem

4.2.1 UML (*Unified Modeling Language*)

4.2.1.1 Use Case



Gambar 4.11 Use Case Diagram Pada Ralali Seller

4.2.1.2 Use Case Description

Berikut merupakan *case description* dari aplikasi “Ralali Seller” yang akan dikembangkan :

1. *Use Case Description Login*

Tabel 4.8 Use Case Description Login

Use Case Name	<i>Login</i>
Scenario	<i>Login</i> ke aplikasi Ralali Seller

Triggering event	Menunggu untuk menampilkan <i>Dashboard report</i>		
Brief Description	Setelah aktor membuka aplikasi “Ralali Seller”, aktor akan diminta untuk menginput <i>E-mail</i> dan <i>password</i> pada <i>form</i> yang sudah disediakan.		
Actors	<i>User (Seller)</i>		
Related Use Cases	<i>Use Case Dashboard Report</i>		
Stakeholders	-		
Pre-condition	Aktor membuka aplikasi untuk pertama kalinya		
Post Condition	Sistem akan menampilkan tampilan “ <i>Dashboard Report</i> ”		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
	1	Aktor membuka aplikasi “Ralali Seller”	2 Menampilkan tampilan <i>Splash Screen</i> dan tampilan <i>Login</i>
	3	Aktor menginput <i>E-mail</i> dan <i>password</i> dan mengklik tombol “ <i>Login</i> ”	4 System akan mengecek kesesuaian <i>E-mail</i> dan <i>password</i> di <i>database</i>
			5 System akan menampilkan <i>error message</i> (jika <i>email</i> dan <i>password</i> tidak cocok), atau akan menyimpan data dan melanjutkan ke halaman <i>Dashboard report</i> (Jika <i>email</i> dan <i>password</i> cocok)

Exception Condition	Alt-Step 3	Jika aktor tidak mengisi salah satu kolom <i>E-mail</i> atau kolom <i>Password</i> yang disediakan, dan aktor menginput email dan password yang tidak valid. Maka akan ditampilkan <i>error message</i>
---------------------	------------	---

2. Use Case Description Notifikasi

Tabel 4.9 Use Case Description Notifikasi

Use Case Name	Notifikasi		
Scenario	Mengirimkan notifikasi ke <i>seller</i>		
Triggering event	Menampilkan notifikasi		
Brief Description	Sistem akan mengirimkan notifikasi bahwa terdapat transaksi orderan baru yang masuk dari <i>buyer</i> .		
Actors	<i>Firebase</i>		
Related Use Cases	<i>Use Case Order</i>		
Stakeholders	<i>Staff</i> Ralali		
Pre-condition	Ada <i>buyer</i> yang melakukan transaksi orderan, dan sudah melakukan pembayaran		
Post Condition	<i>Seller</i> akan mendapatkan notifikasi		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
			1 Mengambil token dari <i>Firebase</i>
			2 Menyimpan token dari <i>Firebase</i> ke Database

		3	Orderan masuk
		4	Mengirimkan notifikasi ke <i>seller</i> yang mendapatkan orderan berdasarkan token <i>user</i> yang didapatkan dari Firebase
Exception Condition			

3. Use Case Description View Dashboard Report

Tabel 4.10 Use Case Description View Dashboard Report

Use Case Name	<i>View Dashboard Report</i>
Scenario	Melihat performa penjualan
Triggering Event	-
Brief Description	<ol style="list-style-type: none"> 1. Aktor ingin melihat performa hasil penjualan tokonya. 2. Selain melihat data performa toko sendiri berdasarkan range waktu yang ditentukan, aktor dapat juga mengakses ke beberapa fitur lainnya sesuai dengan kebutuhan yang dibutuhkan (<i>chat</i>, <i>order</i>, <i>toko</i>) (akan dibahas lebih lanjut lagi di <i>Use Case Description Chatting</i>, <i>Use Case Description Order</i>, <i>Use Case Description View Toko</i>).
Actors	<i>User (Seller)</i>
Related Use Cases	-
Stakeholders	-
Pre-condition	Aktor telah selesai melakukan aktivitas <i>Login</i> .

Post Condition	Aktor dapat melihat performa dirinya selama berjualan di Ralali.com sesuai dengan <i>range</i> yang sudah diatur oleh aktor		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
		1	Menampilkan tampilan <i>dashboard report</i> beserta mengambil dan menampilkan data - data hasil penjualan sang aktor pada bulan terakhir.
Exception Conditions	-		

4. Use Case Description Chatting

Tabel 4.11 Use Case Description Chatting

Use Case Name	<i>Chatting</i>
Scenario	<i>User</i> membalas <i>chat</i>
Triggering Event	-
Brief Description	Aktor ingin merespon chat dari <i>buyer</i>
Actor	<i>User (Seller)</i>
Related Use	-

Cases			
Stakeholders	<i>buyer</i>		
Pre-condition	Dari tampilan <i>Dashboard</i> , aktor mengklik pilihan “ <i>Chat</i> ” yang terdapat di bagian bawah		
Post Condition	Aktor sudah berhasil merespon pertanyaan - pertanyaan dari <i>buyer</i>		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
	1	Aktor mengklik tab “ <i>Chat</i> ”	2 Menampilkan tampilan “ <i>Chat</i> ” serta mengambil dan menampilkan daftar data <i>chat</i>
	3	Aktor memilih salah satu <i>chat</i> dan membalas pertanyaan <i>buyer</i>	4 Sistem mengirimkan pesan dari sang aktor kepada <i>buyer</i> .
Exception Conditions	-		

5. Use Case Description Order

Tabel 4.12 Use Case Description Order

Use Case Name	Order
Scenario	Melihat orderan dan juga memproses
Triggering Event	-
Brief Description	Aktor ingin melihat atau memproses suatu orderan. Di dalam tab “Order” terdapat fitur “Memproses Orderan Yang Belum

	Diproses”, “Sedang Proses”, “Pengiriman Produk”, “Buyer Menerima Produk”, dan juga “Settlement” (akan dibahas lebih lanjut lagi di <i>Use Case Description</i> Memproses Orderan Yang Belum Diproses, <i>Use Case Description</i> Sedang Proses, <i>Use Case Description</i> Pengiriman Produk, <i>Use Case Description</i> Buyer Menerima Produk, dan juga <i>Use Case Description Settlement</i>).		
Actor	<i>User (Seller)</i>		
Related Use Cases	<i>Use Case</i> Memproses Orderan Yang Belum Diproses, <i>Use Case</i> Sedang Proses, <i>Use Case</i> Pengiriman Produk, <i>Use Case</i> Buyer Menerima Produk, dan juga <i>Use Case Settlement</i>		
Stakeholders	-		
Pre-condition	Dari tampilan <i>Dashboard</i> , aktor mengklik pilihan “Order” yang terdapat di bagian bawah		
Post Condition	Aktor dapat melihat orderan yang diterima		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
	1	Aktor membuka halaman “Order”	2 Sistem menampilkan tampilan order, dan segera menampilkan juga tampilan “Belum Diproses” dan juga ditampilkan daftar orderan yang belum diproses oleh Aktor
Exception Conditions	-		

6. *Use Case Description* Memproses Orderan Yang Belum Diproses

Tabel 4.13 *Use Case Description* Memproses Orderan Yang Belum Diproses

Use Case Name	Memproses Orderan Yang Belum Diproses
---------------	---------------------------------------

Scenario	Memproses orderan yang ada di daftar orderan yang belum diproses			
Triggering Event	Sistem memindahkan daftar orderan yang sudah direspon ke tab sedang diproses			
Brief Description	Aktor ingin melihat atau memproses suatu orderan			
Actor	<i>User (Seller)</i>			
Related Use Cases	Sedang Diproses <i>Use Case Description</i>			
Stakeholders	<i>Staff Ralali, buyer</i>			
Pre-condition	Dari tampilan <i>Dashboard</i> , aktor mengklik pilihan “Order” yang terdapat di bagian bawah, ataupun aktor mengklik tombol “Belum Diproses” pada halaman “Order”			
Post Condition	Aktor sudah berhasil merespon suatu orderan yang belum diproses			
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>	
			1	Sistem menampilkan tampilan “Belum Diproses” dan juga menampilkan daftar orderan yang belum diproses oleh Aktor
	2	Aktor dapat mengklik salah satu orderan pada daftar	3	Sistem mengambil dan menampilkan detail pemesanan produk
	4	Aktor dapat membuka salah satu produk pada detail pemesanan produk, dan memberikan	5	Sistem menyimpan respon <i>seller</i> pada penyimpanan sementara dan mengecek apakah semua produk pada produk detail pemesanan tersebut telah direspon semua

		respon		nya. Jika sudah semua maka tombol “Proses Pesanan” akan dienable. Jika belum semua, proses nomor 4 akan dijalankan kembali
	6	Aktor mengklik tombol “Proses Pesanan”	7	Sistem menyimpan respon aktor , memproses orderan dan mengirimkan respon yang diberikan aktor kepada <i>customer service, seller, dan buyer.</i>
Exception Conditions	Alt-Step 4	Aktor akan diminta respon untuk menerima suatu orderan atau menolak suatu orderan beserta alasan penolakan orderan tersebut		

7. *Use Case Description* Melihat Sedang Diproses

Tabel 4.14 Use Case Description Melihat Sedang Diproses

Use Case Name	Melihat Sedang Diproses
Scenario	Memproses orderan yang berada dalam status “Sedang Diproses”
Triggering Event	Sistem akan menampilkan tampilan data - data order detail dari orderan yang dipilih aktor tersebut
Brief Description	<ol style="list-style-type: none"> 1. Aktor ingin melihat dan mengirimkan nomor resi pada orderan yang sedang diproses 2. Di dalam tab “Sedang Diproses” terdapat 3 kondisi yaitu terdapat logo kargo <i>orange</i>, logo kargo abu - abu, dan juga tidak ada logo kargo (akan dibahas lebih lanjut lagi di <i>Use Case Description Mengubah Status Menjadi Siap Di-Pick-Up</i>, <i>Use Case Description Mengisi Nomor Resi Dari Ralali Kargo</i> , <i>Use Case Description Mengisi Nomor Resi</i>).
Actor	<i>User (Seller)</i>

Related Use Case	<i>Use Case Description</i> Mengubah Status Menjadi Siap Di-Pick-Up, <i>Use Case Description</i> Mengisi Nomor Resi Dari Ralali Kargo , <i>Use Case Description</i> Mengisi Nomor Resi		
Stakeholders	-		
Pre-condition	Aktor mengklik tombol “Sedang Diproses” pada halaman “Order”		
Post Condition	Aktor telah melihat daftar orderan yang sedang diproses dan akan memilih salah satu orderan dari daftar tersebut untuk mengisi nomor resi		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
		1	Sistem mengambil dan menampilkan data - data orderan sedang diproses
	2	Kemudian aktor dapat memilih salah satu dari yang ada di daftar (akan dibahas lebih lanjut lagi di <i>Use Case Description</i> Mengubah Status Menjadi Siap Di-Pick-Up, <i>Use Case Description</i> Mengisi Nomor Resi Dari Ralali Kargo , <i>Use Case Description</i> Mengisi Nomor Resi)	
Exception Conditions	-		

8. *Use Case Description* Mengubah Status Menjadi Siap Di-pick-up

Tabel 4.15 Use Case Description Mengubah Status Menjadi Siap Di-pick-up

Use Case Name	Mengubah Status Orderan Menjadi Siap Di- <i>pick-up</i>		
Scenario	Mengubah Status yang ada di list menjadi siap di- <i>pick-up</i>		
Triggering Event	Sistem mengubah warna pada icon Ralali Kargo (hanya pada order yang telah berubah status nya menjadi “siap di- <i>pick-up</i> ”) yang sebelumnya warna orange menjadi warna abu - abu		
Brief Description	Aktor ingin mengkonfirmasi barang siap di- <i>pick-up</i> , dan mendapatkan nomor resi		
Actor	<i>User (Seller)</i>		
Related Use Cases	Mengisi Nomor Resi Dari Ralali Kargo <i>Use Case Description</i>		
Stakeholders	<i>Staff Ralali, Logistics, Buyer</i>		
Pre-condition	Aktor mengklik tombol “Sedang Diproses” pada halaman “Order”		
Post Condition	Aktor telah melakukan konfirmasi barang siap di- <i>pick-up</i> , dan ketika pihak logistik datang untuk melakukan <i>pick-up</i> barang. Aktor akan memberikan nomor resi disaat bersamaan		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
			1 Sistem mengambil dan menampilkan data - data orderan sedang diproses
	2 Kemudian aktor dapat memilih salah satu produk pada produk detail dengan logo kargo <i>orange</i>	3	Sistem menampilkan detail pemesanan produk

	4	Mengklik tombol barang siap di-pick-up	5	Mengirim respon yang diberikan oleh aktor ke <i>Customer Service, buyer, seller</i>
			6	<i>Popup alamat pick-up</i>
	7	Konfirmasi alamat <i>pick-up</i>	8	Mengubah status di Database menjadi siap di- <i>pick-up</i> dan mengirimkan informasi ke pihak logistik
Exception Conditions	-			

9. *Use Case Description* Mengisi Nomor Resi Dari Ralali Kargo

Tabel 4.16 Use Case Description Mengisi Nomor Resi Dari Ralali Kargo

Use Case Name	Mengisi Nomor Resi Dari Ralali Kargo
Scenario	Aktor mengisi nomor resi dimana nomor resi tersebut didapatkan dari pihak Ralali Kargo. Nomor resi tersebut akan diisi di kolom nomor resi di aplikasi Ralali Seller
Triggering Event	Sistem memindahkan daftar yang sudah diisi dengan nomor resi tersebut ke tab pengiriman
Brief Description	Aktor ingin mengisi nomor resi pada orderan yang memakai jasa kurir Ralali kargo
Actor	<i>User (Seller)</i>
Related Use Cases	Pengiriman <i>Use Case Description</i>

Stakeholders	<i>Staff Ralali, Logistics, Buyer</i>		
Pre-condition	Aktor mengklik tombol “Sedang Diproses” pada halaman “Order”		
Post Condition	Aktor telah mengisi dan mengirim nomor resi.		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
			1 Sistem mengambil dan menampilkan data - data orderan sedang diproses
	2 Kemudian aktor dapat memilih salah satu produk pada produk detail dengan logo kargo warna abu - abu	3	Sistem menampilkan detail pemesanan produk
	4 Aktor mengisi nomor resi yang diberikan, dan mengklik tombol “Submit Resi”	5	Sistem mengecek apakah orderan tersebut sudah pernah diinput resi sebelumnya.
		6	Jika hasil pengecekan menunjukkan bahwa orderan belum pernah diisi nomor resi, maka sistem mengirimkan nomor resi tersebut ke pihak <i>Customer Service, buyer, seller</i> . Tetapi apabila orderan tersebut sudah pernah diinput nomor resi sebelumnya. Maka, sistem akan menampilkan <i>error message</i>

	7	Aktor mengklik tombol “OK” (Jika muncul <i>error message</i>)		
Exception Conditions	-			

10. *Use Case* Mengisi Nomor Resi

Tabel 4.17 Use Case Description Mengisi Nomor Resi

Use Case Name	Mengisi Nomor Resi		
Scenario	Aktor mengisi nomor resi yang aktor dapatkan dari kurir ke kolom nomor resi yang tersedia di aplikasi Ralali Seller		
Triggering Event	Sistem memindahkan daftar yang sudah diisi dengan nomor resi tersebut ke tab pengiriman		
Brief Description	Aktor ingin mengisi nomor resi pada orderan		
Actor	<i>User (Seller)</i>		
Related Use Cases	<i>Pengiriman Use Case Description</i>		
Stakeholders	<i>Staff Ralali, Logistics, Buyer</i>		
Pre-condition	Aktor mengklik tombol “Sedang Diproses” pada halaman “Order”		
Post Condition	Aktor telah mengisi dan mengirim nomor resi.		
Flow Of Activities	<i>Actor Action</i> 1		<i>System Action</i> Sistem mengambil dan menampilkan data - data orderan

			sedang diproses
2	Kemudian aktor dapat memilih salah satu produk pada produk detail yang tidak menggunakan logo kargo	3	Sistem menampilkan detail pemesanan produk
4	Aktor menginput nomor resi yang diterima aktor dari jasa kurir, dan mengklik tombol “Submit Resi”	5	Sistem mengecek apakah orderan tersebut sudah pernah diinput nomor resi.
		6	Jika hasil pengecekan menunjukkan bahwa orderan belum pernah diinput nomor resi, maka sistem akan mengirimkan nomor resi yang diinput oleh aktor ke <i>Customer Service, buyer, seller</i> . Tetapi apabila orderan tersebut sudah pernah diinput nomor resi sebelumnya. Maka, sistem akan menampilkan <i>error message</i> .
7	Aktor mengklik tombol “OK” (Jika muncul <i>error message</i>)		
Exception Conditions			-

11. *Use Case Description Mengirimkan Produk*

Tabel 4.18 Use Case Description Mengirimkan Produk

Use Case Name	Mengirimkan Produk			
Scenario	Aktor melihat produk orderan nya yang sedang dalam proses pengiriman			
Triggering Event	-			
Brief Description	Aktor ingin melihat orderan yang sedang dalam proses pengiriman			
Actor	<i>User (Seller)</i>			
Related Use Cases	-			
Stakeholders	-			
Pre-condition	Aktor mengklik tombol “Pengiriman” pada halaman “Order”			
Post Condition	Aktor telah melihat salah satu orderan yang sedang dalam tahap pengiriman			
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>	
		1	Sistem mengambil dan menampilkan data - data orderan yang sedang dalam proses pengiriman	
	2	Aktor dapat memilih salah satu orderan yang ada di daftar	3	Sistem mengambil dan menampilkan detail produk pemesanan dari order yang dipilih oleh aktor
Exception	-			

Conditions	
------------	--

12. *Use Case Description Buyer Menerima Produk*

Tabel 4.19 Use Case Description Buyer Menerima Produk

Use Case Name	<i>Buyer Menerima Produk</i>		
Scenario	Aktor melihat produk orderan yang sudah diterima oleh <i>buyer</i>		
Triggering Event	-		
Brief Description	Aktor ingin melihat orderan yang sudah dalam status “Diterima”		
Actor	<i>User (Seller)</i>		
Related Use Cases	-		
Stakeholders	-		
Pre-condition	Aktor mengklik tombol “Diterima” pada halaman “Order”		
Post Condition	Aktor telah melihat salah satu orderan yang sudah dalam status “Diterima”		
Flow Of Activities	Actor Action 1		System Action
			1 Sistem mengambil dan menampilkan data - data orderan yang sudah dalam status “Diterima”
	2	Aktor dapat memilih salah satu orderan yang ada di daftar	3 Sistem mengambil dan menampilkan detail produk pemesanan dari order yang dipilih oleh aktor

Exception Conditions	-
----------------------	---

13. *Use Case Description Melihat Settlement*

Tabel 4.20 Use Case Description Melihat Settlement

Use Case Name	Melihat <i>Settlement</i>		
Scenario	Aktor melihat orderan - orderan yang sudah masuk ke tahap <i>settlement</i>		
Triggering Event	-		
Brief Description	Aktor ingin melihat orderan yang sudah dalam status “ <i>Settlement</i> ”		
Actor	<i>User (Seller)</i>		
Related Use Cases	-		
Stakeholders	-		
Pre-condition	Aktor mengklik tombol “ <i>Settlement</i> ” pada halaman “Order”		
Post Condition	Aktor telah melihat salah satu orderan yang sudah dalam status “ <i>Settlement</i> ”		
Flow Of Activities	Actor Action		System Action
		1	Sistem mengambil dan menampilkan data - data orderan yang sudah dalam tahap “ <i>Settlement</i> ”
		2	Aktor dapat memilih salah satu orderan yang
		3	Sistem mengambil dan menampilkan detail produk

		ada di daftar	pemesanan dari order yang dipilih oleh aktor
Exception Conditions	-		

14. *Use Case Description View Toko***Tabel 4.21 Use Case Description View Toko**

Use Case Name	<i>View Toko</i>		
Scenario	Aktor melihat data profil toko nya		
Triggering Event	-		
Description	Aktor ingin melihat profil toko nya		
Actor	<i>User (Seller)</i>		
Related Use Cases	<i>Use Case Description Logout</i>		
Stakeholders	-		
Pre-condition	Dari tampilan <i>Dashboard report</i> , aktor mengklik pilihan “Toko” yang terdapat di bagian bawah		
Post Condition	Aktor dapat melihat informasi toko sang aktor		
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>
	1	Aktor membuka halaman “Toko” dimana fitur tersebut terdapat di bagian	2 Sistem menampilkan tampilan Toko, dan terdapat 2 pilihan, dimana aktor dapat memilih tombol “Keluar” untuk <i>Logout</i> ,

		bawah fitur <i>Dashboard</i>		dan juga dapat memilih tombol “Pengaturan Toko” untuk melihat informasi - informasi mengenai toko nya
	3	Aktor memilih “Pengaturan Toko”	4	Sistem menampilkan tampilan Informasi Toko, kemudian mengambil serta menampilkan data - data mengenai toko
	5	Aktor memilih tab “Pengiriman” yang terdapat pada sebelah tab “Informasi”	5	Sistem menampilkan tampilan Pengiriman, kemudian mengambil dan menampilkan data - data gudang sang aktor
	7	Aktor memilih tab “Pembayaran” yang terdapat pada sebelah tab “Pengiriman”	8	Sistem menampilkan tampilan Pengiriman, kemudian mengambil serta menampilkan data - data pembayaran yang aktor daftarkan di Ralali.com
Exception Conditions	Alt-Step 3	Jika aktor mengklik tombol “Keluar”, maka dilihat pada <i>Use Case Description Logout</i>		

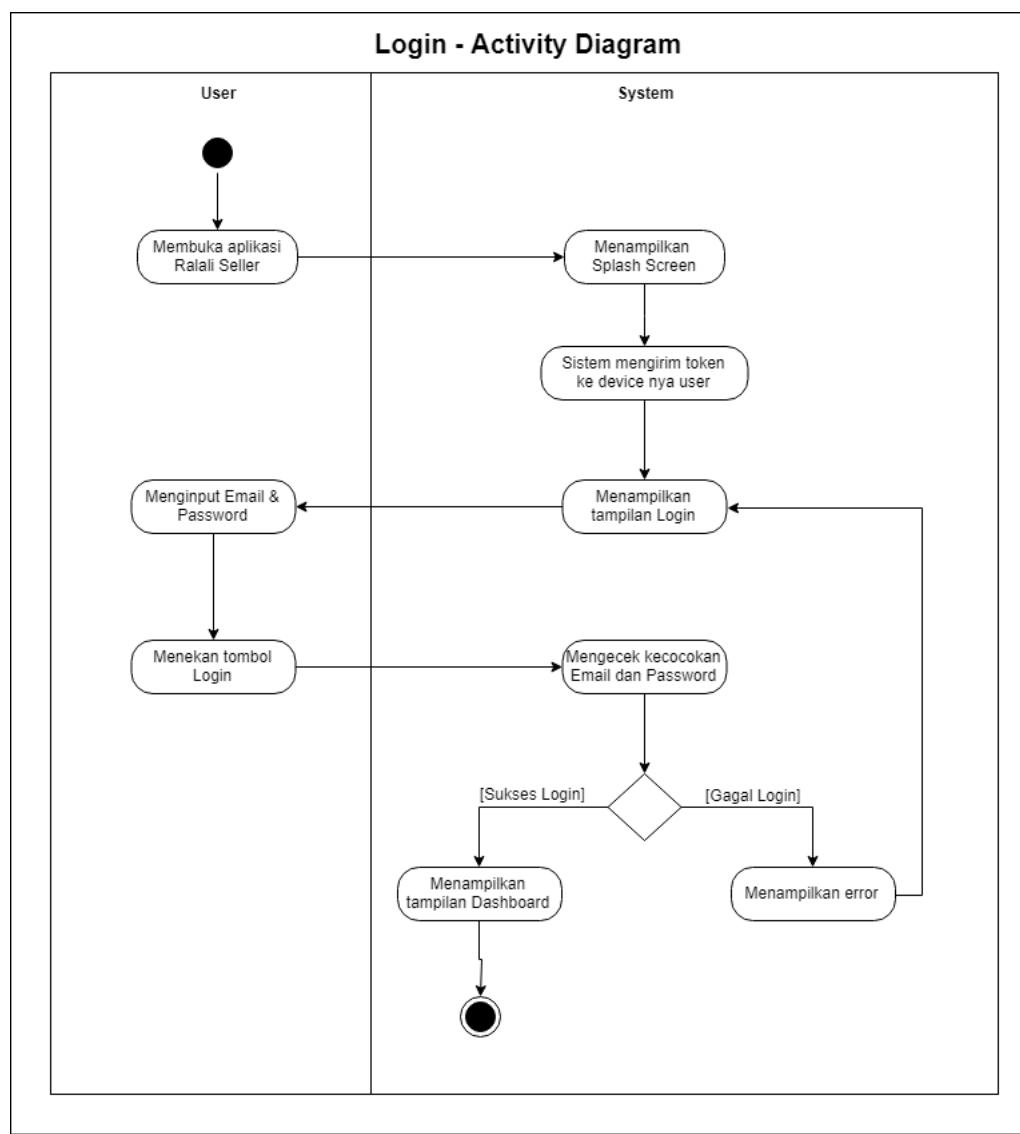
15. Use Case Description Logout

Tabel 4.22 Use Case Description Logout

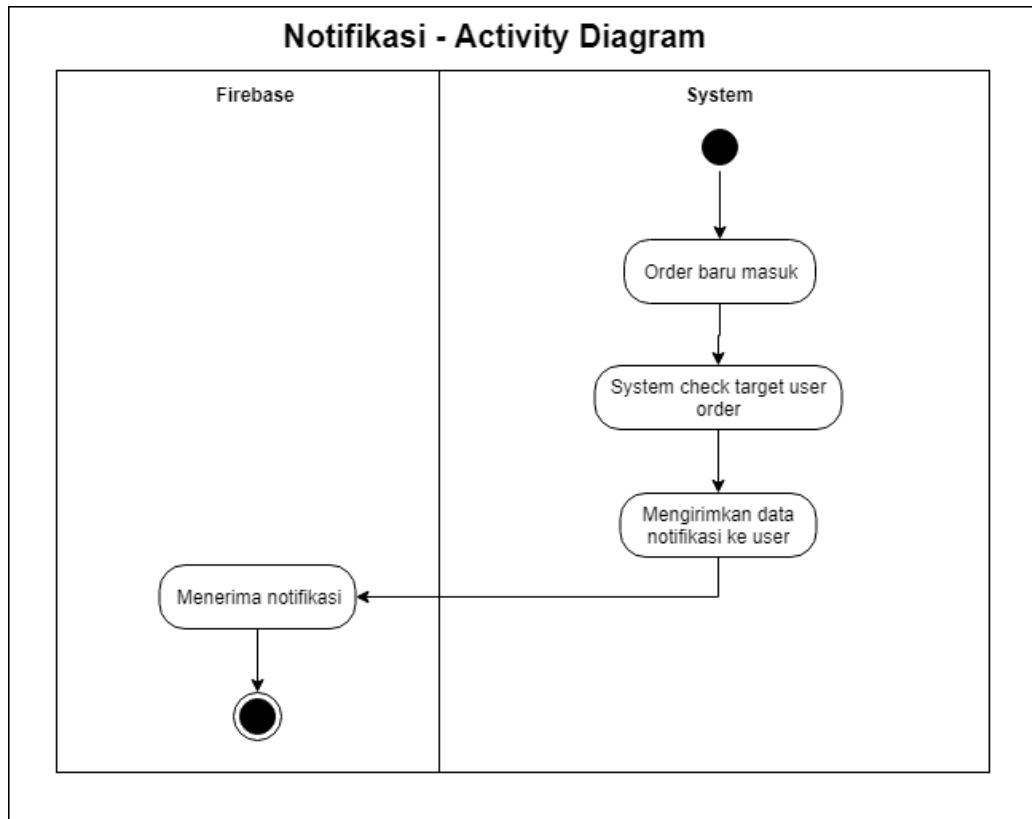
Use Case Name	<i>Logout</i>
Scenario	Aktor melakukan proses keluar dari akun pada aplikasi Ralali <i>Seller</i>
Triggering Event	Menghapus token yang sebelumnya diambil dari Firebase pas login

Brief Description	Aktor ingin melakukan aktivitas Logout dari akun nya pada aplikasi Ralali Seller			
Actor	<i>User (Seller)</i>			
Related Use Cases	-			
Stakeholders	-			
Pre-condition	Dari tampilan <i>Dashboard report</i> , aktor mengklik pilihan “Toko” yang terdapat di bagian bawah, kemudian memilih pilihan “Keluar”			
Post Condition	Aktor telah berhasil melakukan aktivitas <i>Logout</i>			
Flow Of Activities	<i>Actor Action</i>		<i>System Action</i>	
	1	Aktor membuka halaman “Toko” dimana pilihannya terdapat di bagian bawah <i>Dashboard</i>	2	Sistem menampilkan tampilan Toko, dimana terdapat 2 pilihan, yaitu dimana aktor dapat memilih tombol “Keluar” untuk melakukan <i>Logout</i> , dan juga dapat memilih tombol “Pengaturan Toko” untuk melihat informasi - informasi mengenai toko nya
	3	Aktor memilih “Keluar”	4	Sistem memproses keluar dari akun dan menampilkan tampilan <i>Login</i>
Exception Conditions	Alt-Step 3	Jika aktor mengklik tombol “Pengaturan Toko”, maka dapat dilihat pada <i>Use Case Description Toko</i>		

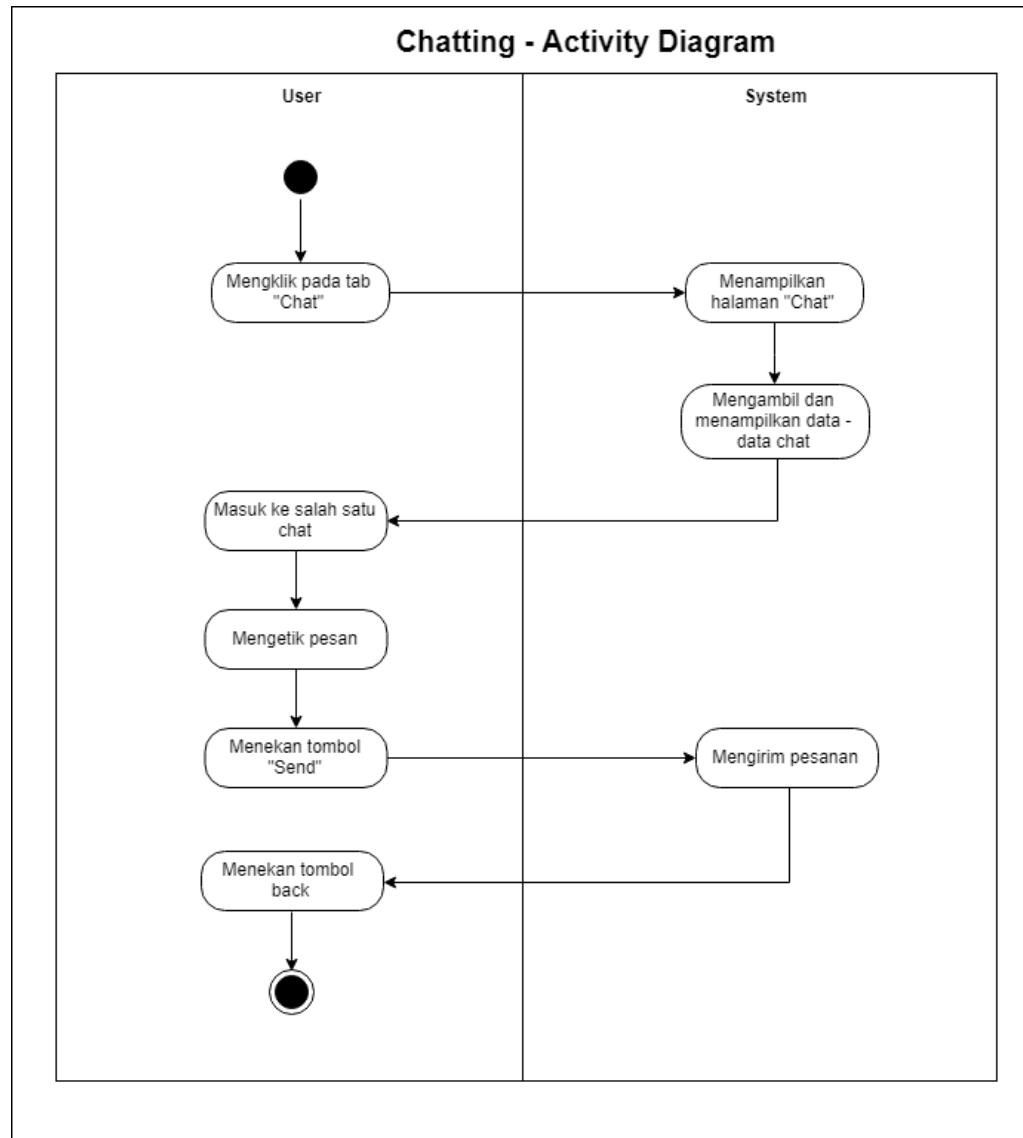
4.2.1.3 Activity Diagram



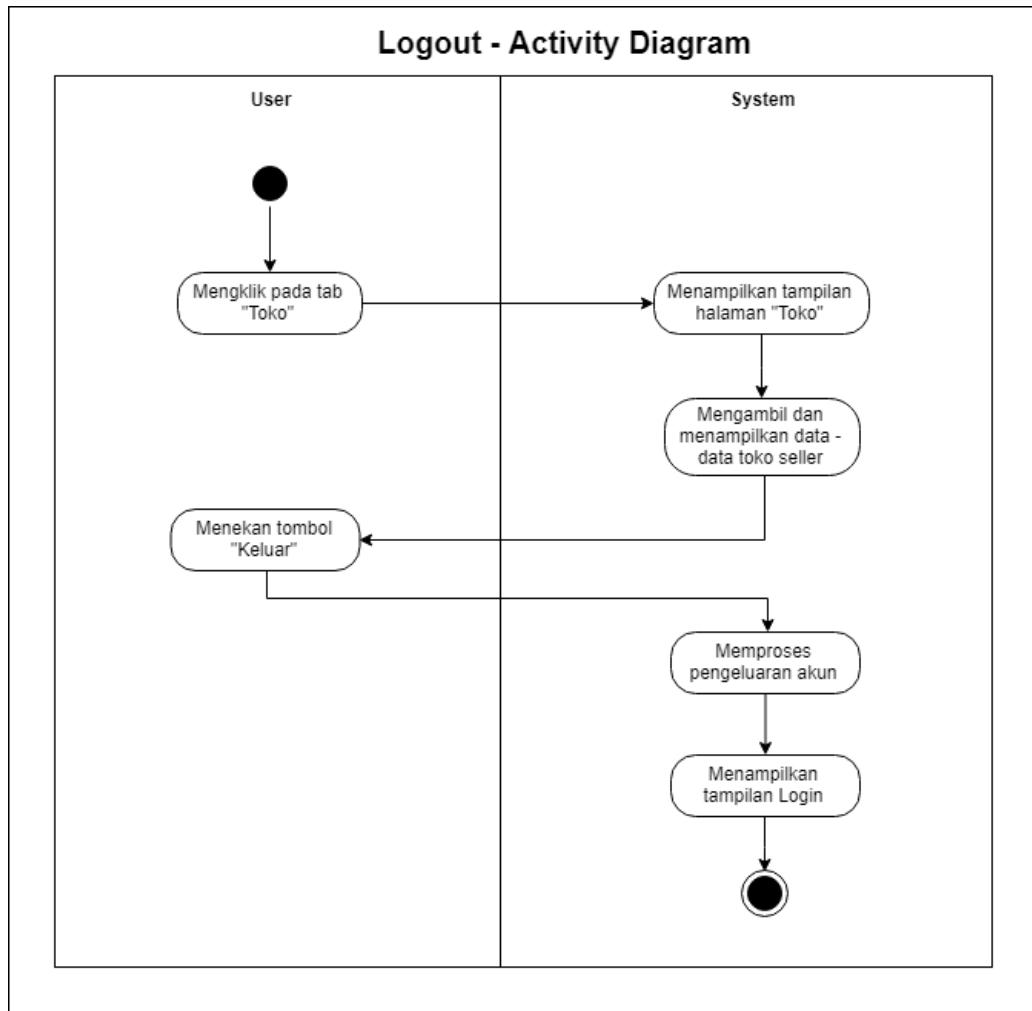
Gambar 4.12 *Login Activity Diagram*



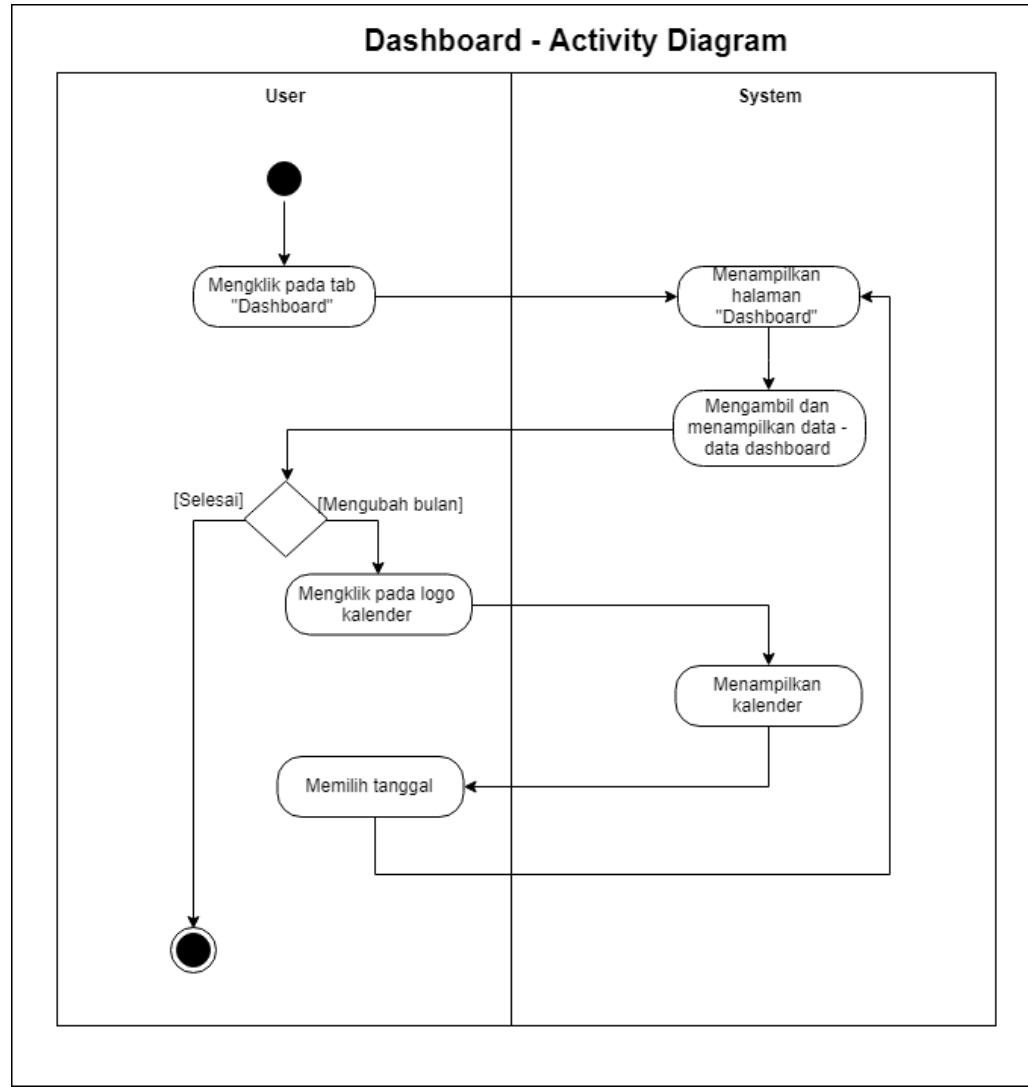
Gambar 4.13 Notifikasi Activity Diagram

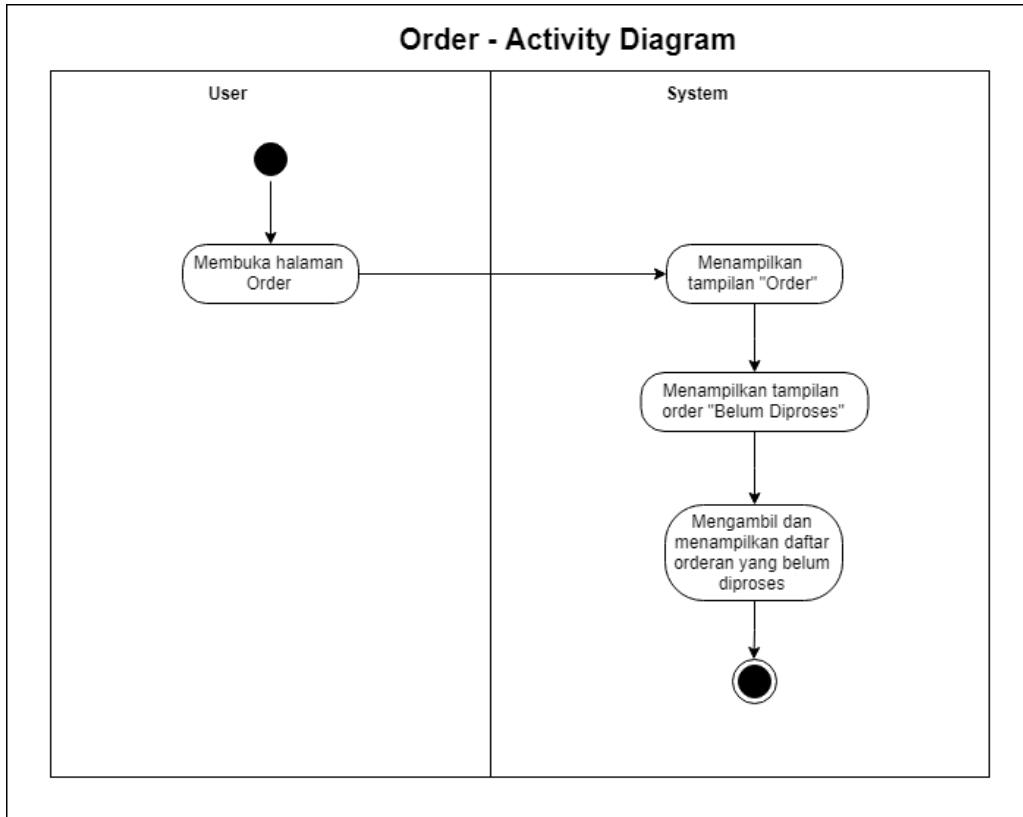


Gambar 4.14 *Chatting Activity Diagram*

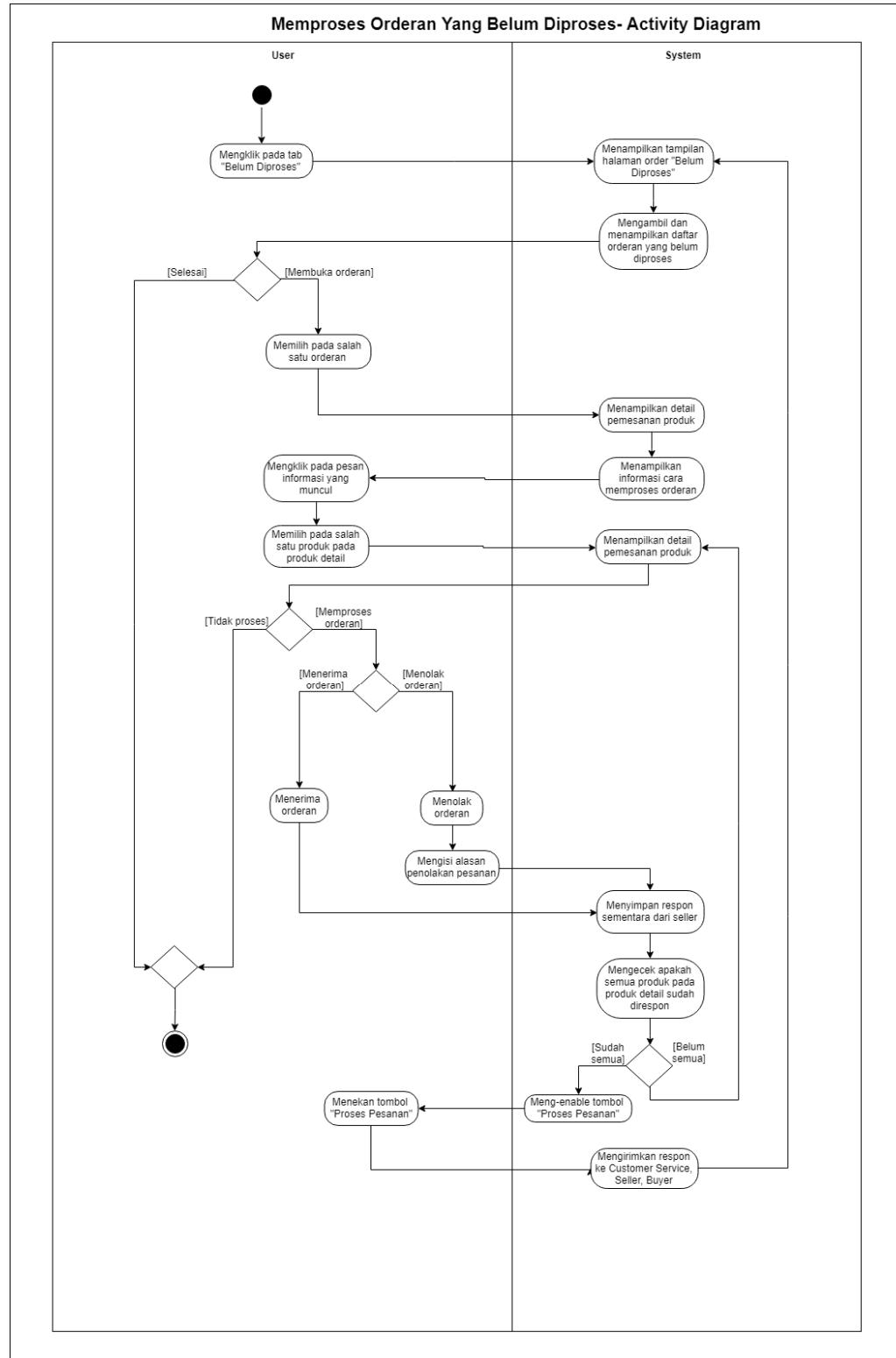


Gambar 4.15 Logout Activity Diagram

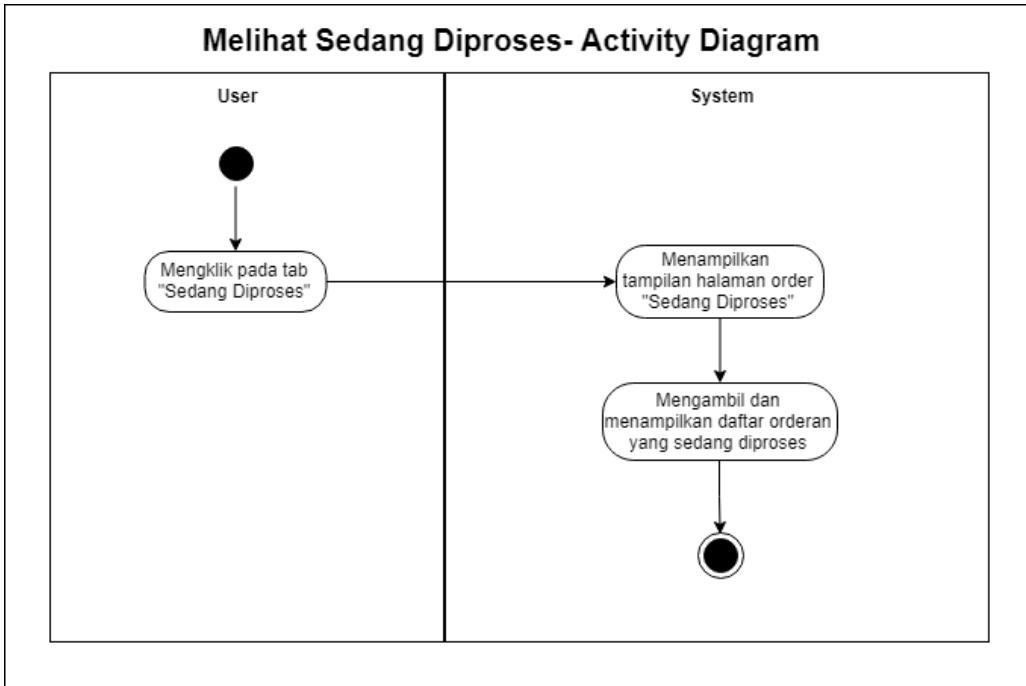




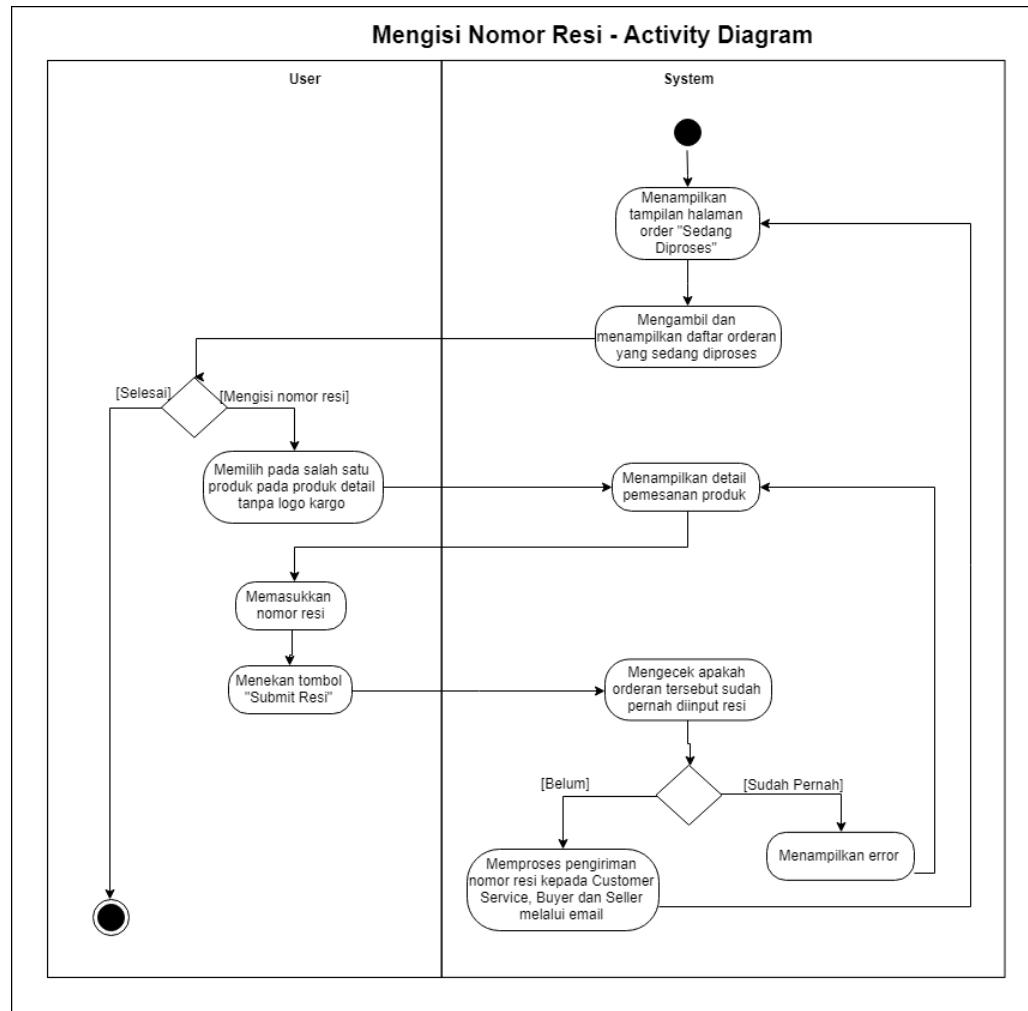
Gambar 4.17 Order Activity Diagram



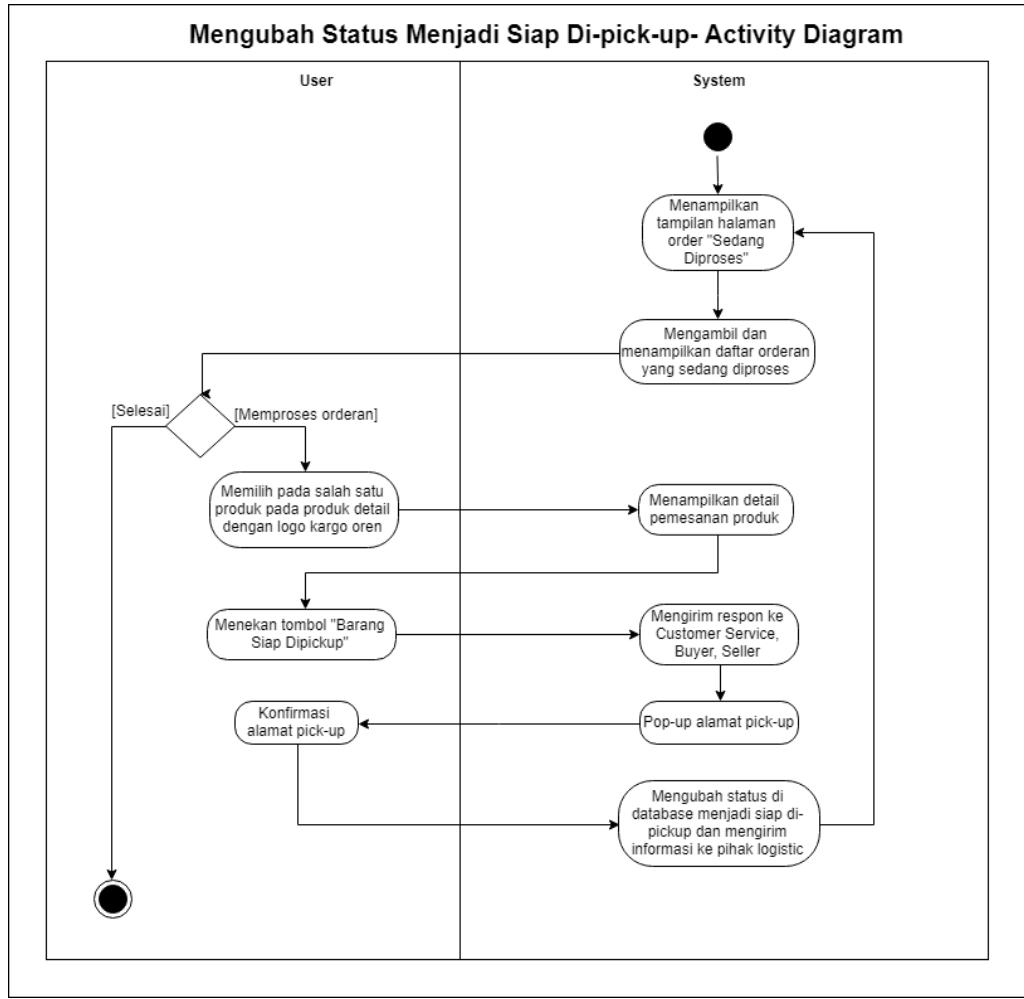
Gambar 4.18 Memproses Orderan Yang Belum Diproses Activity Diagram



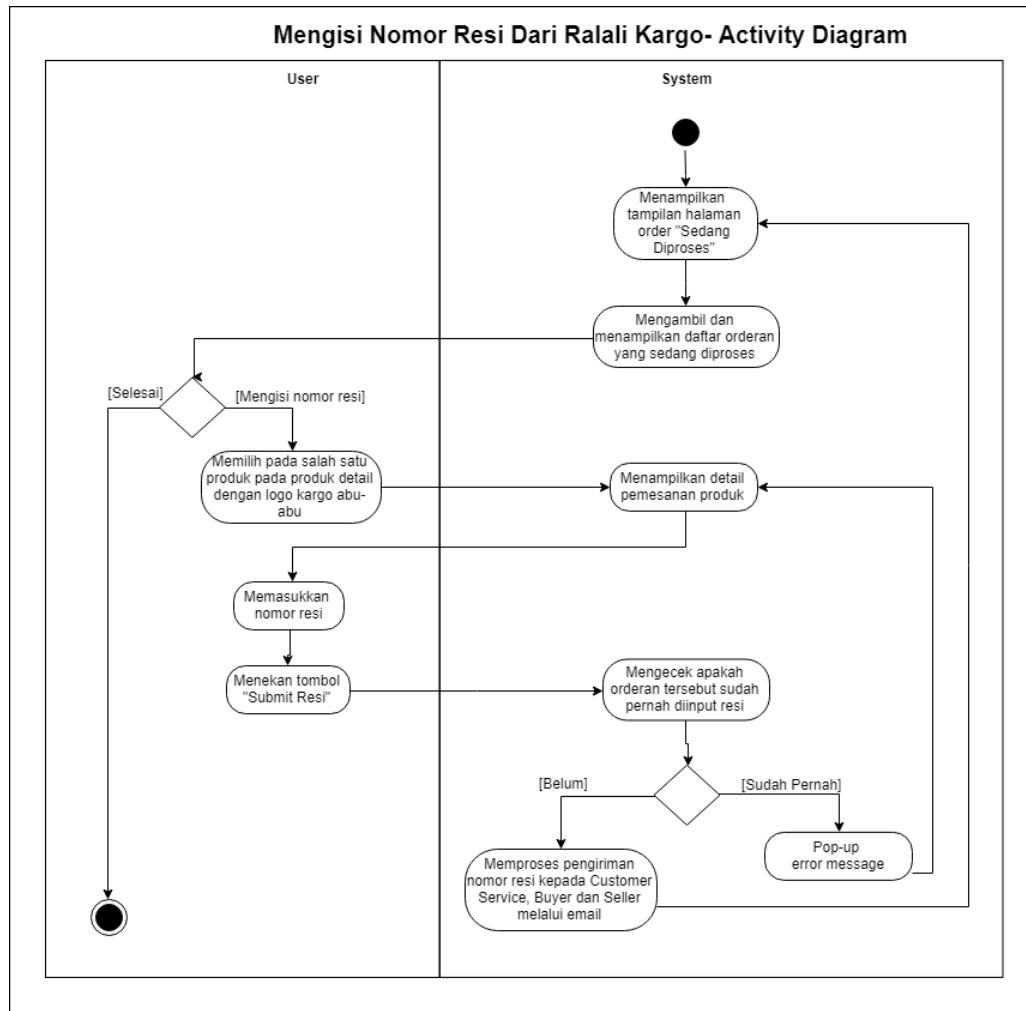
Gambar 4.19 Melihat Sedang Diproses Activity Diagram



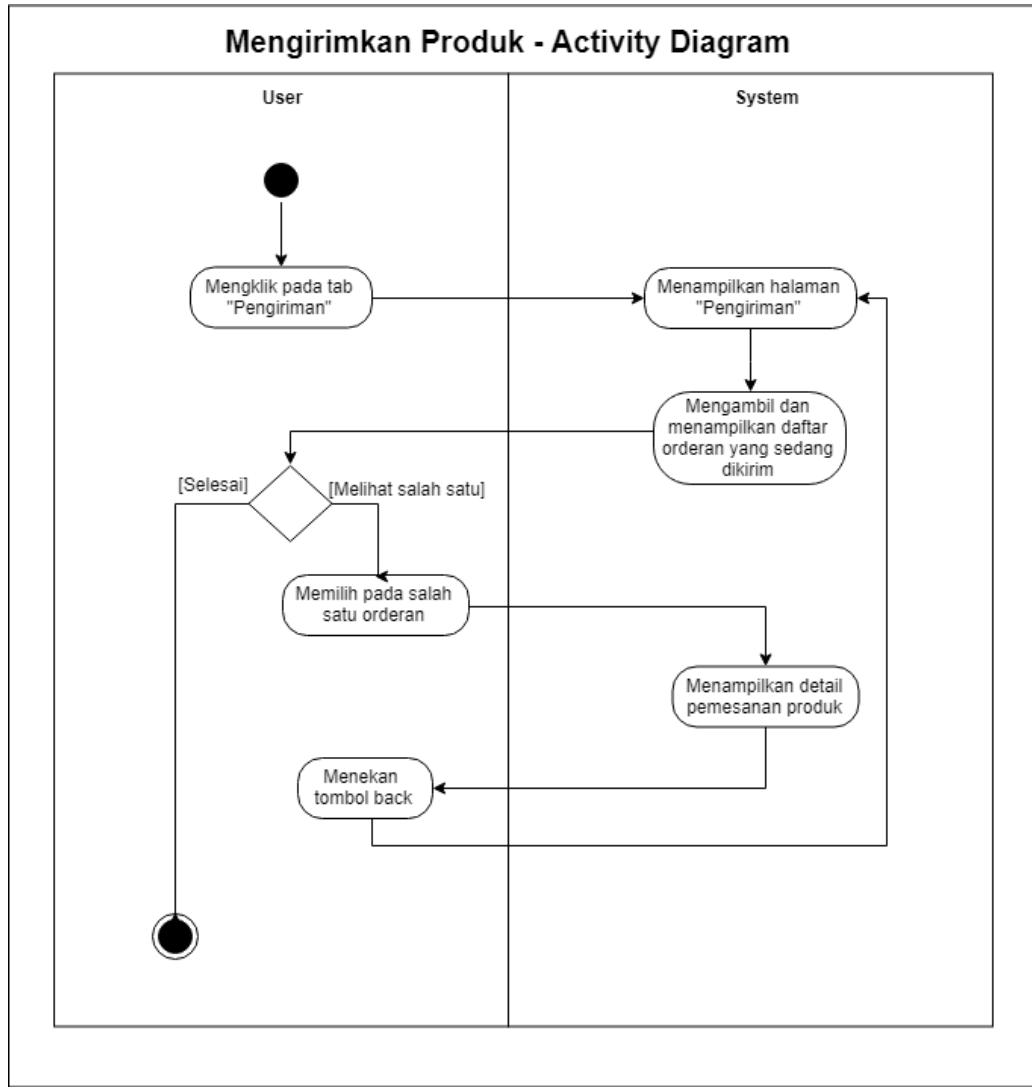
Gambar 4.20 Mengisi Nomor Resi *Activity Diagram*



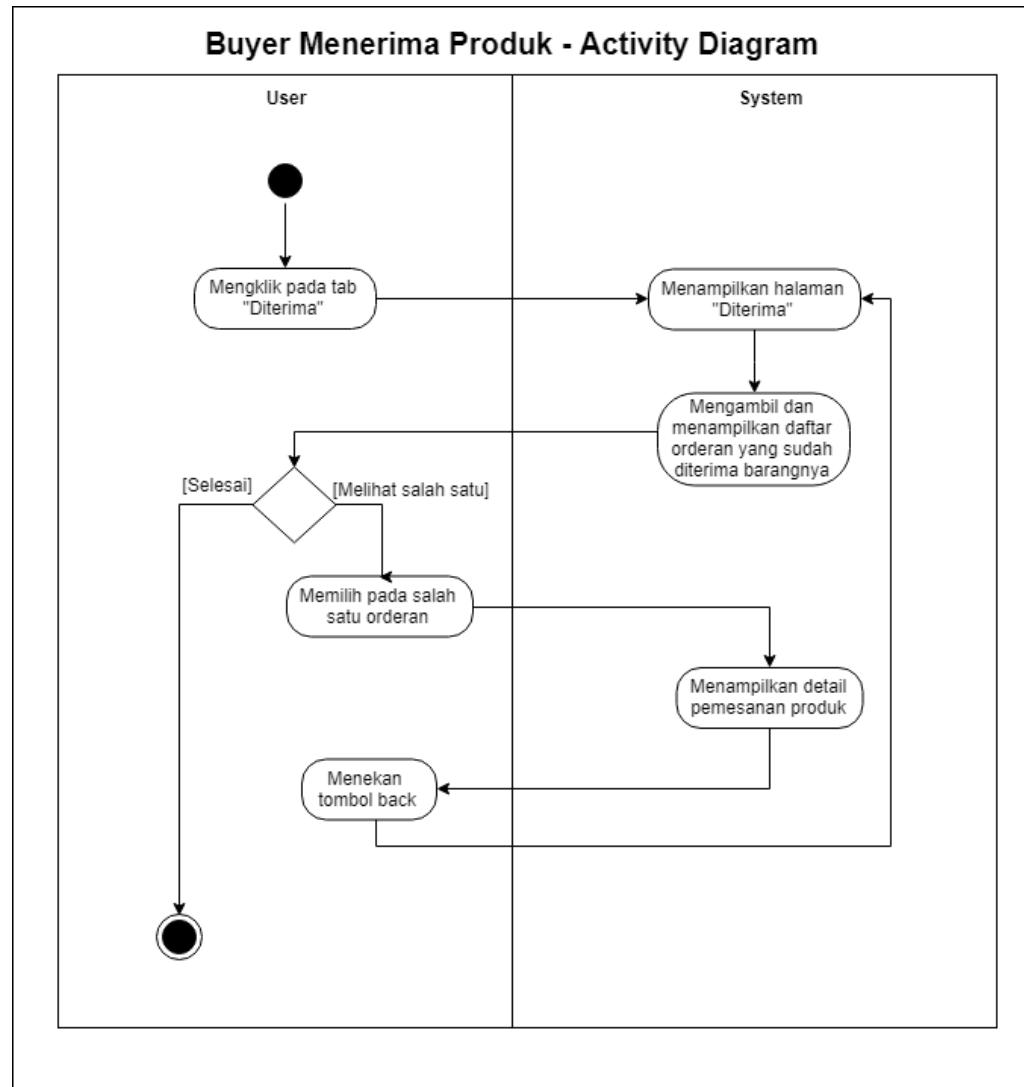
Gambar 4.21 Mengubah Status Menjadi Siap Di-pick-up Activity Diagram



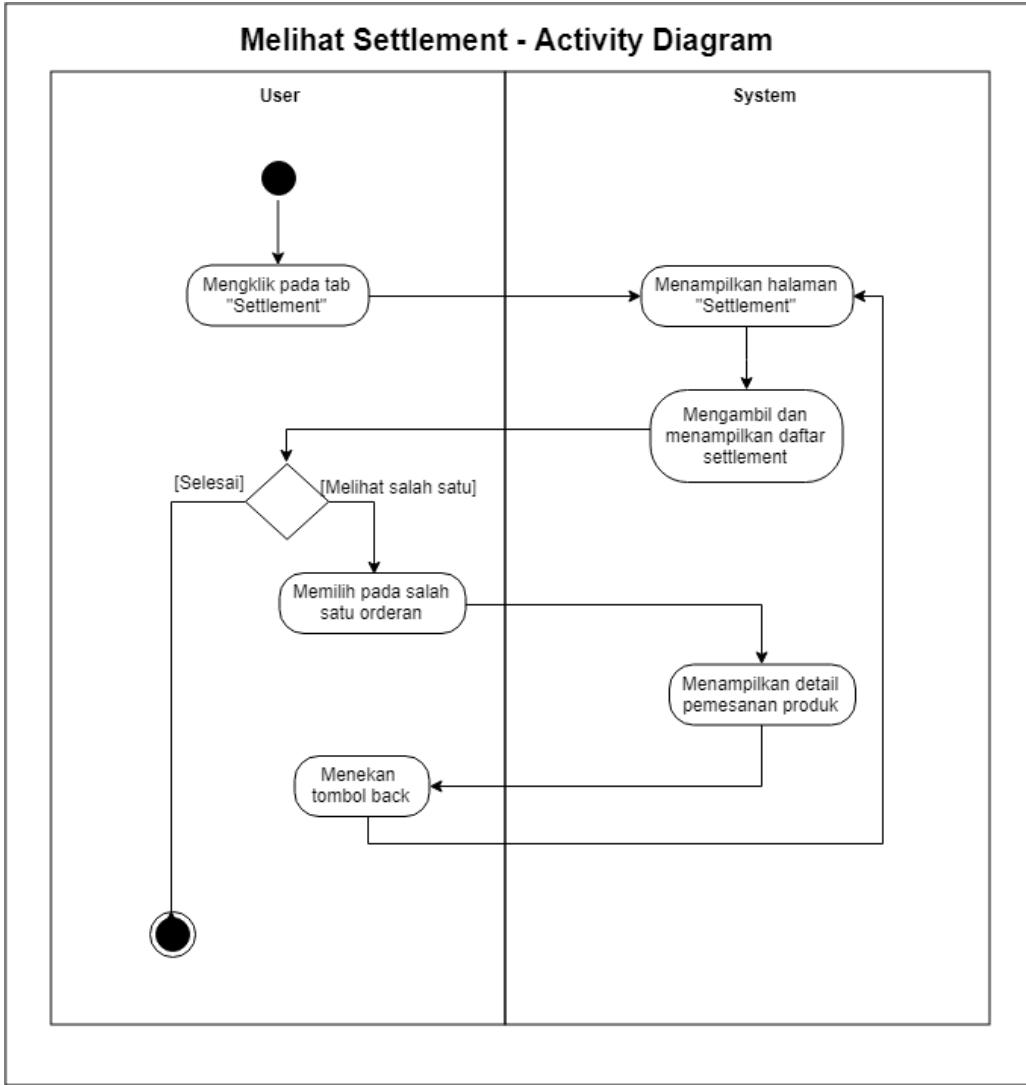
Gambar 4.22 Mengisi Nomor Resi Dari Ralali Kargo Activity Diagram



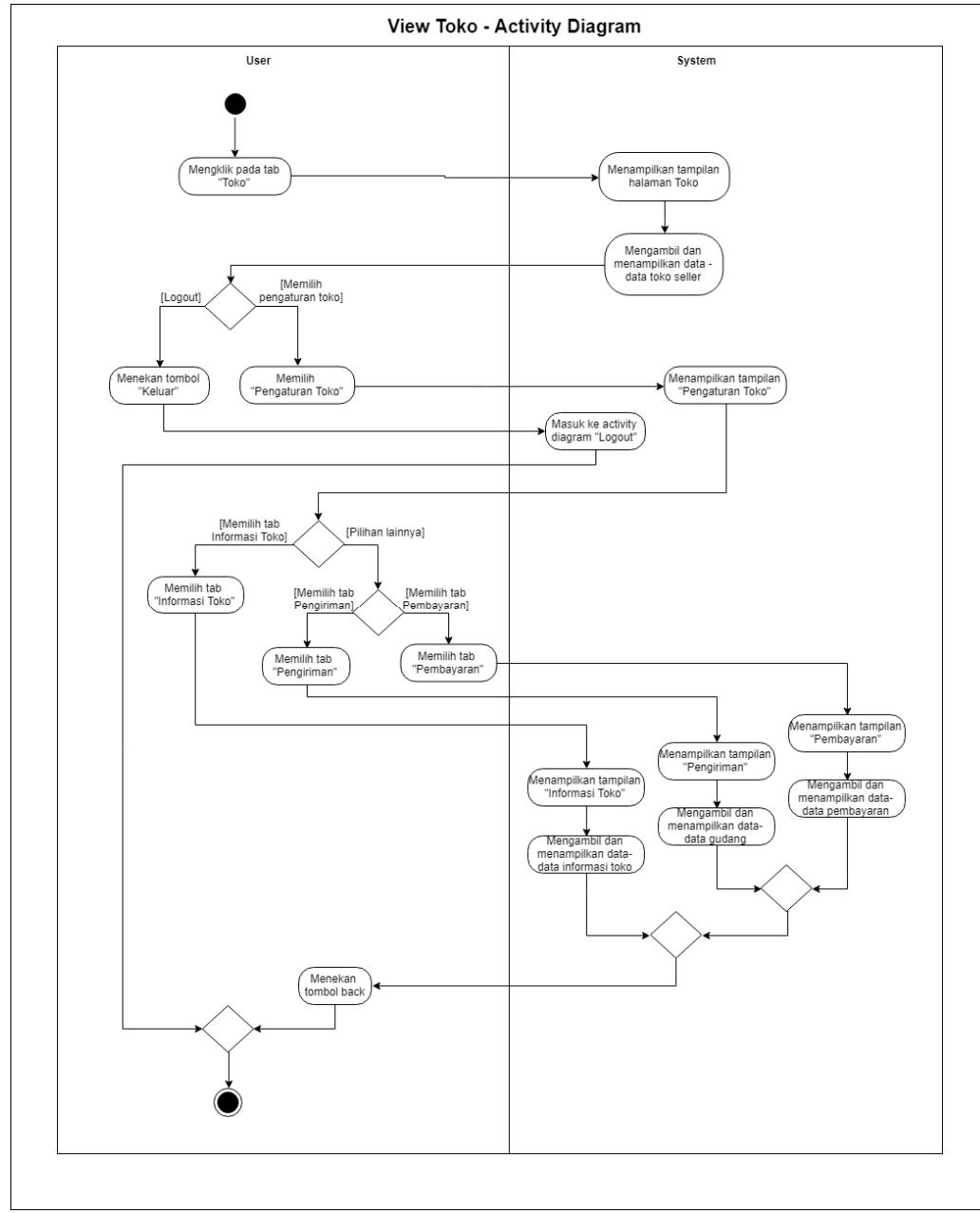
Gambar 4.23 Mengirimkan Produk *Activity Diagram*



Gambar 4.24 Buyer Menerima Produk Activity Diagram

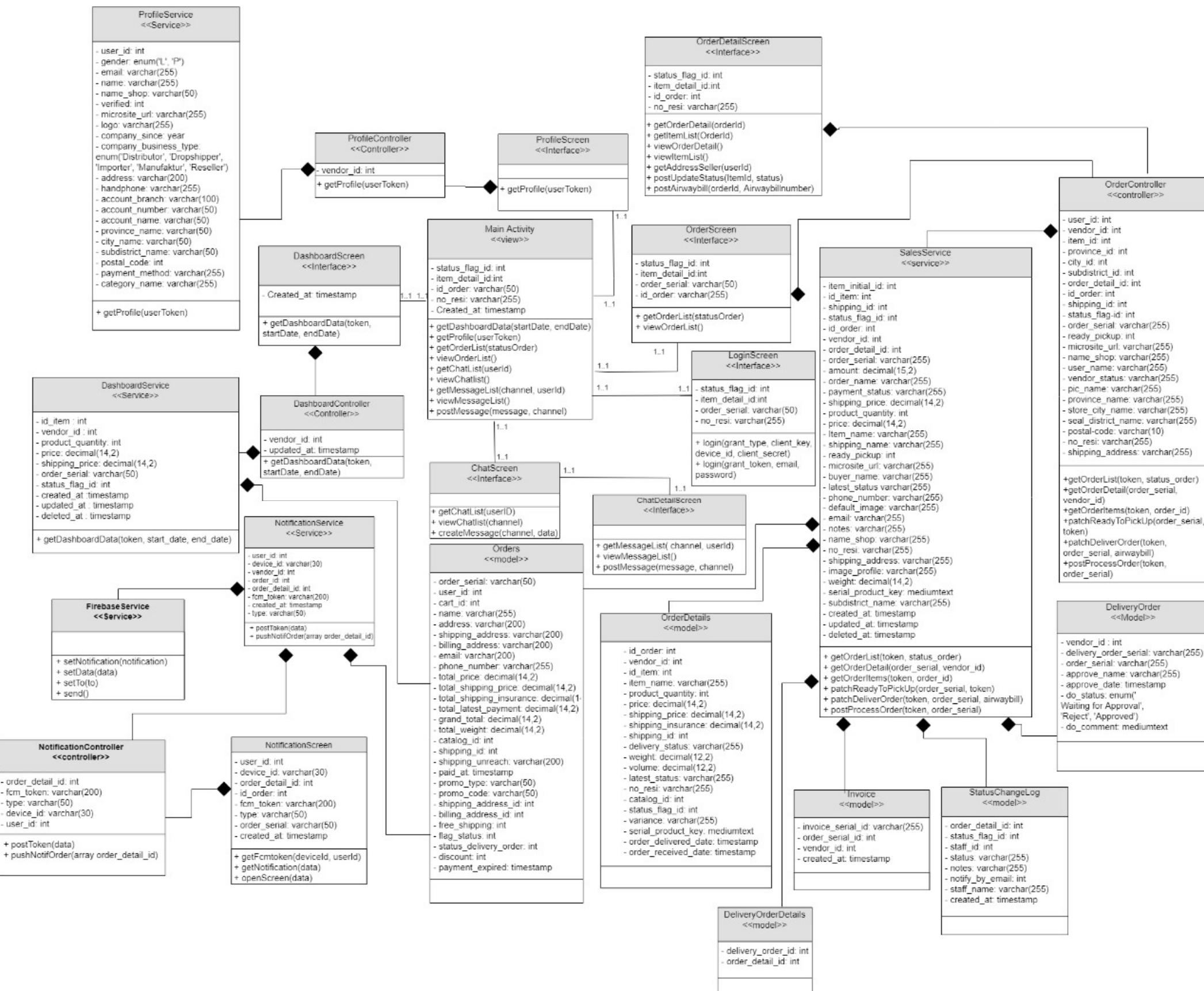


Gambar 4.25 Melihat Settlement Activity Diagram



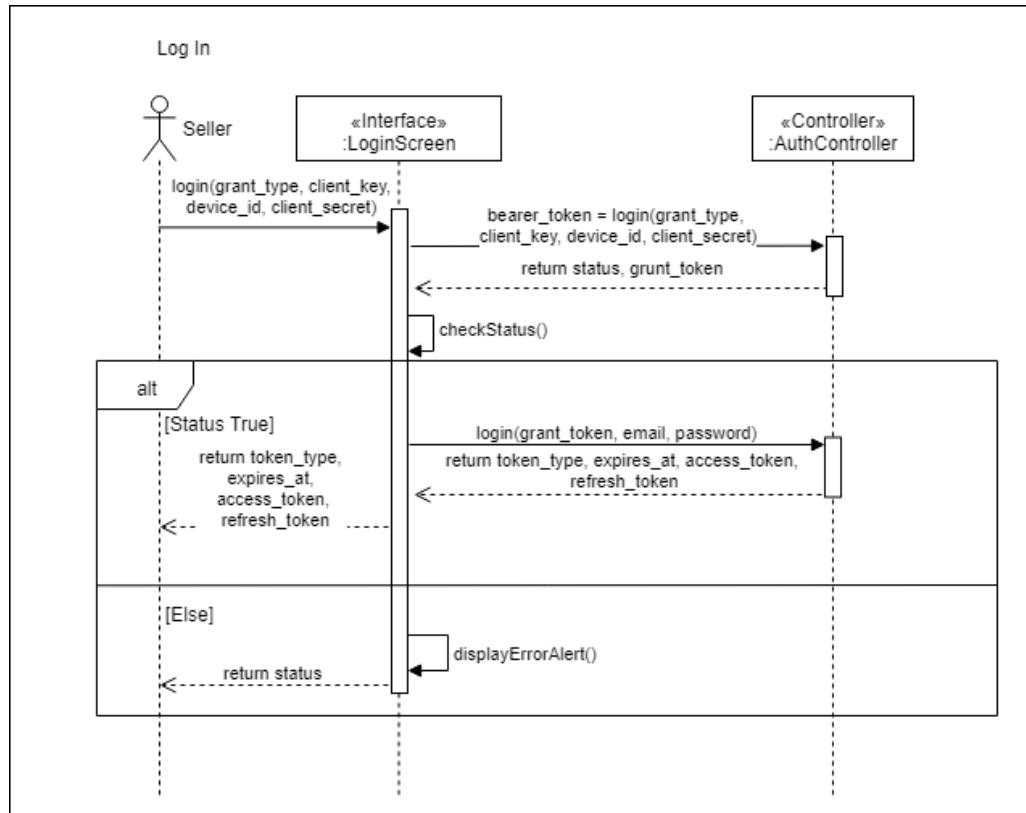
Gambar 4.26 Toko Activity Diagram

4.2.1.4 Class Diagram

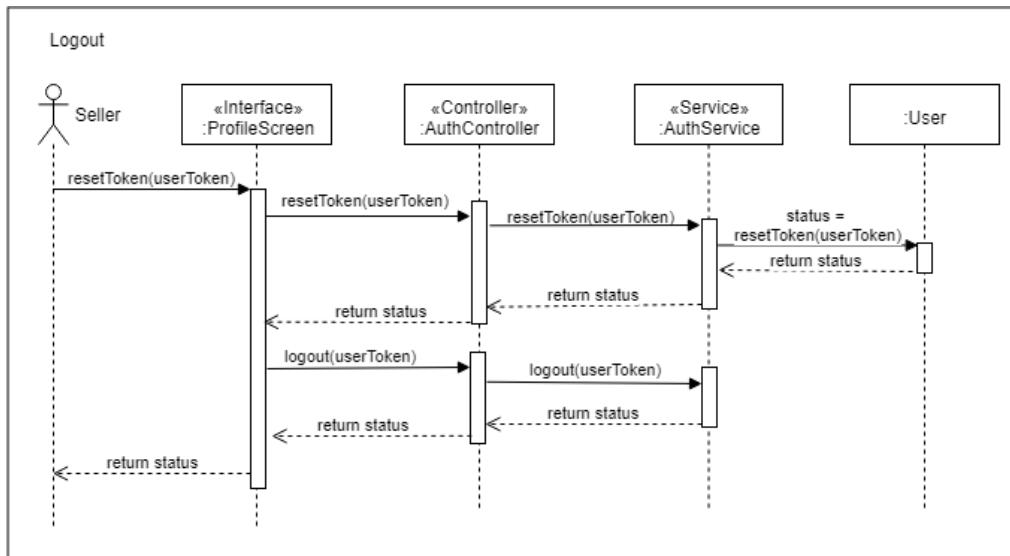


Gambar 4.27 Class Diagram Pada Ralali Seller

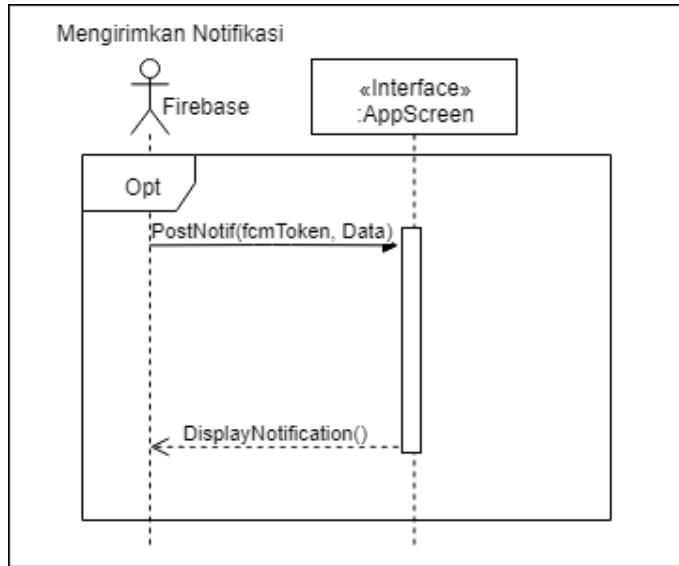
4.2.1.5 Sequence Diagram



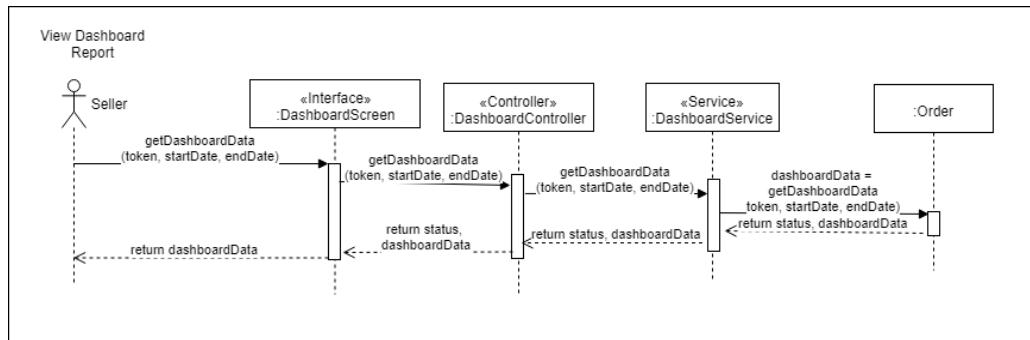
Gambar 4.28 Sequence Diagram Log In



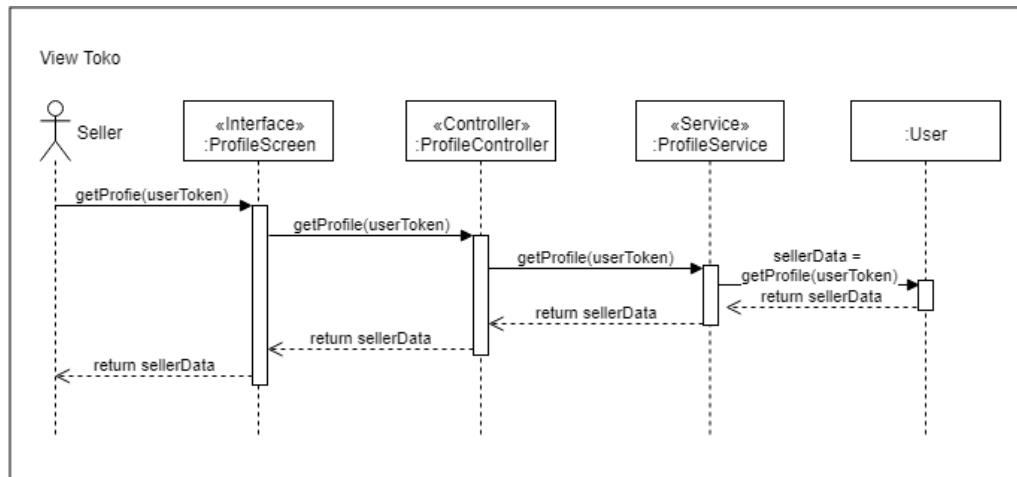
Gambar 4.29 Sequence Diagram Log Out



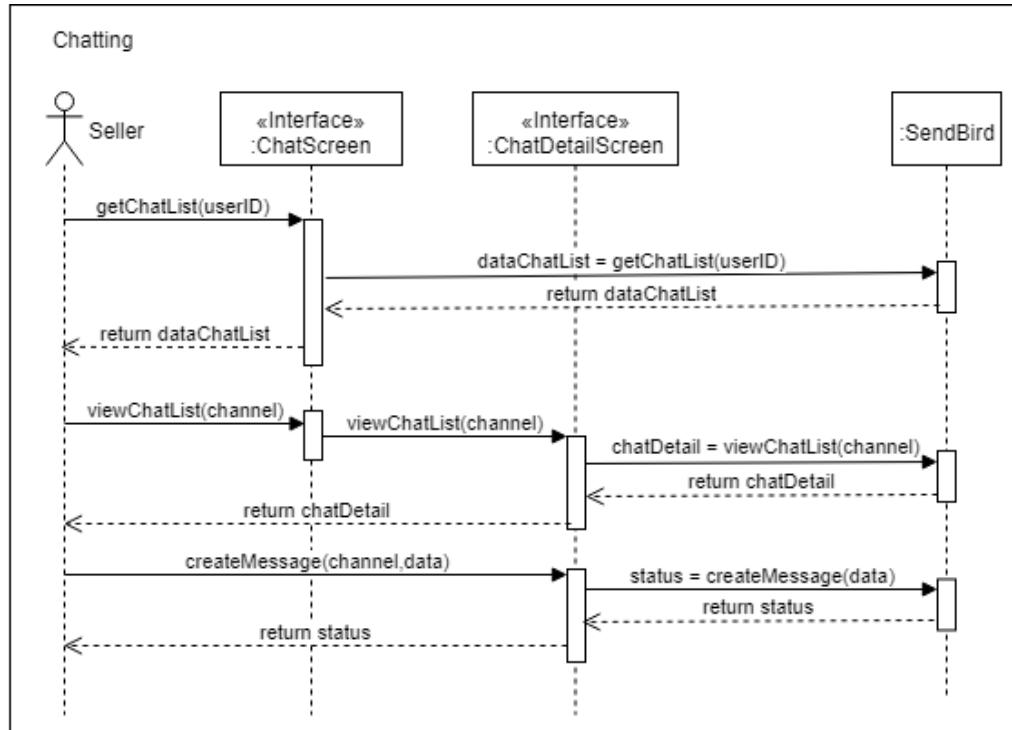
Gambar 4.30 Sequence Diagram Mengirimkan Notifikasi



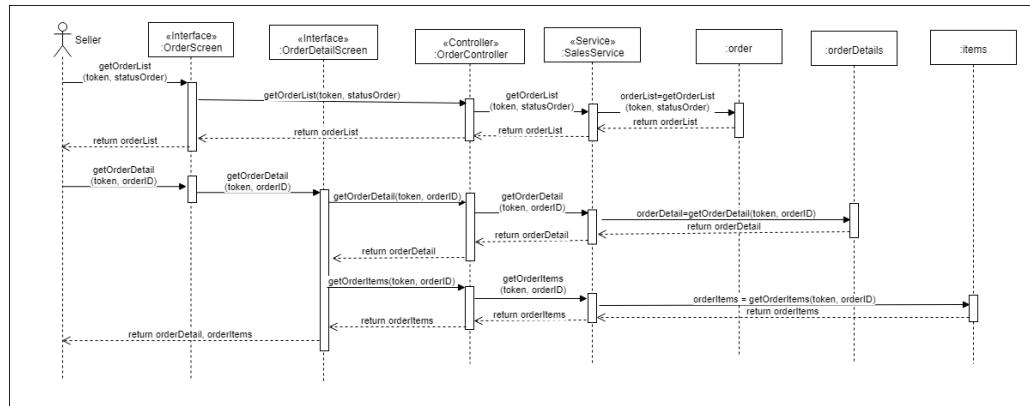
Gambar 4.31 Sequence Diagram View Dashboard Report



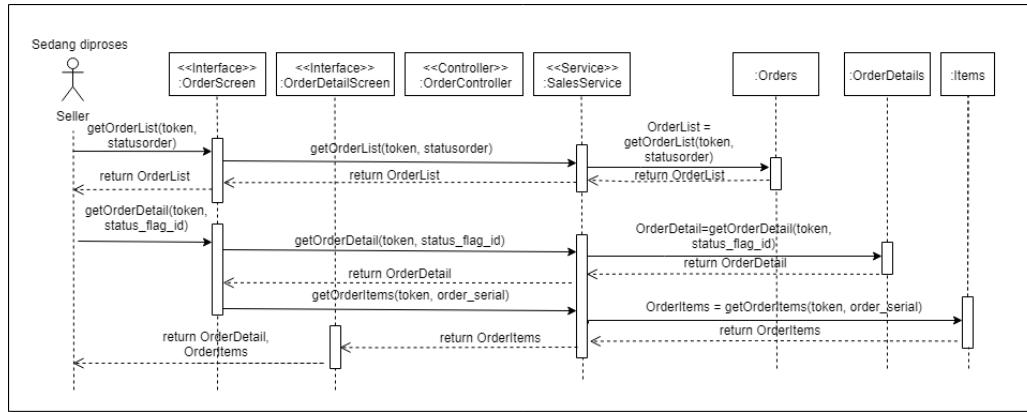
Gambar 4.32 Sequence Diagram View Profile



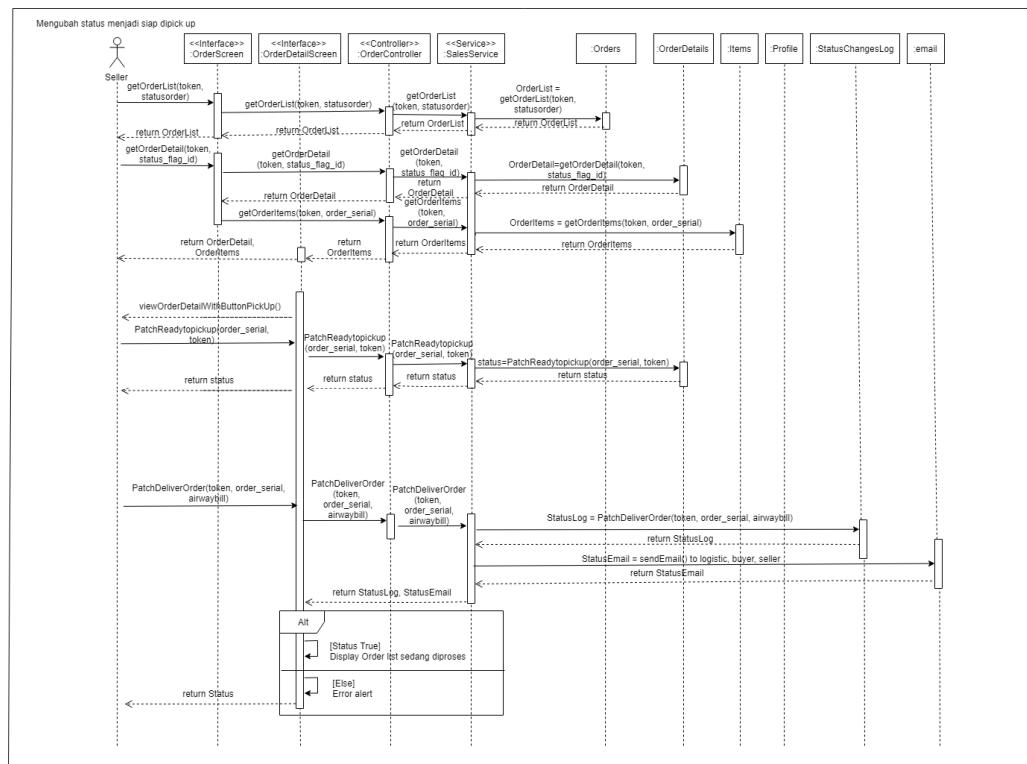
Gambar 4.33 Sequence Diagram Chatting



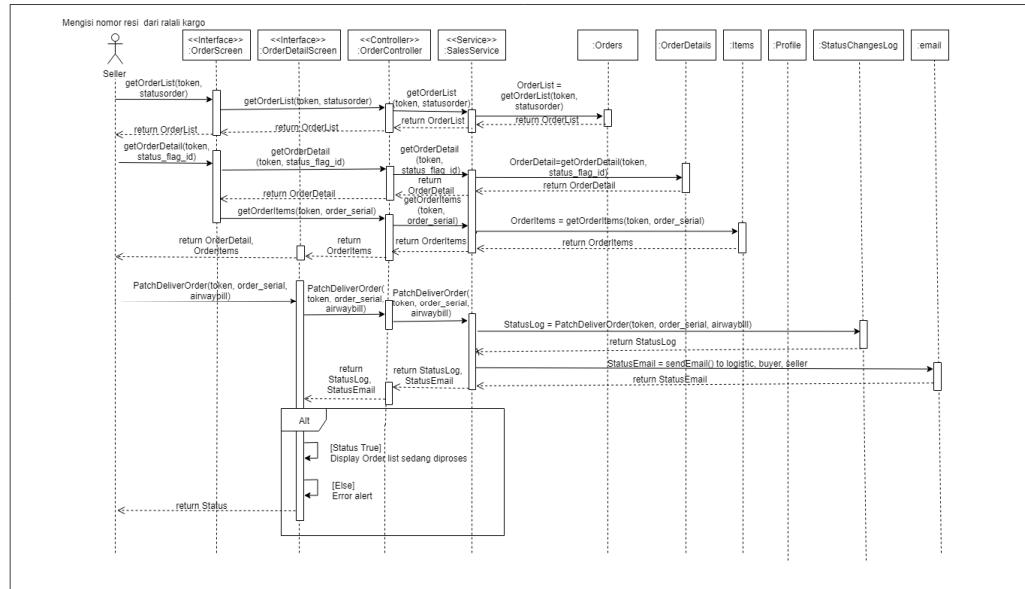
Gambar 4.34 Sequence Diagram Order



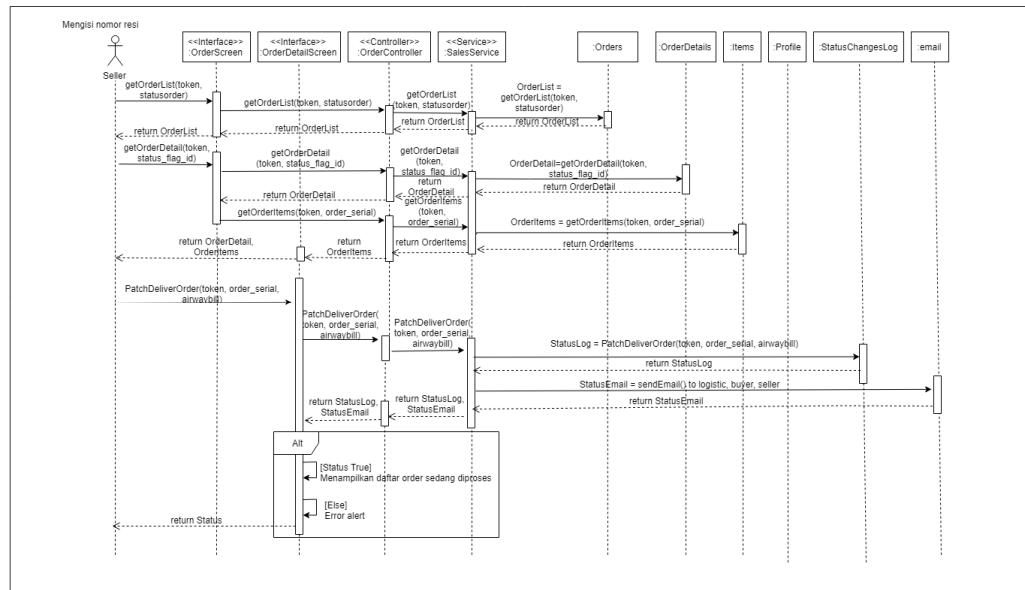
Gambar 4.35 Sequence Diagram Melihat Sedang Diproses



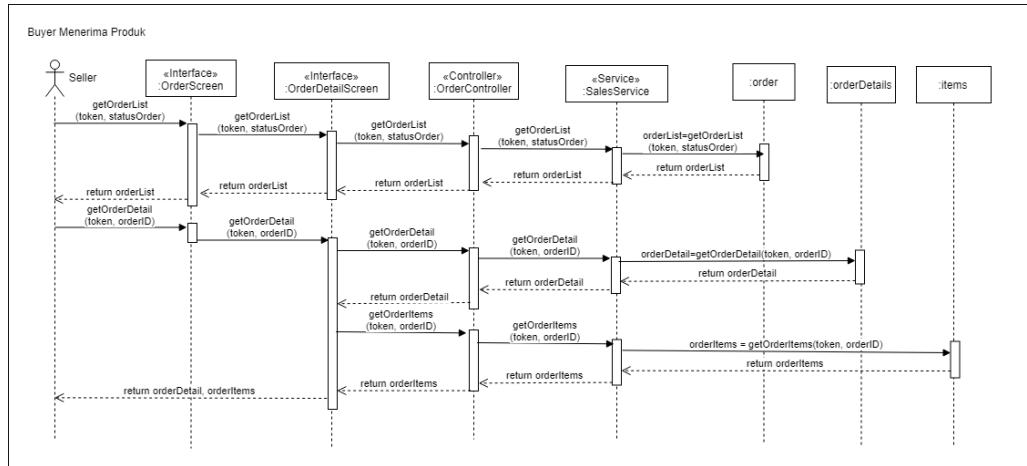
Gambar 4.36 Sequence Diagram Mengubah Status Menjadi Siap Di-pick-up



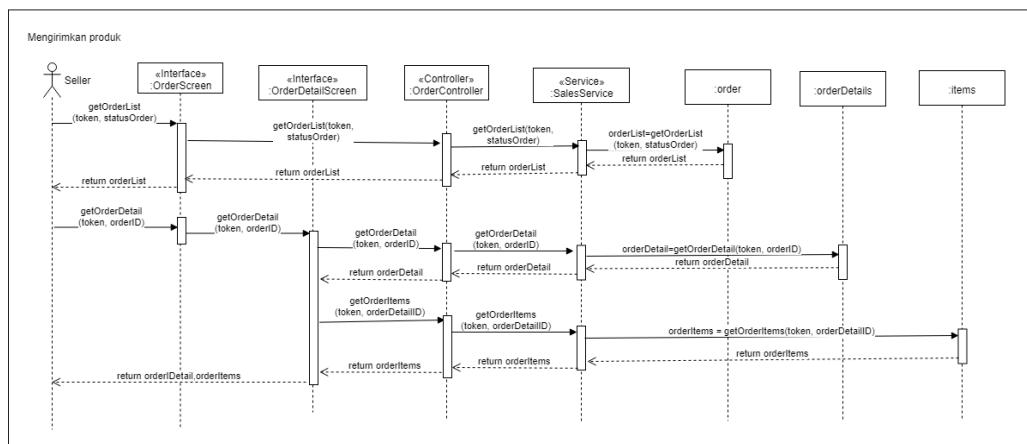
Gambar 4.37 Sequence Diagram Mengisi Nomor Resi Dari Ralali Kargo



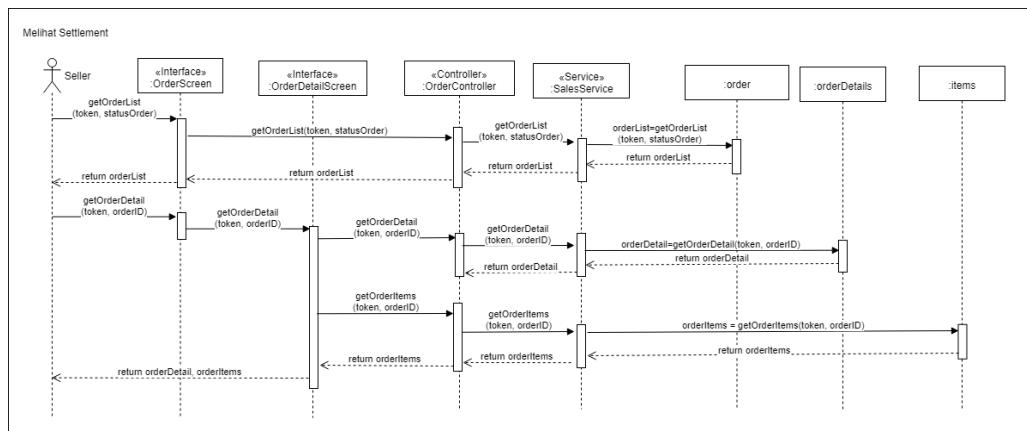
Gambar 4.38 Sequence Diagram Mengisi Nomor Resi



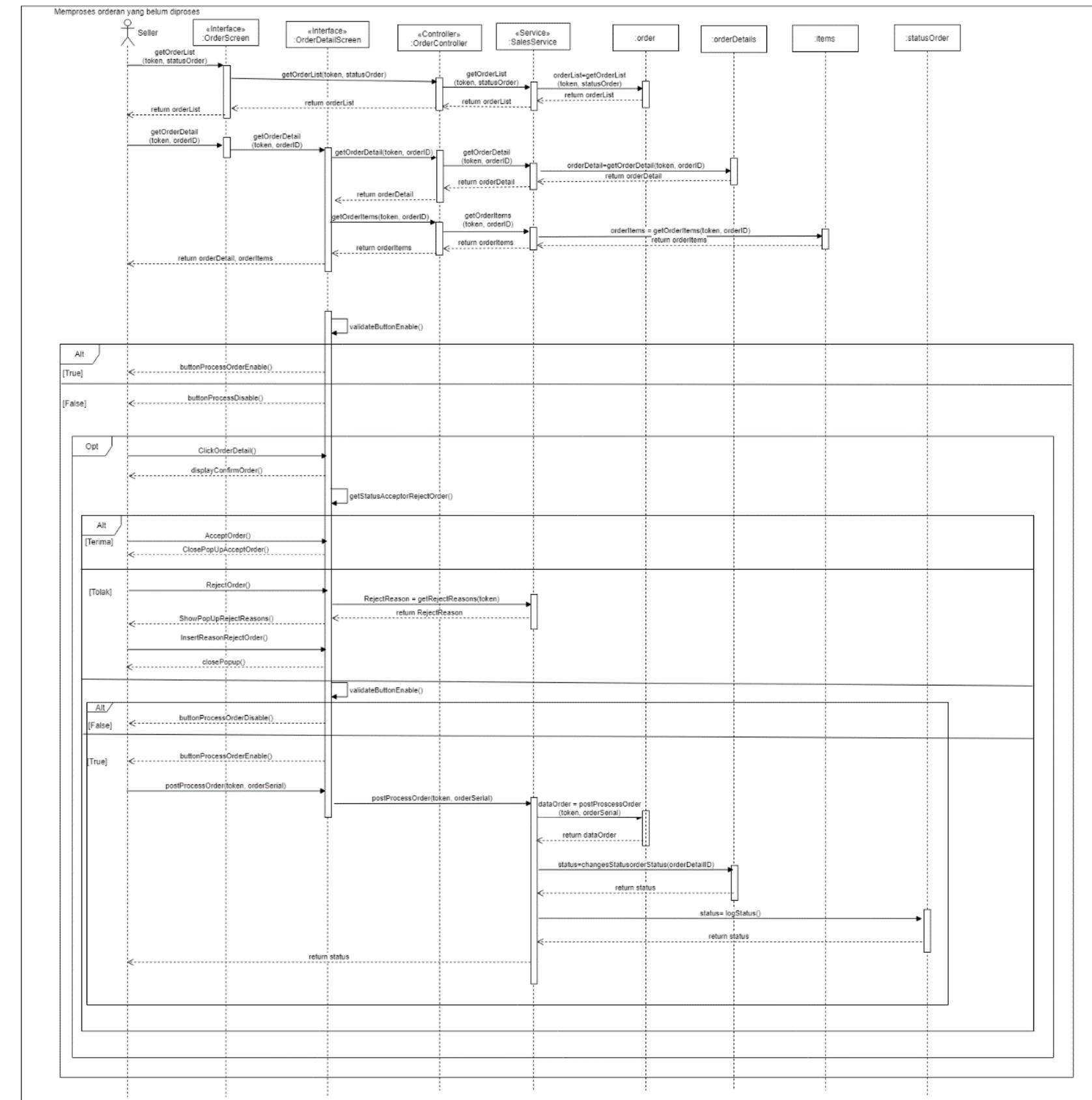
Gambar 4.39 Sequence Diagram Order sedang diterima



Gambar 4.40 Sequence Diagram Order pengiriman

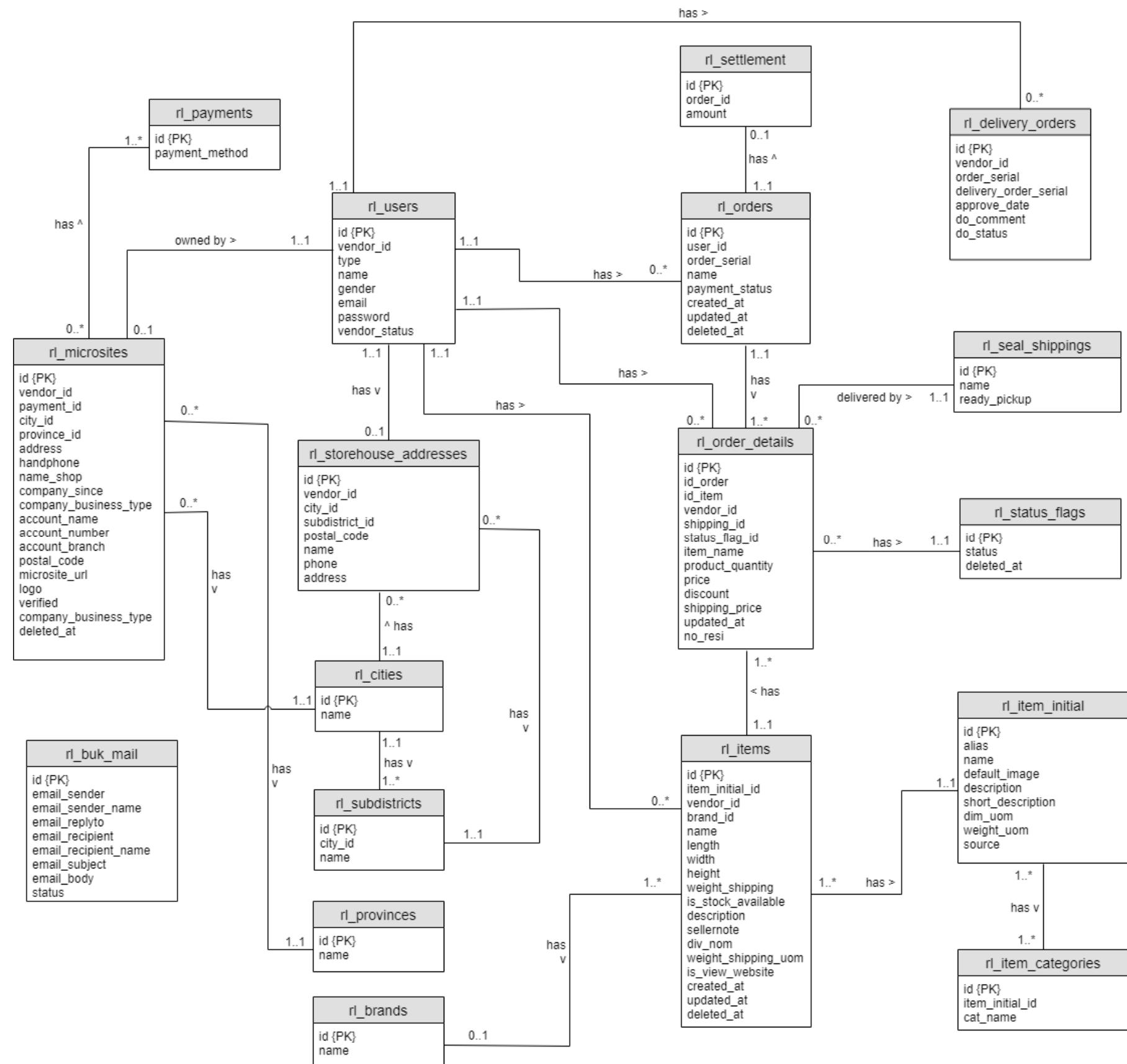


Gambar 4.41 Sequence Diagram Order settlement



Gambar 4.42 Sequence Diagram Memproses Orderan Yang Belum Diproses

4.2.2 ERD (Entity Relationship Diagram)



Gambar 4.43 ERD Ralali Seller

4.2.3 Kamus Data

1. Tabel rl_users

Primary Key : id

Foreign Key : -

Jumlah Field : 8

Tabel 4.23 Kamus Data *rl_users*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi ID nya <i>user</i>
2	vendor_id	int	Berisi ID yang dimiliki oleh <i>user</i> yang merupakan tipe <i>vendor</i>
3	type	varchar(2)	Tipe <i>user</i> apakah seorang <i>customer</i> atau <i>vendor</i>
4	name	varchar(255)	Berisi nama sang <i>user</i>
5	gender	enum('L','P')	Berisi jenis kelamin nya <i>user</i>
6	email	varchar(255)	Berisi <i>Email</i> nya <i>user</i>
7	password	varchar(60)	Berisi <i>Password</i> nya <i>user</i>
8	vendor_status	varchar(2)	Berisi mengenai status <i>user</i> (Reject merupakan status vendor ditolak, Approved merupakan status vendor diterima, Delete merupakan status vendor yang sudah dihapus, Not Yet merupakan status vendor yang belum mengisi data lengkap, Profile undefined, withdraw merupakan vendor yang melakukan pelanggaran, registered merupakan status vendor yang telah diterima oleh Ralali sebagai vendor di Ralali, banned

			merupakan vendor yang mengupload banner, logo, company profile yang belum sesuai dengan kriteria banner vendor Ralali, unverified merupakan vendor yang belum melakukan verifikasi email)
--	--	--	---

2. Tabel rl_storehouse_addresses

Primary Key : id

Foreign Key : -

Jumlah Field : 8

Tabel 4.24 Kamus Data *rl_storehouse_address*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi ID gudang yang dimiliki vendor Ralali
2	vendor_id	int	Berisi ID nya penjual Ralali
3	city_id	int	Berisi ID nya kota gudang
4	subdistrict_id	int	Berisi ID nya kecamatan gudang
5	postal_code	varchar(255)	Berisi kode pos gudang
6	name	varchar(255)	Berisi nama pic gudang
7	phone	varchar(255)	Berisi nomor handphone pic gudang
8	address	varchar(255)	Berisi alamat gudang

3. Tabel *rl_cities*

Primary Key : id

Foreign Key : -

Jumlah Field : 2

Tabel 4.25 Kamus Data *rl_cities*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi IDnya kota
2	name	varchar(255)	Berisi nama kota

4. Tabel *rl_subdistricts*

Primary Key : id

Foreign Key : -

Jumlah Field : 3

Tabel 4.26 Kamus Data *rl_subdistricts*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi IDnya kecamatan
2	city_id	int	Berisi IDnya kota
3	name	varchar(255)	Berisi nama kecamatan

5. Tabel rl_provinces

Primary Key : id

Foreign Key : -

Jumlah Field : 2

Tabel 4.27 Kamus Data *rl_provinces*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi IDnya provinsi
2	name	varchar(255)	Berisi nama provinsi

6. Tabel rl_brands

Primary Key : id

Foreign Key : -

Jumlah Field : 2

Tabel 4.28 Kamus Data *rl_brands*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi IDnya merek
2	name	varchar(255)	Berisi nama merek

7. Tabel *rl_microsites*

Primary Key : id

Foreign Key : -

Jumlah Field : 19

Tabel 4.29 Kamus Data *rl_microsites*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi IDnya toko
2	vendor_id	bigint	Berisi IDnya vendor Ralali
3	payment_id	int	Berisi IDnya pembayaran Ralali
4	city_id	int	Berisi IDnya kota
5	province_id	int	Berisi IDnya provinsi
6	address	varchar(200)	Berisi alamat toko vendor
7	handphone	varchar(100)	Berisi nomor handphone toko
8	name_shop	varchar(50)	Berisi nama toko
9	company_since	year	Berisi tahun berdiri toko
10	company_business_type	enum('Distributor', 'Dropshipper', 'Importer', 'Manufaktur', 'Reseller')	Berisi tipe bisnis toko, boleh dipilih lebih dari satu
11	account_name	varchar(50)	Berisi nama pemilik rekening toko
12	account_number	varchar(50)	Berisi nomor rekening toko
13	account_branch	varchar(100)	Berisi nama cabang bank
14	postal_code	int	Berisi postal kode toko

15	microsite_url	varchar(255)	Berisi alamat url toko
16	logo	text	Berisi logo toko
17	verified	int	Berisi mengenai status toko apakah toko sudah terverifikasi atau belum
18	company_business_type	enum('Distributor','Drops hipper','Importer','Manufacturer','Reseller')	Berisi tipe bisnis toko, boleh dipilih lebih dari satu
19	deleted_at	timestamp	Berisi apakah akun toko masih aktif atau sudah dihapus

8. Tabel rl_payments

Primary Key : id

Foreign Key : -

Jumlah Field : 2

Tabel 4.30 Kamus Data *rl_payments*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi ID pembayaran
2	payment_method	varchar(50)	Berisi mengenai metode pembayaran

9. Tabel *rl_buk_mail*

Primary Key : id

Foreign Key : -

Jumlah Field : 9

Tabel 4.31 Kamus Data *rl_buk_mail*

No	Nama Field	Type/Lebar	Keterangan
1	id	bigint (auto increment)	Berisi ID email yang terkirim
2	email_sender	varchar(50)	Berisi alamat email pengirim(noreply@ralali.com)
3	email_sender_name	varchar(50)	Berisi nama pengirim email(Ralali.com)
4	email_replyto	varchar(50)	Berisi alamat email yang selalu menerima pesan(info@ralali.com)
5	email_recipient	varchar(50)	Berisi alamat email penerima email
6	email_recipient_name	varchar(50)	Berisi nama penerima email
7	email_subject	varchar(255)	Berisi mengenai subjek email yang dikirim oleh Ralali
8	email_body	text	Berisi form email yang dikirim ke penerima email
9	status	int	Berisi mengenai status apakah email tersebut telah dikirim atau belum

10. Tabel rl_settlement

Primary Key : id

Foreign Key : -

Jumlah Field : 3

Tabel 4.32 Kamus Data *rl_settlement*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	Berisi ID nya settlement
2	order_id	int	Berisi ID nya order
3	amount	decimal(15,2)	Berisi total pendapatan vendor selama belanja di Ralali

11. Tabel rl_orders

Primary Key : id

Foreign Key : -

Jumlah Field : 8

Tabel 4.33 Kamus Data *rl_orders*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya order
2	user_id	int	ID nya buyer
3	order_serial	varchar(50)	Order serial order
4	name	varchar(50)	Nama buyer
5	payment_status	enum('Waiting for Payment','Paid','Reject','Top Paid')	Waiting for payment merupakan status transaksi order dimana Ralali

			menunggu pembayaran dari pembeli, Paid merupakan status transaksi order dimana pembeli telah membayar lunas orderannya, Reject merupakan status transaksi orderan yang ditolak oleh vendor Ralali. Top paid merupakan status transaksi order dimana orderan tersebut dibayar belakangan oleh pembeli.
6	created_at	timestamp	Waktu ketika buyer melakukan transaksi orderan
7	updated_at	timestamp	Waktu ketika staff Ralali melakukan konfirmasi pembayaran terhadap status pembayaran yang dilakukan oleh buyer Ralali
8	deleted_at	timestamp	Waktu ketika transaksi orderan direject oleh penjual

12. Tabel rl_order_details

Primary Key : id

Foreign Key : -

Jumlah Field : 13

Tabel 4.34 Kamus Data *rl_order_details*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya order detail
2	id_order	int	ID nya order
3	id_item	int	ID nya item
4	vendor_id	int	ID nya vendor
5	shipping_id	int	ID nya logistik Ralali
6	status_flag_id	int	ID nya status orderan
7	item_name	varchar(255)	Nama item
8	product_quantity	int	Jumlah produk yang tersedia
9	price	decimal(14,2)	Total harga item
10	discount	decimal(12,2)	Diskon terhadap penjualan item
11	shipping_price	decimal(12,2)	Harga ongkos
12	updated_at	timestamp	Waktu ketika vendor Ralali menerima atau menolak orderan item

Primary Key : id

Foreign Key : -

Jumlah Field : 18

Tabel 4.35 Kamus Data *rl_items*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya produk
2	item_initial_id	bigint	ID nya item initial
3	vendor_id	int	ID nya vendor
4	brand_id	int	ID nya brand
5	name	varchar(255)	Nama produk
6	length	varchar(255)	Panjang dimensi produk
7	width	varchar(255)	Lebar dimensi produk
8	height	varchar(255)	Tinggi dimensi produk
9	weight_shipping	decimal(14,2)	Berat produk
10	is_stock_available	enum('0','1')	Status produk tersedia atau tidak
11	description	text	Deskripsi produk
12	sellernote	text	Catatan penjual mengenai produk
13	dim_uom	varchar(50)	Satuan panjang, tinggi, lebar produk
14	weight_shipping_uom	varchar(50)	Satuan berat produk
15	is_view_website	enum('Y', 'N')	Status produk untuk ditampilkan atau tidak
16	created_at	timestamp	Waktu ketika produk diinsert

17	updated_at	timestamp	Waktu ketika produk diupdate
18	deleted_at	timestamp	Waktu ketika produk dihapus

14. Tabel *rl_delivery_orders*

Primary Key : id

Foreign Key : -

Jumlah Field : 7

Tabel 4.36 Kamus Data *rl_delivery_orders*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya order dikirim
2	vendor_id	int	ID nya vendor
3	order_serial	varchar(255)	Order serial transaksi order
4	delivery_order_serial	varchar(255)	Order serial pengiriman orderan
5	approve_date	timestamp	Tanggal penerimaan produk yang dikirim
6	do_comment	mediumtext	Catatan untuk produk yang dikirim
7	do_status	enum('Waiting for Approval', 'Reject', 'Approved')	Status pengiriman barang

15. Tabel *rl_seal_shippings*

Primary Key : id

Foreign Key : -

Jumlah Field : 3

Tabel 4.37 Kamus Data *rl_seal_shippings*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID jenis layanan pengiriman yang dipakai
2	name	varchar(255)	Nama logistik
3	ready_pickup	tinyint(1)	Status pengiriman barang apakah sudah dapat diambil oleh pihak logistik atau tidak

16. Tabel *rl_status_flags*

Primary Key : id

Foreign Key : -

Jumlah Field : 3

Tabel 4.38 Kamus Data *rl_status_flags*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID status flag
2	status	varchar(255)	Status orderan
3	deleted_at	timestamp	Waktu penghapusan status_flag

17. Tabel *rl_item_initial*

Primary Key : id

Foreign Key : -

Jumlah Field : 9

Tabel 4.39 Kamus Data *rl_item_initial*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya item initial
2	alias	varchar(255)	Nama alias produk
3	name	varchar(255)	Nama item initial
4	default_image	mediumtext	Gambar untuk item initial
5	description	mediumtext	Deskripsi item initial
6	short_description	text	Deskripsi singkat untuk item initial
7	dim_uom	varchar(50)	Satuan panjang, lebar, tinggi untuk item initial
8	weight_uom	varchar(50)	Satuan berat untuk item initial
9	source	enum('sp','pim')	Input item dari seller panel atau PIM

18. Tabel *rl_item_categories*

Primary Key : id

Foreign Key : -

Jumlah Field : 3

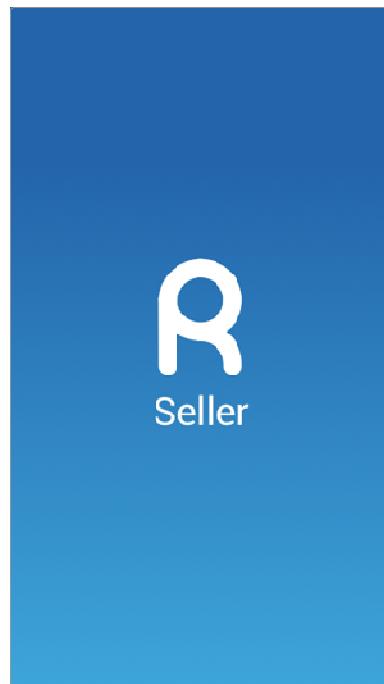
Tabel 4.40 Kamus Data *rl_item_categories*

No	Nama Field	Type/Lebar	Keterangan
1	id	int (auto increment)	ID nya item kategori
2	item_initial_id	bigint	ID nya item initial
3	cat_name	varchar(100)	Nama kategori

4.3 Rancangan Layar

4.3.1 Storyboard

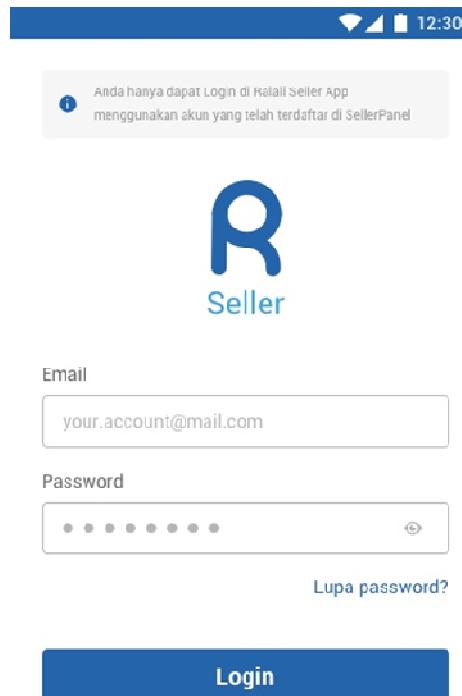
1. Ketika aplikasi Ralali *Seller* dibuka, maka akan muncul *Splash Screen* terlebih dahulu seperti pada gambar berikut ini :



Gambar 4.44 Splash Screen pada Ralali Seller

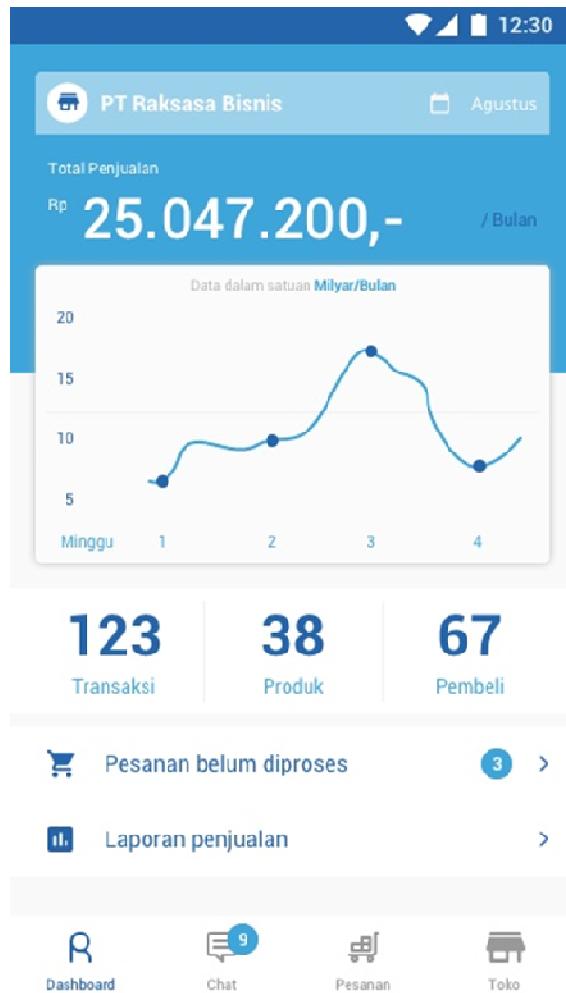
2. Setelah muncul *Splash Screen*, maka selanjutnya akan ditampilkan halaman *Login* pada aplikasi Ralali *Seller*. Jika *user* sudah

terdaftar sebagai *seller* di Ralali.com, maka *user* dapat melakukan aktivitas *login* di aplikasi Ralali *Seller* dengan cara menginput *e-mail* nya serta *password* lalu mengklik tombol “*Login*”. Sistem akan mengecek kevalid-an *email* dan *password* yang diinput dengan email dan password *user* yang terdaftar di *database* Ralali.com. Tampilan fitur *login* dapat dilihat pada gambar dibawah ini :



Gambar 4.45 Login pada Ralali Seller

3. Setelah *user* melakukan aktivitas *login*, maka sistem akan mengambil data dan menampilkan halaman *dashboard report* dimana fitur ini merupakan halaman utama dari aplikasi Ralali *seller*. Tampilan *dashboard report* dapat dilihat pada gambar dibawah ini:

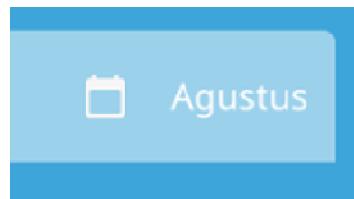


Gambar 4.46 Dashboard pada Ralali Seller

Pada halaman *Dashboard report*, terdapat banyak hal yang dapat dilakukan *user* sesuai dengan kebutuhannya. Diantaranya adalah:

- Memilih tahun dan bulan pada *dashboard report*

Memilih tahun dan bulan, hal ini bertujuan untuk melihat performa penjualan *seller* dalam bulan yang diatur *seller*. *User* dapat melihat performa dirinya selama berjualan di Ralali.com dengan memilih salah satu bulan yang ingin dilihat dengan cara klik pada icon *calendar*, seperti pada gambar di bawah ini :



Gambar 4.47 Icon Calendar

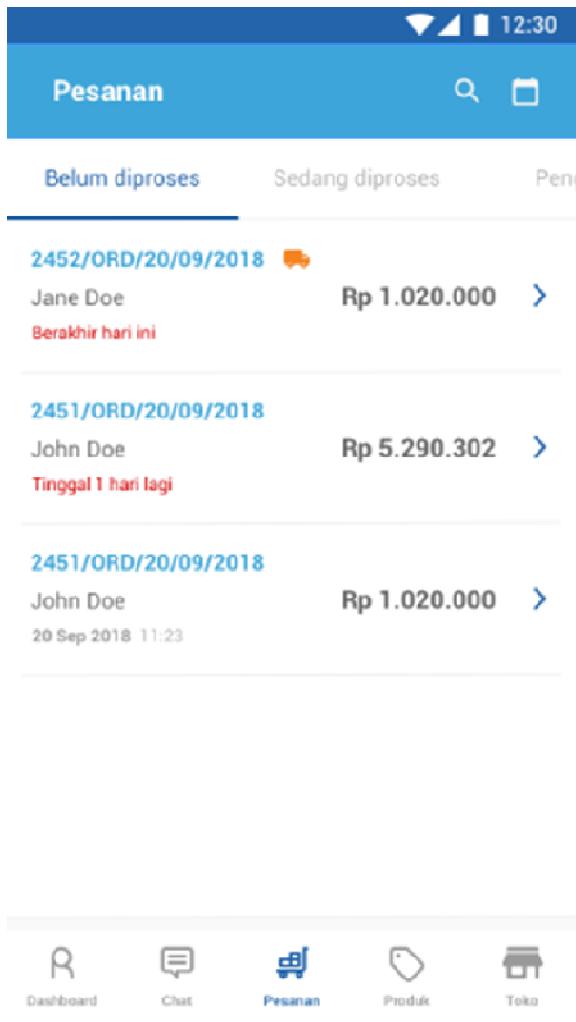
b. Pesanan belum diproses

Menampilkan orderan - orderan yang belum diproses oleh *seller*. *User* dapat menampilkan orderan yang belum diproses dengan cara melakukan aktivitas dengan mengklik pada pilihan “Pesanan belum diproses”, seperti pada gambar di bawah ini:



Gambar 4.47 Pesanan belum diproses pada dashboard

Setelah *user* memilih pilihan “Pesanan belum diproses”, maka akan ditampilkan halaman seperti sebagai berikut:



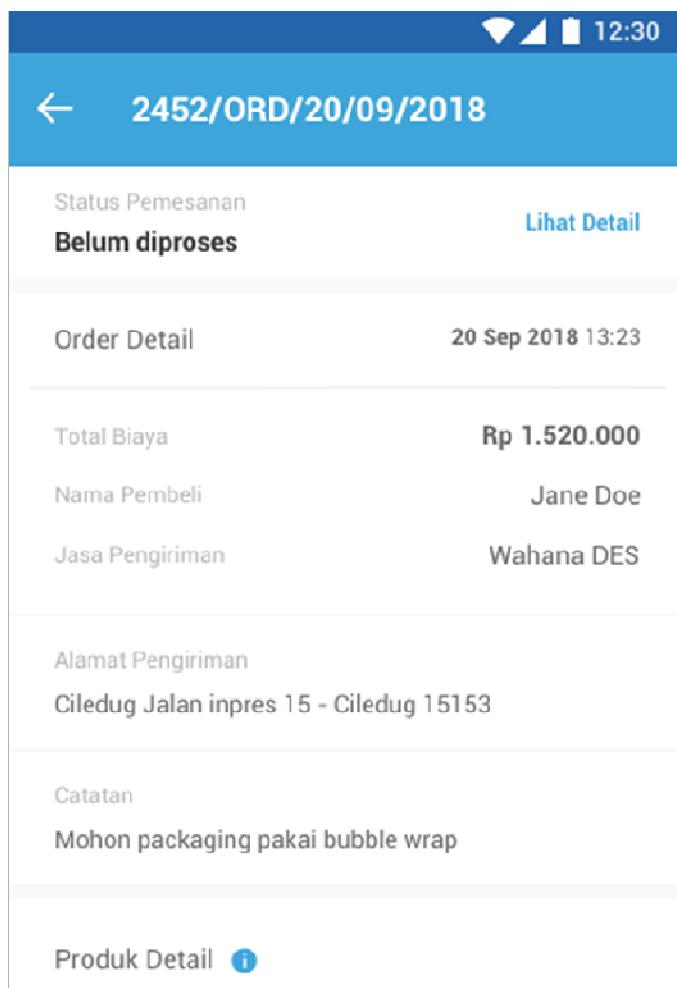
Gambar 4.49 Pesanan belum diproses

User dapat melihat daftar orderan yang belum diproses, nama pembeli, dan juga terdapat *due date* yang berfungsi agar *seller* dapat menangani orderan - orderan yang masuk sebelum orderan - orderan tersebut melewati batas waktu yang telah ditentukan oleh pihak Ralali.

Terdapat juga icon kargo *orange* yang menandakan bahwa *buyer* memilih jasa logistik Ralali kargo untuk mengirimkan barang yang diorder nya.

Pada gambar di atas, dapat diketahui bahwa begitu *user* mengklik pilihan “Pesanan belum diproses”, maka halaman yang dituju adalah sama ketika *user* memilih tab “Pesanan” yang terdapat di tengah paling bawah pada layar.

User dapat mengklik pada salah satu orderan yang ada didaftar “Pesanan belum diproses”, kemudian akan muncul tampilan seperti di bawah ini :



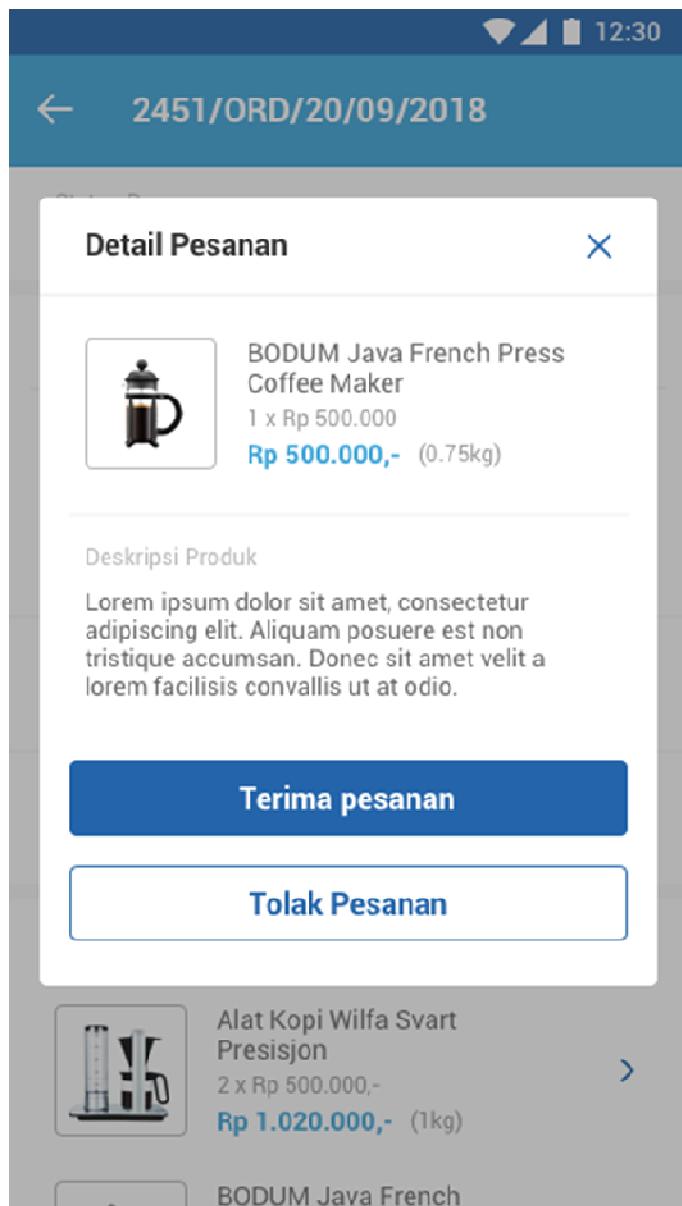
Subtotal	Rp 1.500.000
Ongkos kirim	Rp 20.000
Total pembayaran	Rp 1.520.000

Proses Pesanan

Gambar 4.50 Detail Pesanan belum diproses

Jika dalam satu orderan terdapat 2 *item* (atau lebih) yang merupakan *item* dari satu *seller* yang sama dan juga diorder oleh *buyer* yang sama, maka pada saat *seller* membuka orderan akan muncul 2 atau lebih *item*, dan *seller* harus memproses semua *item* yang terdapat pada orderan tersebut. Tombol “Proses Pesanan” akan aktif (bisa diklik) jika semua *item* yang terdapat pada orderan tersebut sudah diproses semua.

Ketika *seller* mengklik pada salah satu *item* yang terdapat dalam orderan tersebut maka akan muncul dua pilihan, yaitu menerima orderan atau menolak orderan. Kedua pilihan tersebut akan menampilkan tampilan yang berbeda.



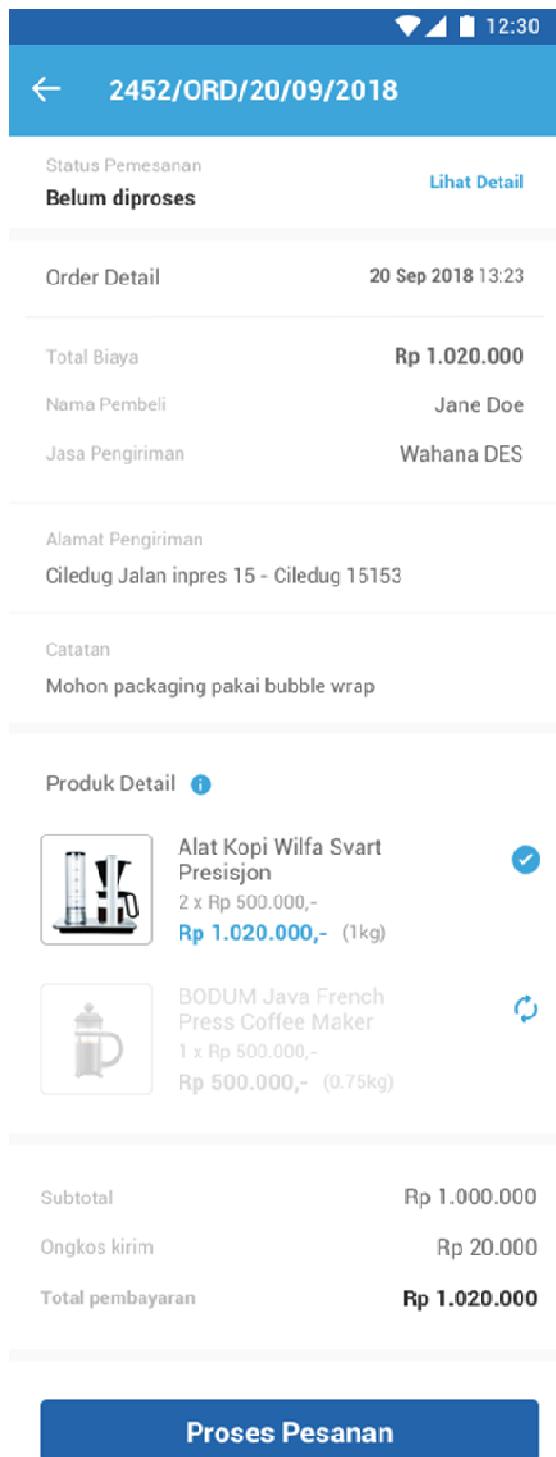
Gambar 4.51 Memproses Pesanan



Gambar 4.52 Menolak Pesanan

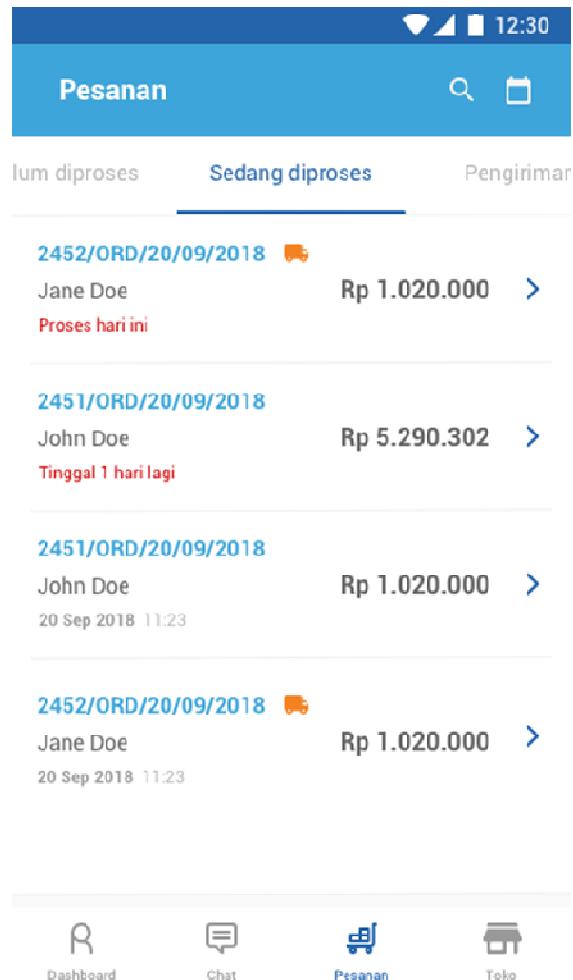
Terdapat pilihan kenapa *seller* melakukan proses penolakan pada orderan yang masuk tersebut, dan juga *seller* dapat menjelaskan lebih lanjut lagi kepada *buyer* dengan cara mengisi pada kolom keterangan yang telah disediakan.

Kemudian, jika *seller* telah memproses semua item yang terdapat dalam *order detail* tersebut, maka tombol “Proses Pesanan” akan di-enable untuk memproses pesanan seperti pada gambar di bawah ini:



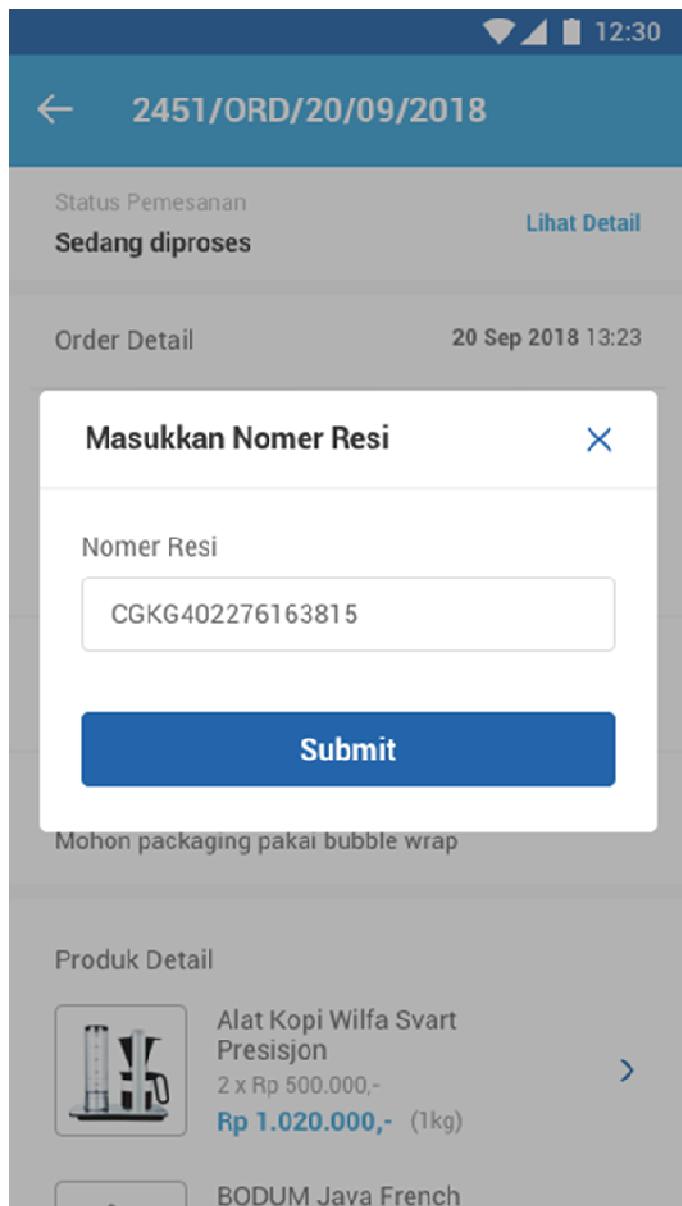
Gambar 4.53 Proses Pesanan

Setelah memproses suatu orderan, maka *seller* dapat berpindah ke tab sebelah yang bernama “Sedang Diproses”

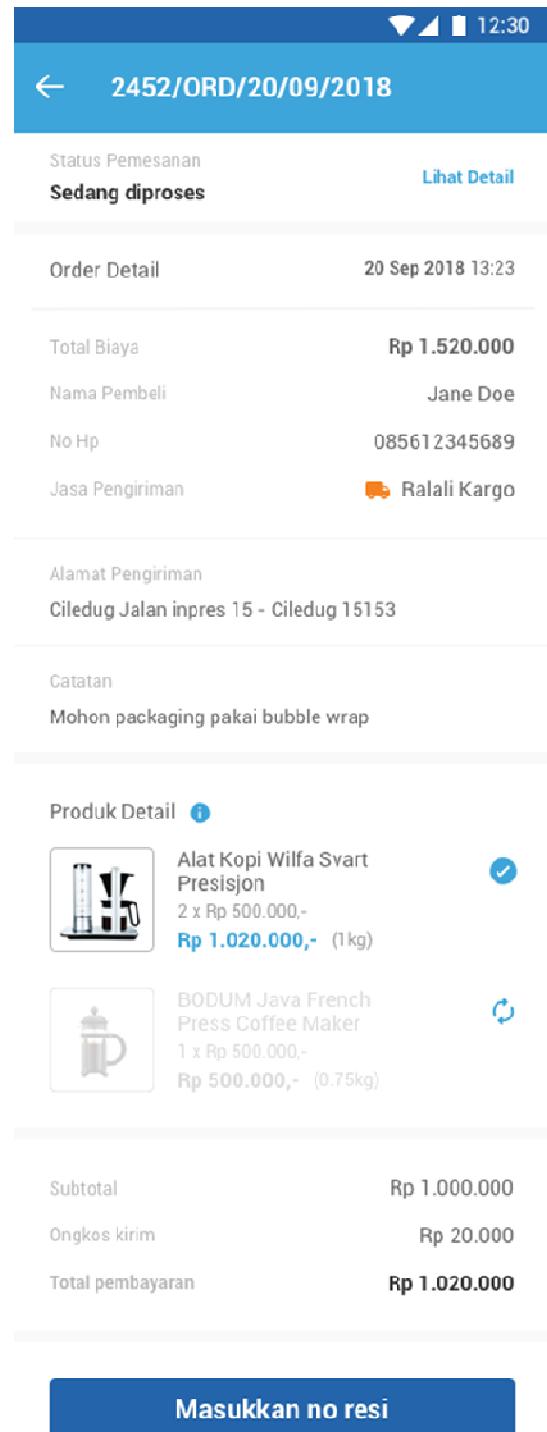


Gambar 4.54 Sedang Diproses

Kemudian *seller* dapat memilih salah satu order pada list, dan mengisi resi pada kolom resi seperti pada gambar di bawah ini (jika tidak terdapat icon Ralali kargo atau terdapat Ralali kargo tapi berwarna abu-abu) :



Gambar 4.55 Pengisian Nomor Resi - Tanpa Logo Ralali Kargo



Gambar 4.56 Pengisian Nomor Resi - Ralali Kargo Abu-Abu

Jika orderan yang kita pilih terdapat Ralali kargo yang berwarna *orange*, maka *seller* dapat melakukan konfirmasi

bahwa *item* yang dipesan oleh *buyer* sudah siap untuk dipickup oleh kargo nya Ralali, dan akan muncul pop-up alamat gudang *seller* yg butuh dikonfirmasi oleh *seller*. Dapat dilihat pada gambar di bawah ini :

← 2452/ORD/20/09/2018

Status Pemesanan **Sedang diproses** Lihat Detail

Order Detail 20 Sep 2018 13:23

Total Biaya	Rp 1.520.000
Nama Pembeli	Jane Doe
No Hp	085612345689
Jasa Pengiriman	Ralali Kargo

Alamat Pengiriman
Ciledug Jalan Inpres 15 - Ciledug 15153

Catatan
Mohon packaging pakai bubble wrap

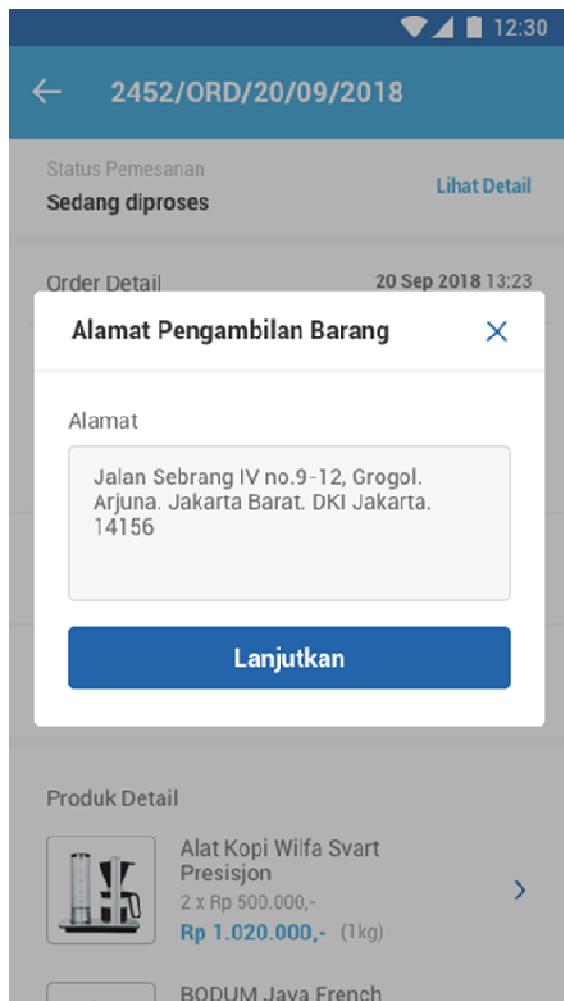
Produk Detail

Alat Kopi Wilfa Svart Presisjon	<input checked="" type="checkbox"/>
2 x Rp 500.000,-	
Rp 1.020.000,- (1kg)	
BODUM Java French Press Coffee Maker	
1 x Rp 500.000,-	
Rp 500.000,- (0.75kg)	

Subtotal	Rp 1.000.000
Ongkos kirim	Rp 20.000
Total pembayaran	Rp 1.020.000

Siap di pick-up

Gambar 4.57 Siap Dipickup



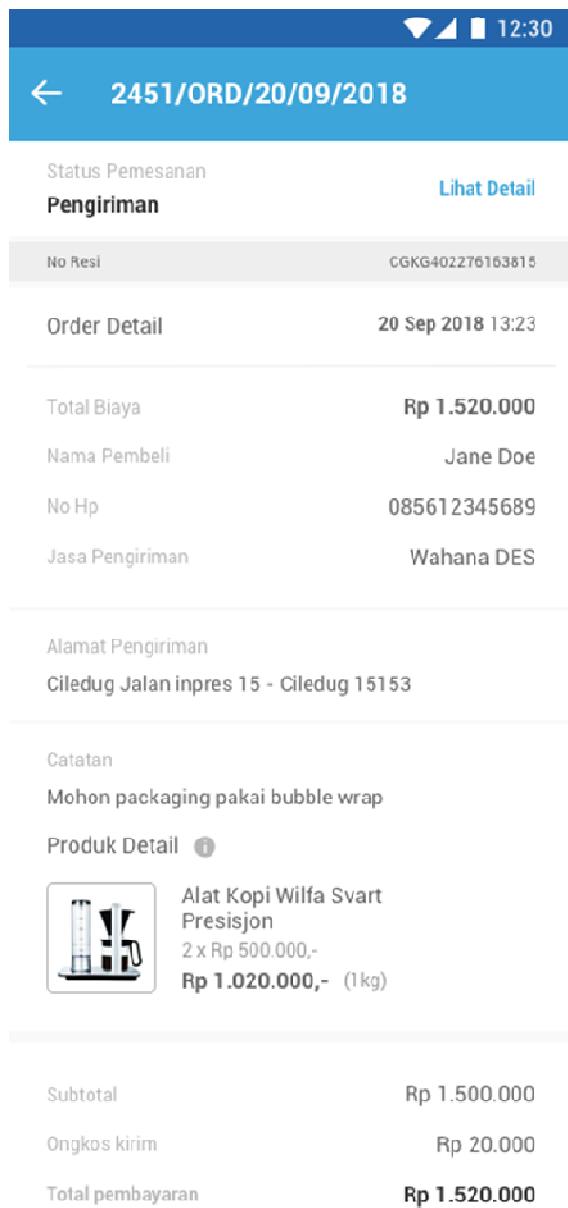
Gambar 4.58 Siap Dipickup-Konfirmasi Alamat

Setelah itu, *seller* dapat pindah ke tab sebelah nya lagi, yaitu tab “Pengiriman”. Dapat dilihat pada gambar dibawah ini :



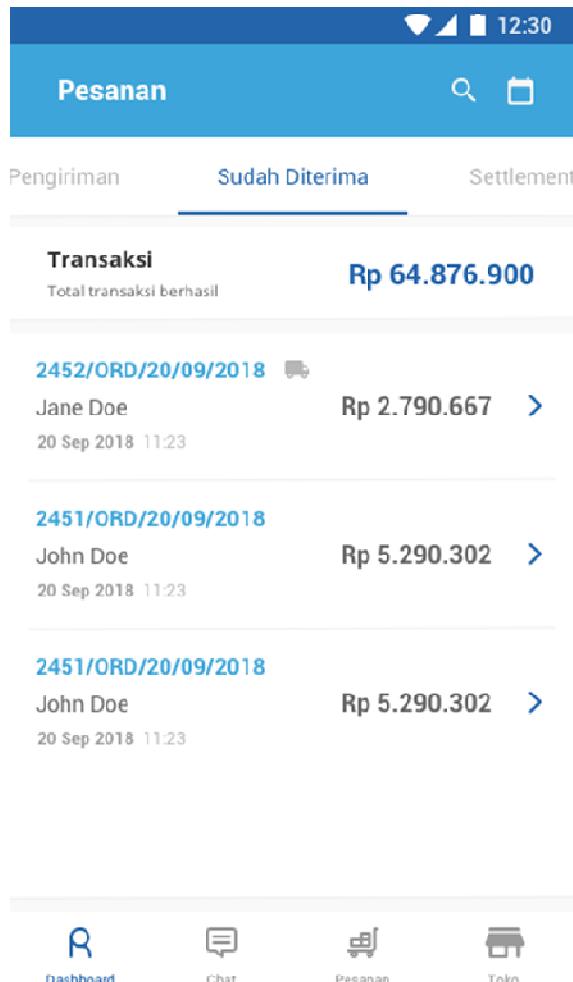
Gambar 4.59 List Pengiriman

Jika kita memilih salah satu orderan yang ada di daftar tersebut, maka akan ditampilkan detail - detail informasi pengiriman, dapat dilihat pada gambar di bawah ini :



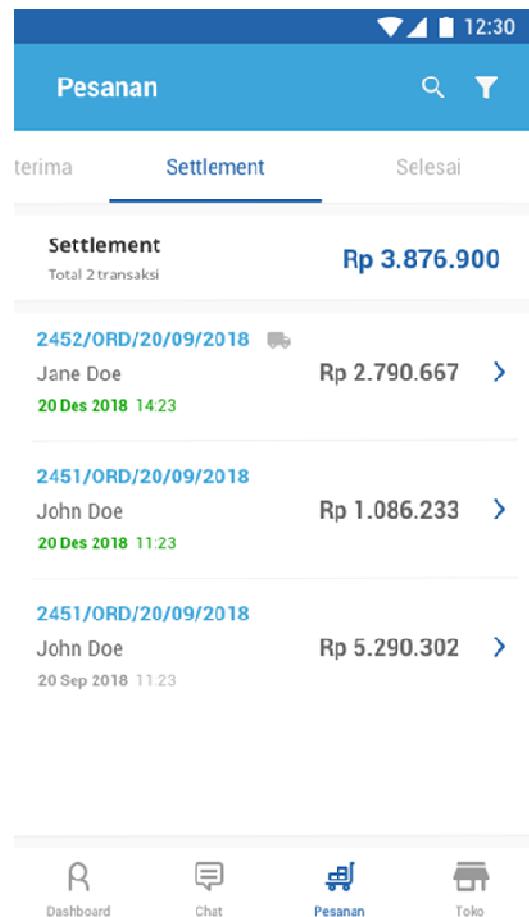
Gambar 4.60 Pengiriman

Jika suatu orderan telah diterima oleh *buyer*, maka orderan tersebut akan berpindah dari list “Pengiriman” dan masuk ke list tab “Diterima”, jika *seller* ingin melihat dapat berpindah ke tab sebelah yaitu tab “Diterima” :



Gambar 4.61 List Diterima

Setelah barang diterima oleh *buyer*, dan *buyer* mengkonfirmasi penerimaan barang, maka *seller* dapat menerima uang dan orderan yang telah diterima tersebut akan dipindahkan ke tab “Settlement”, dapat dilihat pada gambar di bawah ini untuk tampilan halaman “Settlement”:



Gambar 4.62 List Settlement

- c. Kemudian, jika *user* memilih “Laporan Penjualan”, gambar di bawah merupakan gambar tombol untuk memilih “Laporan Penjualan” :



Gambar 4.63 Tombol Laporan Penjualan

Untuk gambaran laporan penjualan belum dirancang oleh pihak UI kantor, dan masih dalam rencana perkembangan di *step* kemudian.

Selain fitur “Pesanan belum diproses”, dan “Laporan Penjualan” yang masih dalam tahap perkembangan, terdapat juga fitur-fitur lainnya seperti “Dashboard”, “Chat”, “Pesanan”, dan “Toko” yang terletak pada bagian paling bawah, berikut adalah gambar nya :



Gambar 4.64 Tab pilihan pada Dashboard

d. *Dashboard*

Ketika *user* sukses melakukan aktivitas *login*, atau user mengklik pada tab “Dashboard”, yang gambar tombolnya seperti di bawah ini :



Gambar 4.65 Tab Dashboard

Setelah itu, akan muncul tampilan *dashboard* yang merupakan tampilan utama dari aplikasi ini, yang dapat dilihat pada gambar 4.25 di atas.

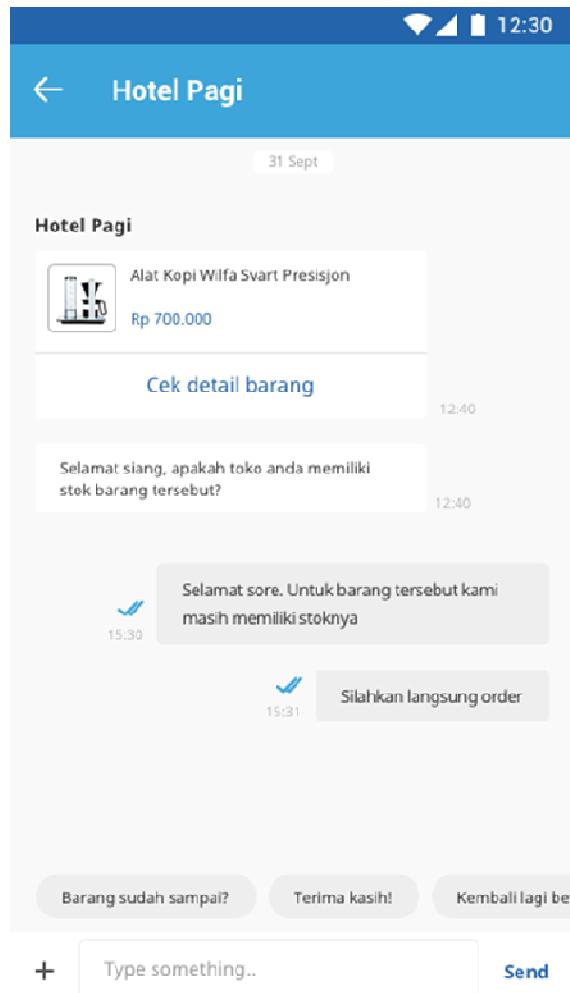
e. *Chat*

Ketika *user* melakukan aktivitas klik pada tab “*Chat*”, maka akan muncul tampilan seperti pada gambar di bawah ini :



Gambar 4.66 List Chats

Kemudian jika *user* memilih salah satu dari *list chat* yang masuk, maka akan muncul tampilan seperti foto di bawah ini :



Gambar 4.67 Chat Open

Dari gambar di atas, dapat diketahui bahwa :

1. ketika ada *customer* yang menanyakan suatu produk tertentu, maka akan muncul gambar produk dan tombol untuk melihat detail nya untuk mempermudah *seller* dalam mengecek produk yang sedang ditanya.
2. Adanya template balasan yang sudah disiapkan untuk mempermudah *seller* dalam menjawab tanpa harus capek mengetik.

f. Toko

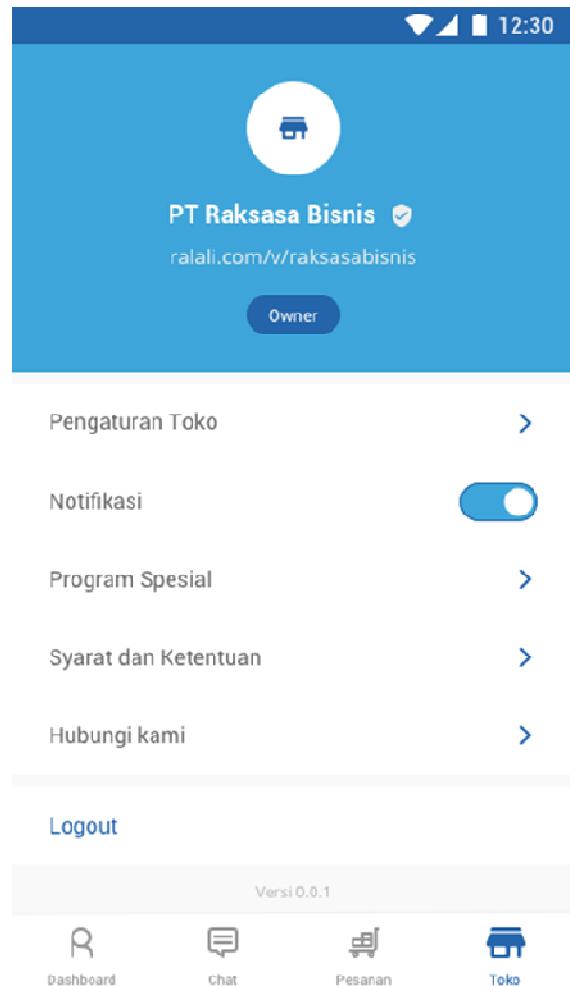
Pada aplikasi Ralali *Seller* , terdapat tab “Toko” seperti pada gambar di bawah ini:



Toko

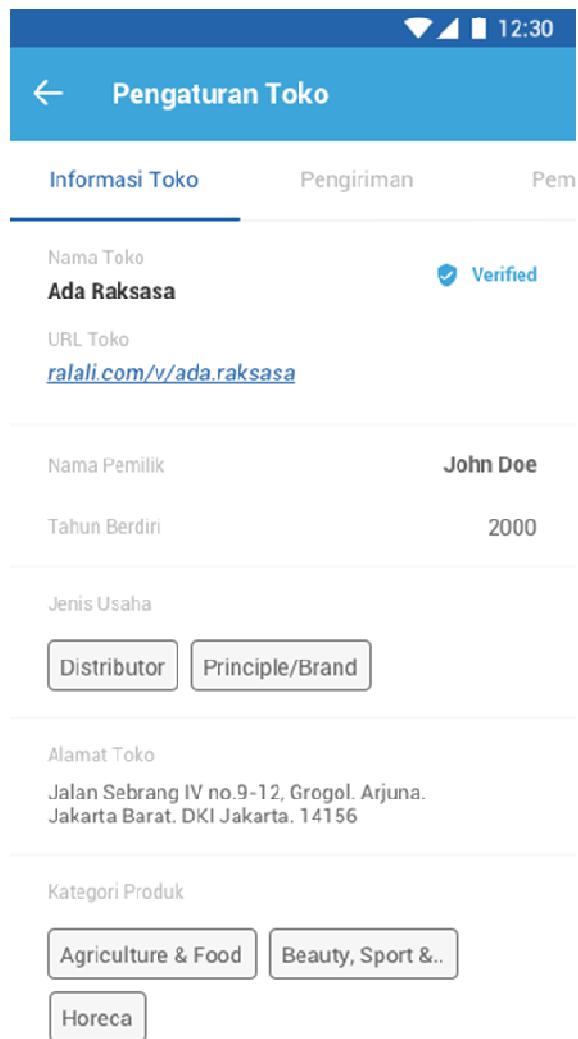
Gambar 4.68 Tab Toko

Setelah melakukan aktivitas klik pada tab “Toko”, akan ditampilkan “Pengaturan Toko” , dapat dilihat pada gambar di bawah ini :



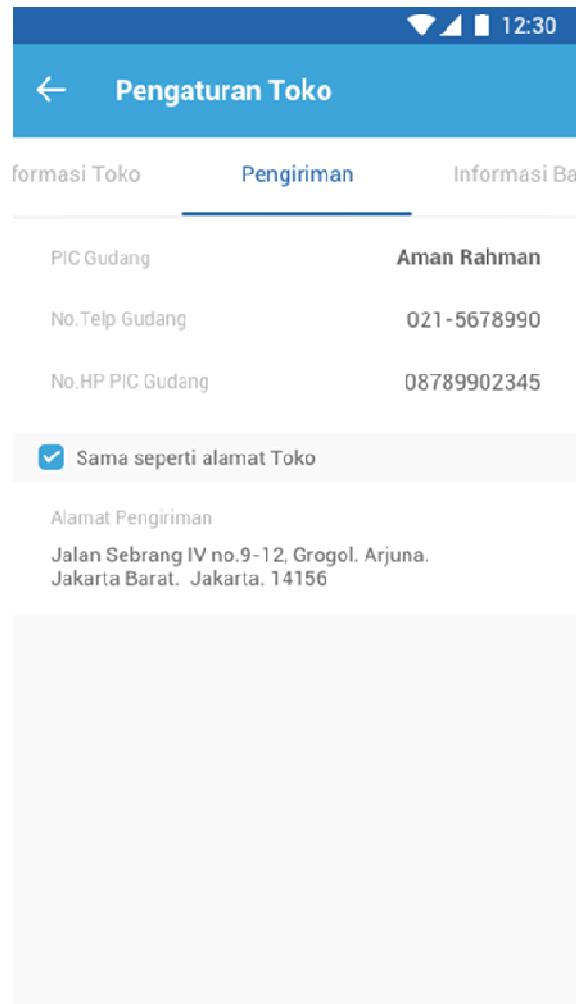
Gambar 4.69 View Toko

Kemudian jika kita pilih tab “Pengaturan Toko”, maka sistem akan menampilkan tampilan pada tab “Informasi Toko” seperti pada gambar di bawah ini :



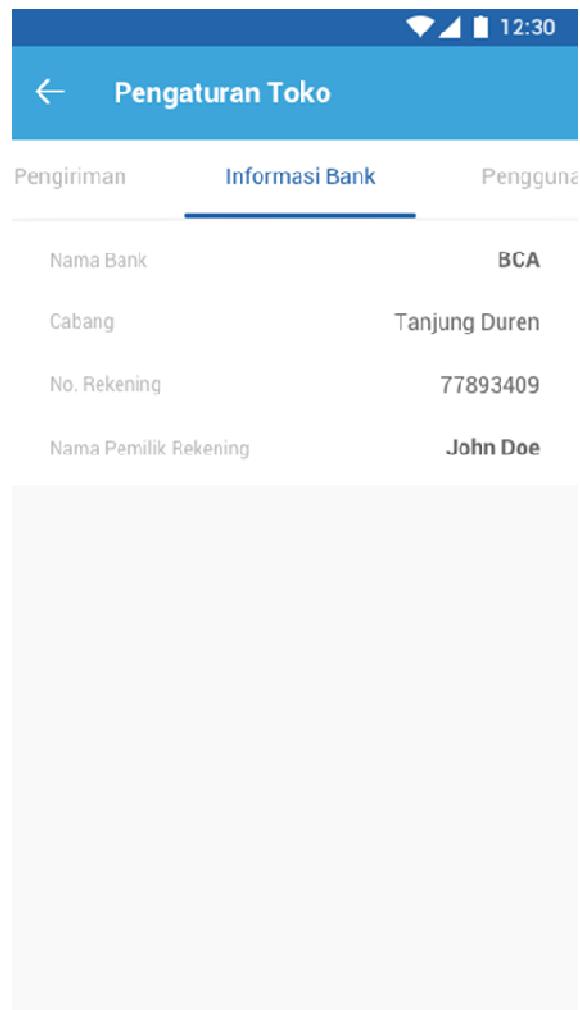
Gambar 4.70 Informasi Toko

Ditampilkan banyak informasi mengenai toko sang *seller*. Kemudian jika kita berpindah ke tab sebelah (tab “Pengiriman”), maka akan ditampilkan informasi - informasi mengenai gudang *seller* yang seller daftarkan :



Gambar 4.71 Gudang Seller

Kemudian jika kita berpindah ke tab sebelah (tab “Pembayaran”), maka akan ditampilkan informasi - informasi mengenai metode pembayaran yang seller daftarkan supaya ketika *buyer* sudah menerima barang orderan, Ralali dapat mengirimkan uang kepada *seller*, berikut adalah gambar nya :



Gambar 4.72 Pembayaran Seller

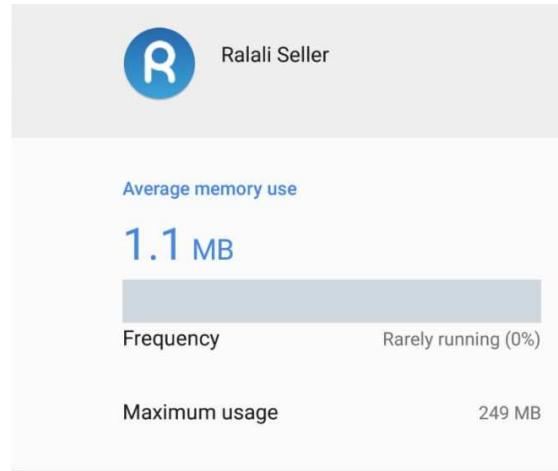
4.4 Implementasi

4.4.1 Spesifikasi yang digunakan

Di bawah ini merupakan spesifikasi minimum yang diperlukan untuk menjalankan aplikasi Ralali Seller. Spesifikasi minimum hardware:

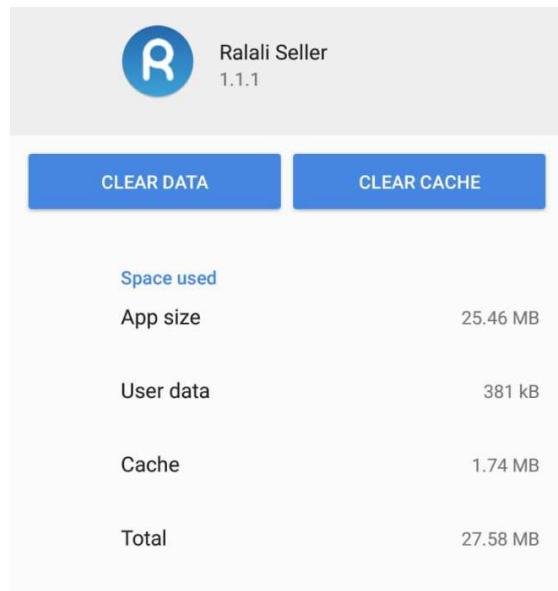
- a. *Random Access Memory (RAM)*

Sesuai dengan gambar dibawah RAM yang digunakan untuk menjalankan aplikasi ini sebaiknya lebih dari 249MB



b. *Memory*

Sesuai dengan gambar dibawah memory yang digunakan untuk menjalan aplikasi ini sebaiknya lebih dari 27.58 MB



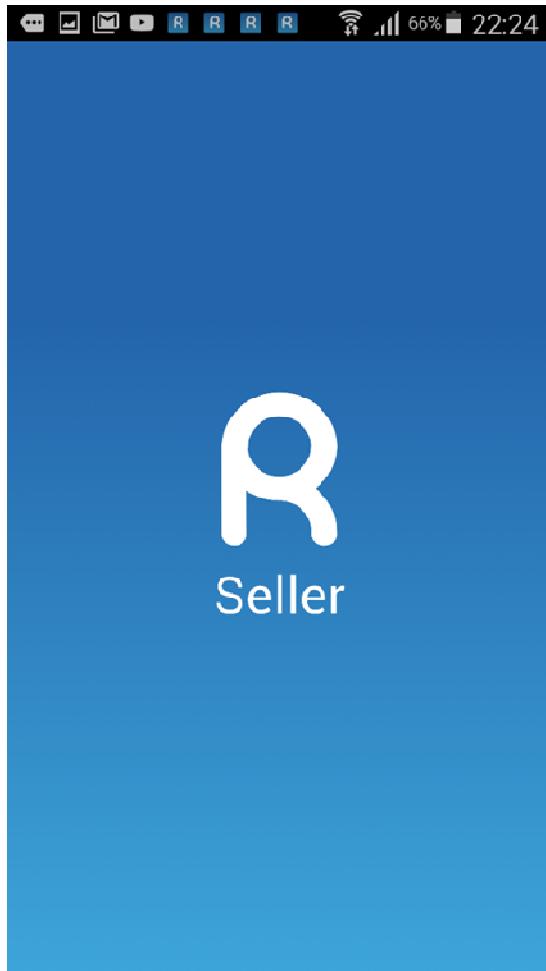
c. *Operating System*

Berdasarkan gambar source code dari Ralali Seller, Minimum *Operating System* untuk menjalankan aplikasi ini adalah yang memiliki SDK Version 16 atau yang biasa disebut Android 4.1(Jelly Bean).

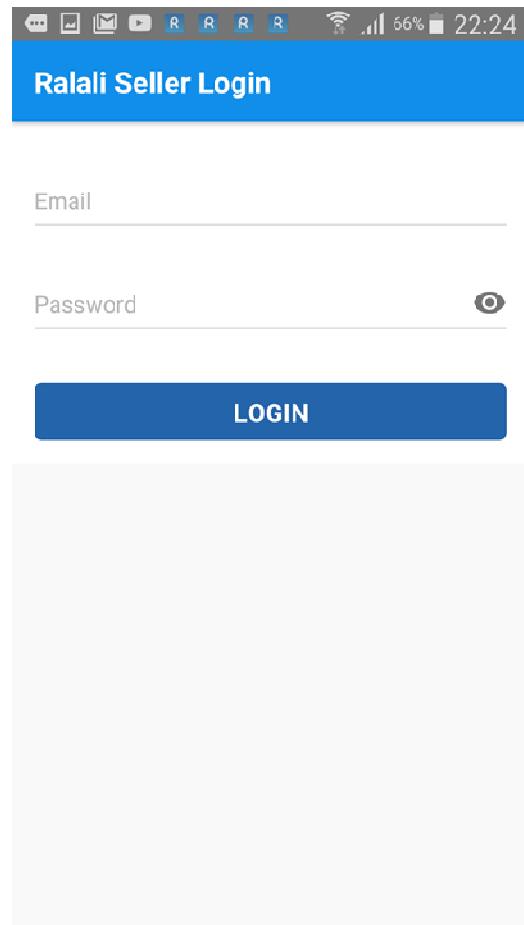
```
defaultConfig {  
    applicationId "com.ralaliseller"  
    minSdkVersion 16  
    targetSdkVersion 27  
    versionCode 4  
    versionName "1.1.1"  
    multiDexEnabled true  
    ndk {  
        abiFilters "armeabi-v7a", "x86"  
    }  
}
```

4.4.2 Dokumentasi Program

4.4.2.1 *Splash Screen & Login*



Gambar 4.73 *Screen Shot Splash Screen*



Gambar 4.74 Screen Shot Login

Ketika menekan Icon ‘Ralali Seller’ pada kumpulan aplikasi yang ada di *handphone user*, maka yang akan muncul pertama kali adalah *splashscreen* yang berisikan logo “ Ralali Seller” berada di tengah tengah dengan *background* biru, setelah *splashscreen* hilang maka user akan berada di *screen* ‘Login’

Pada *screen* ‘Login’ ada 2 *form input text* yaitu *input text* ‘Email’ dan *input text* ‘Password’ di dalam form ‘Password’ ada *button icon* yang bisa menampilkan password apabila kita menekannya.

User diharapkan memasukan email dan *password* pada form yang sudah disediakan lalu ketika sudah selesai memasukan email dan

password, user diharapkan menekan *button* ‘Login’. Jika Email dan *password* sesuai maka *user* diarahkan ke *screen* ‘Dashboard’.

4.4.2.2 Dashboard



Gambar 4.75 Screen Shot Dashboard



Gambar 4.76 Screen Shot Calendar 1

Pada tampilan *Dashboard* terdapat beberapa tampilan yang disajikan seperti Informasi nama toko dan nama bulan sebagai *filter* ketika di klik akan muncul pilihan filter seperti di gambar, selanjutnya ada infomasi total penjualan pengguna di tampilkan menurut bulan yang dipilih, di bawahnya di tampilkan grafik dan angka data penjualan selama 1 bulan, di bawahnya terdapat informasi data penjualan dalam bentuk angka dan lebih spesifik ada transaksi, produk dan pembeli. Selanjutnya terdapat 2 *button* yang bisa di klik yaitu ‘Pesanan belum diproses’ yang akan mengarah ke *tab* order dan laporan penjualan yang akan mengarah ke *screen* fitur sedang dikembangkan.

Selain juga tampilan paling bawah di sajikan navigasi *tab* untuk *user* mudah berpindah sesuai aktifitas yang dia inginkan. Ada 4 tab navigasi yaitu Dashboard, Chat, Order dan Toko.

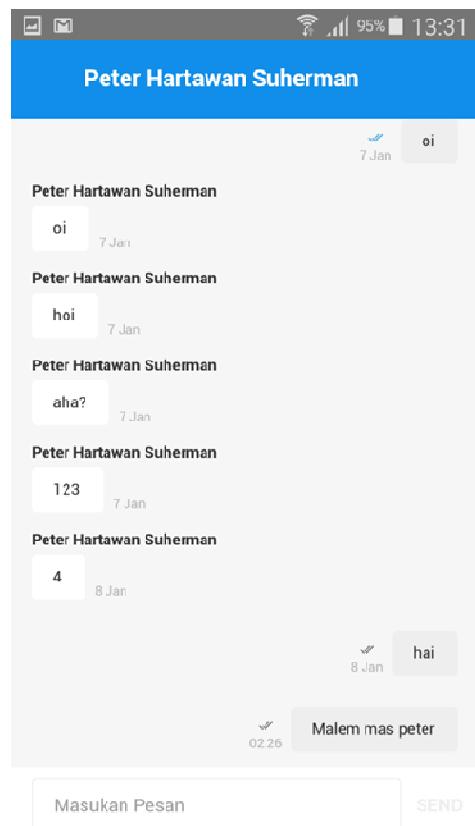


Gambar 4.77 Screen Shot Calendar 2

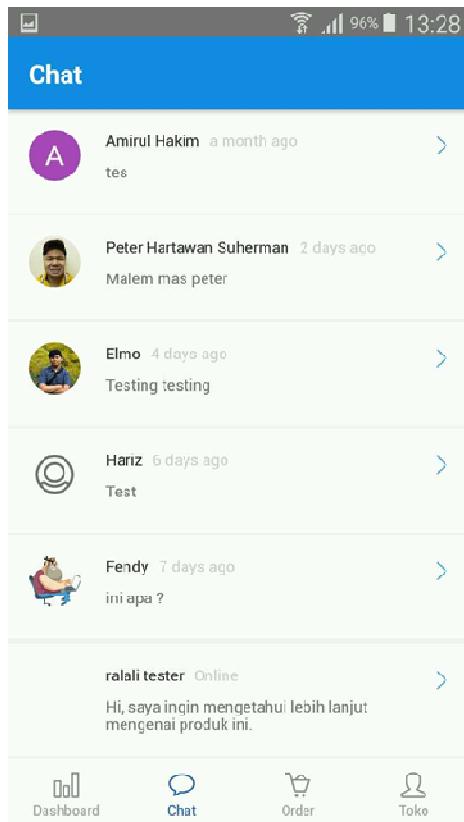


Gambar 4.78 Screen Shot Dashboard 2

4.4.2.3 Chat



Gambar 4.79 Screen Shot Chat



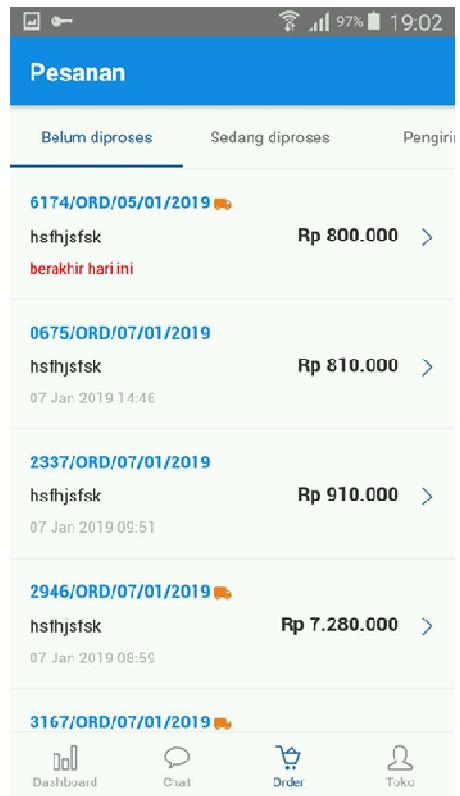
Gambar 4.80 Screen Shot Chat List

Pada *tab* navigasi Chat di sajikan *Chatlist* dengan menampilkan informasi foto, nama, waktu dan pesan terakhir dari *user* lain yang melakukan aktifitas *chatting* dengan *user* yang login.

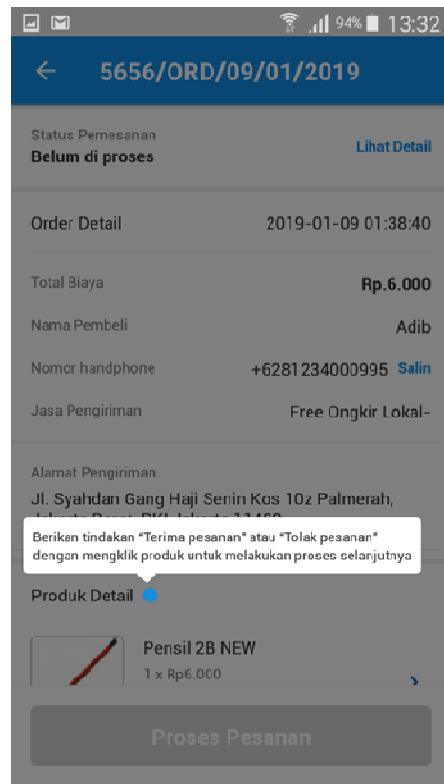
Setelah memilih atau klik salah satu dari *chatlist user* langsung diarahkan ke *screen chat detail* yang menampilkan *history chat* dan bisa melakukan aktifitas *chatting* dengan mengirim dan menerima pesan.

4.4.2.4 Order Management

4.4.2.4.1 Order Belum Diproses



Gambar 4.81 Screen Shot Order Belum Diproses

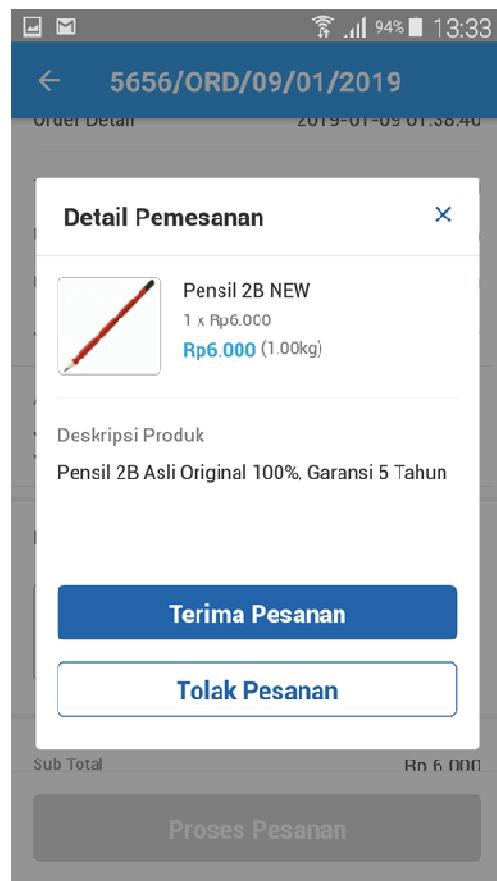


Gambar 4.82 Screen Shot Order belum diproses 2

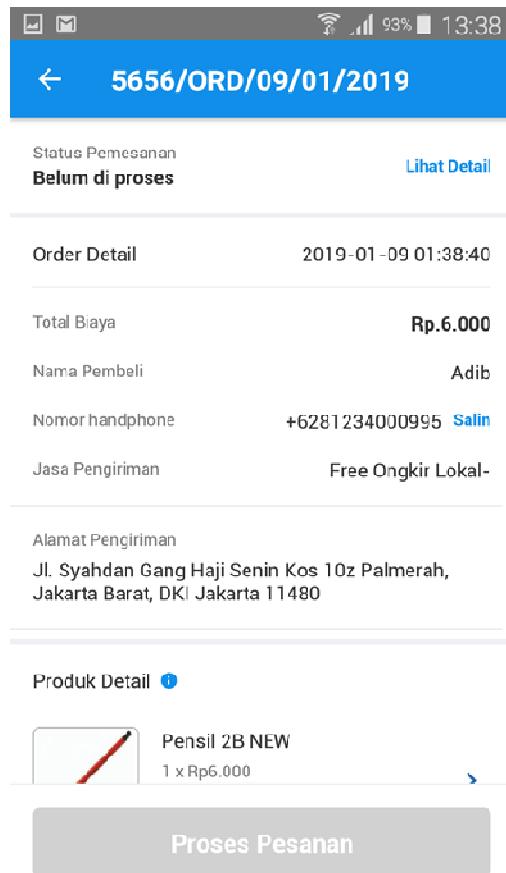
Pada tab navigasi Order setelah di klik akan masuk ke tampilan seperti di gambar 4.68, dengan menampilkan beberapa tab baru di atas diantara nya belum diproses, sedang diproses, pengiriman, diterima, settlement.

Pada setiap bagian tab menampilkan informasi list yang hampir sama diantaranya nomor order yang di ikuti icon truk berwarna orange menunjukan bahwa order tersebut menggunakan pengiriman ralali kargo dan yang tidak ada icon menggunakan jasa pengiriman selain ralali kargo, selanjutnya dibawah nomor order terdapat nama pembeli dan waktu, serta di bagian kanan terdapat keterangan total harga dari *order*.

Ketika *user* melakukan klik pada salah satu order maka akan di arahkan ke orderscreen dan akan muncul *coach-mark* yang menjelaskan untuk menerima pesanan atau tolak pesanan pada section produk.



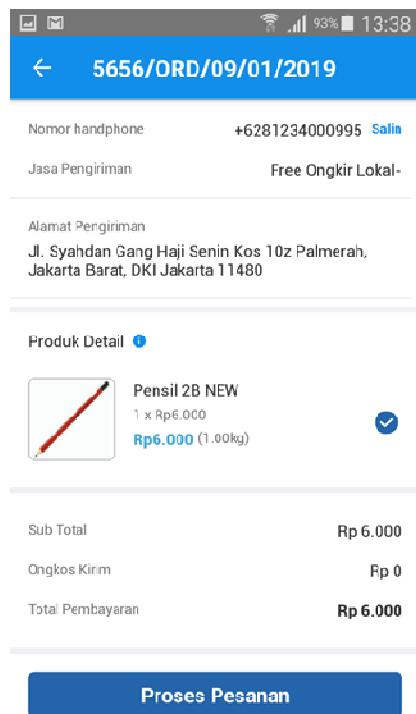
Gambar 4.83 Screen Shot Memproses Pesanan



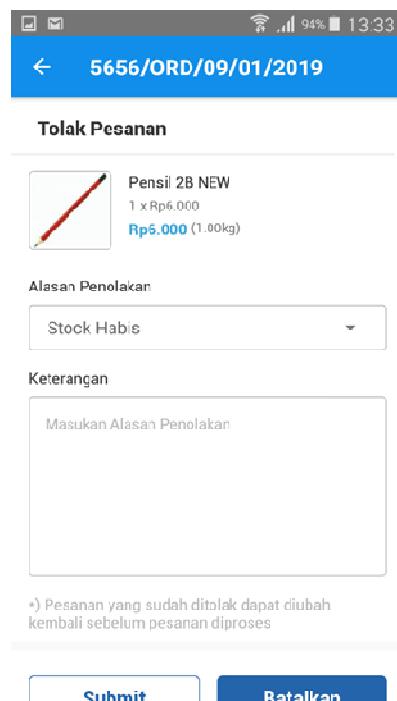
Gambar 4.84 Screen Shot Memproses Pesanan 2

Informasi yang ada dalam layar adalah sebagai berikut:

1. Status pemesanan
2. Order detail (Total biaya, Nama pembeli, No Hp, Jasa Pengiriman)
3. Alamat Pengiriman
4. Catatan order
5. Produk detail yang berisi item-item yang di pilih oleh buyer yang menampilkan informasi (Nama produk, jumlah produk, harga dan status item)
6. Informasi harga (Subtotal, Ongkos kirim, Total Pembayaran)
7. Button Proses Pesanan (Disable jika belum ada order yang di terima/tolak)



Gambar 4.85 Screen Shot Memproses Pesanan 3



Gambar 4.86 Screen Shot Memproses Pesanan 4

Ketika user menekan tombol ‘terima pesanan’ pada *popup* modal, maka akan diarahkan kembali ke *screen* order detail, dan button ‘Proses Pesanan’ akan aktif, jikalau pada modal tadi kita klik button ‘Tolak pesanan’ maka akan diarahkan menuju *screen* tolak pesanan dan di harapkan mengisi alasan penolakan dan keterangannya, setelah itu klik tombol submit maka pesanan akan tertolak dan jika klik batalkan maka akan kembali ke screen order detail.

Ketika menekan tombol proses pesanan yang ada di *screen* order detail yang terjadi adalah order status akan terupdate menjadi order sedang diproses, dan hilang dari order list ‘belum diproses’.

4.4.2.4.2 Order Sedang Diproses



Gambar 4.87 Screen Shot Order Sedang Diproses 1



Gambar 4.88 Screen Shot Order Sedang Diproses 2

Informasi yang ada dalam layar adalah sebagai berikut:

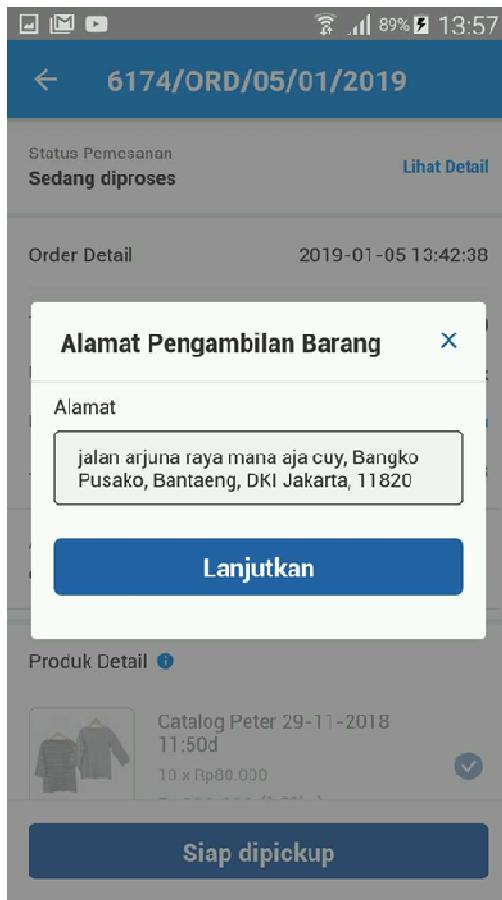
1. Status pemesanan “Sedang diproses”
2. Order detail (Total biaya, Nama pembeli, No Hp, Jasa Pengiriman)
3. Alamat Pengiriman dengan icon berwarna orange
4. Produk detail yang berisi item-item yang di pilih oleh buyer yang menampilkan informasi (Nama produk, jumlah produk, harga dan status item)
5. Informasi harga (Subtotal, Ongkos kirim, Total Pembayaran)
6. Button Proses Siap di pickup

Informasi yang disajikan pada order list sedang diproses hampir sama dengan order list belum diproses, bedanya hanya status icon dari ralali kargo ada yang berwarna

abu-abu itu menandakan kalau order yang menggunakan ralali kargo sudah ber-status ‘siap dipickup’, jikalau masih berwarna orange, harus mengubah status menjadi ‘siap dipickup’ dengan cara masuk ke order detail dan tekan button ‘Siap dipickup’.



Gambar 4.89 Screen Shot Order Sedang Diproses 3



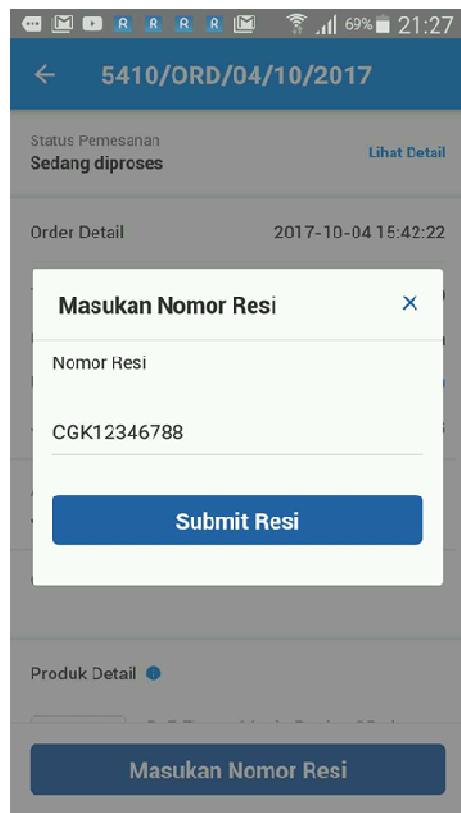
Gambar 4.90 Screen Shot Order Sedang Diproses 4

Ketika menekan tombol ‘Siap dipickup’ maka akan muncul modal yang menampilkan alamat pengambilan barang dan button ‘Lanjutkan’, Ketika menekan tombol ‘Lanjutkan’ maka status Ralali kargo akan ter-update menjadi ‘Siap dipickup’ dan di order detail tampilan akan berubah icon ralali kargo dari warna orange menjadi icon ralali kargo warna abu-abu, dan button ‘Siap dipickup’ berubah menjadi button ‘Masukan Nomor Resi’.

Button ‘Masukan Nomor Resi’ juga akan tampil jika kondisinya, status pengiriman bukan ralali kargo dan order dengan ralali kargo tapi dengan status sudah dirubah menjadi ‘Siap dipickup’ dan juga harus berada di order list sedang diproses.



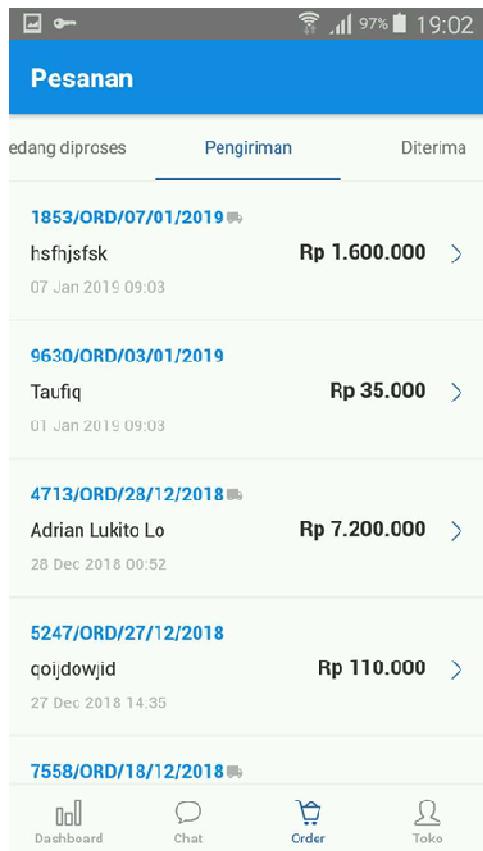
Gambar 4.91 *Screen Shot Order Sedang Diproses 5*



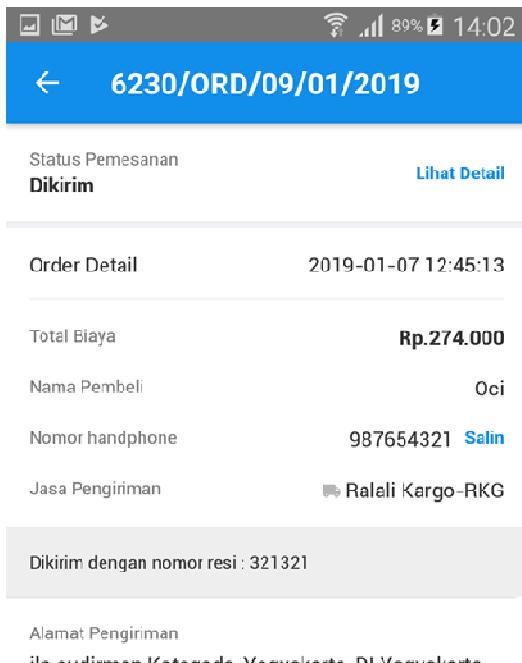
Gambar 4.92 Screen Shot Order Sedang Diproses 6

Ketika menekan tombol ‘Masukan Nomor Resi’ maka akan muncul modal ‘Masukan Nomor Resi’ dan ada input text untuk memasukan nomor resi , ketika kita masukan nomor resi dan menekan button ‘Submit Resi’ maka yang akan terjadi adalah muncul loading page dengan overlay text bertuliskan mohon tunggu, setelah loading selesai makan order akan berubah status menjadi ‘Pengiriman’ dan tampilan akan kembali ke *order list*.

4.4.2.4.3 Order Pengiriman



Gambar 4.93 Screen Shot Pengiriman I



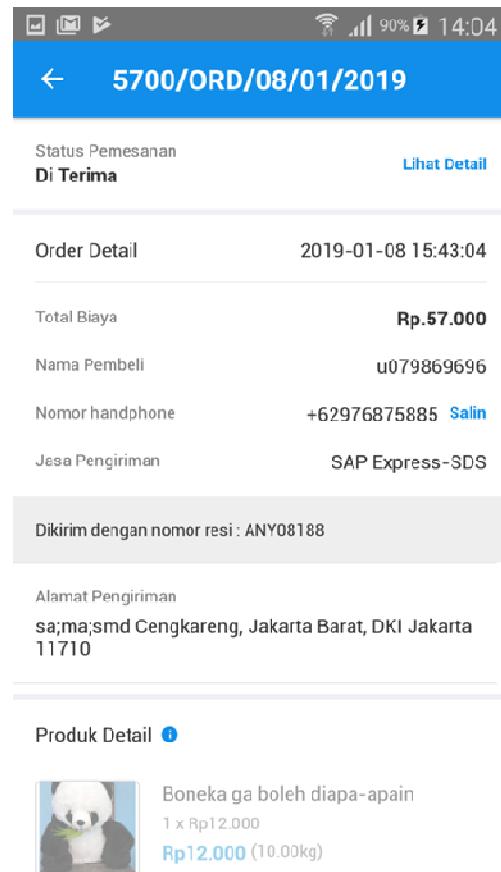
Gambar 4.94 Screen Shot Pengiriman 2

Informasi yang ada dalam layar adalah sebagai berikut:

1. Status pemesanan “Dikirim”
2. Order detail (Total biaya, Nama pembeli, No Hp, Jasa Pengiriman)
3. Nomor resi
4. Alamat Pengiriman
5. Produk detail yang berisi item-item yang di pilih oleh buyer yang menampilkan informasi (Nama produk, jumlah produk, harga dan status item)
6. Informasi harga (Subtotal, Ongkos kirim, Total Pembayaran)

Pada *screen* order detail ini tidak terdapat button yang berada di bawah seperti order detail ‘belum diproses’ dan ‘sedang diproses’ karena pada screen ini tidak ada melakukan aktifitas, hanya sekedar menampilkan informasi.

4.4.2.4.4 Order Diterima



Gambar 4.95 Screen Shot Order Diterima 1

Informasi yang ada dalam layar adalah sebagai berikut:

1. Status pemesanan “Diterima”
2. Order detail (Total biaya, Nama pembeli, No Hp, Jasa Pengiriman)
3. Nomor resi
4. Alamat Pengiriman

5. Produk detail yang berisi item-item yang di pilih oleh buyer yang menampilkan informasi (Nama produk, jumlah produk, harga dan status item)
6. Informasi harga (Subtotal, Ongkos kirim, Total Pembayaran)

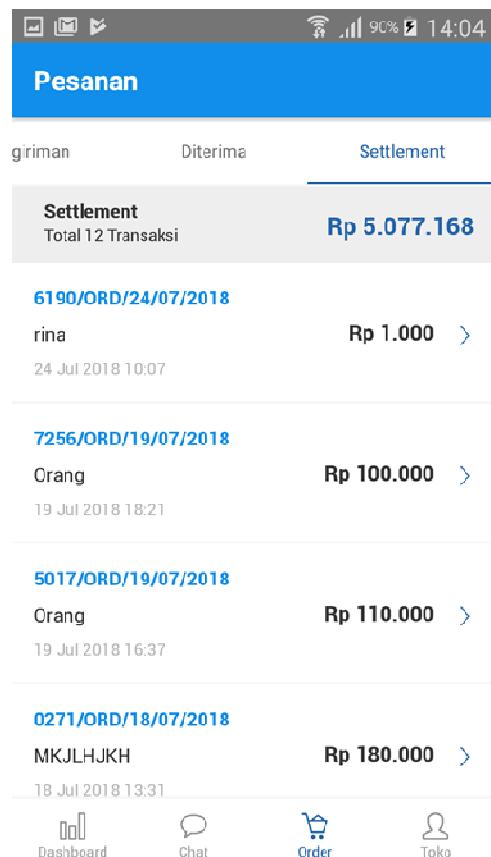


Gambar 4.96 Screen Shot Order Diterima 2

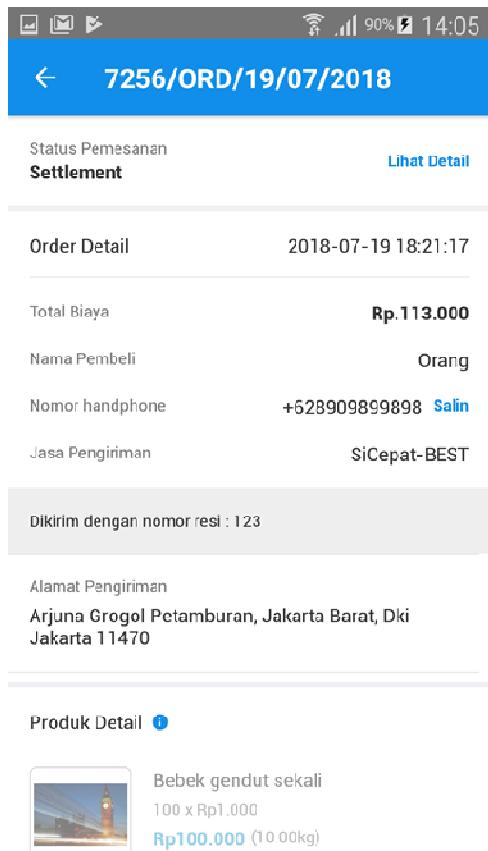
Informasi yang disajikan pada order list ‘Di terima’ tidak jauh beda dengan order list yang lainnya , yang berbeda ada pada bawah tab yang menampilkan total jumlah order dan total nominal order yang berada di tab order list ‘Diterima’ dan juga yang berbeda ada pada *screen* order detail, status pemesanan berubah menjadi Diterima dan juga di bawah jasa pengiriman muncul informasi tentang nomor resi.

Pada *screen* order detail ini tidak terdapat button yang berada di bawah seperti order detail ‘belum diproses’ dan ‘sedang diproses’ karena pada screen ini tidak ada melakukan aktifitas, hanya sekedar menampilkan informasi.

4.4.2.4.5 Order Settlement



Gambar 4.97 Screen Shot Order Settlement 1



Gambar 4.98 Screen Shot Order Settlement 2

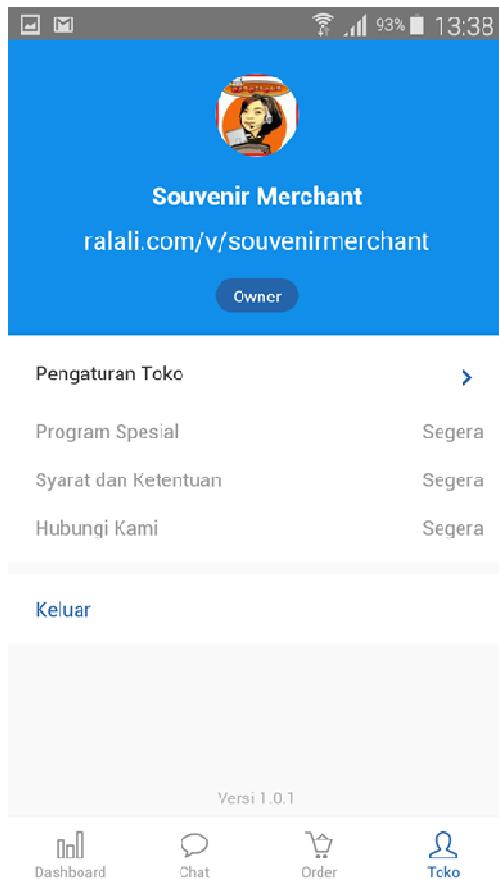
Informasi yang ada dalam layar adalah sebagai berikut:

1. Status pemesanan “Settlement”
2. Order detail (Total biaya, Nama pembeli, No Hp, Jasa Pengiriman)
3. Nomor resi
4. Alamat Pengiriman
5. Produk detail yang berisi item-item yang di pilih oleh buyer yang menampilkan informasi (Nama produk, jumlah produk, harga dan status item)
6. Informasi harga (Subtotal, Ongkos kirim, Total Pembayaran)

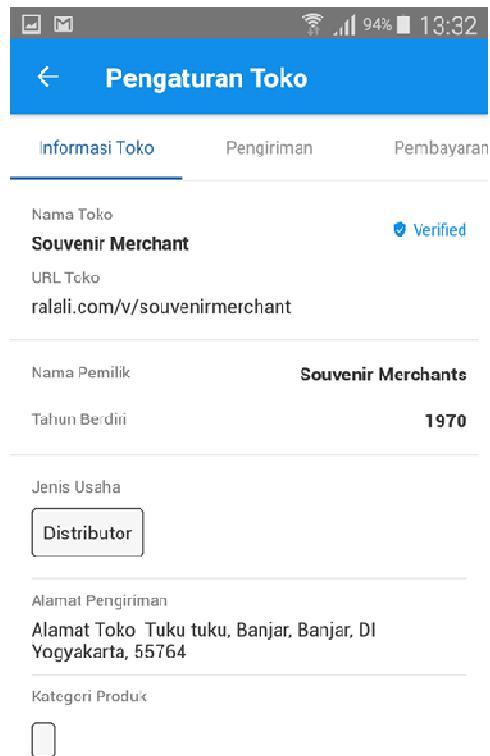
Pada *screen* order detail ini tidak terdapat button yang berada di bawah seperti order detail ‘belum diproses’ dan

‘sedang diproses’ karena pada screen ini tidak ada melakukan aktifitas, hanya sekedar menampilkan informasi.

4.4.2.5 Toko



Gambar 4.99 Screen Shot View Toko 1



Gambar 4.100 Screen Shot View Toko 2

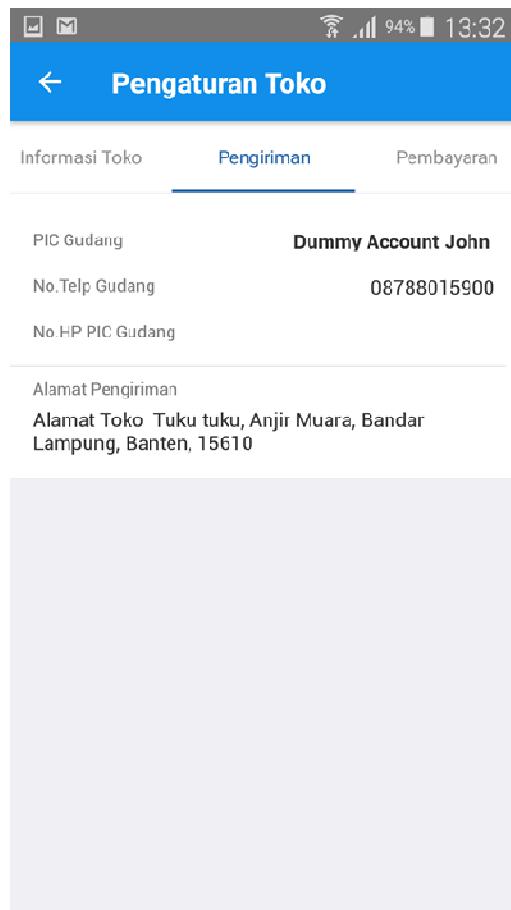
Pada tab navigasi bawah yang terakhir adalah tab ‘Toko’. Ketika menekan tab ‘Toko’ maka akan menampilkan seperti gambar berikut, muncul informasi gambar logo dari akun, nama akun/nama toko, link toko, status *user*.

Selain itu juga ada button ‘Pengaturan Toko’ dan button ‘Keluar’ yang aktif, untuk button ‘Program Spesial’, ‘Syarat dan ketentuan’ dan ‘Hubungi Kami’ Sengaja di non-aktif.

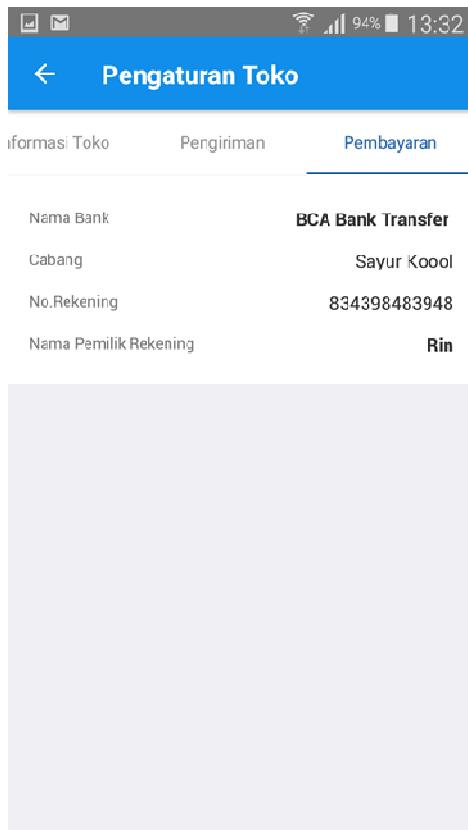
Ketika menekan button ‘Pengaturan Toko’ akan diarahkan ke *screen* ‘Pengaturan Toko yang ada di gambar di atas. Dan ketika kita menekan button ‘keluar’, maka akan melakukan aktifitas ‘Logout’ dan di arahkan kembali ke *screen* ‘Login’.

Pada *screen* ‘Pengaturan Toko’ akan menampilkan 3 tab yaitu ‘Informasi Toko’, ‘Pengiriman’ dan ‘Pembayaran’. Pada tab ‘Informasi Toko’ akan menampilkan informasi seperti Nama toko,

Status toko, link toko, nama pemilik, jenis usaha, alamat pengiriman dan kategori produk.



Gambar 4.101 *Screen Shot View Toko 3*



Gambar 4.102 Screen Shot View Toko 4

Pada *screen* ‘Pengaturan Toko’ selain tab ‘Informasi Toko’ juga ada 2 tab yang lain yaitu tab ‘Pengiriman’ yang berisi informasi nama PIC gudang, nomor telepon gudang, nomor handphone PIC gudang dan alamat pengiriman. Sedangkan pada tab ‘Pembayaran’ berisi informasi nama bank, cabang bank, nomor rekening bank, nama pemilik rekening bank.

4.5 Testing

4.5.1 Testing Yang Dilakukan Dan Hasilnya

Testing dilakukan oleh *user* dengan memeriksa setiap modul pada aplikasi. Metode yang digunakan untuk menguji fungsi-fungsi yang ada di dalam sistem agar berjalan seperti yang diharapkan adalah *Black Box Testing*, yang berfokus pada keperluan fungsional dari aplikasi tanpa melihat proses

detail yang terjadi di belakangnya. Tabel-tabel berikut ini adalah hasil pengujian modul yang sudah dilakukan.

Test Case 1 – Login

Aktor: User Seller

Tabel 4.41 Test Case Login

No.	Action	Expected Result	Status
1.	User mengakses aplikasi	Aplikasi terbuka dan menampilkan splashscreen setelah itu menampil Login Screen yang berisi form email dan password	Berhasil
2.	User memasukan email dan password pada form yang terlah tersedia dan menekan tombol login	Jika berhasil: Aplikasi menampilkan halaman dashboard	Berhasil
		Jika gagal: Aplikasi akan menampilkan pesan error pada halaman login	

Test Case 2 – Dashboard

Aktor: User Seller

Tabel 4.42 Test Case Dashboard

No.	Action	Expected Result	Status
1.	User mengakses Halaman Dashboard	Menampilkan informasi data penjualan user dalam grafik dan angka, serta total transaksi, produk dan pembeli, serta menampilkan filter dalam berdasarkan bulan, dan 2 button ‘Pesanan belum diproses’ dan ‘Laporan Penjualan’	Berhasil

2.	<i>User</i> menekan tombol filter berdasarkan bulan	Menampilkan popup modal yang menyajikan pilihan tahun dan bulan	Berhasil
3.	<i>User</i> menekan atau memilih bulan yang tersedia di popup filter	Tampilan akan kembali ke dashboard dan menghilangkan popup serta data tampilan yang ada di dashboard akan berubah sesuai dengan pilihan bulan user	Berhasil
4.	<i>User</i> menekan tombol ‘Pesanan belum di proses’	Tampilan akan di arahkan ke screen ‘Order list’	Berhasil
5.	<i>User</i> menekan tombol ‘Laporan Penjualan’	Tampilan akan di arahkan ke screen ‘Fitur sedang dikembangkan’	Berhasil

Test Case 3 – Chat

Aktor: User Seller

Tabel 4.43 Test Case Chat

No.	Action	Expected Result	Status
1.	<i>User</i> menekan tab Chat	Berpindah ke screen Chat dengan menampilkan Chat list	Berhasil
2.	<i>User</i> menekan atau memilih salah satu dari chat list button	Berpindah ke screen Chat detail dan menampilkan <i>message list</i> dari user dan juga menampilkan input text dan button send untuk user mengirim pesan	Berhasil
3.	<i>User</i> mengirim pesan dengan cara memasukan text dalam form input text dan menekan tombol send	Pesan text akan di tampilkan pada message list dan dikirim ke user target	Berhasil

Test Case 4 – View Order List

Aktor: User Seller

Tabel 4.44 Test Case Order list

No.	Action	Expected Result	Status
1.	<i>User</i> menekan tab Order	Berpindah ke screen Order dengan menampilkan 5 tab order list yaitu ‘Belum diproses’, ‘Sedang diproses’, ‘Pengiriman’, ‘Settlement’ dan ‘Settlement’	Berhasil
2.	<i>User</i> menekan tab order list ‘Belum diproses’	Berpindah ke tab order list ‘Belum diproses’ dan menampilkan order-order yang berstatus ‘Belum diproses’	Berhasil
3.	<i>User</i> menekan tab order list ‘Sedang diproses’	Berpindah ke tab order list ‘Sedang diproses’ dan menampilkan order-order yang berstatus ‘Sedang diproses’	Berhasil
4.	<i>User</i> menekan tab order list ‘Pengiriman’	Berpindah ke tab order list ‘Pengiriman’ dan menampilkan order-order yang berstatus ‘Pengiriman’	Berhasil
5.	<i>User</i> menekan tab order list ‘Diterima’	Berpindah ke tab order list ‘Diterima’ dan menampilkan order-order yang berstatus ‘Diterima’	Berhasil
6.	<i>User</i> menekan tab order list Settlement	Berpindah ke tab order list ‘Settlement’ dan menampilkan order-order yang berstatus ‘Settlement’	Berhasil

Test Case 5 – Process Order

Aktor: User Seller

Tabel 4.45 Test Case Process Order

No.	Action	Expected Result	Status
1.	<i>User</i> menekan tab Order list ‘Belum diproses’	Menampilkan order-order yang berstatus ‘Belum diproses’	Berhasil
2.	<i>User</i> menekan atau memilih salah satu order dari order list ‘Belum diproses’	Berpindah ke screen order detail, yang pertama akan ditampilkan user adalah <i>coachmark</i> cara memproses order setelah itu menampilkan informasi sesuai dengan order yang dipilih. Selain informasi tentang detail order juga ditampilkan item-item produk yang ada pada order tersebut, dan juga di bawahnya terdapat button ‘Proses Pesanan’ yang masih non-aktif	Berhasil
3.	<i>User</i> menekan salah satu item produk yang ada pada order detail	Tampilan akan memunculkan pop-up modal yang berisikan informasi tentang produk dan 2 button pililah ‘Terima Pesanan’ dan ‘Tolak Pesanan’	Berhasil
4.	<i>User</i> menekan tombol ‘Terima Pesanan’ pada pop-up modal	Tampilan diarahkan kembali ke order detail dan button ‘Proses Pesanan’ akan aktif apabila item hanya 1, apabila item produk lebih dari satu maka harus melakukan aktivitas terima atau tolak pesanan terlebih dahulu pada	Berhasil

		semua item produk	
5	<i>User</i> menekan tombol ‘Tolak Pesanan’ pada pop-up modal	Tampilan akan diarahkan ke screen tolak pesanan yang akan menampilkan form pilihan alasan penolakan dan keterangan serta ada 2 button ‘Submit’ dan button ‘Batalkan’	Berhasil
6.	<i>User</i> menekan tombol ‘Batalkan’ pada screen ‘Tolak Pesanan’	Tampilan akan diarahkan kembali ke order detail	Berhasil
7.	<i>User</i> memilih asalan penolakan dan memasukan text pada kolom keterangan serta menekan tombol ‘Submit’ pada screen ‘Tolak Pesanan’	Tampilan diarahkan kembali ke order detail dan button ‘Proses Pesanan’ akan aktif apabila item hanya 1, apabila item produk lebih dari satu maka harus melakukan aktifitas terima atau tolak pesanan terlebih dahulu pada semua item produk	Berhasil
8.	<i>User</i> menekan tombol ‘Proses Pesanan’	<p>Tampilan akan muncul loading indicator dengan overlay screen warna hitam.</p> <p>Jika berhasil:</p> <p>Tampilan akan diarahkan ke Order list</p> <p>Jika gagal:</p> <p>Tampil popup error message</p>	Berhasil

Test Case 7 – Update Status Order ‘Siap dipickup’

Aktor: User Seller

Tabel 4.46 Test Case Order Siap dipickup

No.	Action	Expected Result	Status
1.	Di dalam list order ‘Sedang diproses’ user memilih atau menekan order yang ada icon truk berwarna orange	Berpindah ke screen order detail dengan informasi pengiriman adalah ‘Ralali Kargo’ dan dengan button bawah adalah ‘Siap dipickup’	Berhasil
2.	User menekan button ‘Siap dipickup’	Tampilan akan muncul popup modal yang menampilkan informasi alamat pengiriman dan button ‘Lanjutkan’	Berhasil
3.	User menekan button ‘Lanjutkan’	<p>Tampilan akan muncul loading indicator dengan overlay screen warna hitam.</p> <p>Jika berhasil:</p> <p>Tampilan akan diarahkan ke Order detail dengan perubahan warna icon ‘Ralali kargo’ menjadi abu-abu dan button bawah berubah jadi ‘Masukan Nomor Resi’</p> <p>Jika gagal:</p> <p>Tampil popup error message</p>	Berhasil

Test Case 8 – Input Nomor Resi

Aktor: User Seller

Tabel 4.47 Test Case Input Nomor Resi

No.	Action	Expected Result	Status
1.	User menekan atau memilih salah satu order dari order list pada tab ‘Sedang diproses’	Berpindah ke screen order detail dengan informasi pengiriman adalah ‘Ralali Kargo’ yang sudah berstatus	Berhasil

	yang tanpa icon truk atau yang ber-icon truk warna abu-abu	'Siap dipickup' atau pengiriman lain dan dengan button bawah adalah 'Masukan Nomor Resi'	
2.	User menekan button 'Masukan Nomor Resi'	Tampilan akan muncul popup form input nomor resi dan button 'Submit Resi'	Berhasil
3.	User memasukan nomor resi pada form input dan menekan button 'Submit Resi'	Tampilan akan muncul loading indicator dengan overlay screen warna hitam. Jika berhasil: Tampilan akan diarahkan ke Order list	Berhasil
		Jika gagal: Tampil popup error message	

Test Case 9 – View Order Detail

Aktor: User Seller

Tabel 4.48 Test Case View Order Detail

No.	Action	Expected Result	Status
1	User menekan atau memilih salah satu order dari order list pada tab 'Belum diproses'	Berpindah ke screen order detail, yang pertama akan di tampilkan user adalah <i>coachmark</i> cara memproses order setelah itu menampilkan informasi sesuai dengan order yang dipilih. Selain informasi tentang detail order dan status pemesanan 'Belum diproses', juga di tampilkan item-item produk yang ada pada order tersebut, dan juga di paling bawah terdapat button 'Proses Pesanan' yang masih non-aktif	Berhasil
2.	User menekan atau memilih salah satu	Berpindah ke screen order detail dengan informasi pengiriman adalah status 'Sedang	Berhasil

	order dari order list pada tab ‘Sedang diproses’	diproses’ dan juga menampilkan button bawah ‘Siap dipickup’ atau button ‘Masukan nomor resi’	
3.	<i>User</i> menekan atau memilih salah satu order dari order list pada tab ‘Pengiriman’	Berpindah ke screen order detail dengan informasi pengiriman adalah status ‘Dikirim’ dan juga menampilkan nomor resi yang sudah di input serta detail informasi order lainnya dan juga button bawah tidak muncul	Berhasil
4.	<i>User</i> menekan atau memilih salah satu order dari order list pada tab ‘Diterima’	Berpindah ke screen order detail dengan informasi pengiriman adalah status ‘Diterima’ dan juga menampilkan nomor resi yang sudah di input serta detail informasi order lainnya dan juga button bawah tidak muncul	Berhasil
5.	<i>User</i> menekan atau memilih salah satu order dari order list pada tab ‘Settlement’	Berpindah ke screen order detail dengan informasi pengiriman adalah status ‘Settlement’ dan juga menampilkan nomor resi yang sudah di input serta detail informasi order lainnya dan juga button bawah tidak muncul	Berhasil

Test Case 10 – View Profile Detail

Aktor: User Seller

Tabel 4.49 Test Case View Profile Detail

No.	Action	Expected Result	Status
1.	<i>User</i> menekan tab Toko	Tampilan akan diarahkan ke screen profile dan akan menampilkan informasi dari user seperti foto, nama toko, url toko, status toko, dan juga ada 5 button dibawah itu tetapi hanya button ‘Pengaturan Toko’ dan button ‘Keluar’ yang aktif dan berfungsi.	Berhasil
2.	<i>User</i> menekan button ‘Pengaturan	Tampilan akan diarahkan ke screen Profile detail dengan 3 tab yaitu , Infomasi Toko, pengiriman, pembayaran. Secara default tampilan akan diarahkan	Berhasil

	Toko'	ke tab 'Informasi Toko' yang berisikan infomasi tentang user secara lebih detail.	
3.	User menekan tab 'Pengiriman' pada screen profile detail	Tampilan akan menampilkan infomasi tentang PIC gudang, nomor telepon gudang, nomor handphone gudang, serta alamat pengiriman gudang	Berhasil
4.	User menekan tab 'Pembayaran' pada screen profile detail	Tampilan akan menampilkan informasi tentang rekening bank user diantaranya nama bank, cabang bank, nomor rekening, dan nama pemilik rekening	Berhasil
5.	User menekan button 'Keluar' pada screen profile	Tampilan akan diarahkan ke halaman login kembali, jikalau ingin masuk kembali ke dashboard user di harapkan untuk mengisi form yang ada di halaman login	Berhasil

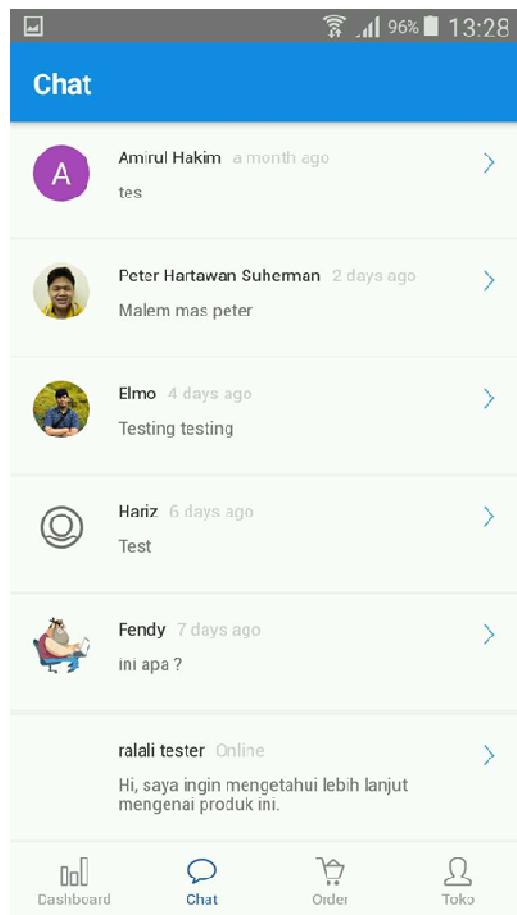
4.6 Evaluasi

Evaluasi diperlukan untuk menilai/tolak ukur kemampuan , berhasil atau tidaknya metode yang digunakan. dengan adanya evaluasi kita menjadi tau apa yang perlu ditingkatkan, diperbaiki , dinilai cukup/kurang. tanpa adanya evaluasi kita tidak tau kemajuan/ kemunduran yang terjadi pada sebuah proses.

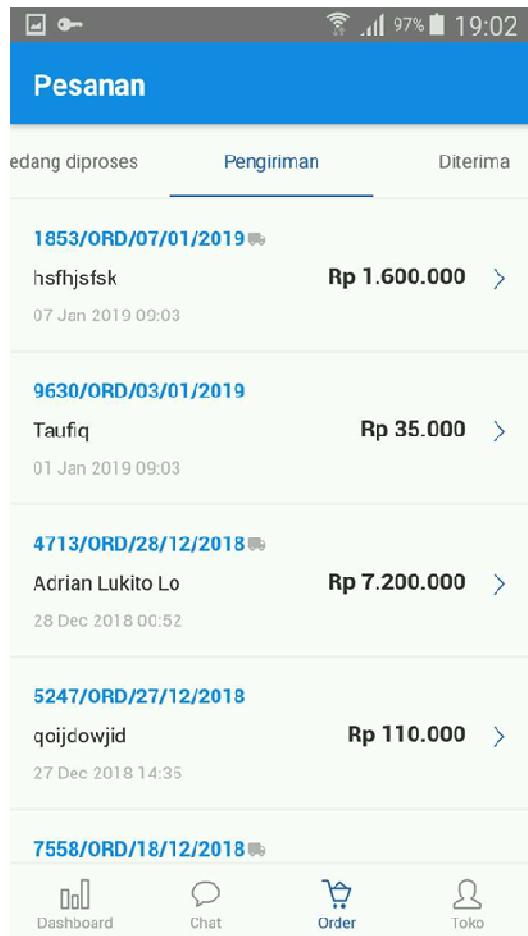
4.6.1 Evaluasi Berdasarkan *Eight Golden Rules*

1. *Strive for Consistency*

Tampilan layar dari aplikasi berbasis mobile *Ralali Seller* dirancang dengan warna, tata letak komponen yang konsisten, serta penggunaan bahasa yang seragam pada setiap halamannya. Contohnya, pada komponen list dari chatlist dan orderlist menampilkan konsistensi penempatan item-item informasinya serta terdapat icon next yang menunjukan bahwa didalam list tersebut masih terdapat tampilan berikutnya.



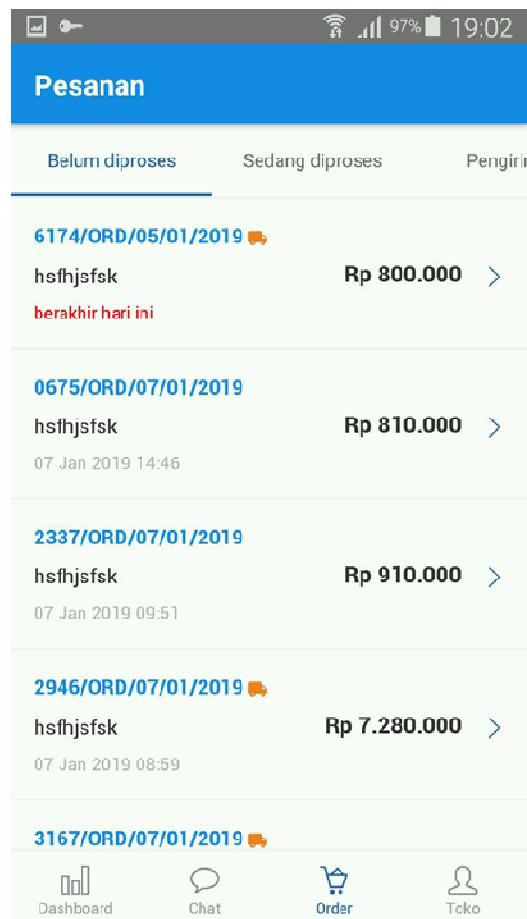
Gambar 4.103 SS Evaluasi Chat



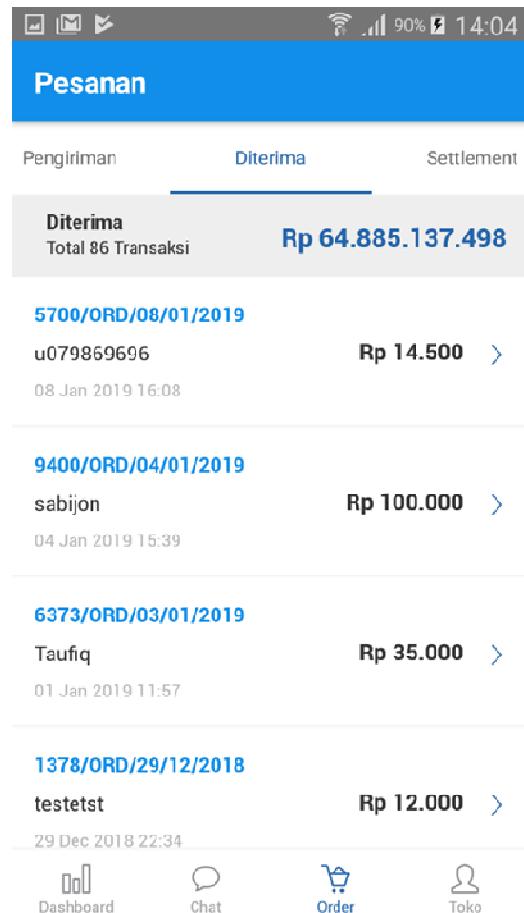
Gambar 4.104 SS Evaluasi Pengiriman

2. *Enable frequent users to use shortcuts*

Aplikasi *Ralali Seller* ini akan digunakan oleh user yang diharapkan mudah dalam pemproses suatu pesanan dari user lainnya, maka dari itu dibutuhkan *shortcut* untuk mempercepat aktifitas user .contohnya, tab bagian bawah navigasi dan tab dalam halaman order diharapkan mempermudah user dalam berpindah halaman.



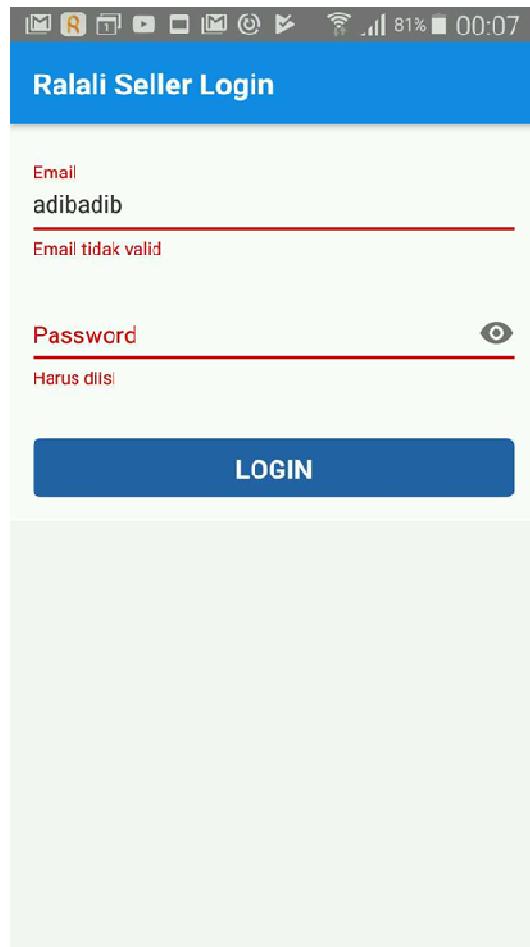
Gambar 4.105 SS Evaluasi Belum Diproses



Gambar 4.106 SS Evaluasi Diterima

3. Offer informative feedback

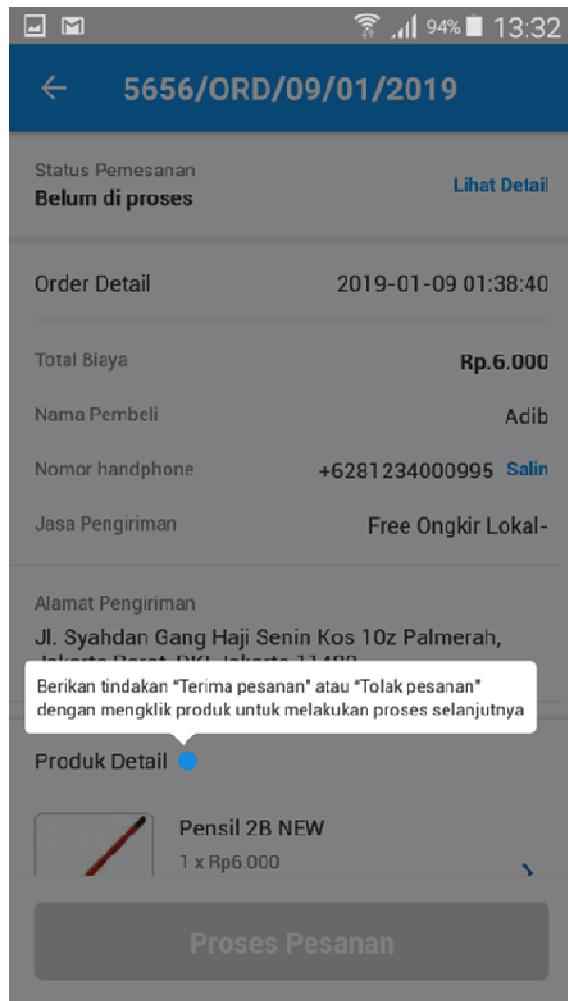
Pada halaman *Login* di aplikasi *Ralali Seller*, pengguna harus memasukkan data pada email dan password. Untuk memberikan informasi kepada pengguna, setiap pengisian sebuah kolom akan ada keterangan validasi ketika data yang dimasukkan oleh *user* tidak sesuai dengan ketentuan. Sehingga, pengguna akan mendapatkan umpan balik dari aplikasi dan dapat menentukan langkah selanjutnya agar dapat menyelesaikan tujuannya.



Gambar 4.107 SS Evaluasi *Login*

4. *Design dialog to yield closure*

Memberikan dialog untuk menghasilkan suatu aksi telah diterapkan pada saat user sampai di halaman order detail dan saat menunggu pengiriman nomor resi.



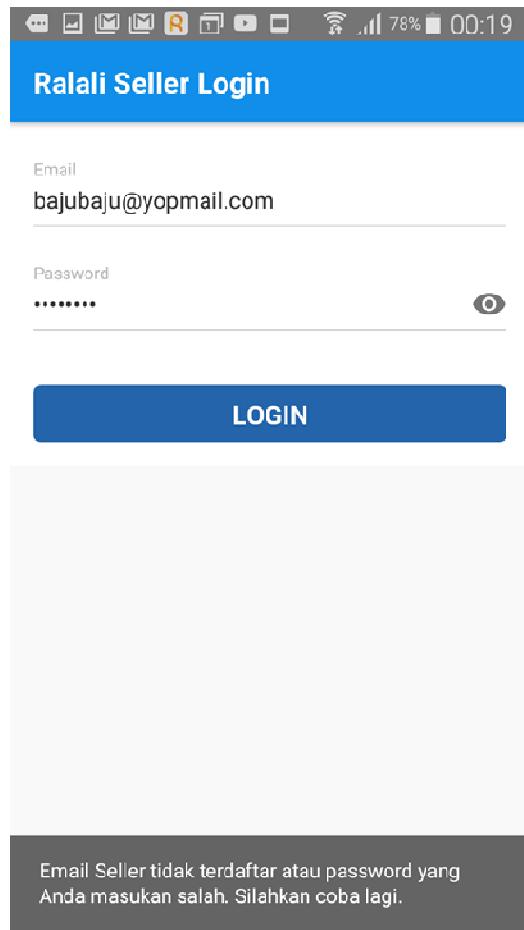
Gambar 4.108 SS Evaluasi Dialog



Gambar 4.109 SS Evaluasi Pemasukan Nomor Resi

5. *Offer simple error handling*

Aplikasi harus bisa menangani kesalahan yang ada dalam penggunaannya baik kesalahan dari user maupun dari sistem . Jika kesalahan dibuat, sistem harus dapat mendeteksi kesalahan dan menawarkan mekanisme sederhana yang dapat dipahami untuk menangani kesalahan. Berikut adalah contoh *error handling* yang ada pada halaman *login* dan halaman awal jika menekan tombol *back* dua kali secara tidak sengaja



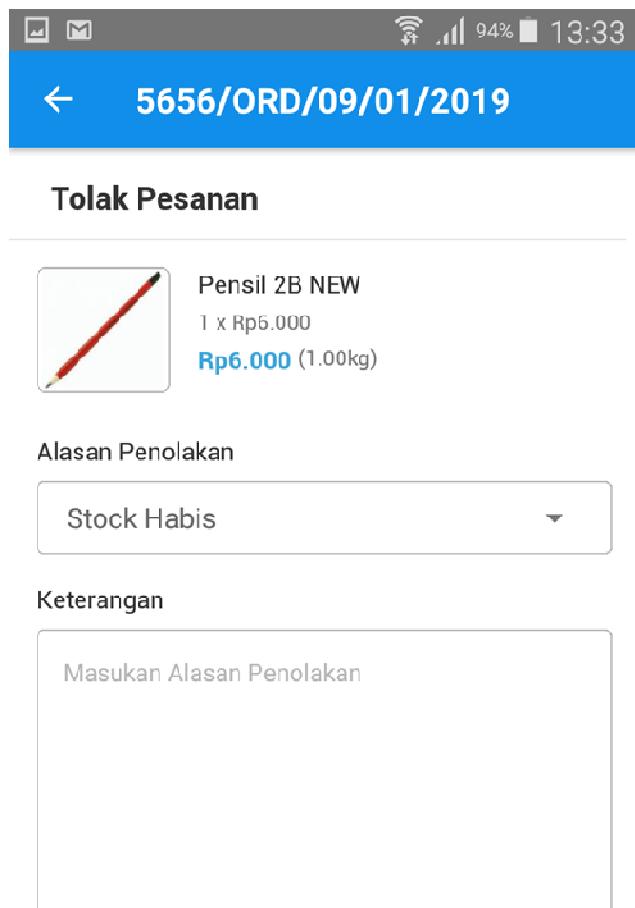
Gambar 4.110 SS Evaluasi *Login*



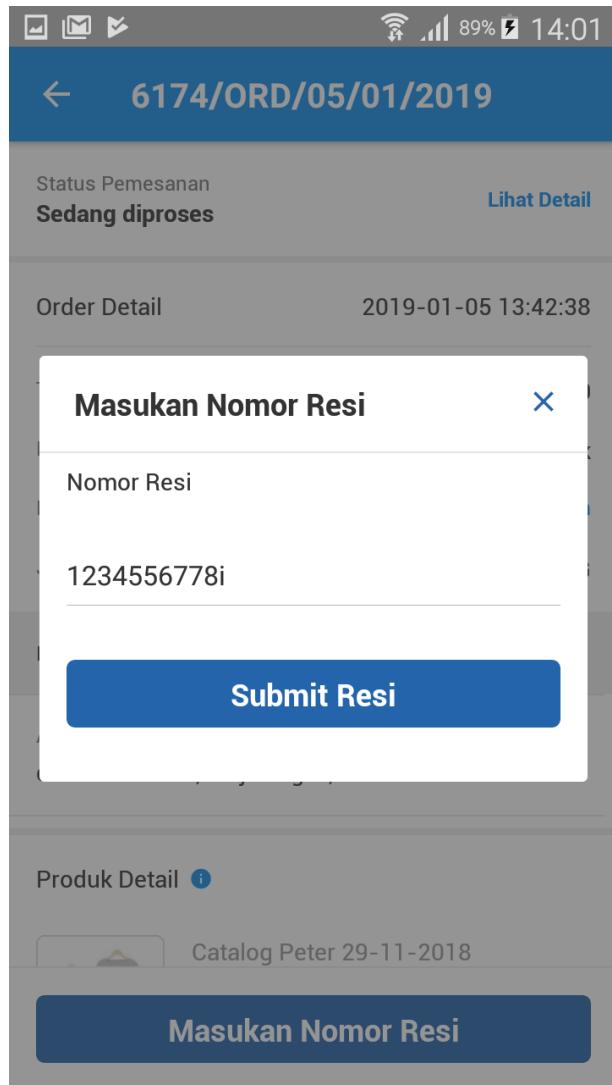
Gambar 4.111 SS Evaluasi Dashboard

6. Permit easy reversal of actions

Aplikasi Ralali Seller mengizinkan pengembalian aksi sederhana, yang bertujuan membatalkan aksi jika user tidak berkenan atas suatu aksi yang telah user lakukan, contohnya ada pada halaman tolak pesanan yang menawarkan tombol batalkan untuk membatalkan aksi tolak pesanan tersebut dan mengembalikan ke halaman sebelumnya. Selain itu juga ada pada halaman order detail pada saat akan memasukan nomor resi user diberikan pop up modal yang menempatkan icon close pada ujung atas guna user membatalkan aksi input nomor resi dan menutup pop up modal.



Gambar 4.112 SS Evaluasi Tolak Pesanan



Gambar 4.113 SS Evaluasi Pengisian nomor resi

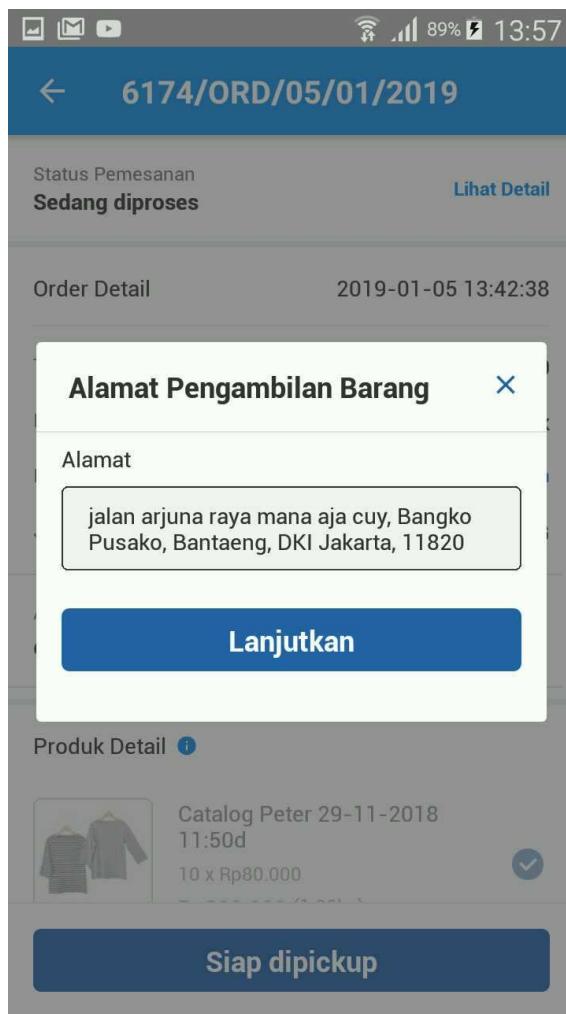
7. *Support internal locus of control*

Aplikasi *Ralali Seller* belum dapat mendukung untuk *user* dapat mengendalikan aplikasi dari dalam sistem untuk aksi-aksi tertentu. Contoh dalam pengaturan bahasa pada profile, *sort*, *filter* dan lain sebagainya

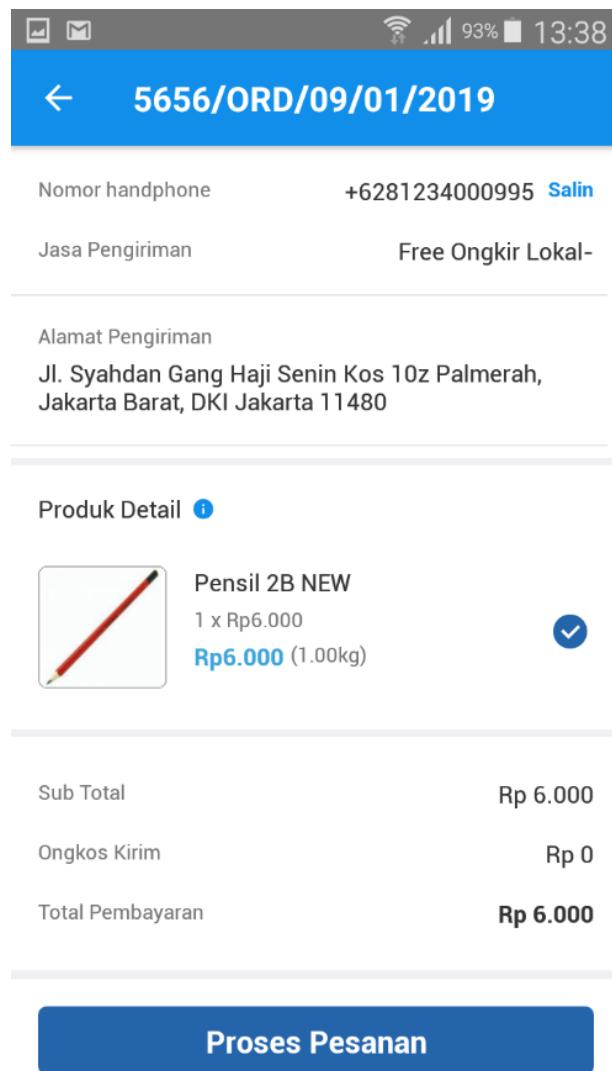
8. *Reduce short-term memory load*

Aplikasi ini mendukung untuk keterbatasan pengingatan manusia dengan informasi tampilan yang pada aksi yang akan dan sudah dilakukan. Contohnya, terdapat informasi berupa icon *check* yang terletak pada halaman order detail pada *section* produk yang

menjelaskan tentang sudah atau tidak nya suatu barang diterima atau ditolak. Dan juga terletak pada pop-up modal ketika menekan tombol ‘siap dipickup’ yang menampilkan alamat pengiriman barang tanpa perlu user melihat kembali alamat pengiriman user pada halaman profile



Gambar 4.114 SS Evaluasi Alamat *Pickup*



Gambar 4.115 SS Evaluasi Proses Pesanan

4.6.2 Evaluasi Berdasarkan 5 Faktor Manusia Terukur

1. *Time to Learn*

Jumlah waktu yang dibutuhkan untuk memahami aplikasi *Ralali Seller* ini diminimalkan dengan desain antarmuka yang sederhana dan tata letak komponen yang konsisten. Dengan ini, *user* dapat menghemat banyak waktu dalam memahami penggunaan dari aplikasi ini.

2. *Speed of Performance*

Aplikasi ini membantu *user* untuk melakukan sebuah aksi secara cepat dan *real time*. Contohnya, pada halaman chat *user* dapat

melakukan aksi chatting secara real time . selain itu juga *user* cukup melakukan membuka order detail dan menerima pesanan dalam hitungan detik dengan aplikasi ini, sehingga pesanan akan segera diproses.

3. *Rate of Errors by Users*

Tingkat kesalahan yang dilakukan oleh *user* diminimalkan dengan adanya bantuan sistem yang mampu menanggulangi kesalahan dalam penggunaan. Aplikasi menyediakan pencegahan kesalahan yang informatif, seperti keterangan validasi yang muncul di bawah kolom teks saat data yang dimasukkan oleh pengguna tidak sesuai dengan ketentuan.

4. *Retention Over Time*

Tampilan desain yang konsisten serta informasi yang efektif dapat membantu *user* dalam keterbatasan daya ingat pada saat melakukan aksi pada aplikasi ini.

5. *Subjective Satisfaction*

Berdasarkan hasil interview, *user* sudah puas dengan fitur-fitur yang terdapat pada aplikasi *Ralali Seller* karena dapat membantu *user* dalam melakukan *order processing* dan *engagement* kepada *user* lainnya .

4.6.3 Evaluasi Berdasarkan *User Interview*

Berikut adalah hasil evaluasi menggunakan metode wawancara yang dilakukan pada tanggal 18 Januari 2019 di *Ralali Office* dengan *user*, yaitu *Team Leader Technology Seller*, *Head of Community Seller*, dan *Seller* .

Tabel 4.50 Evaluasi Berdasarkan *User Interview*

Pertanyaan	Jawaban
Bagaimana pendapat Anda mengenai aplikasi <i>Ralali Seller</i> ?	<i>User 1:</i> Menurut saya aplikasi Ralali Seller sudah berjalan dan mudah untuk digunakan, serta memberikan feedback positif bagi seller.

	<p><i>User 2:</i> Overall bisa membantu seller untuk memprioritaskan hal yang perlu dikerjakan segera, jadi secara langsung dapat meningkatkan experience buyer ketika belanja di Ralali.com</p> <p><i>User 3:</i> Aplikasi Ralali Seller mempermudah para Seller untuk mengelola toko kapanpun dan dimanapun dengan menggunakan gadget mereka</p>
Bagaimana pendapat Anda mengenai desain dari aplikasi <i>Ralali Seller</i> ?	<p><i>User 1:</i> Design dari aplikasi Ralali Seller sudah user-friendly khususnya bagi seller.</p> <p><i>User 2:</i> Secara design clean dan mudah di mengerti</p> <p><i>User 3:</i> pendapat Saya mengenai desain dari aplikasi Ralali Seller yaitu, harusnya warnanya mungkin bisa disamakan dengan warna Ralali.com, jadi tidak biru, <i>Overall</i> Okelahh</p>
Apakah menurut Anda, Fitur pada Ralali Seller mempermudah user dalam <i>order processing</i> dan <i>engagement</i> dengan <i>buyer</i> ?	<p><i>User 1:</i> Saat ini untuk fitur order processing sudah membantu dari sisi kecepatan untuk meningkatkan kecepatan peningkatan keputusan bagi seller, hal ini dinilai dari persentase uninstall yang rendah dan persentase retensi pengguna yang tinggi.</p> <p><i>User 2:</i> Memudahkan, tapi perlu dipertajam untuk push notification, karena bagaimanapun, online dipilih karena kecepatan dalam respon. Tanpa push notification, seller tidak tahu ada hal baru yg masuk.</p> <p><i>User 3:</i> menurut saya, fitur pada <i>Ralali Seller</i> mempermudah user dalam <i>order processing</i> dan <i>engagement</i> dengan <i>buyer</i></p>
Apakah terdapat masalah yang muncul setelah adanya aplikasi	<p><i>User 1:</i> Sejauh ini belum ada masalah yang terjadi dari sisi bisnis, justru feedback positive yang dapat</p>

<i>Ralali Seller?</i>	dilihat dari adanya aplikasi ini.
	<p><i>User 2:</i> Seller masih banyak yg tidak mengerti soal dashboard, karena seller mengira dashboard tersebut memunculkan semua periode (padahal di seller apps hanya menampilkan data diperiode bulan berjalan).</p> <p>Selain itu ketika buka seller apps, terkadang loading timenya lama (blank screen saja)</p>
	<p><i>User 3:</i> masalah yang muncul setelah adanya aplikasi Ralali Seller yaitu terkadang Aplikasinya pada saat dibuka blank dan hanya tampilan putih saja pada layarnya</p>
Menurut Anda, apa puas dengan adanya aplikasi <i>Ralali Seller</i> ?	<p><i>User 1:</i> Menurut saya dengan adanya aplikasi Ralali Seller sangat membantu dan saya puas dengan aplikasi ini</p>
	<p><i>User 2:</i> Jika dikasih nilai 1 s/d 5 (1 paling tidak puas, 5 paling puas), maka nilai saya 3</p> <p>Karena fitur di seller apps, masih banyak yg belum ready, dan banyak seller kita yg menanyakan hal tersebut.</p>
	<p><i>User 3:</i> Cukup puas, dikarenakan hal yang ditunggu lama akhirnya datang juga</p>
Apakah ada saran terkait aplikasi <i>Ralali Seller</i> ?	<p><i>User 1:</i> Saran saya agar aplikasi ini dapat mencangkup lingkup yang lebih luas, melihat trend penggunaan mobile dan mobilitas user kami yang tinggi.</p>

	<p><i>User 2:</i> push notification lebih dimaksimalkan kembali</p>
	<p><i>User 3:</i> Saran dari saya mengenai Ralali Seller App yaitu mungkin dapat ditambahkan untuk menambahkan fitur <i>upload product</i>, dikarenakan banyak seller yang mempertanyakan hal tersebut</p>