BAB I

PENDAHULUAN

A. Latar Belakang

Seiring dengan perkembangan teknologi, kebutuhan masyarakat terhadap layanan pengiriman barang semakin meningkat. Layanan kurir kini menjadi salah satu sektor yang penting dalam mendukung berbagai aktivitas bisnis, terutama di era digital saat ini. Dengan semakin berkembangnya e-commerce, masyarakat cenderung mengandalkan layanan kurir untuk mengirimkan berbagai produk, baik dalam skala kecil maupun besar.

Namun, meskipun berbagai aplikasi pengiriman barang sudah tersedia, beberapa kendala masih sering ditemui, seperti ketidakmampuan aplikasi untuk diakses di berbagai perangkat, ketergantungan pada koneksi internet yang kuat, serta keterbatasan fitur yang ada pada aplikasi yang memerlukan pengunduhan secara khusus di platform tertentu. Oleh karena itu, solusi yang dapat diakses dengan mudah di berbagai perangkat tanpa harus mengunduh aplikasi khusus, dan tetap memberikan pengalaman yang baik bagi pengguna, sangat dibutuhkan.

Progressive Web App (PWA) muncul sebagai solusi inovatif untuk menjawab kebutuhan tersebut. PWA merupakan teknologi web modern yang memungkinkan aplikasi web berperilaku layaknya aplikasi native di perangkat seluler. PWA memiliki kelebihan seperti akses yang mudah melalui peramban web, kemampuan untuk berjalan secara offline, serta konsumsi sumber daya perangkat yang lebih efisien dibandingkan aplikasi native. Dengan PWA, pengguna tidak

perlu lagi melakukan pengunduhan aplikasi dari toko aplikasi, namun tetap mendapatkan fitur dan performa yang menyerupai aplikasi native.

Dalam konteks ini, Aplikasi Shenior Kurir diusulkan sebagai sebuah aplikasi jasa antar barang berbasis PWA. Aplikasi ini dirancang untuk memberikan kemudahan bagi pengguna dalam melakukan pemesanan dan pelacakan pengiriman barang tanpa harus mengunduh aplikasi khusus. Dengan menggunakan teknologi PWA, aplikasi ini diharapkan dapat memberikan pengalaman pengguna yang optimal, terutama dalam hal kecepatan akses, kemampuan offline, dan kemudahan integrasi dengan berbagai perangkat.

Pembangunan aplikasi Shenior Kurir diharapkan tidak hanya mampu memudahkan pengguna dalam mengirimkan barang, tetapi juga dapat menjadi solusi efisien dalam menyelesaikan masalah yang dihadapi oleh layanan kurir konvensional saat ini, seperti pengaturan rute yang kurang optimal, keterbatasan aksesibilitas, dan rendahnya efisiensi operasional. Dengan demikian, aplikasi ini diharapkan mampu meningkatkan kualitas layanan kurir dan memberikan manfaat bagi masyarakat luas.

B. Rumusan Masalah

1. Bagaimana merancang dan membangun aplikasi kurir berbasis PWA yang dapat memberikan pengalaman pengguna (user experience) setara dengan aplikasi native?
2. Bagaimana memastikan aplikasi PWA dapat diakses dan digunakan secara optimal di berbagai platform (desktop, mobile) tanpa memerlukan instalasi aplikasi tambahan?
3. Bagaimana meningkatkan efisiensi proses pemesanan dan pelacakan pengiriman barang melalui aplikasi Shenior Kurir?

C. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Lingkup Fitur Aplikasi: Aplikasi ini hanya akan menyediakan fitur dasar seperti pemesanan layanan pengiriman barang, pelacakan status pengiriman secara real-time, dan pemberitahuan (notifikasi) terkait pengiriman.
2. Ketersediaan Platform: Aplikasi Shenior Kurir dikembangkan sebagai Progressive Web App (PWA) yang diakses melalui peramban (browser) pada perangkat mobile dan desktop. Pengembangan aplikasi native khusus untuk Android dan iOS tidak akan dilakukan.

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan membangun aplikasi Shenior Kurir berbasis Progressive Web App (PWA) yang mampu memberikan pengalaman pengguna setara dengan aplikasi native, namun dengan keunggulan

aksesibilitas langsung melalui peramban tanpa perlu instalasi. Aplikasi ini diharapkan dapat diakses secara optimal di berbagai perangkat, baik mobile maupun desktop, dengan fitur utama seperti pemesanan layanan pengiriman barang, pelacakan real-time, serta notifikasi yang informatif. Selain itu, aplikasi ini juga memanfaatkan fitur offline PWA, memungkinkan pengguna untuk tetap mengakses beberapa fungsi aplikasi meskipun koneksi internet terbatas. Penelitian ini juga bertujuan untuk mengoptimalkan performa aplikasi agar tetap ringan, cepat, dan efisien dalam penggunaan daya serta data.

E. Manfaat Penelitian

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Bagi akademik

   Sebagai bahan referensi bagi penulis lain untuk mengembangkan kemampuan di bidang yang sama.

2. Bagi penulis

   Penulis dapat mengimplementasikan ilmu yang didapat selama proses perkuliahan.

3. Bagi pengembang/pemerintah

   Sebagai media untuk memudahkan dalam mengambil keputusan yang lebih tepat dan terinformasi terkait penentuan lokasi tanah strategis untuk pembangunan perumahan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Progressive Web Apps (PWA)*

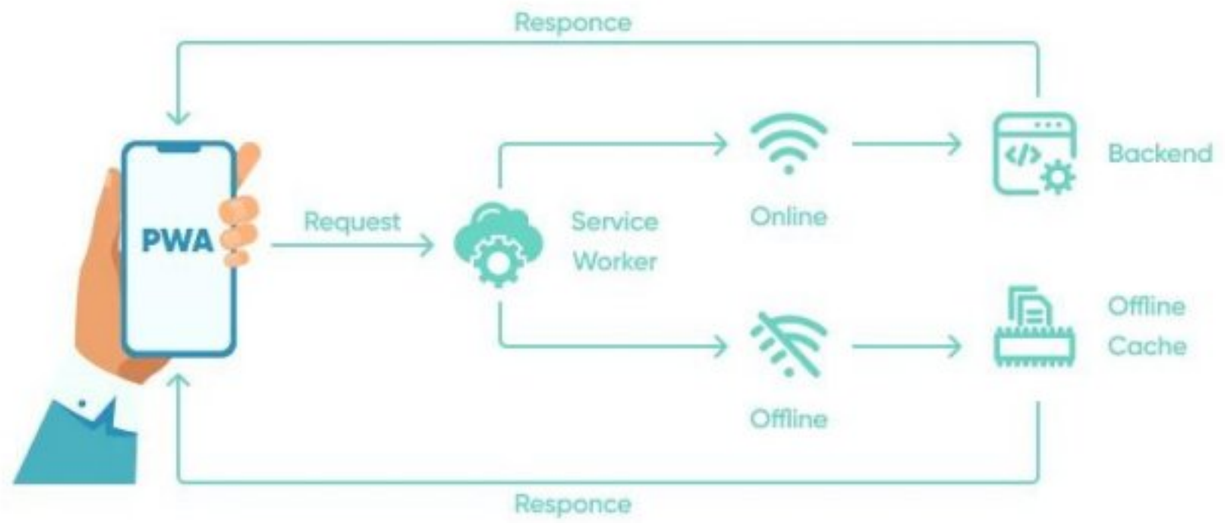


Gambar 2. 1 *Progressive Web Apps*

Progressive Web Apps adalah untuk teknologi untuk membuat aplikasi web yang memberikan tampilan dan nuansa aplikasi web konvensional pada aplikasi asli. Pengembang dapat menyempurnakan aplikasi web tradisional dengan semua komponen dan fungsionalitas aplikasi seluler dengan menggunakan aplikasi web progressive. Ini memperluas pengalaman pengguna (*UX*) yang luar biasa dan meningkatkan kegunaan. Pengembangan aplikasi ini menciptakan pengalaman web dengan fungsionalitas aplikasi. *PWA* adalah aplikasi web yang memanfaatkan kemampuan khusus browser untuk berperilaku lebih seperti aplikasi *native* atau mobile app saat digunakan dengan browser yang compatible (Wargo, 2020). Sedangkan menurut (Ater, 2017) *PWA* adalah aplikasi web generasi baru yang menggabungkan fitur dan sumber daya aplikasi native dengan aplikasi web standar. *PWA* adalah situs web standar yang dibuat dengan teknologi *HTML*, *CSS* dan *Javascript* yang kita kenal dan sukai sebagai web developer, tetapi *PWA* selangkah lebih maju dan

memberi pengguna pengalaman yang lebih baik (*Alan Hume*, 2017). Sebuah *PWA* harus dapat berjalan dengan baik di berbagai perangkat dan ukuran layar, memberikan pengalaman yang mirip dengan aplikasi native, selalu mendapatkan pembaruan terbaru, mudah ditemukan oleh pengguna, dapat diinstal di perangkat pengguna, dan dapat dibagikan dengan pengguna lain.

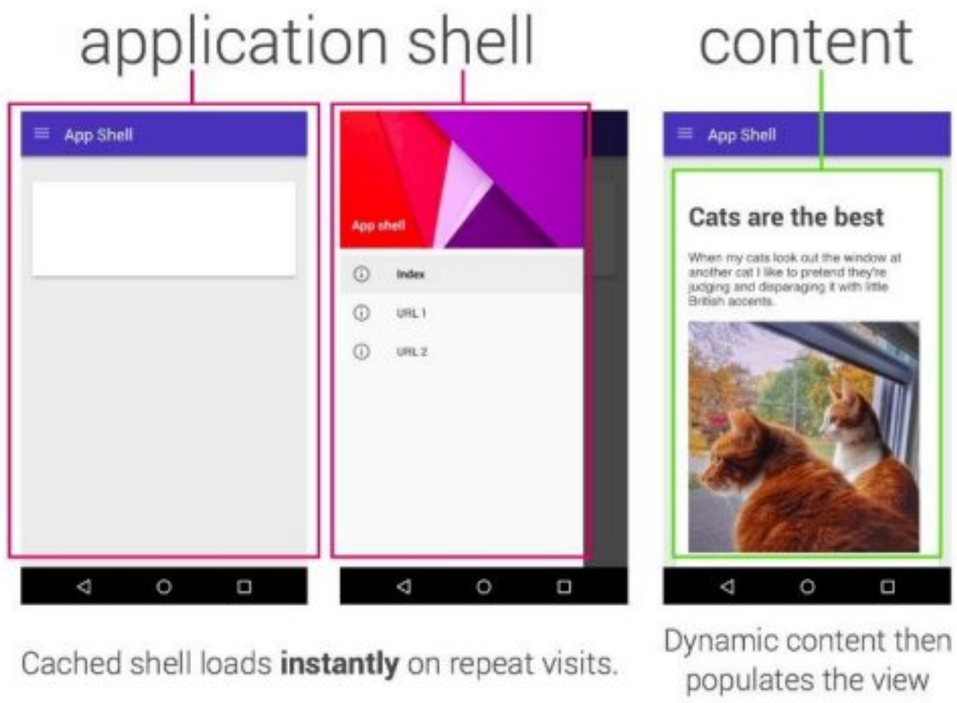
PWA dapat meningkatkan fungsionalitas web app yang sudah ada, baik itu dibangun menggunakan *framework* atau murni *JavaScript* (*Vanilla JS). PWA* dapat diterapkan pada *website* yang dibangun menggunakan *React*, *Angular*, atau *Vue*. Anda bahkan dapat memasukkan fitur ini ke dalam *website* yang sudah aktif tanpa banyak mengubah kode.



Gambar 2. 2 Arsitektur Teknologi *PWA*

Ada beberapa komponen yang harus digunakan saat membangun *PWA*. Beberapa komponen harus digunakan setiap kali membuat *PWA*, sedangkan yang lain adalah opsional. Komponen wajib digunakan setiap kali membuat *PWA*, sedangkan yang opsional adalah komponen yang tidak mempengaruhi kinerja *PWA* tetapi dapat digunakan untuk meningkatkan fiturnya. Berikut komponen wajib yang dimaksud:

a. *Application shell*



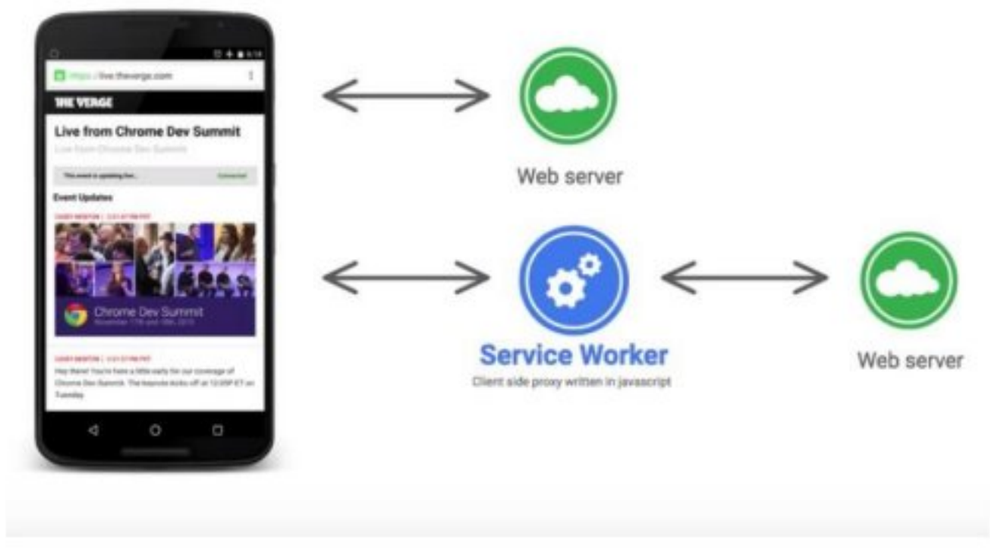
Gambar 2. 3 *Application Shell*

Application Shell, juga disebut sebagai "shell aplikasi", adalah kerangka antarmuka aplikasi yang terdiri dari berbagai komponen halaman dan aset lainnya. Mereka dapat muncul secara instan saat aplikasi dibuka karena telah disimpan lebih dahulu di dalam cache.

b. *Web app manifest*

File JSON yang disebut *Web App Manifest* yang mengatur bagaimana bagaimana aplikasi ditampilkan dan dijalankan di sisi pengguna. Dalam kebanyakan kasus, *file* ini disebut manifest.json. *Browser* akan membaca *file manifest*, mengunduh sumber daya, dan menampilkan konten saat program pertama kali diluncurkan di *browser*.

c. *Service worker*



Gambar 2. 4 Cara kerja *Service Worker*

Service worker adalah *file* yang ditulis dalam bahasa javascript yang berjalan di latar belakang *browser* dan terpisah dari *script* lain pada halaman *web* tersebut. Dengan adanya *service worker*, *browser* dapat menampilkan konten bahkan dalam mode jaringan *offline*.

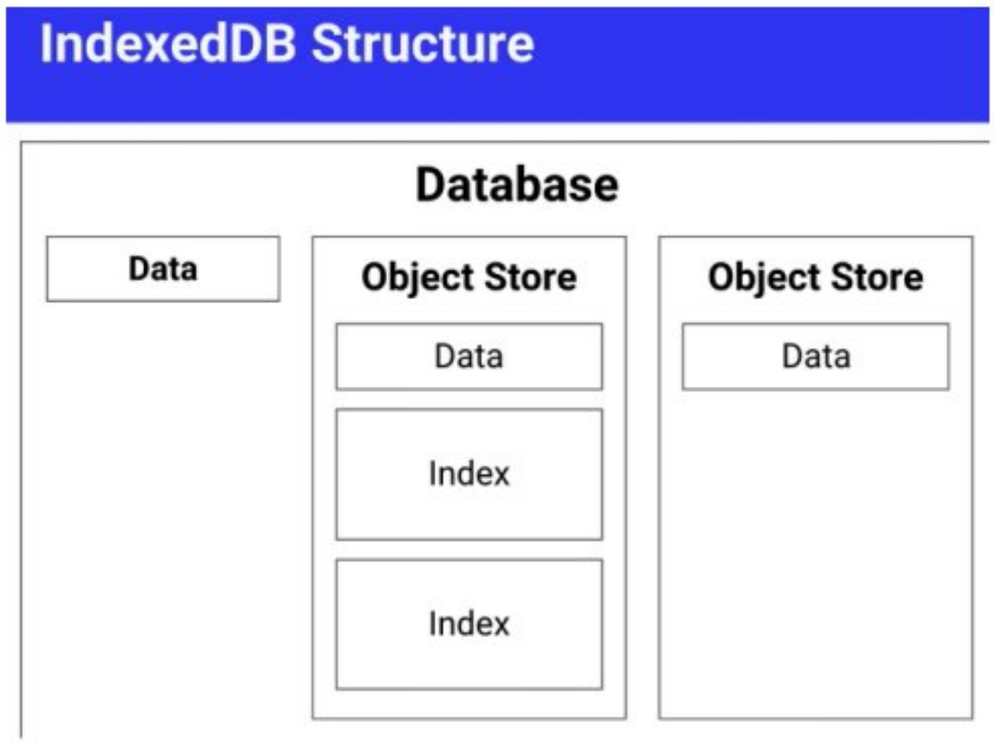
d. *Cache application programming interface*



Gambar 2. 5 *Cache API*

Cache Application Programming Interface (*API*) adalah ruang penyimpanan lokal yang digunakan *browser* untuk menyimpan sumber daya sehingga dapat digunakan kembali selama panggilan berikutnya tanpa membuat permintaan jaringan tambahan.

e. *IndexedDB*

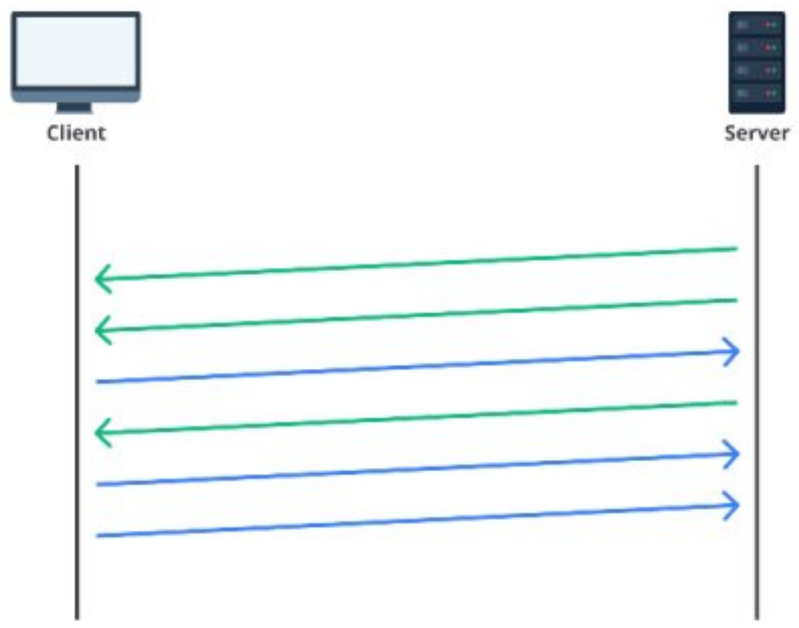


Gambar 2. 6 Struktur *IndexedDB*

Data apa pun dapat disimpan sebagai objek JavaScript di sistem penyimpanan lokal untuk *browser* yang disebut *IndexedDB*, yang berbasis NoSQL. Selain berperan sebagai *cache*, *IndexedDB* juga dapat digunakan untuk menyimpan *item* tertentu.

f. *Web Socket*

Standar baru untuk komunikasi *real-time* di *web* atau aplikasi seluler disebut *Web Socket* yang banyak digunakan saat ini untuk membuat aplikasi obrolan waktu nyata, umpan berita, dan bahkan *game* multipemain daring. *Web Socket* menjadi standar dan komponen HTML5 di dalam *browser* itu sendiri.



Gambar 2. 7 *Web Socket Communication*

Client dan *server* harus selalu terkoneksi agar dapat menerima sinyal komunikasi dari kedua belah pihak saat menggunakan pola komunikasi aktif dua arah (*full-duplex*) seperti ini. Kami dapat membangun komunikasi waktu nyata karena kami selalu terhubung. *Web Socket* dapat digunakan untuk komunikasi waktu nyata, *streaming* langsung, *game* multipemain, dan progres waktu nyata.

g. *Notifications*

Pesan *pop-up* yang dikenal sebagai notifikasi dapat ditampilkan di *gadget*. Situs *web* yang mendukung *PWA* mendukung tampilan notifikasi. Memberi tahu pengguna tentang informasi apa pun sangat penting. Metode efektif untuk mendorong orang kembali ke situs *web* adalah pemberitahuan.

2. Layanan Pengiriman Barang

Layanan pengiriman barang adalah suatu bentuk layanan yang bertujuan untuk mengirimkan barang dari satu tempat ke tempat lain, baik dalam skala lokal, nasional, maupun internasional. Layanan ini melibatkan berbagai proses bisnis yang saling berkaitan, termasuk pemesanan, penjemputan barang, pengiriman, hingga penerimaan barang oleh penerima akhir. Dalam dunia yang semakin terhubung secara digital, layanan pengiriman barang menjadi sangat penting dalam mendukung berbagai sektor ekonomi, terutama dalam industri e-commerce.

1. Proses Layanan Pengiriman Barang

Proses pengiriman barang secara umum terdiri dari beberapa tahap utama:

Pemesanan (*Order*): Pengguna melakukan pemesanan layanan pengiriman barang melalui platform yang disediakan, baik melalui aplikasi mobile, website, atau platform lainnya.

Penjemputan Barang (*Pickup*): Kurir atau penyedia layanan menjemput barang dari pengirim di lokasi yang telah ditentukan.

Pengiriman (*Delivery*): Barang yang telah dijemput akan dikirimkan ke alamat tujuan sesuai dengan rute pengiriman yang telah diatur.

Pelacakan (*Tracking*): Selama proses pengiriman, pengguna dapat memantau status barang melalui fitur pelacakan (*tracking*) *real-time* yang disediakan oleh penyedia layanan.

Penerimaan Barang (*Receipt*): Barang diterima oleh penerima di alamat tujuan, dan status pengiriman diperbarui di sistem layanan.

2. Komponen Utama Layanan Pengiriman Barang

a. Kurir

Kurir adalah tenaga operasional yang bertanggung jawab untuk mengantarkan barang dari pengirim ke penerima. Mereka berperan penting dalam memastikan barang tiba dengan selamat dan tepat waktu di lokasi tujuan.

b. Pusat Distribusi

Pusat distribusi berfungsi sebagai titik transit untuk penyortiran dan pengaturan rute barang sebelum barang dikirim ke tujuan akhirnya. Di pusat distribusi, barang yang datang dari berbagai lokasi dikumpulkan, disortir, dan dikelompokkan sesuai dengan rute dan tujuan pengiriman.

c. Rute Pengiriman

Rute pengiriman harus diatur sedemikian rupa untuk memastikan efisiensi waktu dan biaya. Optimalisasi rute melibatkan penentuan jalur pengiriman yang paling cepat dan hemat biaya, sering kali dibantu oleh algoritma pengaturan rute yang kompleks.

3. Jenis-jenis Layanan Pengiriman Barang

Pengiriman Ekspres: Layanan pengiriman yang menawarkan kecepatan lebih cepat dengan tarif yang lebih tinggi. Layanan ini biasanya dipilih untuk pengiriman barang yang mendesak atau membutuhkan waktu sampai yang cepat.

Pengiriman Reguler: Layanan ini menyediakan pengiriman barang dengan waktu yang lebih fleksibel dan biaya yang lebih terjangkau. Layanan ini sering digunakan untuk pengiriman yang tidak mendesak.

Pengiriman Same Day: Layanan pengiriman yang memastikan barang dikirimkan dan diterima oleh penerima di hari yang sama. Layanan ini banyak digunakan di kota-kota besar dengan sistem pengiriman yang cepat.

Pengiriman Internasional: Pengiriman barang antarnegara yang melibatkan proses bea cukai dan pengaturan logistik lintas negara.

4. Manajemen Layanan Pengiriman Barang

Manajemen pengiriman barang melibatkan pengelolaan sumber daya, tenaga kerja, serta teknologi untuk memastikan bahwa barang dapat dikirimkan dengan efisien dan tepat waktu. Faktor-faktor yang diperhatikan dalam manajemen pengiriman barang antara lain:

Efisiensi Waktu dan Biaya: Mengoptimalkan rute pengiriman dan meminimalkan biaya operasional untuk menjaga keuntungan.

Pelayanan Pelanggan: Menyediakan layanan yang responsif dan ramah pelanggan, terutama dalam hal menangani keluhan atau masalah terkait pengiriman.

Kualitas Pengiriman: Memastikan barang yang dikirim tidak mengalami kerusakan selama proses pengiriman dan tiba dalam kondisi baik.

5. Teknologi dalam Layanan Pengiriman Barang

Dengan perkembangan teknologi, layanan pengiriman barang kini didukung oleh berbagai teknologi digital seperti:

Sistem Pelacakan *Real-time*: Menggunakan teknologi *GPS*, pengguna dapat melacak posisi barang yang sedang dikirim secara real-time. Ini meningkatkan transparansi dan kepercayaan pengguna terhadap layanan yang diberikan.

Automasi Rute Pengiriman: Algoritma yang secara otomatis menentukan rute terbaik untuk pengiriman barang. Teknologi ini memanfaatkan data real-time seperti kondisi lalu lintas dan jarak tempuh untuk mengoptimalkan efisiensi.

Aplikasi *Mobile* dan Web: Penggunaan aplikasi berbasis mobile dan web untuk memudahkan pengguna dalam memesan layanan pengiriman, melacak status pengiriman, serta mengatur jadwal pengambilan dan pengiriman barang.

6. Tantangan dalam Layanan Pengiriman Barang

Beberapa tantangan utama dalam layanan pengiriman barang antara lain:

Logistik dan Infrastruktur: Di beberapa wilayah, terutama di daerah terpencil, keterbatasan infrastruktur dapat menjadi hambatan dalam pengiriman barang secara efisien.

Keamanan dan Risiko: Pengiriman barang berisiko mengalami kehilangan atau kerusakan selama proses pengiriman. Oleh karena itu, penyedia layanan harus menerapkan sistem keamanan yang baik untuk meminimalkan risiko.

Kepuasan Pelanggan: Pelanggan memiliki harapan yang tinggi terhadap kecepatan, keakuratan, dan keamanan pengiriman barang. Penyedia layanan harus selalu berusaha untuk memenuhi ekspektasi tersebut agar dapat mempertahankan loyalitas pelanggan.

3. *SHENIOR KURIR*



Gambar 2. 8 Shenior Kurir

Pada akhir tahun 2019, dunia menghadapi tantangan besar dengan munculnya wabah penyakit yang disebabkan oleh virus corona, yang lebih dikenal sebagai *COVID*-19. Sejak pertama kali terdeteksi di kota *Wuhan*, Provinsi *Hubei*, *Tiongkok*, pada awal tahun 2020, *COVID*-19 segera menjadi perhatian publik global. Dengan ribuan jiwa yang meninggal akibat virus ini, banyak negara, termasuk Indonesia, terpaksa merespons secara serius. Pada tanggal 11 Maret 2020, Organisasi Kesehatan Dunia (*WHO*) menetapkan wabah ini sebagai pandemi global.

Pandemi *COVID*-19 telah memberikan dampak signifikan terhadap kondisi ekonomi dan sosial di Indonesia sejak akhir tahun 2019. Perekonomian di seluruh wilayah Indonesia terancam, dan banyak daerah mengalami kondisi yang lebih buruk dibandingkan sebelumnya. Dalam

upaya untuk menekan angka penyebaran virus, pemerintah Indonesia mengambil langkah-langkah agresif.

Di tengah situasi ini, banyak pekerja mengalami pemutusan hubungan kerja (*PHK*), terutama di kota Parepare. Melihat peluang dari meningkatnya angka pengangguran dan kebutuhan masyarakat, Supriyadi mendirikan Shenior Kurir pada bulan November 2020. Usaha ini didirikan untuk memberikan kesempatan kerja kepada para pengangguran dan membantu masyarakat yang terpaksa tinggal di rumah selama masa pandemi.

Seiring dengan perkembangan usaha, Supriyadi—yang dikenal juga dengan nama Beth—mengambil inisiatif untuk membuka cabang di Mamuju dan Majene. Shenior Kurir menyediakan berbagai layanan, termasuk pengantaran makanan dan minuman, belanja di pasar, serta layanan antar-jemput. Dengan demikian, Shenior Kurir tidak hanya berkontribusi dalam menciptakan lapangan kerja, tetapi juga membantu memenuhi kebutuhan masyarakat di tengah tantangan yang dihadapi akibat pandemi.

4. *Node JS*

Node.js adalah runtime environment bersifat *open-source* dan *cross-platform* yang memungkinkan Anda menjalankan kode JavaScript di luar browser. Node.js dibangun di atas ***V8 JavaScript engine*** milik *Google Chrome*, yang membuat eksekusi kode JavaScript menjadi sangat cepat dan efisien.

Dengan *Node.js*, *JavaScript* tidak hanya dipakai untuk membuat antarmuka pengguna di sisi klien, tetapi juga dapat digunakan untuk membangun server, aplikasi backend, *API*, hingga alat otomasi.

Fitur *Node JS*

a. *Asynchronous* dan *Non-blocking I/O*

   Semua operasi *I/O* (misalnya membaca *file*, *query database*, komunikasi jaringan) bersifat *non-blocking*. Artinya, *Node.js* tidak akan menunggu satu operasi selesai sebelum memulai operasi lainnya. Hal ini membuat *Node.js* sangat efisien dan cocok untuk aplikasi yang membutuhkan skalabilitas tinggi.

b. *Single-threaded Event Loop*

   Node.js menggunakan satu thread utama dengan event loop untuk menangani banyak koneksi secara bersamaan. Model ini lebih ringan dibandingkan pendekatan tradisional berbasis *thread pool*.

c. *NPM* (*Node Package Manager*)

   Node.js dilengkapi dengan *NPM*, sistem manajemen paket terbesar di dunia, yang menyediakan ribuan library siap pakai untuk mempermudah pengembangan aplikasi.

d. Cepat dan Efisien

   Karena menggunakan *V8* engine yang dioptimasi, eksekusi JavaScript di *Node.js* sangat cepat. Selain itu, *non-blocking I/O* membuat penggunaan sumber daya menjadi lebih efisien.

Kelebihan *Node JS*

a. Performanya tinggi untuk aplikasi *real-time* dan banyak koneksi sekaligus.
b. Bahasa pemrograman yang seragam, yaitu *JavaScript*, di sisi klien maupun *server*.
c. Ekosistem paket yang luas melalui *NPM*.
d. Komunitas aktif yang terus mengembangkan modul baru.
e. Mudah dipelajari bagi developer *JavaScript*.

Kekurangan *Node JS*

a. *REST API* dan *Backend Web*

   Misalnya *Express.js* untuk membuat *API server*.

b. Aplikasi *Real-time*

   Seperti chat *server* (menggunakan *Socket.IO*).

c. *Alat Command Line* (*CLI*)

   Banyak *tools* modern dibuat dengan *Node.js*.

d. *Proxy* atau *Middleware*

   Untuk menangani permintaan ke berbagai layanan backend.

Node.js adalah solusi modern untuk membuat aplikasi backend dan *real-time* berbasis *JavaScript* dengan performa tinggi. Berkat arsitektur *non-blocking* dan ekosistem paket yang luas, *Node.js* menjadi pilihan populer bagi banyak developer di berbagai skala proyek.

5. *Bun*

Bun adalah *runtime JavaScript* modern, *all-in-one*, yang dirancang untuk menjadi **sangat cepat dan efisien**. Bun dikembangkan oleh *Jarred Sumner*, dan mulai populer sejak diperkenalkan karena performanya yang diklaim **lebih cepat daripada *Node.js* dan *Deno***.

Bun tidak hanya sekadar *runtime JavaScript*, tetapi juga mencakup:

a. *JavaScript/TypeScript runtime*

b. Bundler (penggabung *file*)

c. *Transpiler* (konversi *TypeScript* atau *JSX* ke *JavaScript*)

d. Package manager (pengganti *npm*/*yarn*)

Dengan satu alat, developer bisa menjalankan, membundel, dan mengelola dependensi proyek *JavaScript*/*TypeScript* secara terpadu.

Berikut beberapa fitur yang menjadi keunggulan Bun:

a. Sangat Cepat

Bun ditulis dalam bahasa pemrograman *Zig* dan menggunakan *JavaScriptCore* engine (*engine JavaScript* milik *WebKit*) yang membuat eksekusi kode lebih cepat dibandingkan *V8* (yang digunakan *Node.js*).

b. *Bundler* dan *Transpiler* Bawaan

Bun bisa membundel *JavaScript*/*TypeScript*/*JSX* tanpa memerlukan *Webpack*, *esbuild*, atau *Babel*

c. *Package Manager* Bawaan

Bun memiliki *package* manager sendiri (*bun install*) yang kompatibel dengan *package.json* dan lebih cepat dibandingkan *npm* atau *yarn*.

d. *Built-in Tooling*

Termasuk testing framework (*bun test*) dan *support native TypeScript*.

e. Kompatibilitas dengan *API Node.js*

Banyak modul *Node.js* dapat dijalankan di Bun.

Kelebihan *Bun*

a. Performa sangat cepat (instalasi paket, menjalankan skrip, *bundling*).

b. Semua alat terintegrasi dalam satu *executable* (*bun*).

c. *Transpiler native TypeScript* dan *JSX* tanpa konfigurasi tambahan.

d. *CLI* sederhana dan intuitif.

e. Ukuran bundling kecil dan cepat dijalankan.

Kekurangan *Bun*

a. Belum semua modul *Node.js* kompatibel 100%, meskipun terus ditingkatkan.

b. Masih relatif baru, ekosistemnya belum sebesar *Node.js*.

c. Dokumentasi masih berkembang.

d. Belum mendukung semua *API* standar *Node.js* (beberapa modul native belum tersedia).

Bun adalah runtime modern untuk *JavaScript* dan *TypeScript* yang menggabungkan kecepatan tinggi dengan tooling terintegrasi (*runtime*, *bundler*, *transpiler*, dan *package manager*).

Walaupun masih dalam tahap perkembangan, *Bun* sudah menjadi alternatif menjanjikan bagi *developer* yang menginginkan performa optimal dan pengalaman pengembangan yang lebih sederhana.

6. *Typescript*

TypeScript adalah bahasa pemrograman open-source yang dikembangkan oleh *Microsoft*. *TypeScript* merupakan **superset dari *JavaScript***, artinya semua kode *JavaScript* valid juga merupakan kode *TypeScript*. Namun, *TypeScript* menambahkan fitur utama yang tidak dimiliki *JavaScript* secara native: ***static typing*** (pengetikan statis).

Dengan *TypeScript*, Anda bisa menentukan tipe data variabel, parameter fungsi, dan nilai kembalian fungsi. Hal ini membuat kode lebih mudah dipelihara, terbaca, dan lebih aman dari bug.

Fitur Utama *TypeScript*

a. *Static Typing*

   Anda bisa mendefinisikan tipe data secara eksplisit sehingga banyak kesalahan dapat terdeteksi saat proses kompilasi, sebelum kode dijalankan.

b. *Type Inference*

   Jika tipe data tidak ditentukan secara eksplisit, *TypeScript* tetap bisa mencoba "menebak" tipe data secara otomatis.

c. *Interface* dan *Type Alias*

   Menyediakan cara mendefinisikan bentuk objek dan kontrak data.

d. Enum

   Mendefinisikan kumpulan nilai konstan.

e. *Generics*

Membuat fungsi atau *class* yang fleksibel dengan tipe data yang tetap aman.

f. Support Modern *JavaScript*

TypeScript mendukung semua fitur *ECMAScript* terbaru dan akan dikompilasi menjadi *JavaScript* versi yang kompatibel dengan browser.

Kelebihan *Typescript*

a. Deteksi error lebih awal saat proses kompilasi.

b. Kode lebih jelas dan terdokumentasi melalui deklarasi tipe.

c. Skalabilitas lebih baik, cocok untuk proyek besar.

d. *Tooling* yang kuat (*auto-complete*, *refactoring*, navigasi kode) di editor modern seperti *Visual Studio Code*.

e. Kompatibel dengan *JavaScript*, sehingga mudah migrasi bertahap.

Kekurangan *Typescript*

a. Perlu proses kompilasi menjadi *JavaScript* sebelum dijalankan.

b. Sedikit tambahan kurva belajar, terutama jika belum familiar dengan konsep *typing*.

c. Beberapa library *JavaScript* tidak memiliki tipe deklarasi lengkap, meski sekarang sudah banyak diperbaiki melalui *DefinitelyTyped*.

TypeScript adalah pilihan populer untuk pengembangan aplikasi skala menengah hingga besar karena pengetikan statis yang kuat, deteksi error lebih dini, dan pengalaman pengembangan yang lebih baik. Jika Anda sudah menggunakan *JavaScript*, beralih ke *TypeScript* bisa dilakukan secara bertahap tanpa perlu memulai dari awal.

7. *Javascript*

JavaScript adalah bahasa pemrograman tingkat tinggi, dinamis, dan berbasis prototipe yang terutama digunakan untuk membuat website interaktif. Awalnya dikembangkan oleh *Netscape* pada tahun 1995, *JavaScript* kini menjadi salah satu bahasa paling populer di dunia pengembangan web.

JavaScript berjalan di sisi klien (*browser*), tetapi dengan hadirnya runtime seperti *Node.js*, *JavaScript* juga banyak digunakan di sisi server.

Fitur Utama *Javascript* :

a. Interaktif dan Dinamis

JavaScript memungkinkan halaman web merespons tindakan pengguna secara *real-time*, misalnya validasi form, animasi, dan manipulasi elemen *HTML*.

b. *Interpreted Language*

JavaScript dieksekusi langsung oleh browser tanpa proses kompilasi.

c. *Event-driven*

JavaScript menggunakan event (klik, *hover*, *submit*, dll.) untuk memicu logika tertentu.

d. Multiplatform

Dapat berjalan di berbagai browser, sistem operasi, dan perangkat.

e. Object-Oriented dan Functional

Mendukung paradigma pemrograman berbasis objek dan fungsional.

Kelebihan *JavaScript*

a. Mudah Dipelajari

   Sintaks *JavaScript* relatif sederhana dibandingkan banyak bahasa lain.

b. Didukung Semua *Browser* Modern

   Tanpa perlu plugin tambahan.

c. Ekosistem *Library* yang Besar

   Misalnya *jQuery*, *React*, *Angular*, *Vue*.

d. Digunakan di *Frontend* dan *Backend*

   Dengan *Node.js*, *JavaScript* menjadi *full-stack*.

e. Komunitas Aktif

   Tersedia banyak tutorial, dokumentasi, dan forum diskusi.

Kekurangan *JavaScript*

a. Dinamisme Tipe Data

   Karena bersifat *loosely typed, error runtime* lebih sering terjadi.

b. Keamanan

   Rentan disalahgunakan (*XSS*, *CSRF*) jika tidak ditulis dengan benar.

c. Ketergantungan *Browser*

   Kadang perilaku *JavaScript* berbeda di masing-masing browser.

JavaScript digunakan untuk:

a. Interaktivitas Website

   Slider, *popup*, validasi form.

b. *Single Page Application* (*SPA*)

   Dengan *framework* seperti *React*, *Angular*, *Vue*

c. *Server-side Development*

Menggunakan *Node*.js.

d. *Mobile App Development*

Contoh: *React Native*.

e. *Desktop App Development*

Contoh: *Electron*

Seiring waktu, *JavaScript* terus berevolusi melalui standar *ECMAScript* (*ES*). Versi-versi terbaru seperti ***ES6* (*ECMAScript* 2015)** membawa banyak fitur modern, seperti:

a. *let & const*

b. *Arrow function*

c. *Class*

d. M*odules*

e. *Promise & async/await*

JavaScript adalah tulang punggung web modern. Dengan dukungan luas di *browser* dan *server*, *JavaScript* telah menjadi salah satu bahasa pemrograman terpenting dan paling banyak digunakan saat ini.

8. *Socket.IO*

Socket.IO adalah *library JavaScript* yang digunakan untuk membangun komunikasi *real-time* antara klien (*browser*, aplikasi *desktop*, atau *mobile*) dan server melalui *WebSocket* atau protokol *fallback* lain (seperti *long polling*).

Socket.IO membuat pengembangan aplikasi *real-time* menjadi jauh lebih sederhana dibandingkan menggunakan *WebSocket* mentah. *Library* ini menyediakan *API* yang mudah digunakan, dukungan *fallback* otomatis, dan fitur tambahan seperti broadcast dan rooms.

Socket.IO berjalan di ***Node.js*** pada sisi *server*, dan tersedia *library* klien untuk *browser*, *Node.js*, maupun platform lain.

Fitur Utama *Socket.IO*

a. *Real-time Bidirectional Communication*

   Klien dan *server* bisa saling mengirim data kapan saja secara instan.

b. *Auto-reconnection*

   Secara otomatis mencoba menyambung kembali jika koneksi terputus.

c. *Multiplexing*

   Mendukung *namespace*, yaitu memisahkan beberapa saluran komunikasi dalam satu koneksi.

d. *Rooms*

   Mendukung pengelompokan koneksi dalam "*rooms*" untuk broadcast ke grup tertentu.

e. *Fallback Mechanism*

Jika *WebSocket* tidak didukung, *Socket.IO* otomatis menggunakan metode lain seperti *HTTP long polling*.

Kelebihan *Socket.IO*

a. *API* yang Mudah Digunakan

   Lebih sederhana dibandingkan *WebSocket* murni.

b. Fallback Otomatis

   Tidak perlu pusing kompatibilitas browser.

c. Mendukung *Broadcast*

   Dapat mengirim pesan ke semua klien atau klien tertentu.

d. Event-based Communication

   Menggunakan event custom untuk komunikasi yang lebih fleksibel.

e. Komunitas Aktif dan Dokumentasi Lengkap

Kekurangan *Socket.IO*

a. Lebih Berat daripada *WebSocket* Murni

   Karena ada lapisan tambahan (*fallback*, *protocol negotiation*).

b. Bergantung pada *Library* Khusus

   Klien harus menggunakan *library Socket.IO* juga.

c. Versi *Server* dan Klien Harus Kompatibel

   Kadang versi berbeda tidak bisa saling terhubung.

Socket.IO banyak digunakan dalam aplikasi:

a. Chat *real-time*

b. Notifikasi instan

c. *Live updates* (misalnya *dashboard data*)

d. Kolaborasi dokumen (contoh: *Google Docs clone*)

e. *Game multiplayer* berbasis *web*

Socket.IO adalah solusi powerful untuk membangun aplikasi *real-time* berbasis *event* dengan sedikit usaha tambahan dibandingkan *WebSocket* mentah. Dengan fitur *fallback* otomatis, reconnect, dan broadcast, *Socket.IO* menjadi pilihan populer untuk chat, notifikasi, dan aplikasi interaktif modern.

9. *HTML*



Gambar 2. 9 *Hypertext Markup Language*

Hypertext Markup Language atau *HTML*, adalah bahasa *markup* standar industri untuk membangun halaman *web*. Ini menawarkan kumpulan *tag* dan properti yang menentukan pengorganisasian halaman *web*, konten, dan format halaman *web*. Untuk merepresentasikan elemen pada halaman *web*, *HTML* menggunakan struktur hierarkis yang dikenal sebagai *Document Object Model* (*DOM*). Teks, foto, tautan, multimedia, dan elemen lainnya semuanya dapat dimuat dalam setiap elemen, yang dibatasi oleh *tag* pembuka dan penutup.

Istilah "*HyperText*" mengacu pada kemampuan *HTML* untuk membuat tautan yang memungkinkan pengguna dengan mudah dan cepat berpindah dari satu situs *web* ke situs *web* lainnya (Duckett & Wiley, 2011). *HyperText*

Markup Language, juga dikenal sebagai HTML, adalah bahasa global *World Wide Web*. Seperti yang tersirat dari namanya *HyperText Markup Language*, HTML adalah bahasa *markup* bukan bahasa pemrograman (Hartl & Donahoe, 2016).

Dokumen yang dibuat dengan menggunakan bahasa bahasa markup seperti *HTML* memiliki bagian yang direnderpada hasil akhir serta bagian yang menginstruksikan program rendering cara menginterpretasikan teks yang tersisa. Proses mengubah dokumen teks yang berisi teks *HTML* menjadi, misalnya, representasi visualnya di layar disebut sebagai "*rendering*" (Nematrian, 2020).

Perkembangan *HTML* sangat pesat. Dari permulaan *HTML* 1.0 yang sederhana, yang mencakup elemen dasar seperti judul dan paragraf, hingga penambahan tabel dan gambar pada *HTML* 2.0, dan akhirnya ke *frame* dan dukungan *form* yang disempurnakan pada *HTML* 3.2. Dengan *CSS*, konten dan presentasi sekarang dapat dipisahkan dalam *HTML* 4.01, yang juga meningkatkan komponennya. Dengan elemen semantiknya, dukungan multimedia, bentuk yang lebih baik, fitur aksesibilitas, dan kemampuan *mobile-friendly*, *HTML5* menandai periode transformasi.

Versi *HTML* terbaru, yang dikenal sebagai *HTML5*, digunakan untuk membuat dan mengatur dokumen di internet. Ini mencakup elemen dan fitur baru yang memungkinkan pembuatan halaman *web* yang lebih dinamis dan menarik, termasuk dukungan untuk audio dan video, kanvas untuk grafik dan

animasi, dan peningkatan kompatibilitas dengan perangkat seluler (Mclaughlin, 2011).

Fungsionalitas baru yang diperkenalkan oleh *HTML5* memungkinkan pembuatan halaman *web* yang lebih dinamis dan interaktif, yaitu:

a. Dukungan audio dan video.
b. Kanvas visual dan animasi.
c. Dukungan yang ditingkatkan untuk *mobile*..
d. *Semantic Element* baru termasuk *nav*, *article*, *sections* dan *headers*.
e. Kontrol untuk form, yaitu tanggal, waktu, email, pencarian, dll.
f. Penyimpanan *web* untuk menyimpan informasi secara lokal.
g. Fitur *drag* and *drop*.

10. *CSS*

CSS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten *web* di layar, printer, sintesis ucapan, pembaca layar, dan jendela obrolan. *CSS* digunakan oleh semua *browser* pada semua ukuran layar di semua jenis perangkat *IoT*, termasuk ponsel, komputer, video game, televisi, jam tangan, dan konsol otomatis (Meyer & Weyl, 2014). Sedangkan menurut (Duckett, 2022) *CSS* adalah bahasa pemrograman *web* yang digunakan untuk mengendalikan tampilan halaman *web*. (Castro, 2022) menjelaskan bahwa *CSS* memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. *CSS* terdiri dari dua bagian utama, yaitu deklarasi dan selektor. Deklarasi adalah bagian yang menentukan properti dari suatu halaman *web*. Selektor adalah bagian yang menentukan elemen *web* mana yang akan diatur

oleh deklarasi. *CSS* juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. Beberapa properti *CSS* yang umum digunakan antara lain:

a. Ukuran: properti ini digunakan untuk mengatur ukuran halaman *web*, seperti ukuran font, ukuran gambar, dan ukuran jarak antar elemen.
b. Warna: properti ini digunakan untuk mengatur warna halaman *web*, seperti warna font, warna latar belakang, dan warna *border*.
c. Font: properti ini digunakan untuk mengatur font elemen *web*, seperti jenis font, ukuran font, dan gaya font.
d. Tata letak: properti ini digunakan untuk mengatur tata letak elemen *web*, seperti posisi elemen, ukuran elemen, dan jarak antar elemen.

11. *Unified modelling language*

UML adalah serangkaian alat yang digunakan untuk menyederhanakan suatu sistem atau perangkat lunak berbasis objek. Singkatan dari *Unified Modelling Language*, *UML* juga berperan dalam mempermudah pengembangan aplikasi secara berkelanjutan. Sistem atau aplikasi yang tidak didokumentasikan dengan baik seringkali menghambat proses pengembangan karena memaksa para pengembang untuk melakukan penyelidikan dan memahami kode program yang kompleks. *UML* juga berfungsi sebagai alat untuk men-*transfer* pengetahuan tentang suatu sistem atau aplikasi dari satu pengembang ke pengembang lainnya. Keberadaan *UML* tidak hanya mempermudah komunikasi antara para pengembang, tetapi juga

memungkinkan orang di luar lingkaran pengembangan, seperti pihak bisnis atau siapapun, untuk memahami suatu sistem dengan lebih baik.

UML (*Unified Modelling Language*), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. *UML* menyediakan format standar untuk merancang model sistem. Namun, karena *UML* memanfaatkan kelas dan operasi sebagai bagian dari konsep intinya, lebih sesuai digunakan untuk pengembangan perangkat lunak dengan bahasa berbasis objek seperti C++, Java, C#, atau VB.NET. Meskipun begitu, *UML* masih bisa digunakan untuk memodelkan aplikasi prosedural dalam bahasa seperti *VB* atau C.

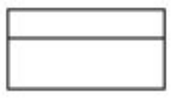
UML mulai dikembangkan oleh *Object Management Group* sejak versi 1.0 dirilis pada bulan Januari 1997. Dalam pengembangan berbasis objek, terdapat beberapa prinsip kunci yang harus dipahami, seperti Objek, Kelas, Abstraksi, Enkapsulasi, Pewarisan, dan Polimorfisme. Adapun daftar simbol UML yaitu:

Tabel 2. 1 *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
4		*Generalization*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.

NO.	GAMBAR	NAMA	KETERANGAN
5		*Include*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Extend*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Association*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2. 2 *Symbol Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.

NO.	GAMBAR	NAMA	KETERANGAN
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi

Tabel 2. 4 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 5 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

B. Kajian Hasil Penelitian Terdahulu

Untuk dapat digunakan sebagai data pendukung, diperlukan berbagai penelitian sebelumnya. Studi sebelumnya yang berkaitan dengan masalah yang dibahas dalam penelitian ini merupakan salah satu informasi pendukung yang peneliti butuhkan. Dalam hal ini, referensi yang berkaitan dengan tantangan teknologi informasi dari studi sebelumnya digunakan. Oleh karena itu penelitian dilakukan secara *online* dengan menggunakan berbagai temuan penelitian yang disajikan dalam makalah akhir, tesis, atau publikasi.

(Wulandari, Deviana et al., 2019) STMIK AKAKOM YOGYAKARTA. dengan judul penelitian "PROGRESSIVE WEB APPS (PWA) UNTUK APLIKASI PELACAKAN PENGIRIMAN BARANG (STUDI KASUS JASA PENGIRIMAN PT.ORIFLAME)". Dalam penelitiannya bertujuan untuk mampu menerapkan teknologi PWA dalam pembuatan website pelacakan pengiriman.

(Reza Maulana et al., 2020) Universitas Islam Negeri Alauddin Makassar dengan judul penelitian "PENERAPAN PROGRESSIVE WEB APPS (PWA) PADA SISTEM LAYANAN KONSULTASI KEPADA PAKAR SYARI'AH

ISLAM”. Dalam penelitian ini bertujuan untuk bagaimana menerapkan layanan konsultasi dakwah.

Novelty penelitian "Rancang Bangun Aplikasi Ensiklopedia Digital Masakan Nusantara Menggunakan Teknologi Progressive Web Apps" terletak pada penggunaan teknologi PWA untuk membangun aplikasi. PWA adalah teknologi web terbaru yang memungkinkan aplikasi web untuk diinstal di perangkat pengguna dan memiliki pengalaman pengguna yang mirip dengan aplikasi native. Hal ini merupakan hal yang baru karena belum ada penelitian sebelumnya yang menggunakan teknologi PWA untuk membuat aplikasi ensiklopedia digital masakan nusantara.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian eksperimental, yaitu penelitian yang pengumpulan datanya melalui pencatatan secara langsung dari hasil percobaan yang dilakukan.

B. Lokasi dan Waktu Penelitian

Penelitian dilakukan di Kota Parepare. Penelitian ini akan berlangsung selama ± 2 (dua) bulan di tahun 2024.

C. Alat dan Bahan Penelitian

1. Perangkat keras

a. Laptop Asus ROG Strix G512LI-I75TB6T dengan spesifikasi hardware:

- *Processor* : Intel® Core™ i5-10300H CPU @ 2.50GHz
- *Installed RAM* : 8GB
- *SSD* : 512GB

2. Perangkat lunak

a. Windows 10

b. Visual Studio Code

c. XAMPP

d. MySQL

e. Google Chrome

D. Teknik Pengumpulan Data

1. Studi pustaka

Studi pustaka merupakan strategi pengumpulan data dari berbagai macam sumber literatur seperti, buku, jurnal, *e-book*, youtube, .ataupun *website-website* yang relevan dengan masalah yang diidentifikasi.

Observasi

Observasi merupakan teknik pengumpulan data dengan cara mengamati secara langsung maupun tidak langsung. Observais tidak langsung dilakukan dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

2. Dokumentasi

Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis, dokumen elektronik, atau dokumen audio visual.

E. Tahapan Penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu:

1. Pengimpulan data

   Mengumpulkan data-data dan informasi untuk di jadikan acuan dalam membangun aplikasi yang dirancang

2. Analisis Sistem

   Mengidentifikasikan dan mengenali masalah yang ada kemudian mencari alternatif pemecahan.

3. Pengumpulan Data

Pada tahap ini peneliti melakukan mengumpulkan data melalui Internet .

4. Analisis

   Stelah masalah ditentukan dan analisi data sudah dilakukan maka perlu dilakukan pembangunan atau mendesain system tentang masalah yang temukan.

5. Perancangan

   Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah, desain tampilan dan alur penggunaan aplikasi.

6. Pengujian

   Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat menggunakan *black box, white box* dan pengujian *marker.* Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

7. Implementasi

   Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh pengguna.

F. Jenis Data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan data sekunder yang relevan dengan masalah yang akan dibahas :

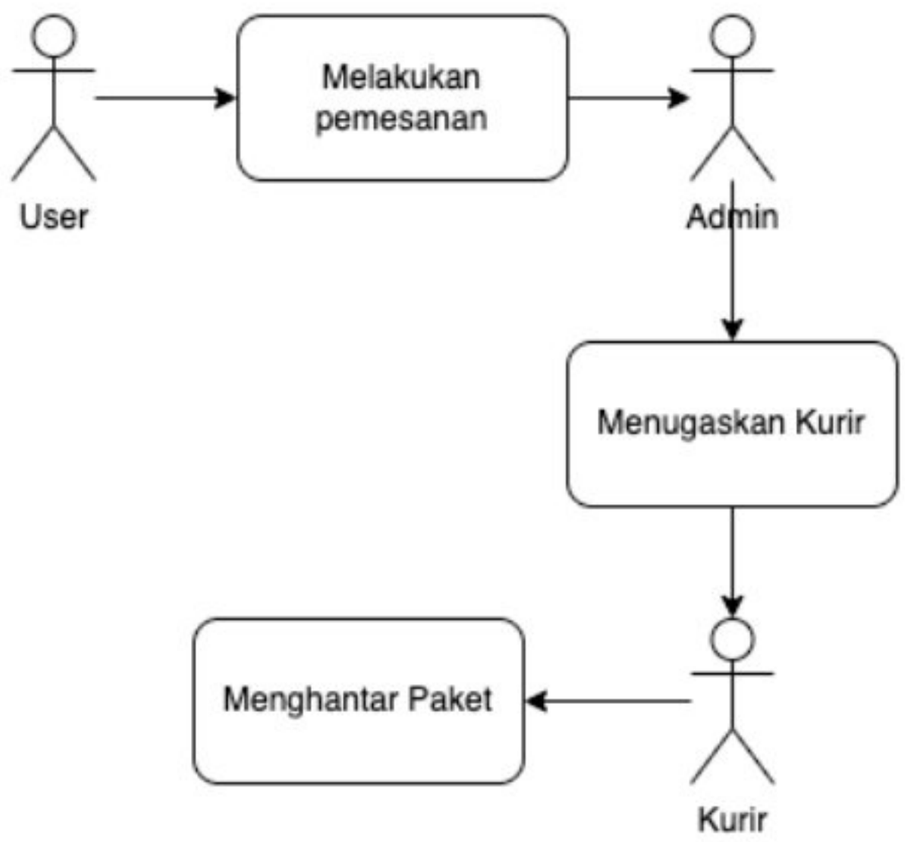
1. Data Primer adalah yang berasal atau data yang diperolah langsung dari sumber-sumber yang berada di lapangan yang berasal dari wawancara dengan pihak-pihak terkait dengan penelitian ini.

2. Data Sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Penelitia ini mendapatkan data yang sudah jadi yang sidah dikumpulkan dari pihak lain dengan berbagai cara atau metode lainnya atau metode baik secara komersial maupun non komersial.

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem yang berjalan



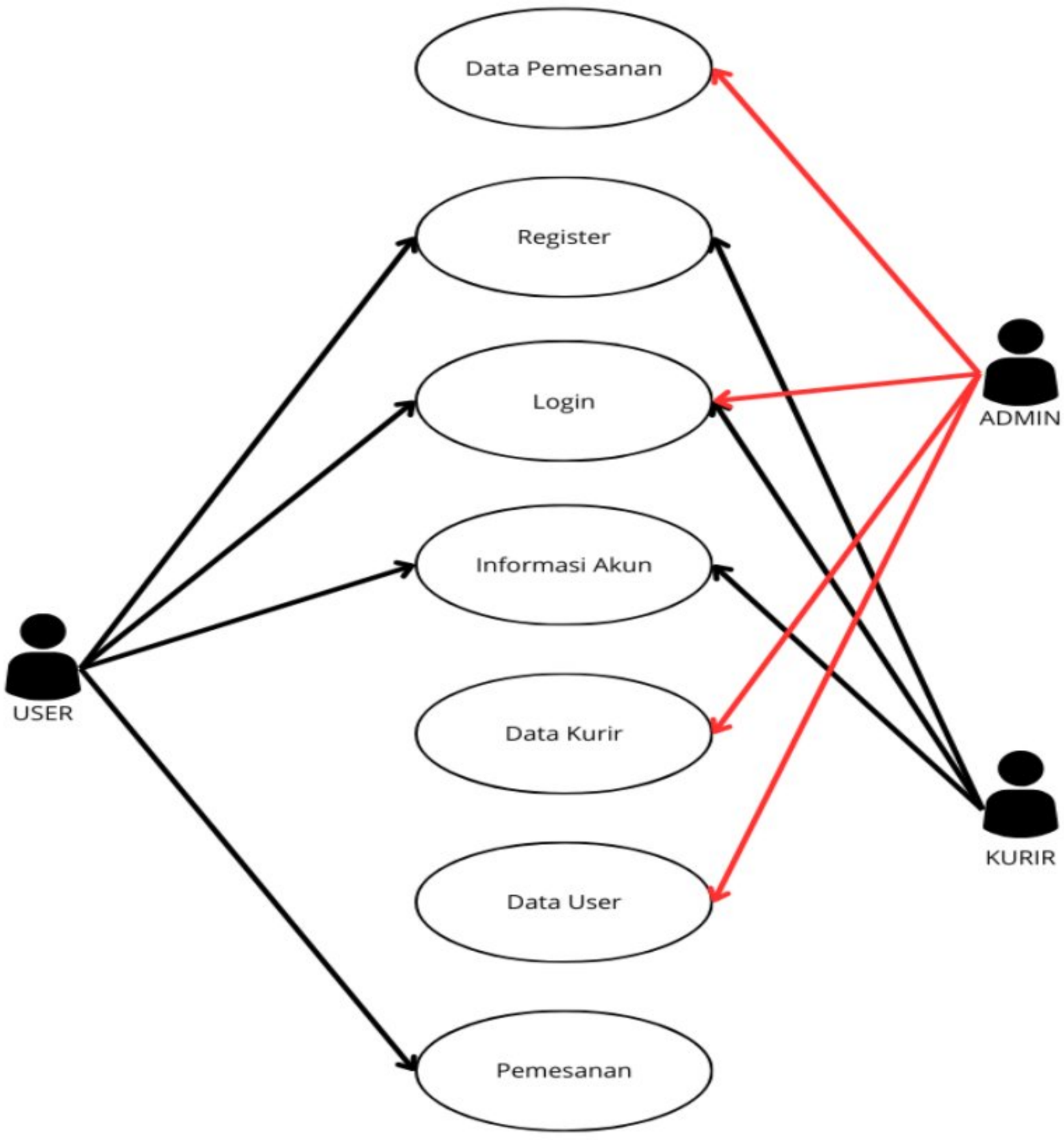
Gambar 4.1 Sistem Yang Berjalan

Berikut ini adalah sistem yang ada di Shenior Kurir:

1) Pelanggan melakukan pemesanan melalui whatsapp admin shenior kurir
2) Shenior Kurir membagikan informasi pelanggan ke Grup yang berisikan Kurir dari shenior Kurir
3) Menunggu respon kurir yang mau mengambil pemesanan pelanggan
4) Setelah ada kurir yang mau mengambil pemesanan pelanggan
5) Kurir menghubungi pelanggan untuk memastikan pemesanan dan alamat dan sebagainya
6) Kurir Mengambil barang yang mau diantarkan ke pelanggan
7) Kurir mengantarkan ke alamat pelanggan dan melakukan pembayaran cash

B. Sistem yang diusulkan

Adapun sistem yang diusulkan yaitu:

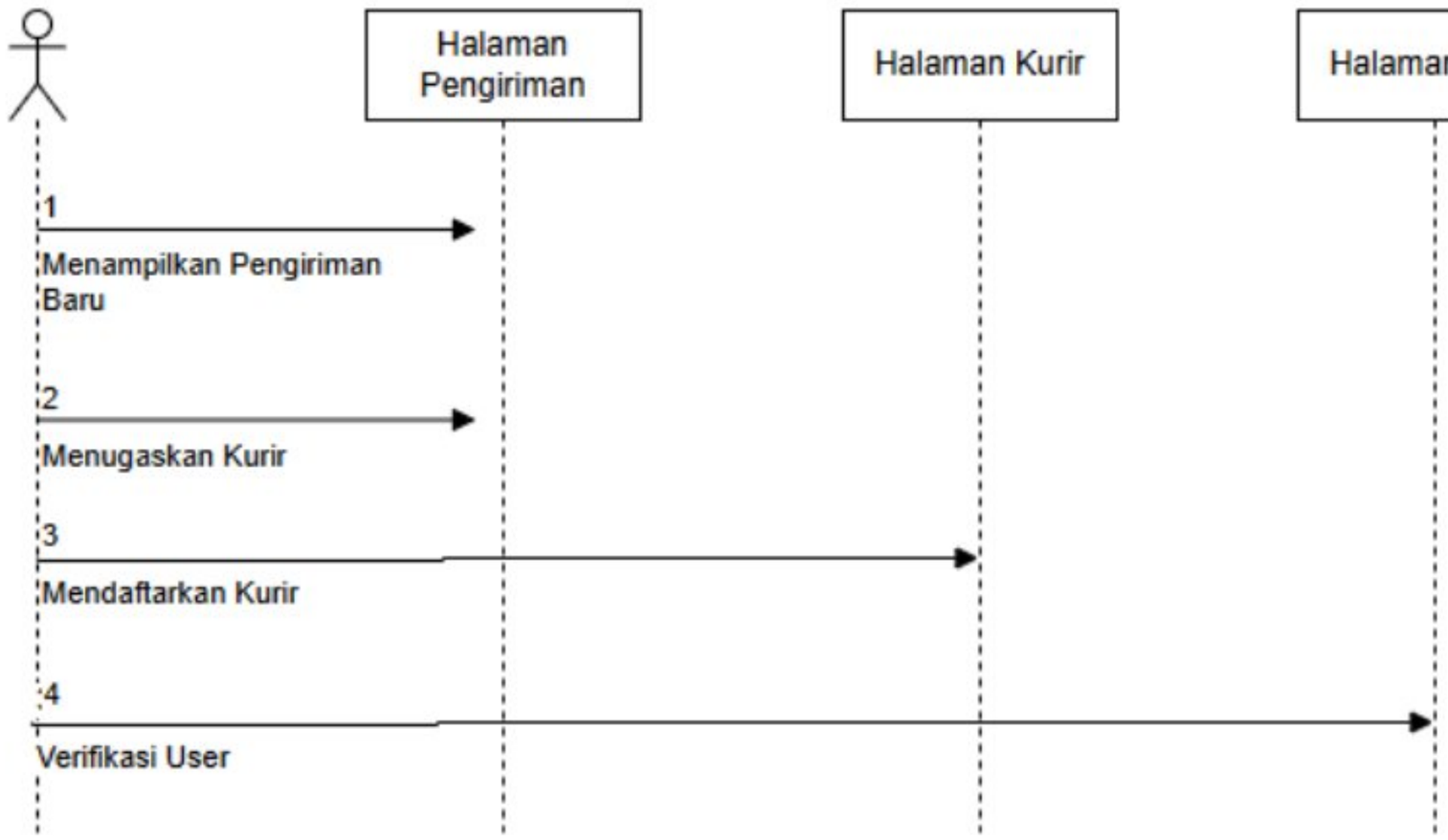


Gambar 4.2 Sistem Yang Diusulkan

Diatas merupakan sistem yang diusulkan, dimana admin dapat melihata data pemesanan, melihat data kurir, melakukan pendaftaran kurir dan juga data user. Manakala kurir bisa melihat informasi akun dan harus terdaftar sebelumnya. User dapat melakukan pemesanan dan juga melihat informasi akun. Ketiga aktor harus melakukan login untuk mengakses sistem

C. Sequence Diagram

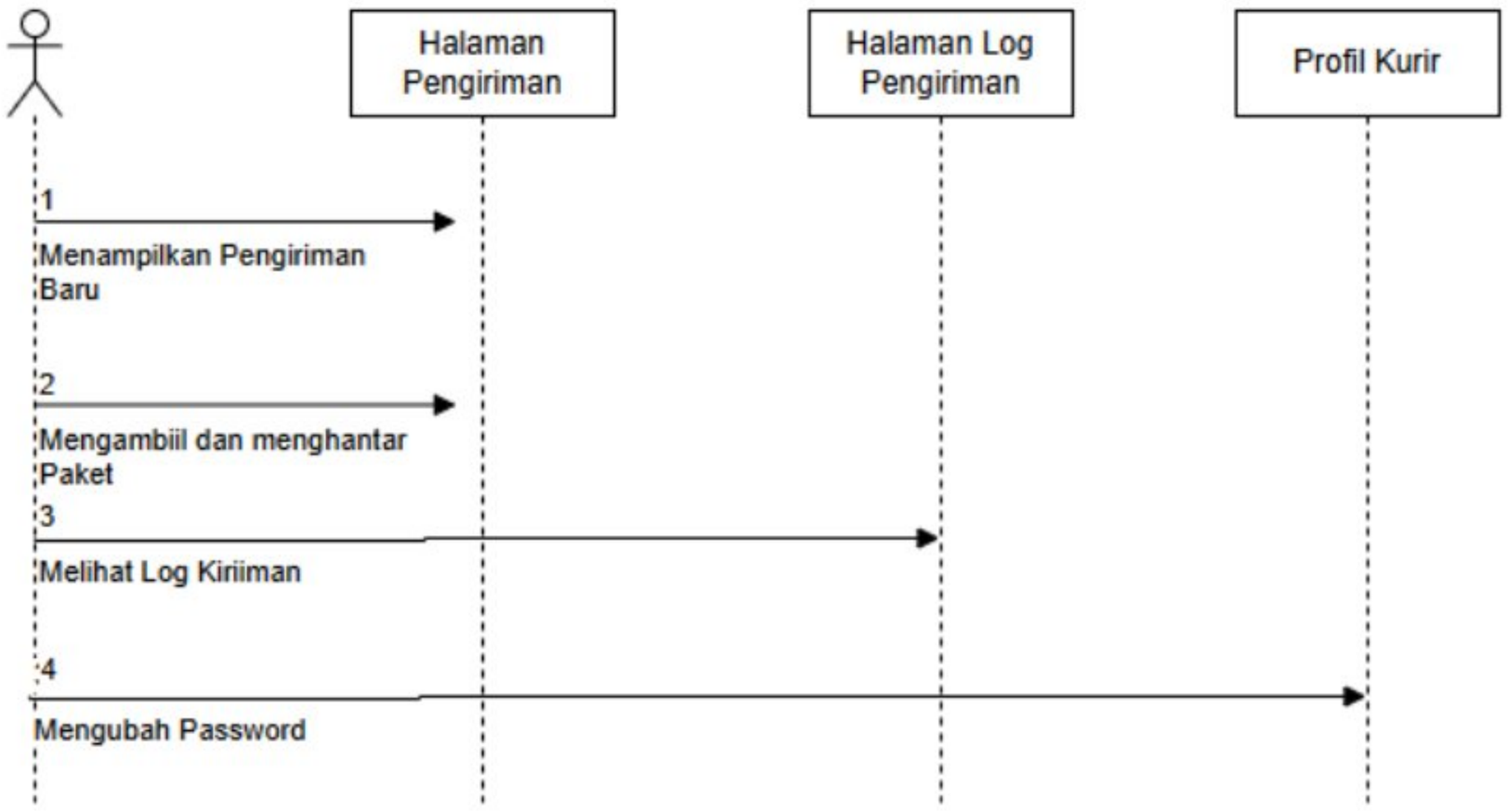
1. Admin



Gambar 4.3 Sequence Diiagram Admin

Diatas merupakan sequence diagram admin, disini terlihat admin dapat menugaskan kurir pada halaman pengiriman, mendaftarkan kurir baru pada halaman kurir dan memverifikasi user pada halaman user.

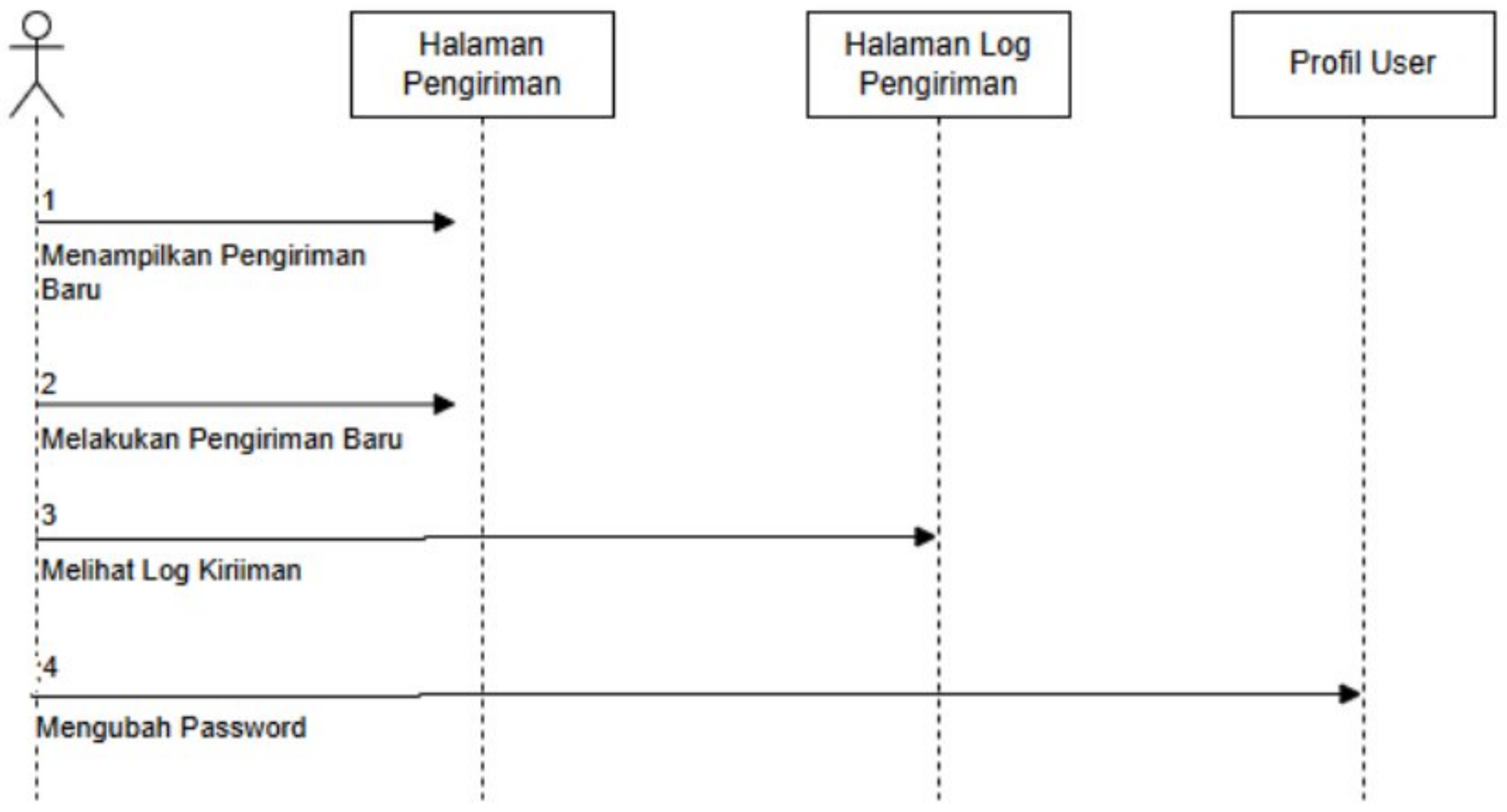
2. Kurir



Gambar 4.4 Sequence Diiagram Kurir

Diatas merupakan sequence diagram kurir, disini terlihat kurir dapat mengambil dan menghantar paket pada halaman pengiriman. melihat log kiriman pada halaman log pengiriman dan mengganti password login pada halaman profil kurir.

3. User

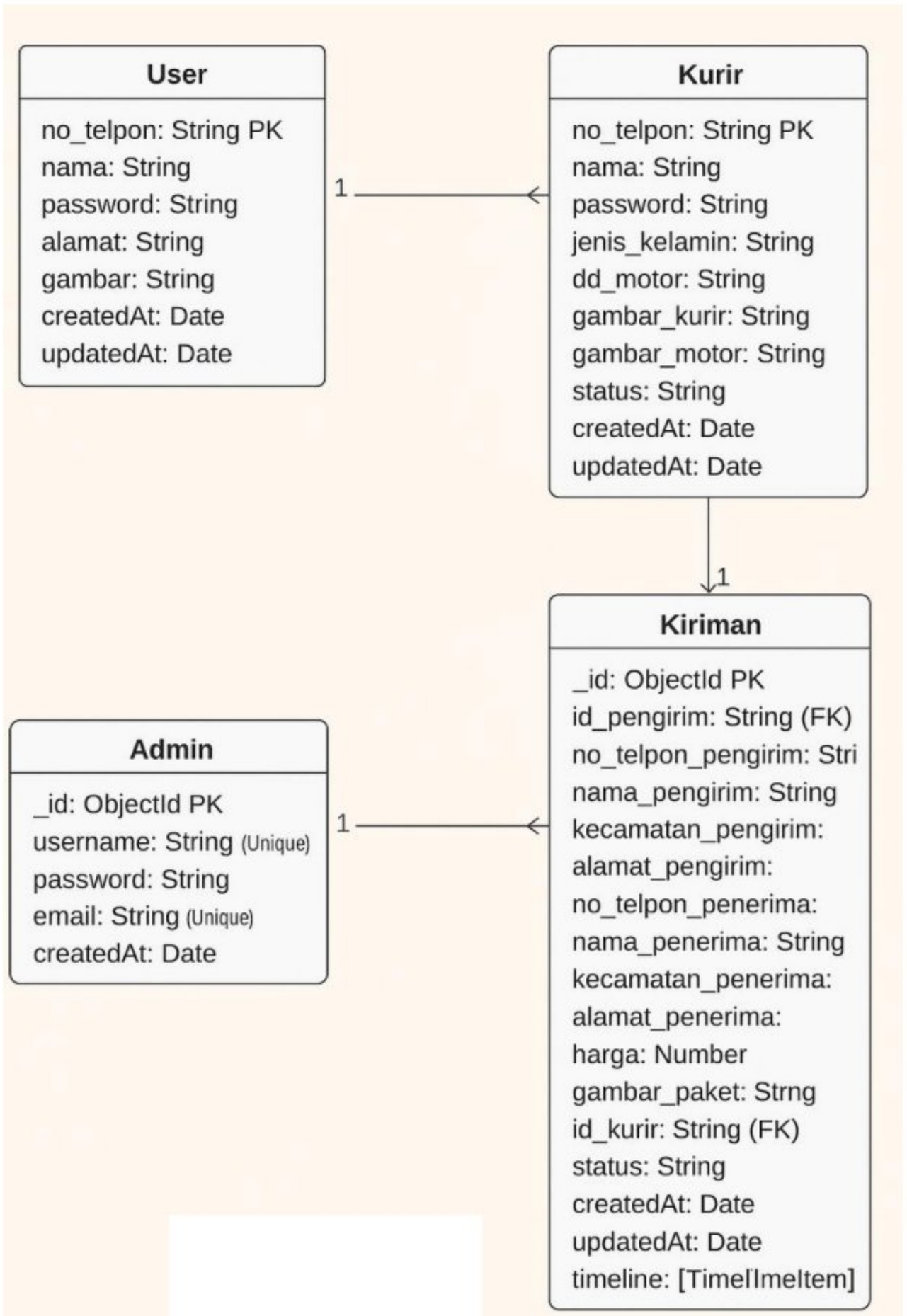


Gambar 4.5 Sequence Diagram User

Diatas merupakan sequence diagram user, disini terlihat user dapat melakukan pengiriiman baru pada halaman pengiriman. melihat log kiriman pada halaman log pengiriman dan mengganti password login pada halaman profil user.

D. ERD dan Kamus Data

1. ERD (Entity Relationship Diagram)



Gambar 4.6 ERD Sistem

Diatas merupakan ERD system, dimana setiap collection saling terkait antara satu dengan lainnya. Dari gambar diatas terlihat empat collection yaitu collection user, kurir , kiriman dan admin.

2. Kamus Data

Tabel 3.1 Kamus Data Admin

Nama Field	Tipe Data	Deskripsi	Constraint
_id	ObjectId	ID unik untuk dokumen admin	Otomatis (Mongoose)
username	String	Nama pengguna admin	Wajib, Unik
password	String	Kata sandi admin	Wajib
email	String	Alamat email admin	Wajib, Unik
createdAt	Date	Tanggal pembuatan akun admin	Default: Date.now

Tabel 4.2 Kamus Data Kurir

Nama Field	Tipe Data	Deskripsi	Constraint
_id	ObjectId	ID unik untuk dokumen kurir	Otomatis (Mongoose)
no_telpon	String	Nomor telepon kurir	Wajib, Unik
nama	String	Nama kurir	Wajib
password	String	Kata sandi kurir	Opsional
jenis_kelamin	String	Jenis kelamin kurir	Wajib
dd_motor	String	Nomor plat motor kurir	Wajib, Unik
gambar_kurir	String	URL/path gambar kurir	Wajib
gambar_motor	String	URL/path gambar motor kurir	Wajib
status	String	Status kurir (misalnya, aktif)	Opsional
createdAt	Date	Tanggal pembuatan akun kurir	Otomatis (Mongoose)
updatedAt	Date	Tanggal pembaruan akun kurir	Otomatis (Mongoose)

Tabel 4.3 Kamus Data User

Nama Field	Tipe Data	Deskripsi	Constraint
_id	ObjectId	ID unik untuk dokumen user	Otomatis (Mongoose)
no_telpon	String	Nomor telepon pengguna	Wajib, Unik
nama	String	Nama pengguna	Wajib
password	String	Kata sandi pengguna	Wajib
alamat	String	Alamat pengguna	Opsional
gambar	String	URL/path gambar pengguna	Opsional
createdAt	Date	Tanggal pembuatan akun pengguna	Otomatis (Mongoose)
updatedAt	Date	Tanggal pembaruan akun pengguna	Otomatis (Mongoose)

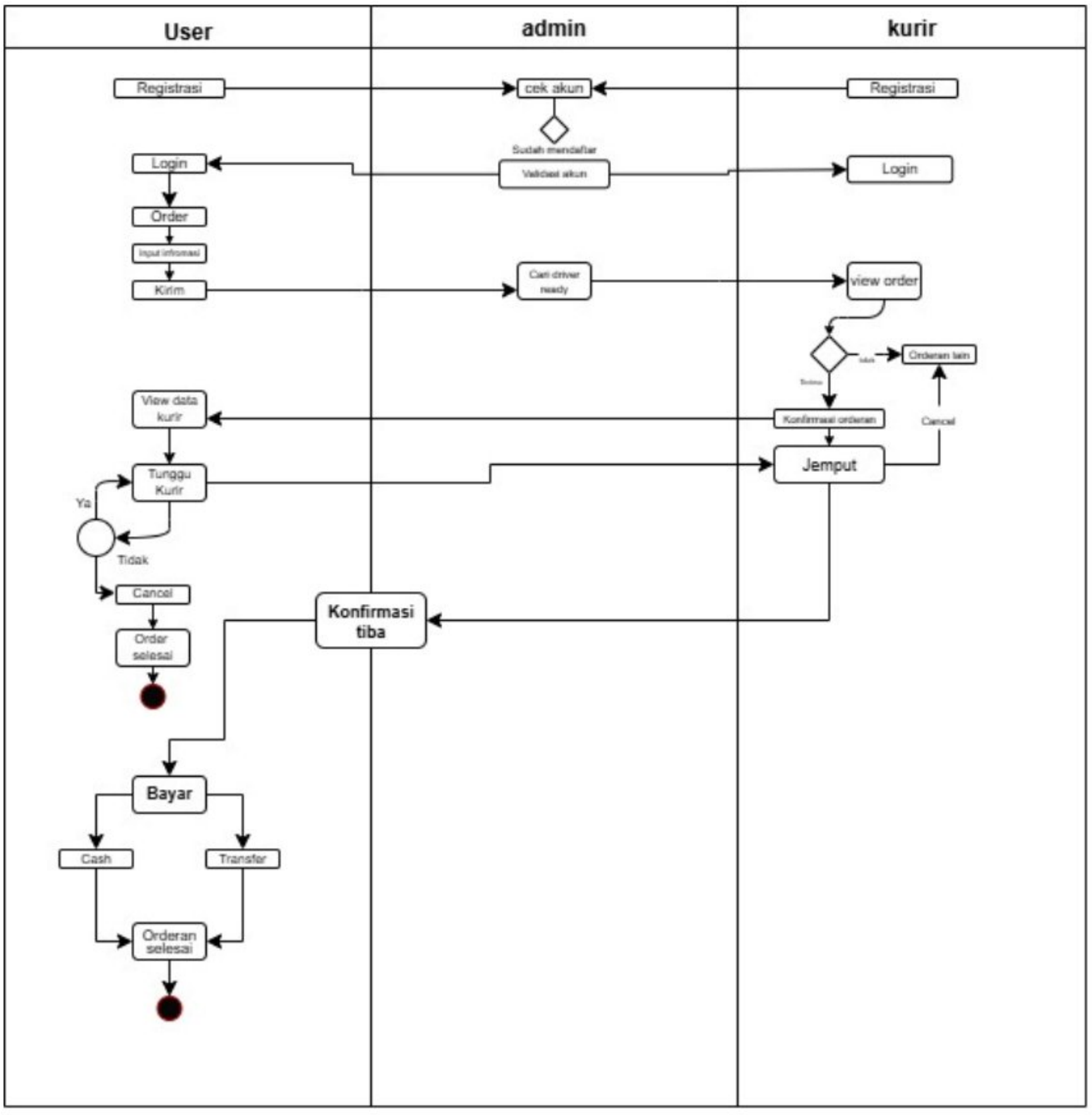
Kamus Data: Kiriman

Tabel 4.4 Kamus Data Kiriman

Nama Field	Tipe Data	Deskripsi	Constraint
_id	ObjectId	ID unik untuk dokumen kiriman	Otomatis (Mongoose)
id_pengirim	String	ID pengirim (merujuk ke User.no_telpon)	Wajib
no_telpon_pengirim	String	Nomor telepon pengirim	Wajib
nama_pengirim	String	Nama pengirim	Wajib
kecamatan_pengirim	String	Kecamatan pengirim	Wajib
alamat_pengirim	String	Alamat lengkap pengirim	Wajib
no_telpon_penerima	String	Nomor telepon penerima	Wajib
nama_penerima	String	Nama penerima	Wajib
kecamatan_penerima	String	Kecamatan penerima	Wajib
alamat_penerima	String	Alamat lengkap penerima	Wajib
harga	Number	Harga pengiriman	Opsional

Nama Field	Tipe Data	Deskripsi	Constraint
gambar_paket	String	URL/path gambar paket	Wajib
id_kurir	String	ID kurir (merujuk ke Kurir.no_telpon)	Opsional
status	String	Status pengiriman	Default: "Menunggu Admin Memproses"
createdAt	Date	Tanggal pembuatan kiriman	Otomatis (Mongoose)
updatedAt	Date	Tanggal pembaruan kiriman	Otomatis (Mongoose)
timeline	[TimelineItem]	Array riwayat status pengiriman	Default: [{status, waktu}]

3. Activity Diagram



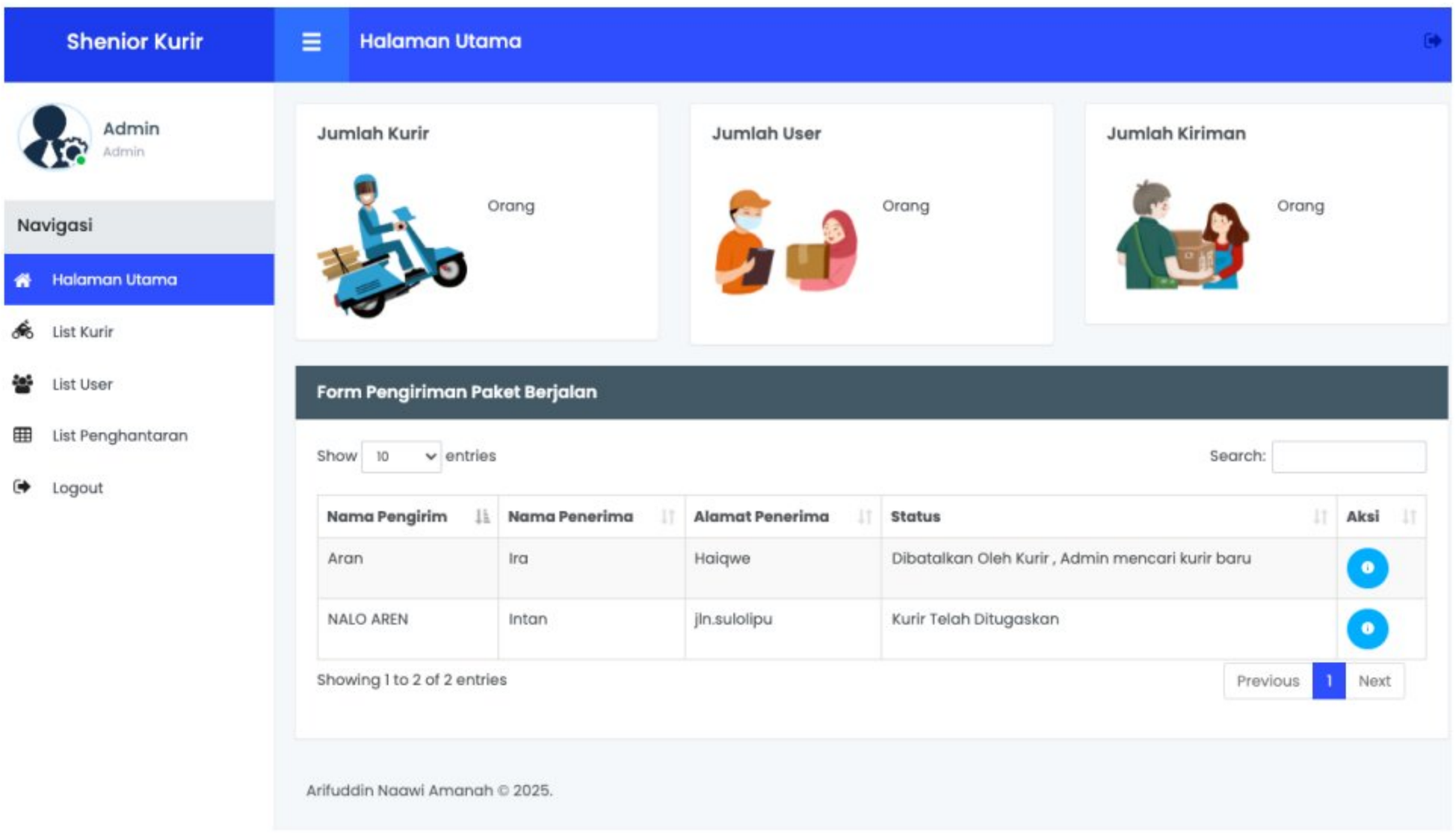
Gambar 4.7 Activity diagram

Diagram ini menampilkan bagaimana aktivitas-aktivitas pada suatu proses saling terhubung dan kapan terjadi perpindahan dari satu aktivitas ke aktivitas lain (baik secara sekuensial maupun paralel)

E. Detail Sistem

1. Admin

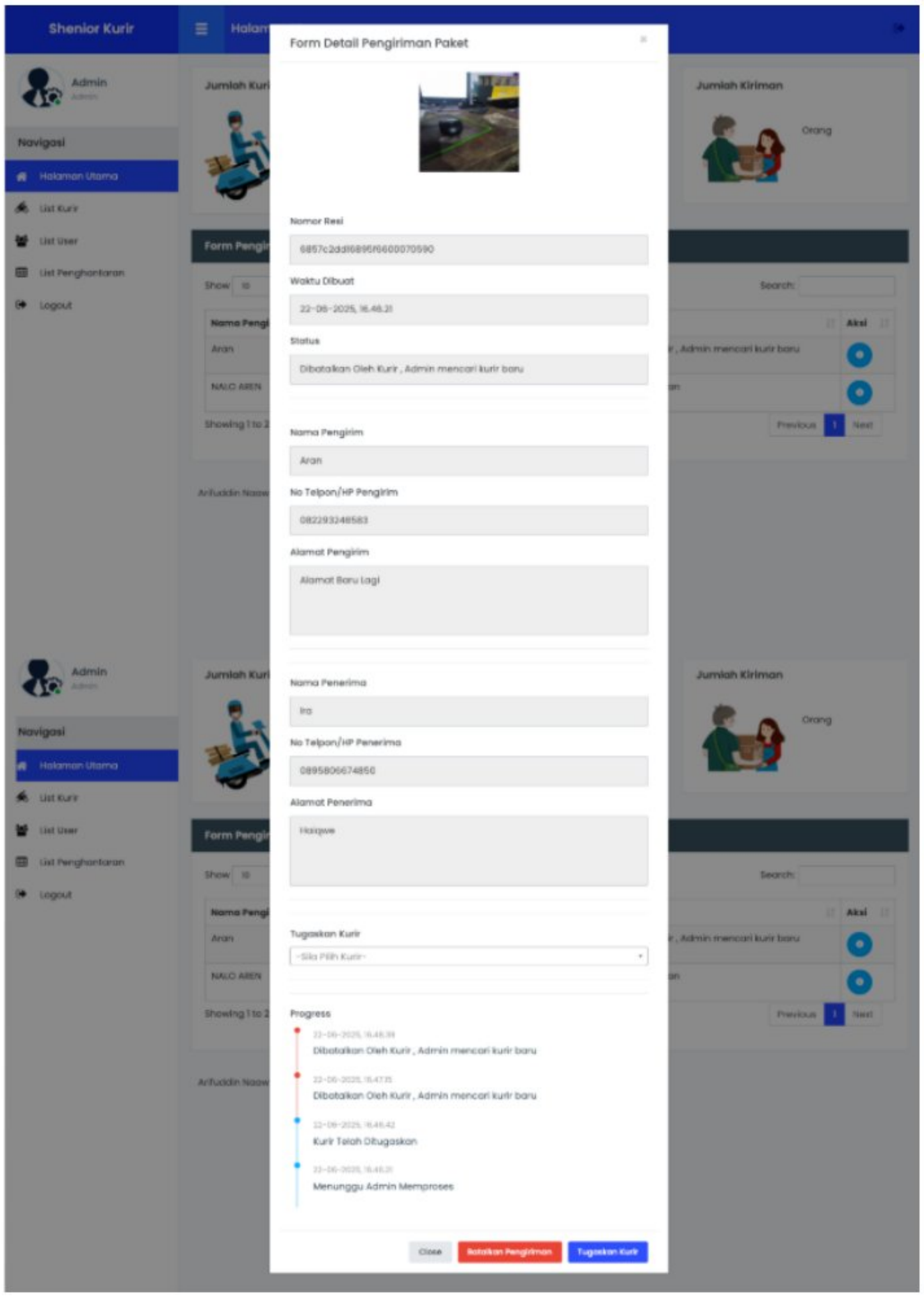
   a. Halaman Utama



Gambar 4.8 Halaman Utama Admin

Diatas merupakan halaman utama admin, dimana pada halaman ini admin dapat melihat list pengiriman yang sedang berjalan, yang baru terdaftar , informasi berkaitan paket yang dikirim, menugaskan kurir untuk mengantar paket dan lainnya.

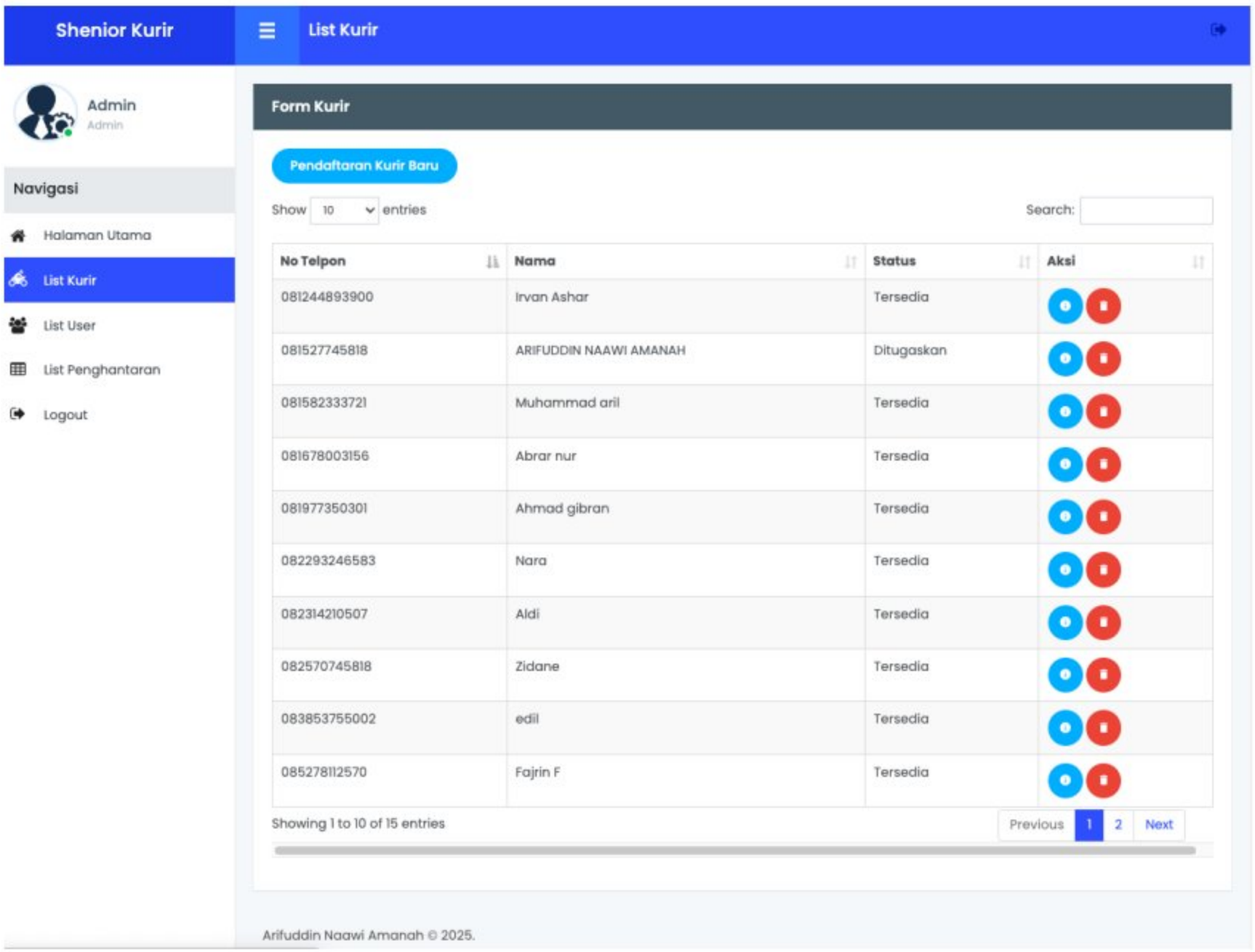
b. Dialog Detail Pengiriman Paket



Gambar4.9 Dialog Detail Pengiriman Paket

Diatas merupakan dialog detail pengiriman paket, disini admin dapat melihat informasi seputar pengiriman paket seperti nama pengirim dan penerima, alamat pengirim dan penerima, progres penghantaran paket dan memberikan tugas kepada kurir.

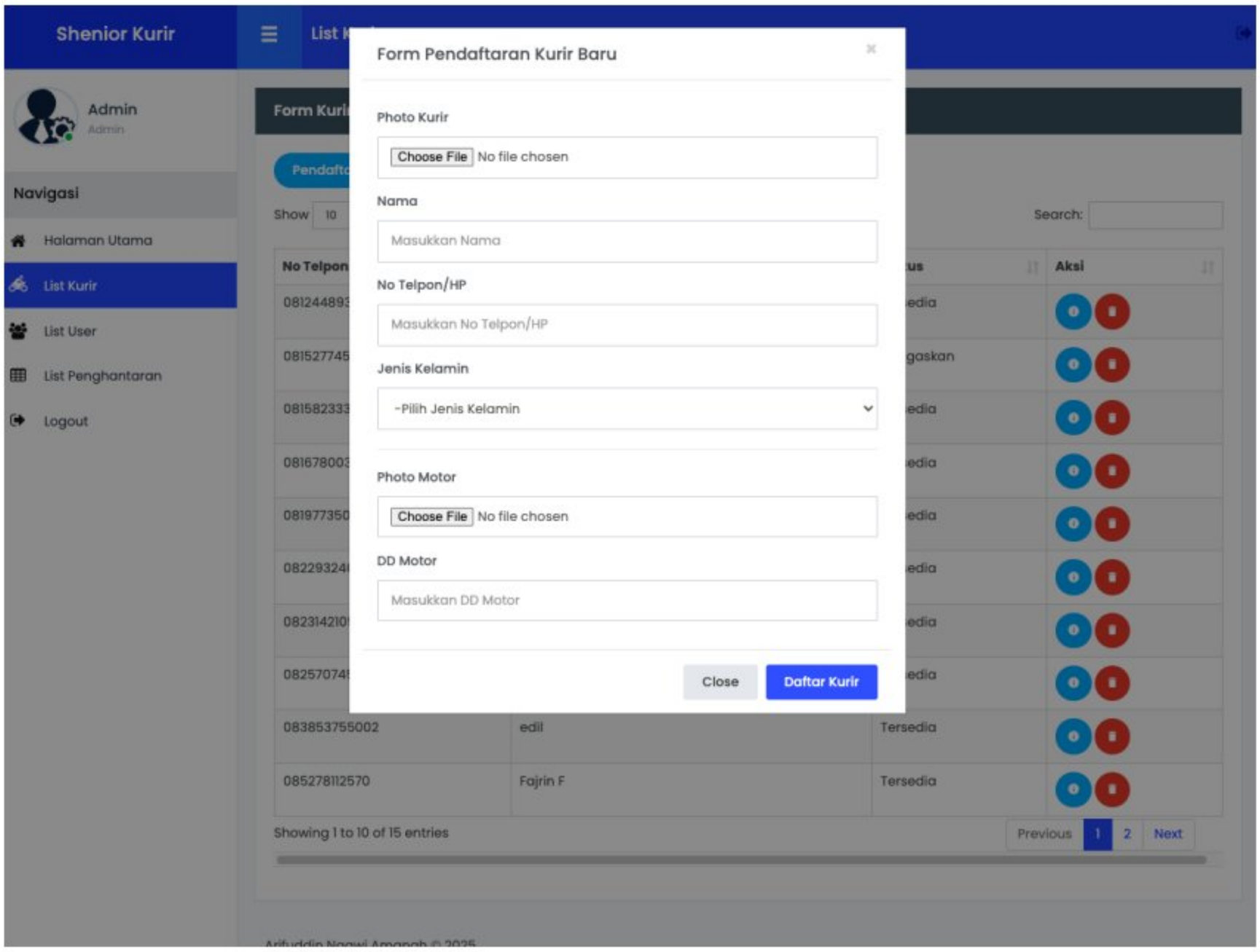
c. List Kurir



Gambar 4.10 Halaman List Kurir

Diatas merupakan halaman list kurir, dimana disini admin dapat melihat list kurir, status kurir jika bersedia atau ditugaskan, medaftarkan kurir baru dan lainnya.

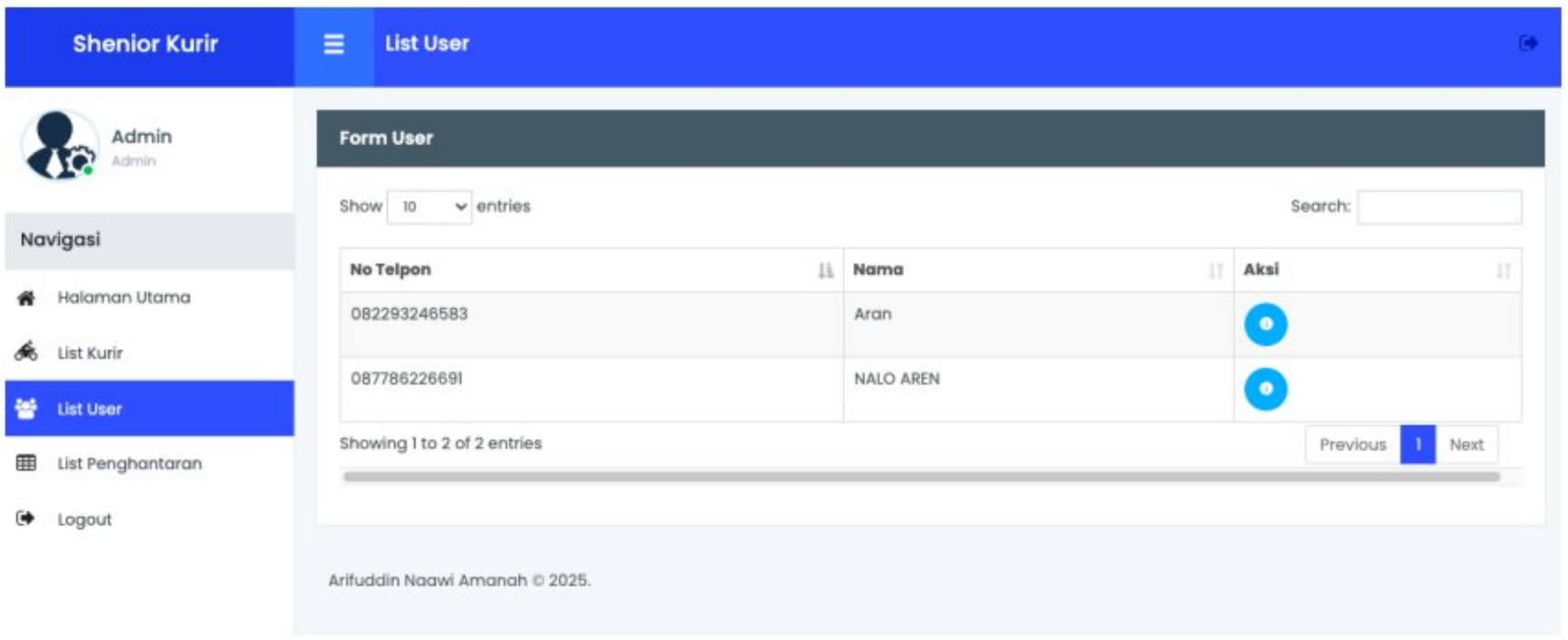
d. Form Pendaftaran Kurir



Gambar 4.11 Form Pendaftaran Kurir

Diatas merupakan form pendaftarn kurir, disini admin melakukan pendaftaran kurir dengan memasukkan foto motor, foto kurir ,nama kurir dan juga dd motor

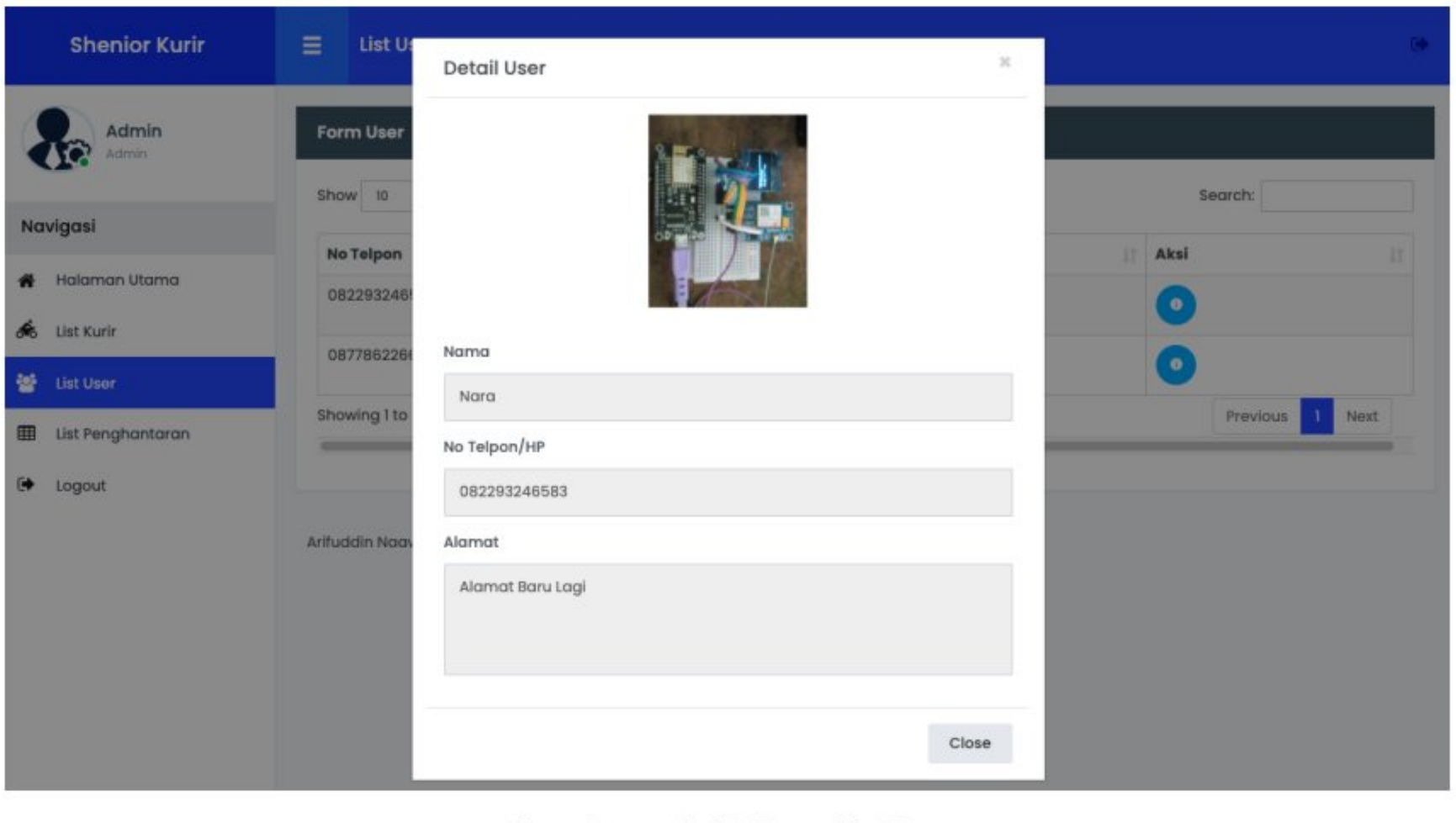
e. List User



Gambar 4.12 List User

Diatas merupakan halaman list kurir, disini admin dapat melihat list user yang telah terdaftar dalam sistem

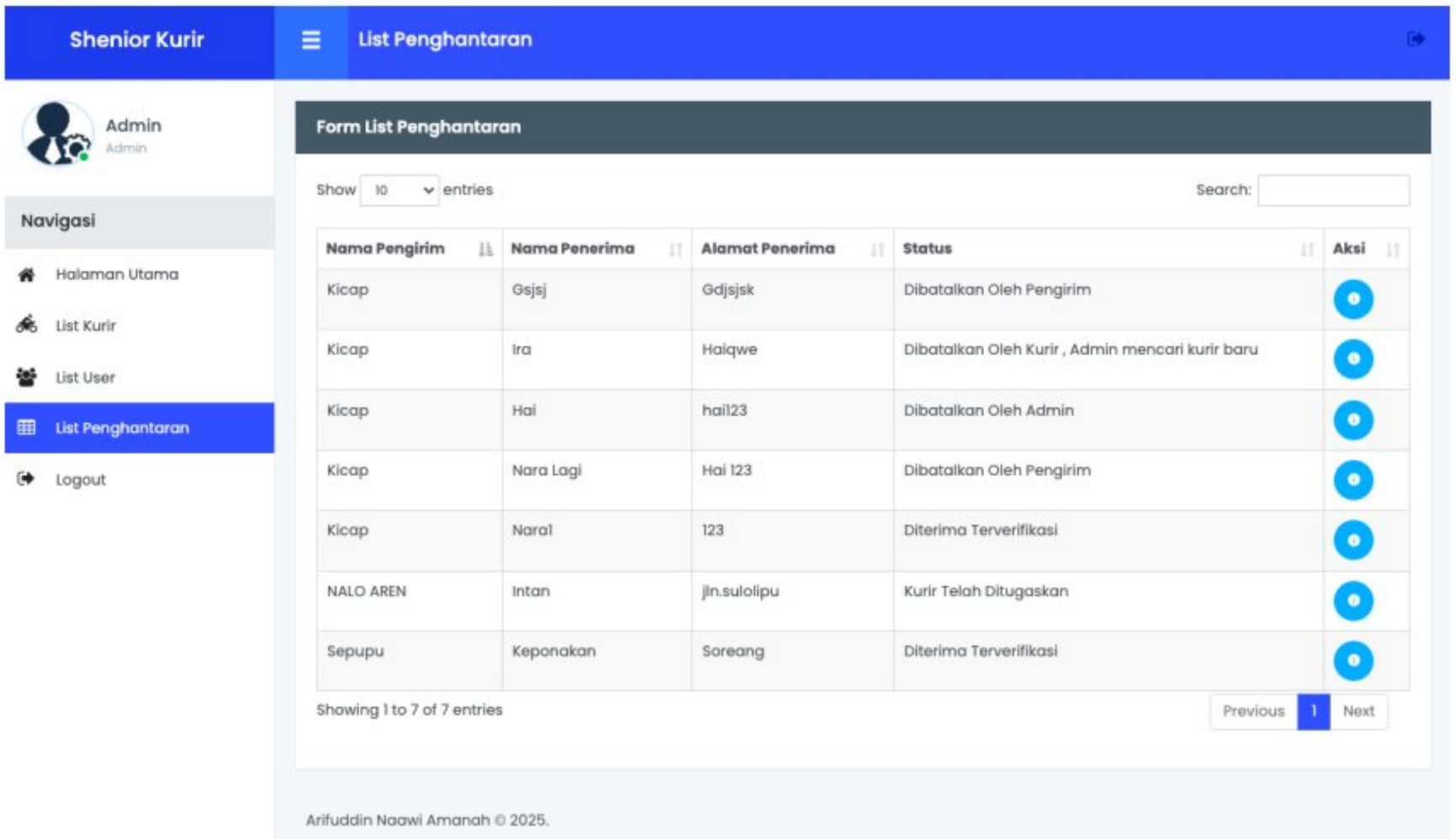
f. Detail User



Gambar 4.13 Detail User

Diatas merupakan dialog detail user, disini admin dapat melihat gambar user, nama user, nomor telpon user dan juga alamat user.

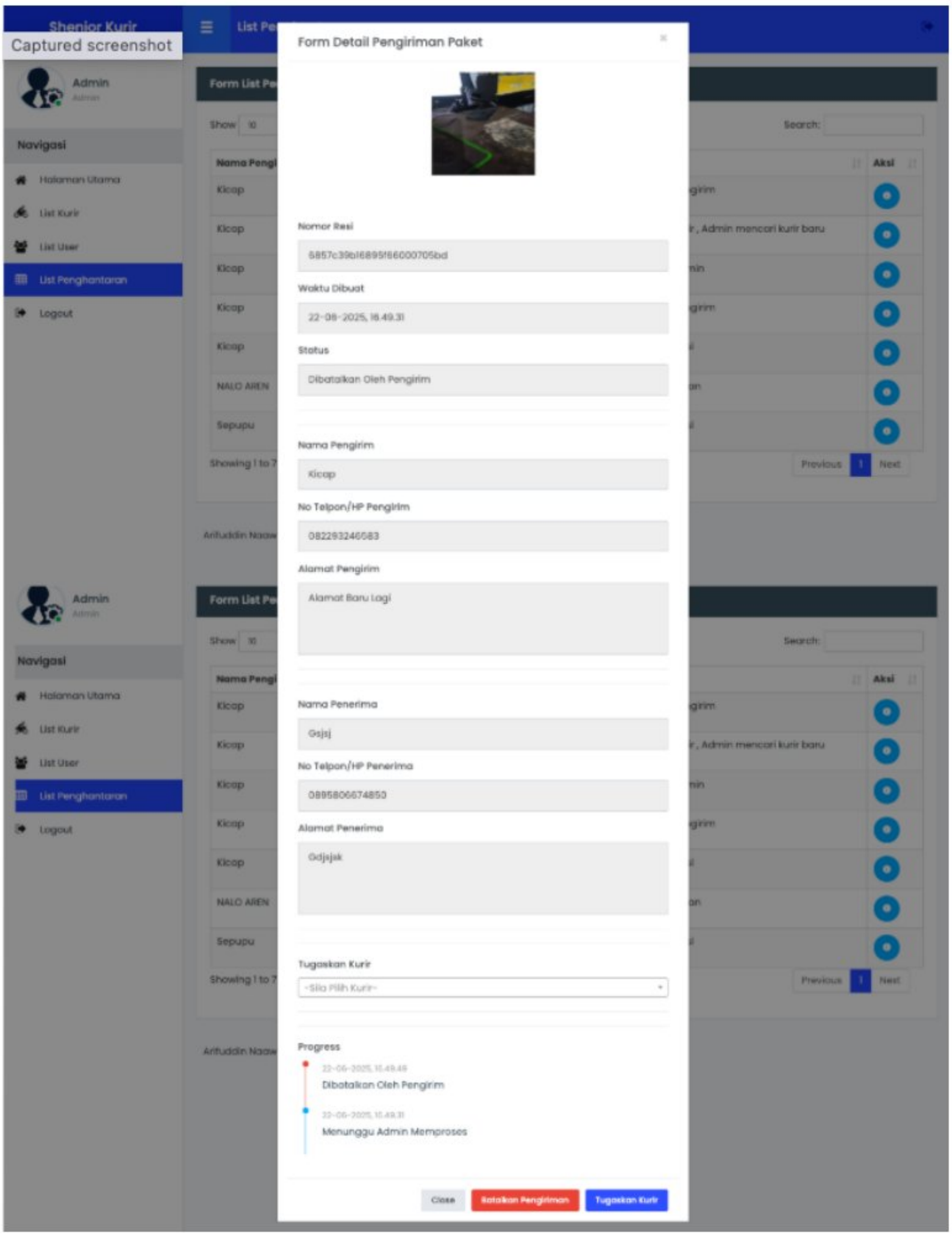
g. List Penghantaran



Gambar 4.14 List Penghantaran

Diatas merupakan list penghantaran yang telah dilakukan maupun yang berjalan pada waktunya, disini admin dapat melihat status penghantaran, menugaskan kurir, melihat progres penghantaran dan lainnya

h. Detail Penghantaran Paket

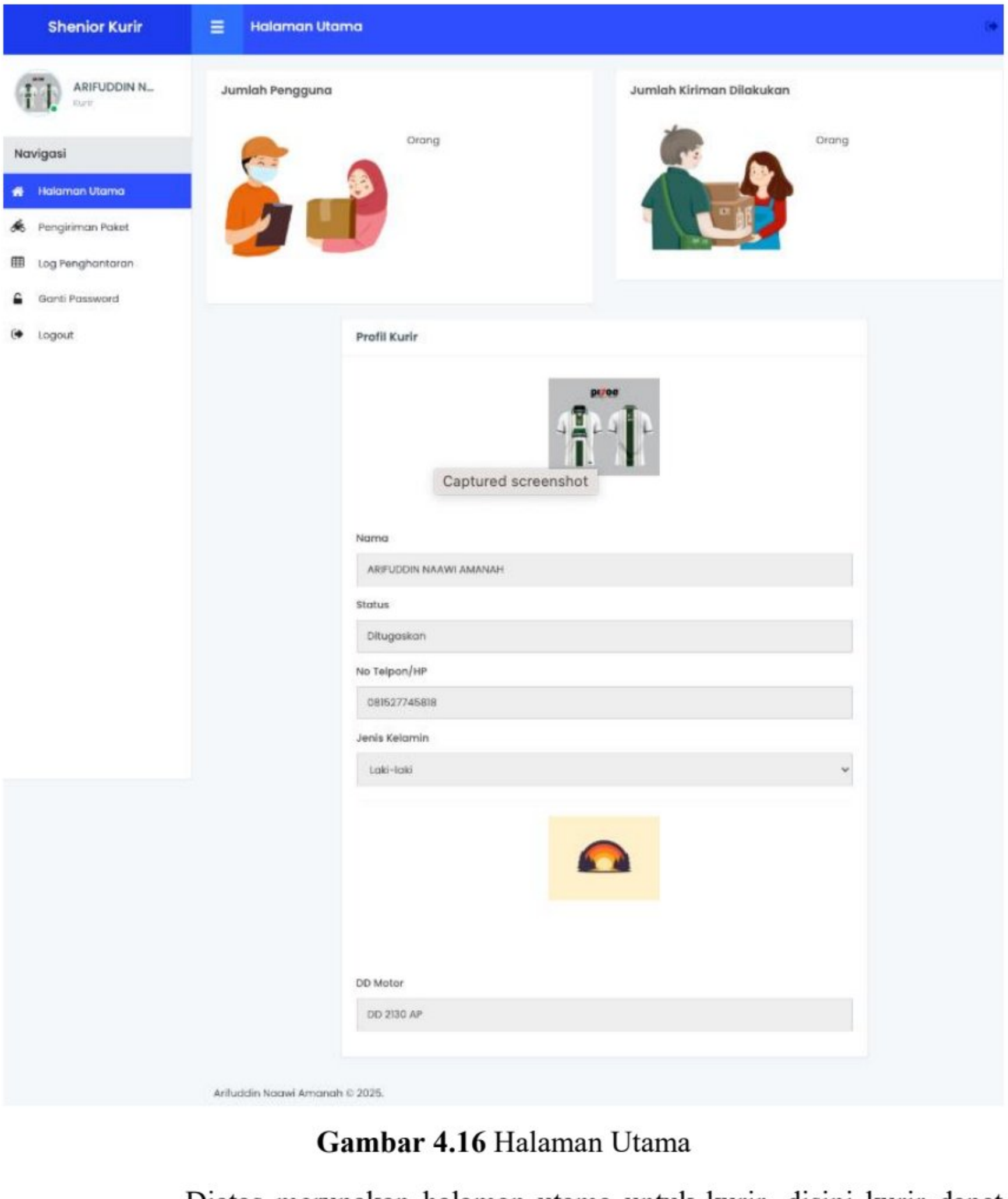


Gambar 4.15 Detail Penghantaran Paket

Diatas merupakan dialog detail pengiriman paket, disini admin dapat melihat informasi seputar pengiriman paket seperti nama pengirim dan penerima, alamat pengirim dan penerima, progres penghantaran paket dan memberikan tugas kepada kurir.

2. Kurir

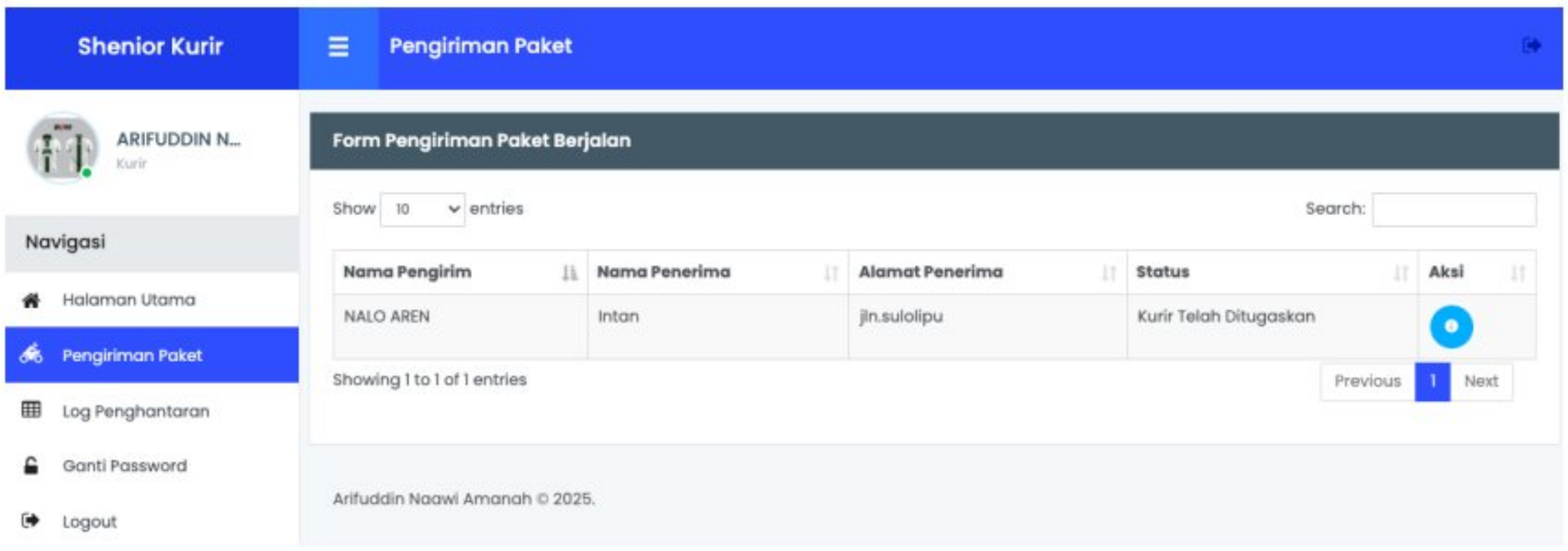
a. Halaman Utama



Gambar 4.16 Halaman Utama

Diatas merupakan halaman utama untuk kurir, disini kurir dapat melihat jumlah pengguna dan juga jumlah kiriman yang telah dilakukan oleh kurir tersebut. Kurir juga dapat melihat detail berkaitan kurir tersebut.

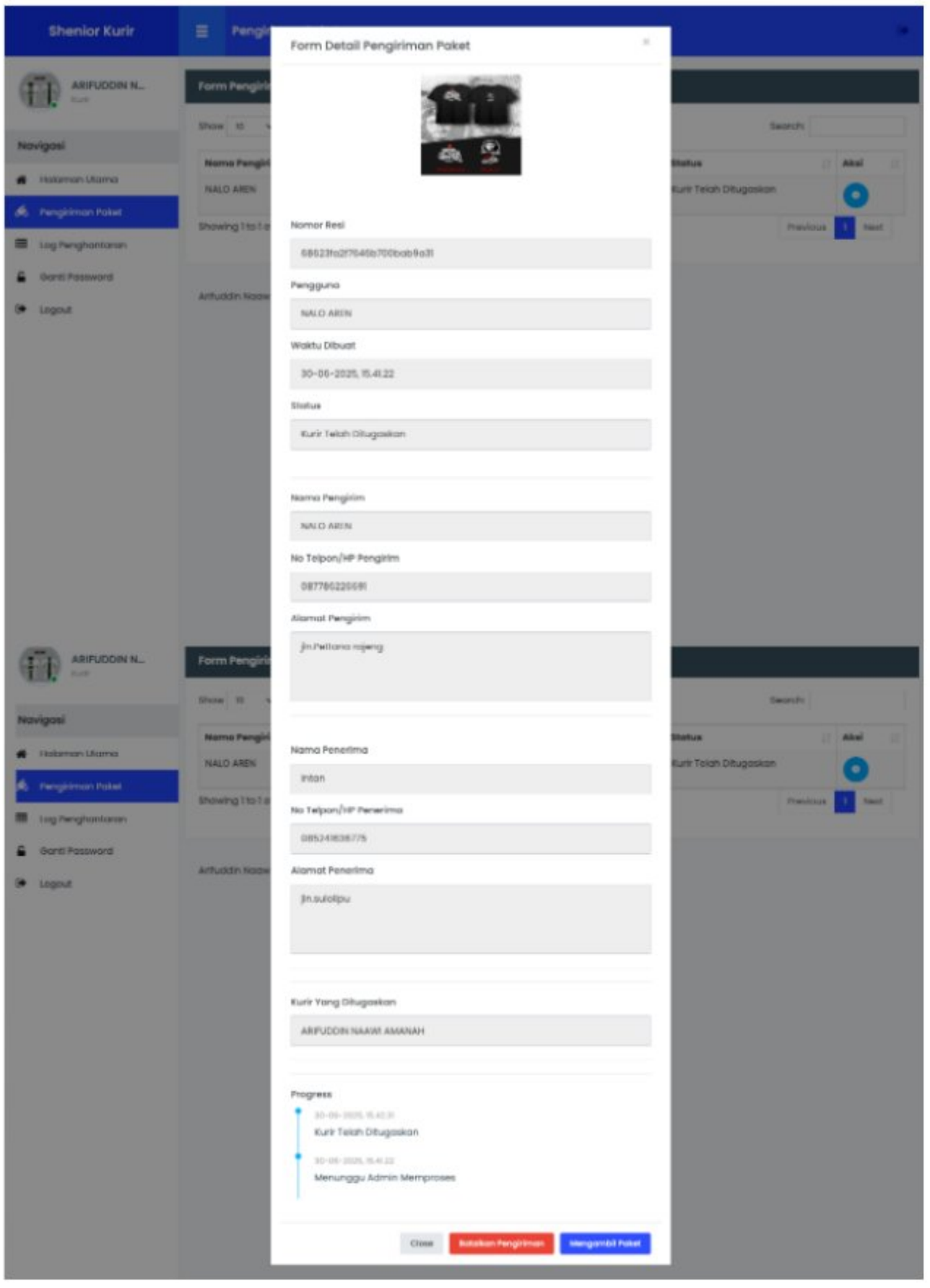
b. Pengiriman Paket



Gambar 4.17 Pengiriman Paket

Diatas merupakan halaman pengiriman paket, jika admin telah menugaskan kurir tersebut, maka kurir akan mendapat notifikasi baik dalam sistem maupun melalui Whatsapp, kurir kemudian dapat melihat informasi pengiriman paket seperti nama pengirim dan penerima, nomor telepon pengirim dan peneriman, dan alamat pengirim dan penerima, serta foto paket yang ingin dihantar. Ini untuk memudahkan kurir untuk menghubungi pengirim maupun penerima untuk mengambil paket maupun menghantar paket

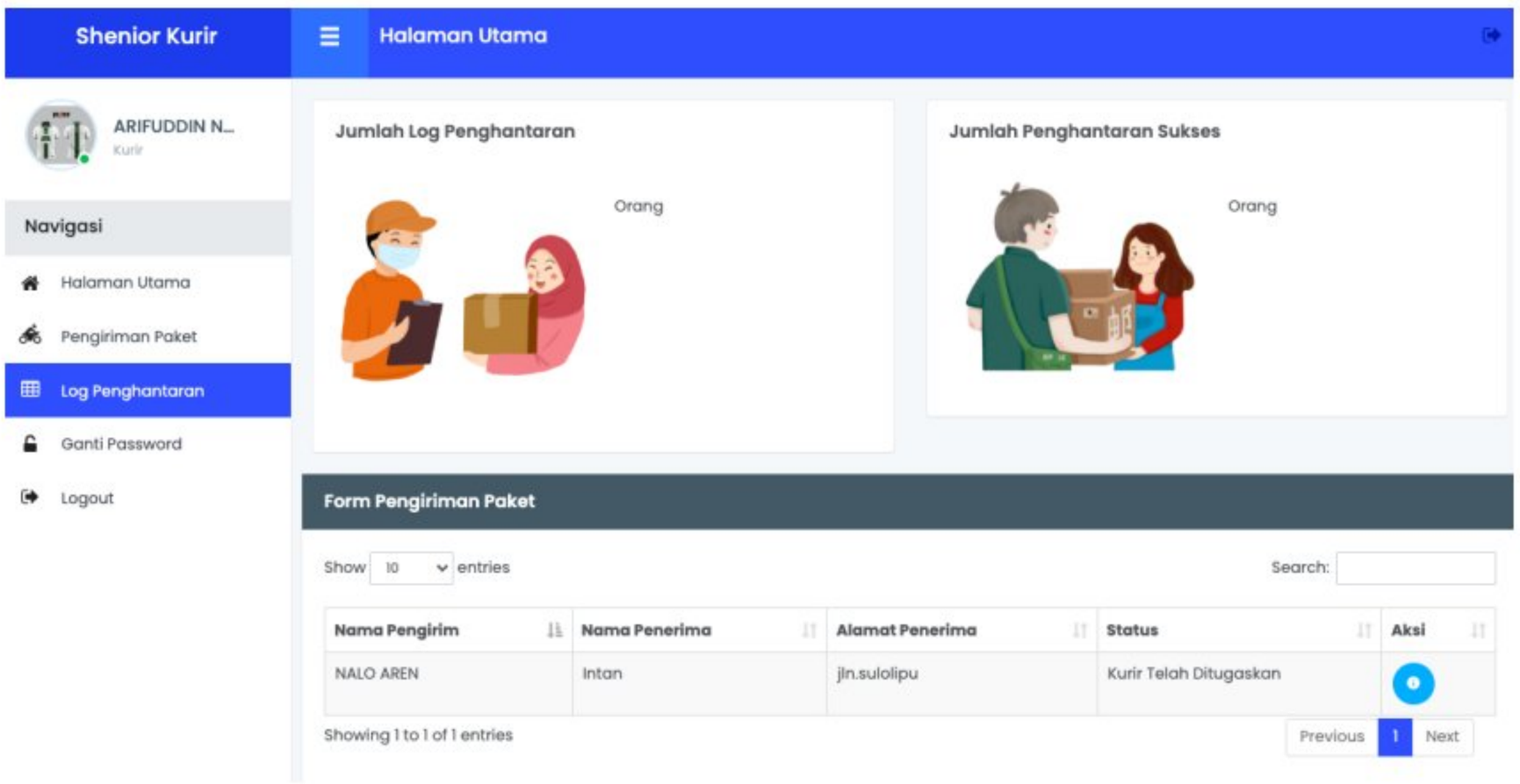
c. Informasi Pengiriman Paket



Gambar 4.18 Dialog Informasi Pengiriman Paket

Diatas merupakan dialog detail pengiriman paket, disini kurir dapat melihat informasi seputar pengiriman paket seperti nama pengirim dan penerima, alamat pengirim dan penerima, progres penghantaran paket dan melakukan aksi seperti membatalkan penerimaan paket dengan alasan, mengambil paket, dan menghantar paket yang mana notifikasi diterima oleh user dan admin

d. Log Penghantaran

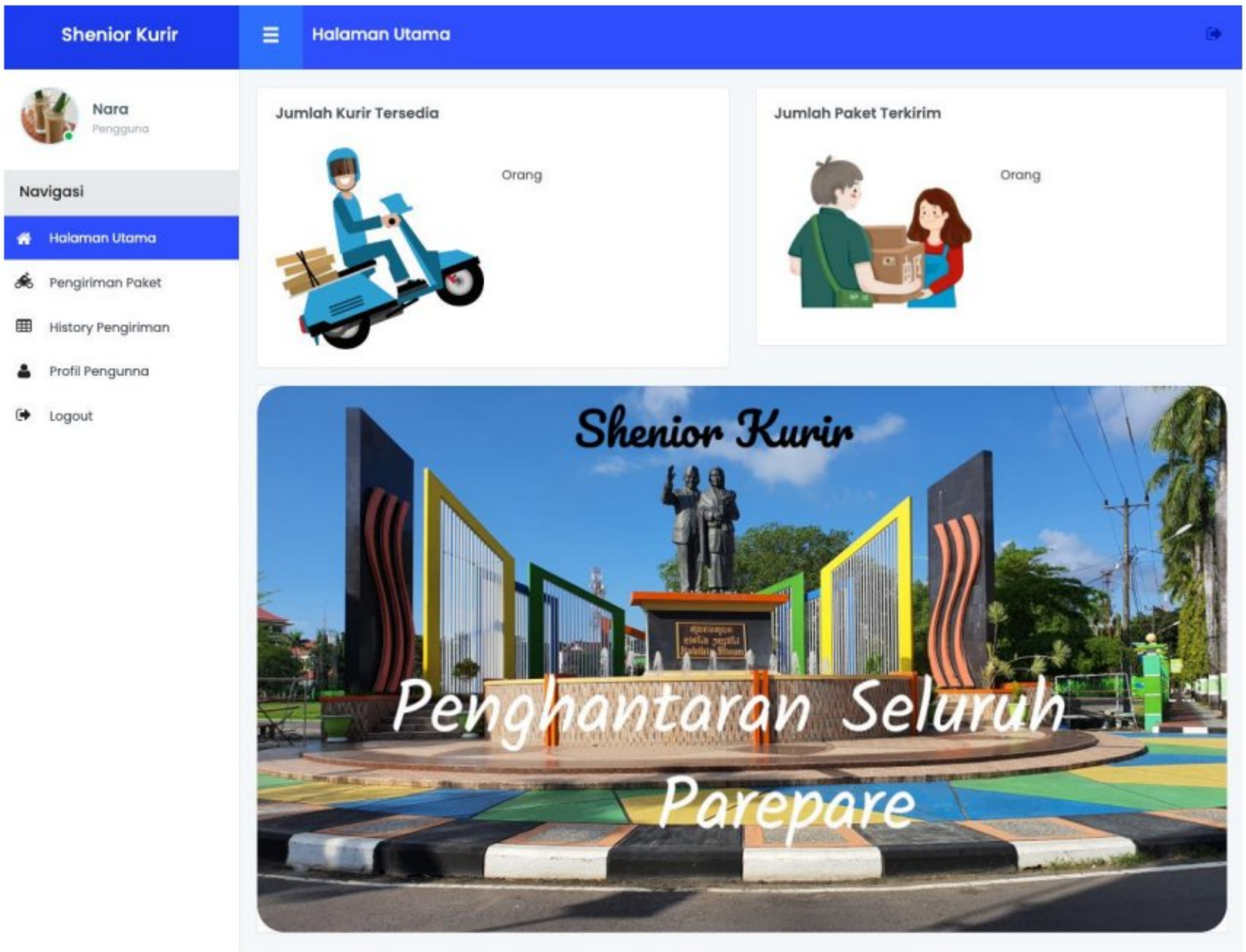


Gambar 4.19 Log Penghantaran

Diatas merupakan halaman Log Penghantaran, disini admin dapat melihat log penghantaran yang telah dilakukan sebelumnya ataupun yang sedang berjalan.

3. User

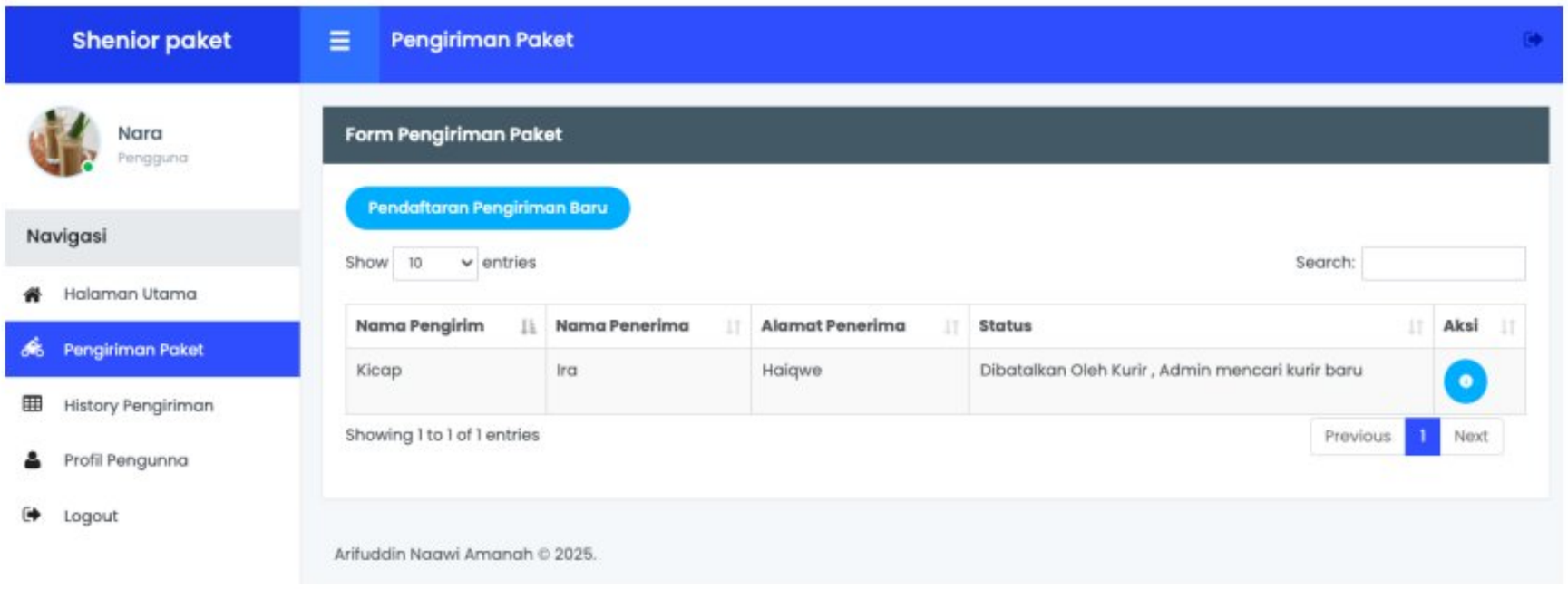
a. Halaman Utama



Gambar 4.20 Halaman Utama

Diatas merupakan halaman utama untuk user, disini user dapat melihat jumlah kurir yang tersedia , dan juga jumlah paket terkirim yang dilakukan oleh user.

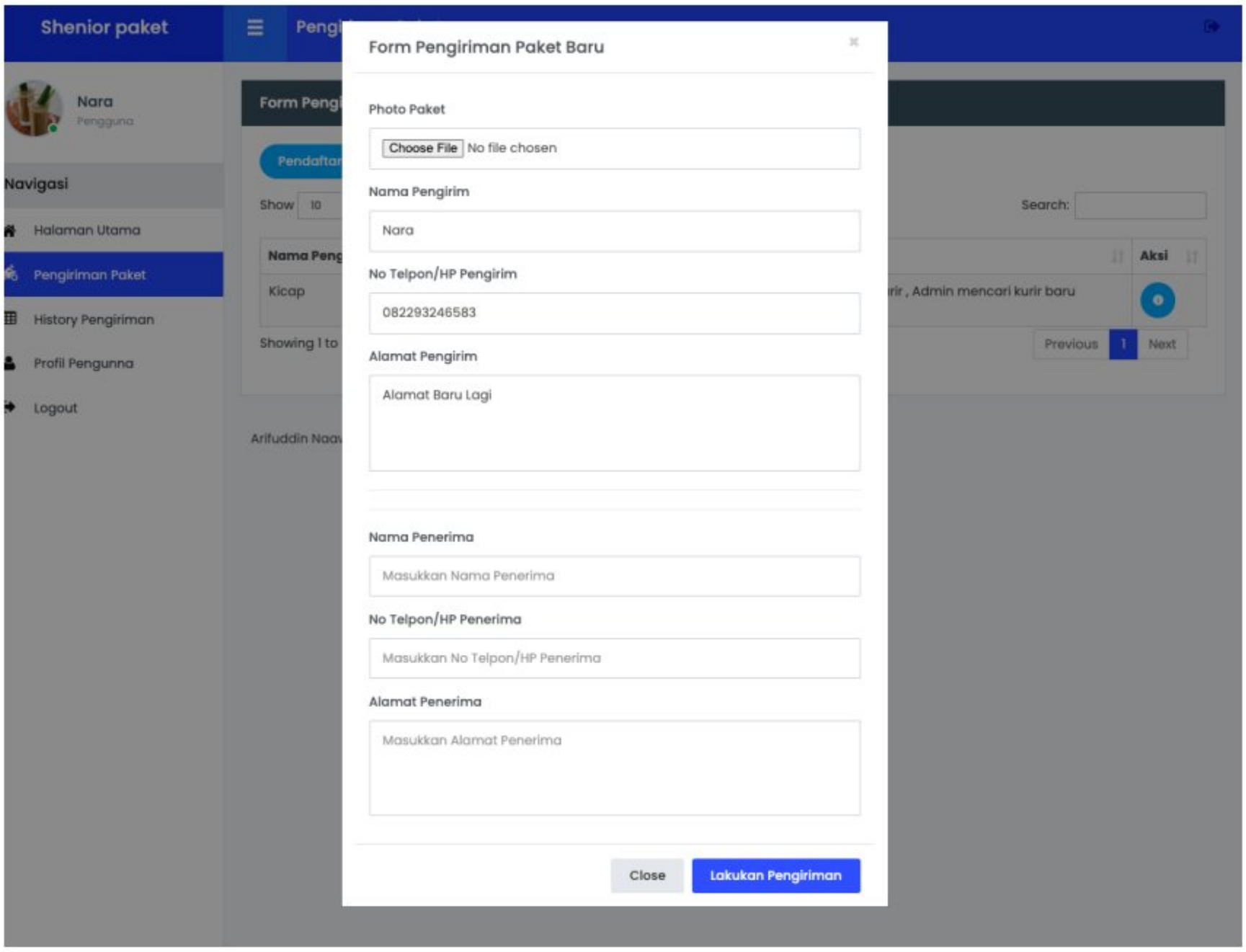
b. Pengiriman Paket



Gambar 4.21 Pengiriman Paket

Diatas merupakan halaman pengiriman paket, disini user dapat melihat list progres pengiriman yang sedang berjalan, dan juga melakukan pengiriman paket baru, jika ada perubahan status berkaitan pengiriman paket, maka user akan menerima notifikasi di sistem dan juga Whatsapp.

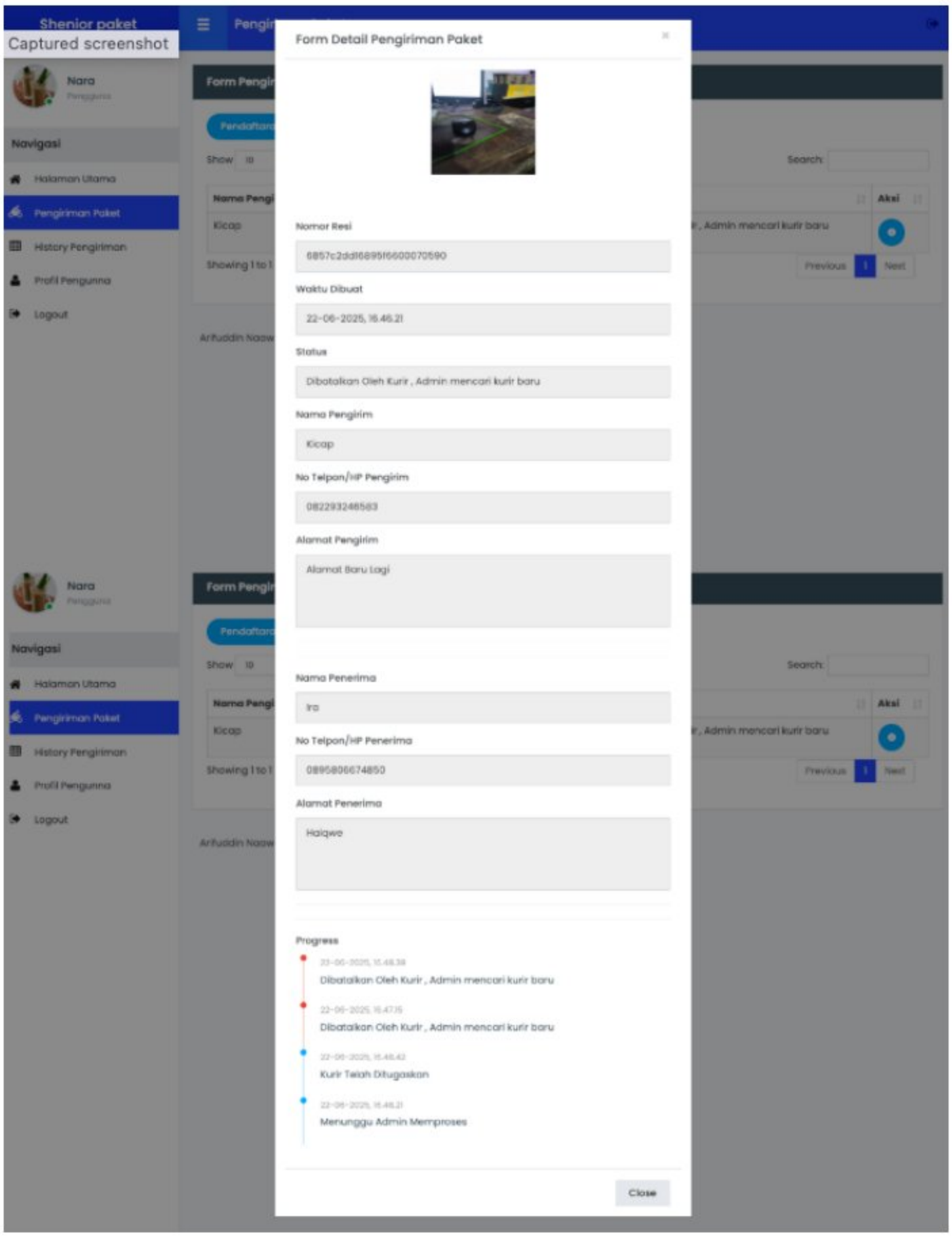
c. Form Pengiriman Paket



Gambar 4.22 Form Pengiriman Paket

Diatas merupakan form pengiriman paket, disini kurir dapat mendaftarkan pengiriman paket dengan memasukkan foto paket, nama pengirim, nomor telpon pengirim alamat pengirim, nama penerima , nomor telpon penerima dan alamat penerima. Tujuan data ini adalah untuk memudahkan kurir dalam menghubungi baik pengirim dan penerima jika ada kendala dalam pengiriman.

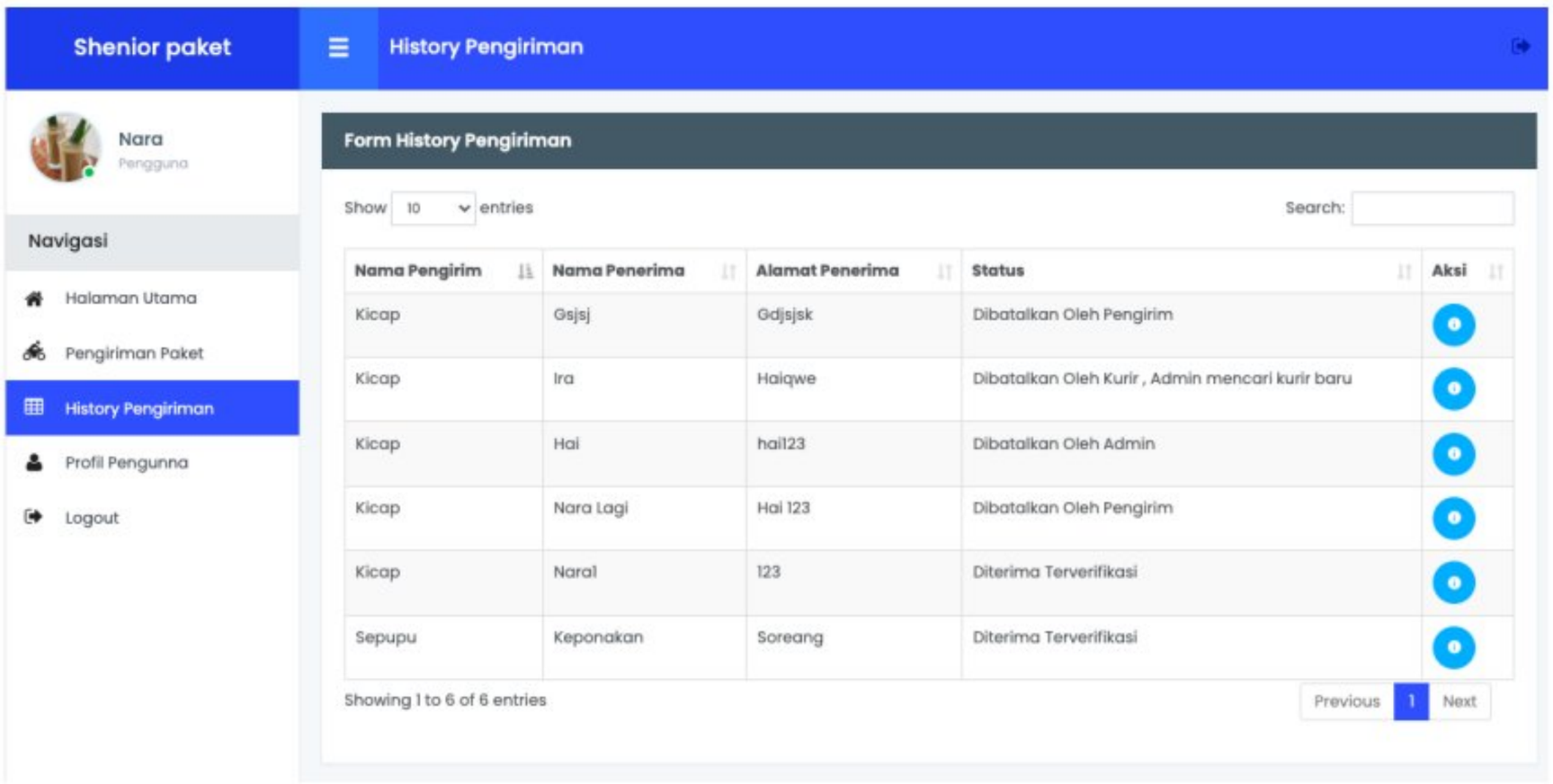
d. Form Detail Pengiriman Paket



Gambar 4.23 Form Detail Pengiriman Paket

Diatas merupakan form detail pengiriman paket, disini user dapat melihat status pengiriman paket, jika ada perubahan status pengiriman paket, maka user akan menerima notifikasi baik dalam sistem maupun melalui whatsapp

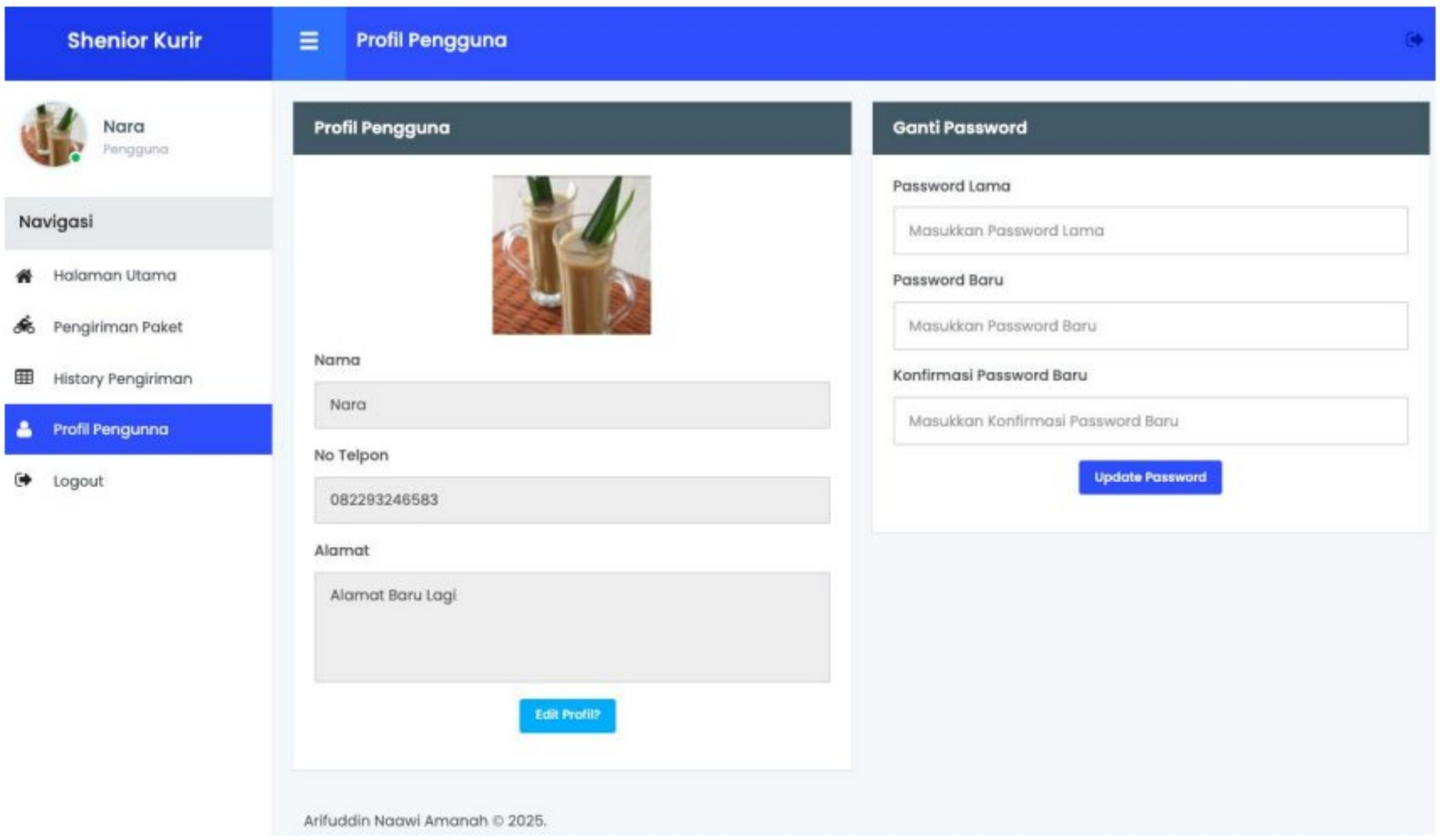
e. History Pengiriman



Gambar 4.24 History Pengiriman

Diatas merupakan history pengiriman ,baik yang telah berlaku maupun yang sedang berjalan, disini user dapat melihat informasi pengiriman yang telah dilakukan.

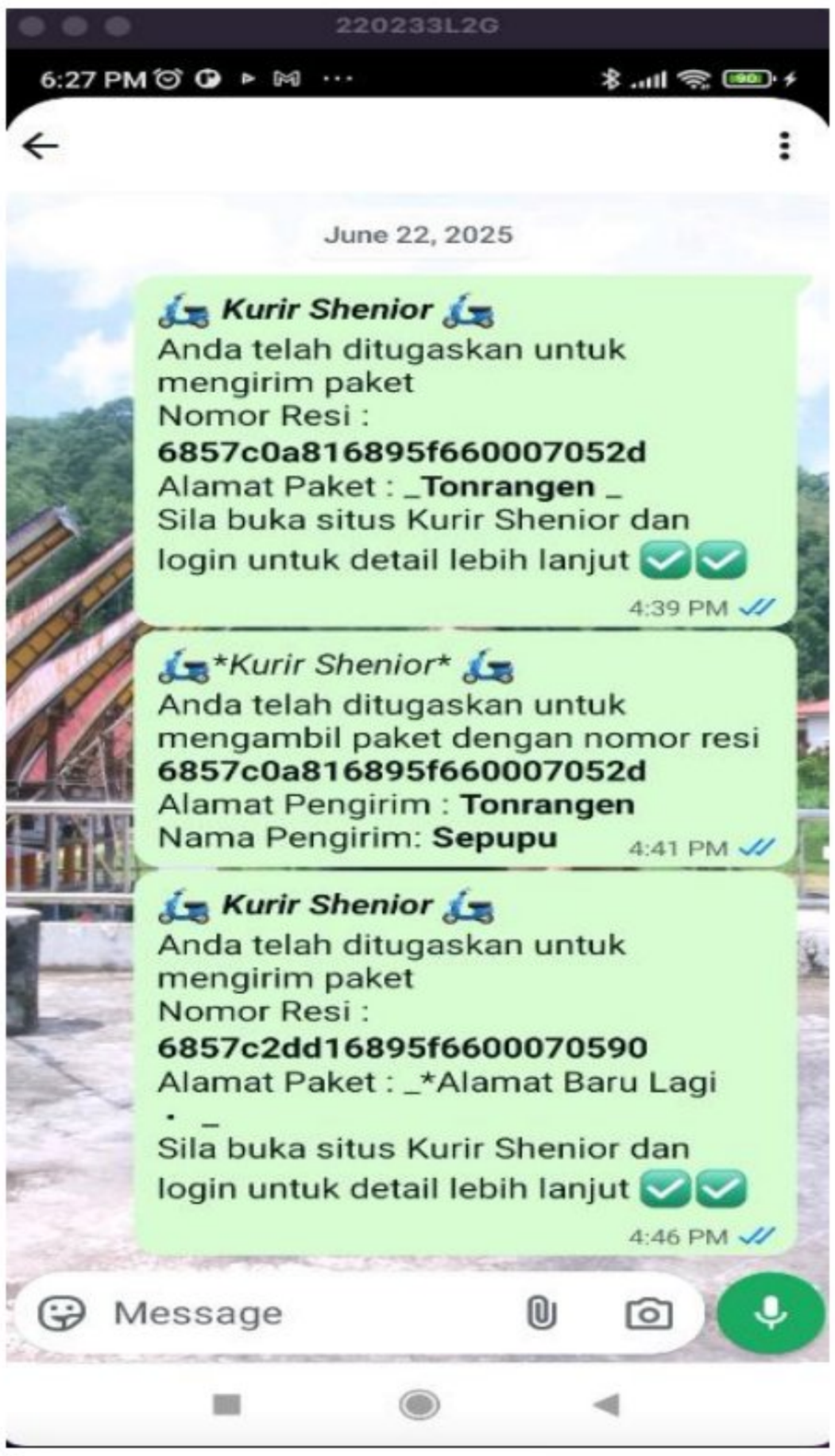
f. Profil Pengguna



Gambar 4.25 Profil Pengguna

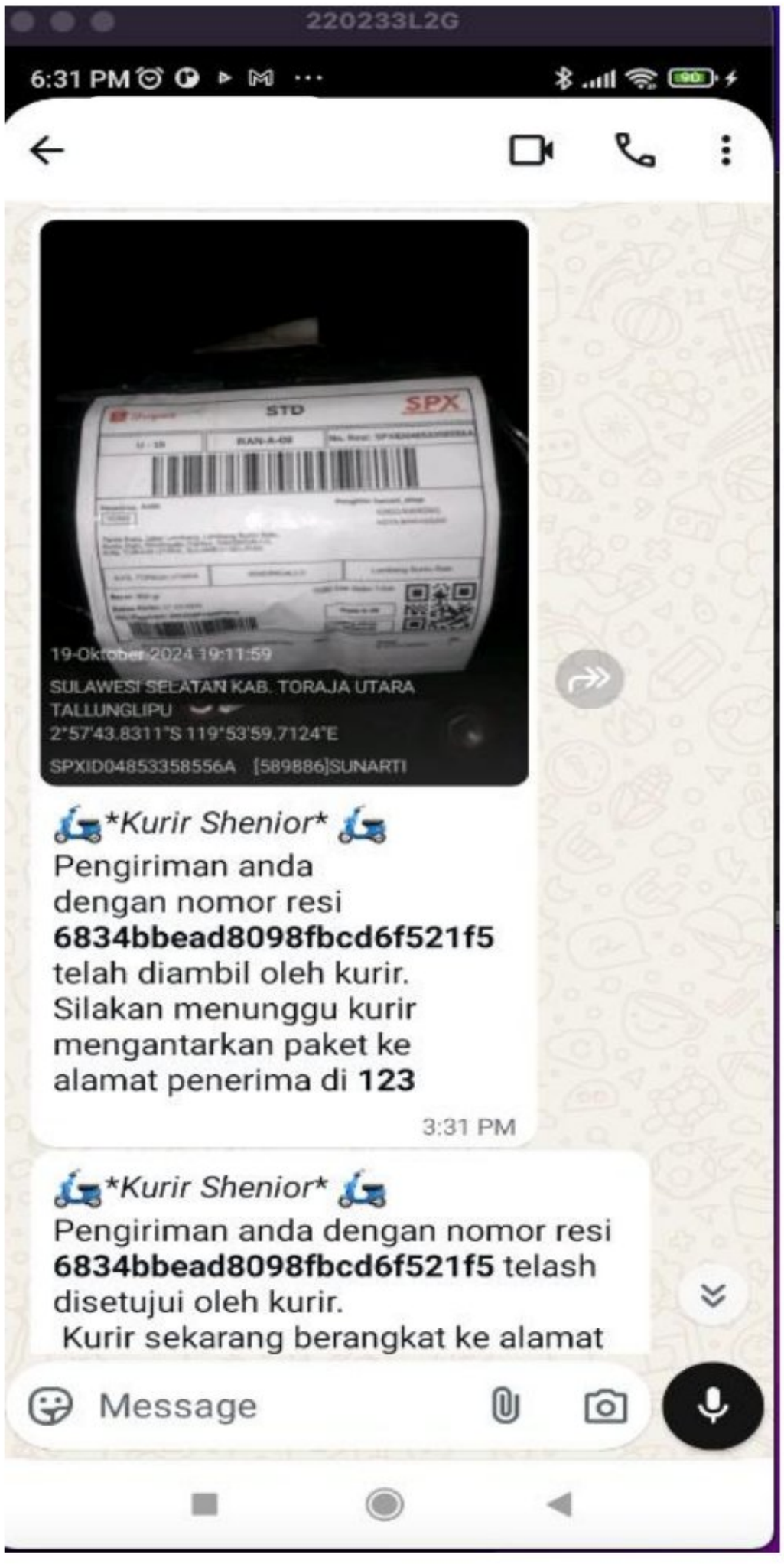
F. Notifikasi Whatsapp

Selain notifikasi dikirimkan ke sistem jika ada perubahan status pengiriman paket, sistem juga mengirim notifikasi kepada whatsapp user dan juga kurir yang terlibat. Semua notifikasi dapat dilihat pada gambar dibawah.



Gambar 4.26 Notifikasi Ke Kurir

Diatas merupakan notifikasi ke kurir, seperti notifikasi kurir ditugaskan, mengambi paker, mengantar paket dan lainnya.



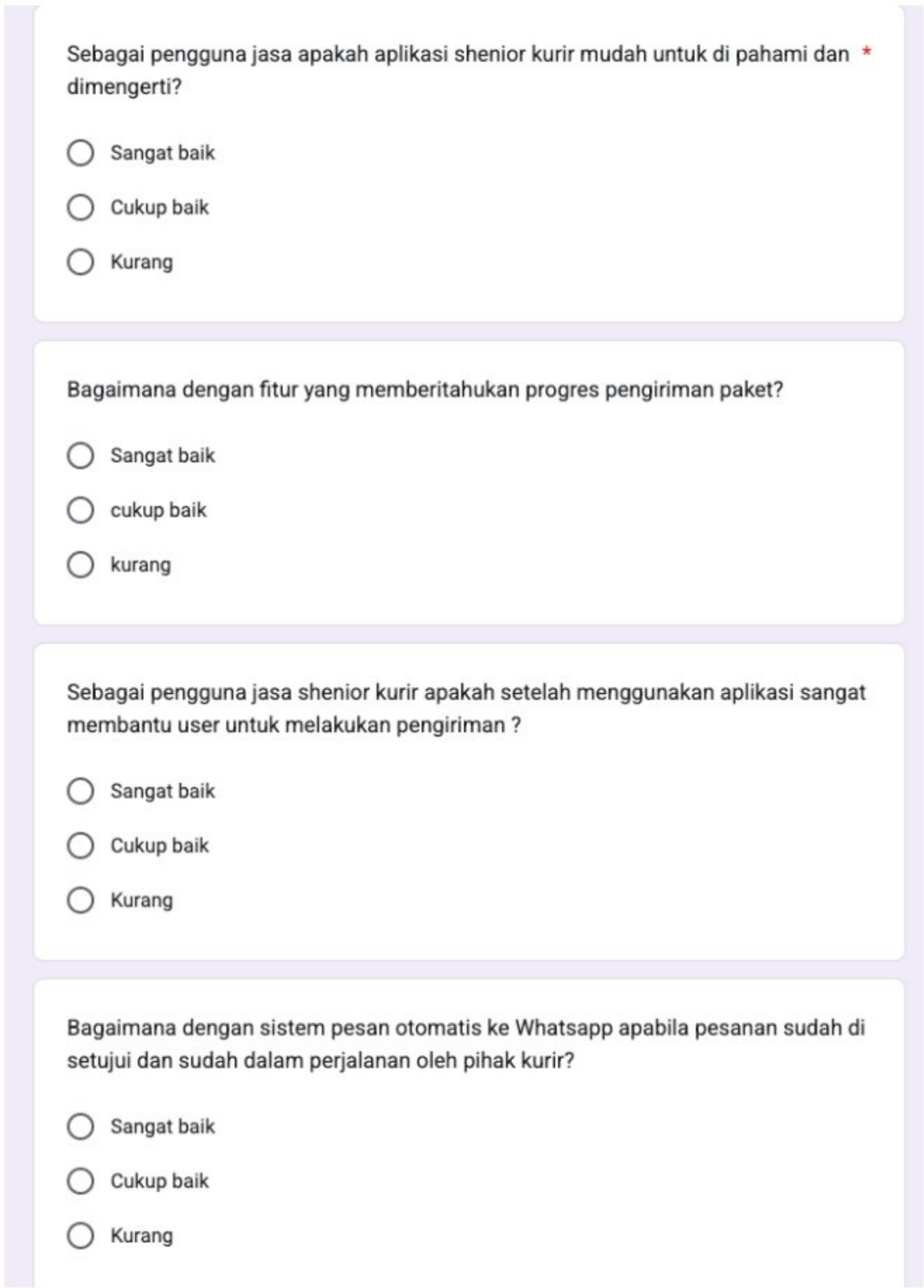
Gambar 4.27 Notifikasi Ke User

Diatas merupakan notifikasi kepada user, seperti notifikasi paket diverifikasi oleh admin, kurir telah ditugaskan, kurir dalam perjalanan mengambil paket, dan kurir dalam perjalanan menghantar paket

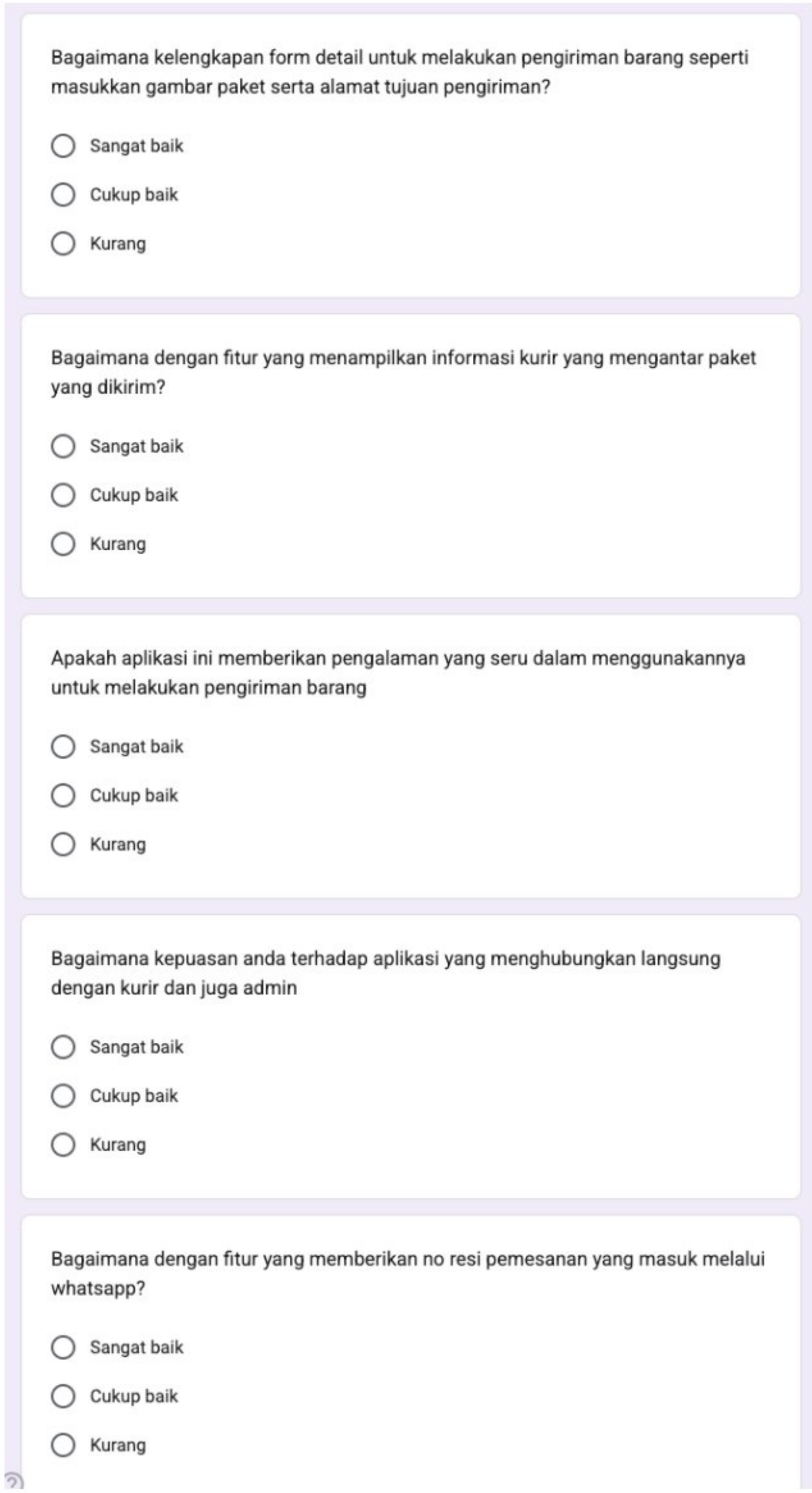
G. Kuestioner

Peneliti juga membuat kuestioner baik untuk kurir maupun pengguna aplikasi, dimana pada kuestioner ini menanyakan baik kurir dan pengguna berkaitan kepuasan menggunakan jasa Kurir Shenior. Dibawah merupakan kuestioner kurir dan pengguna berserta jawabannya.

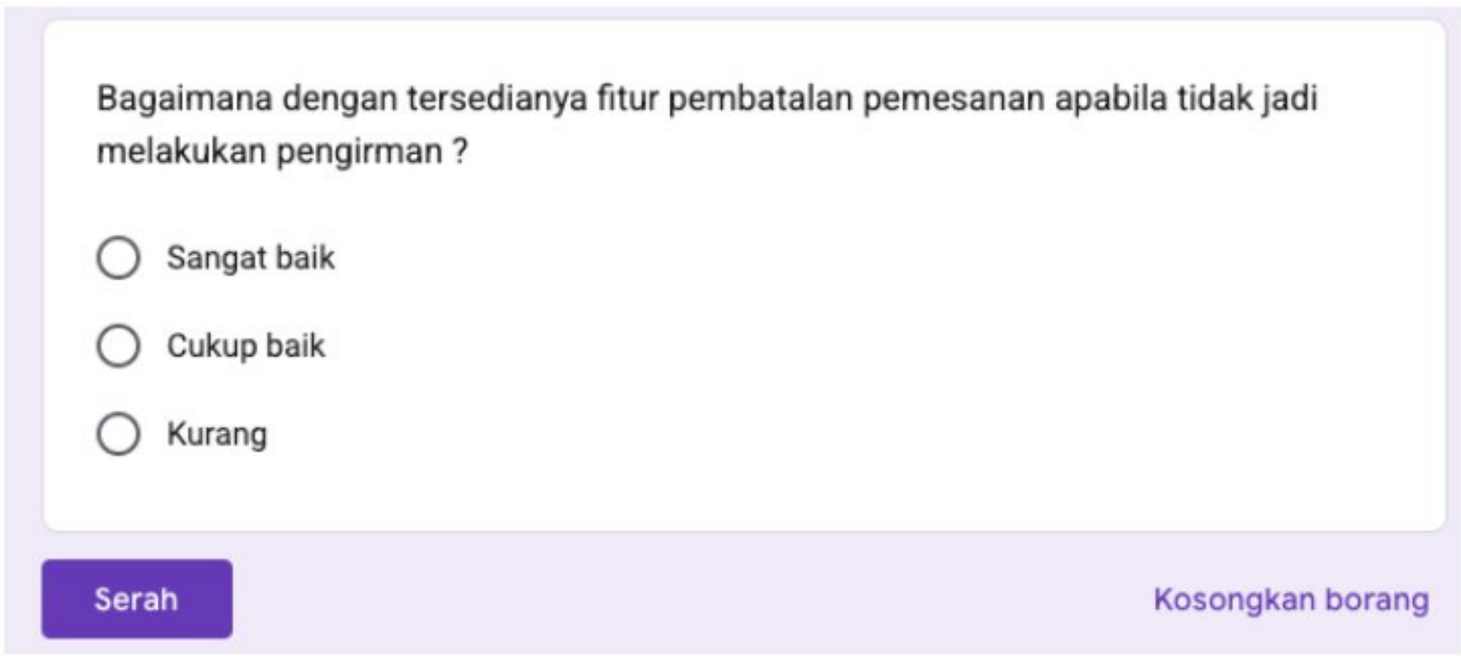
1. Kurir



Gambar 4.28 Pertanyaan Kurir 1

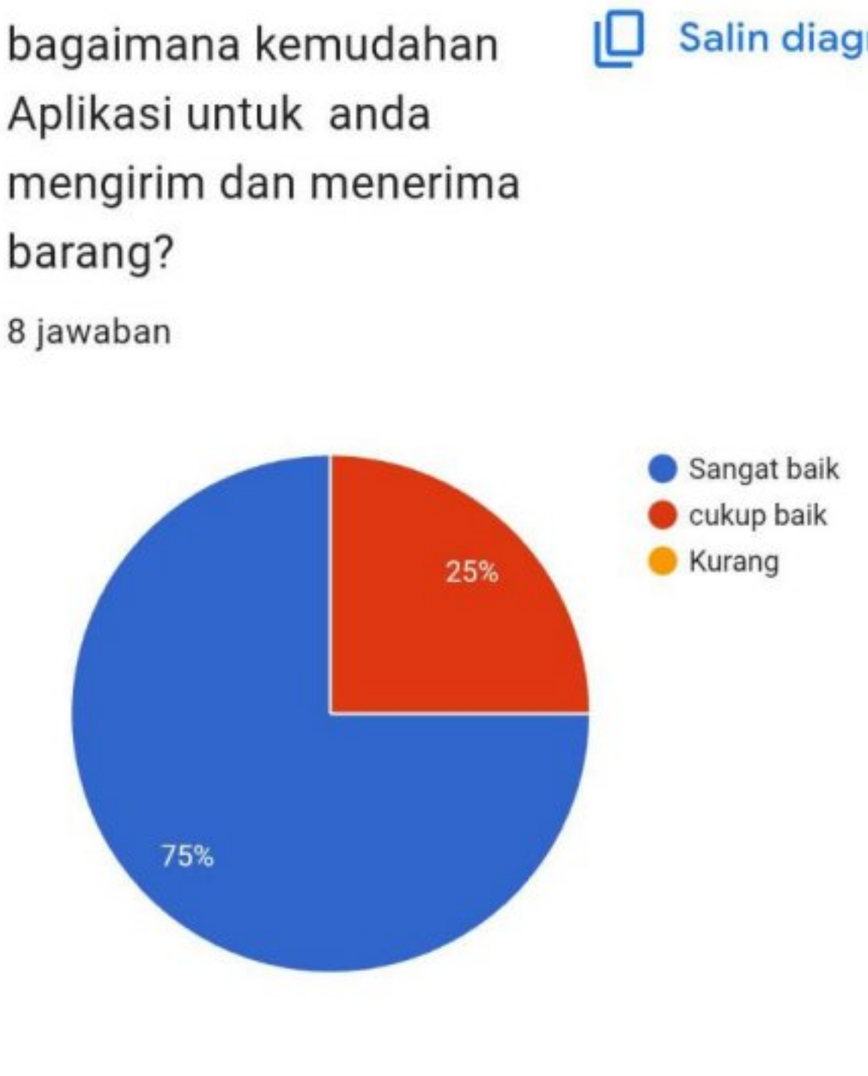


Gambar 4.29 Pertanyaan Kurir 2



Gambar 4.30 Pertanyaan Kurir 3

Diatas merupkan kuestioner *google form* yang diberikan untuk kurir, disini terdapat sepuluh pertanyaan yang akan dijawab kurir, tujuan pertanyaan ini adalah untuk menghasilkan tingkat kepuasan penggunaan aplikasi di kalangan kurir, dan di bawa merupakan hasil dari kuestioner yang diberikan.

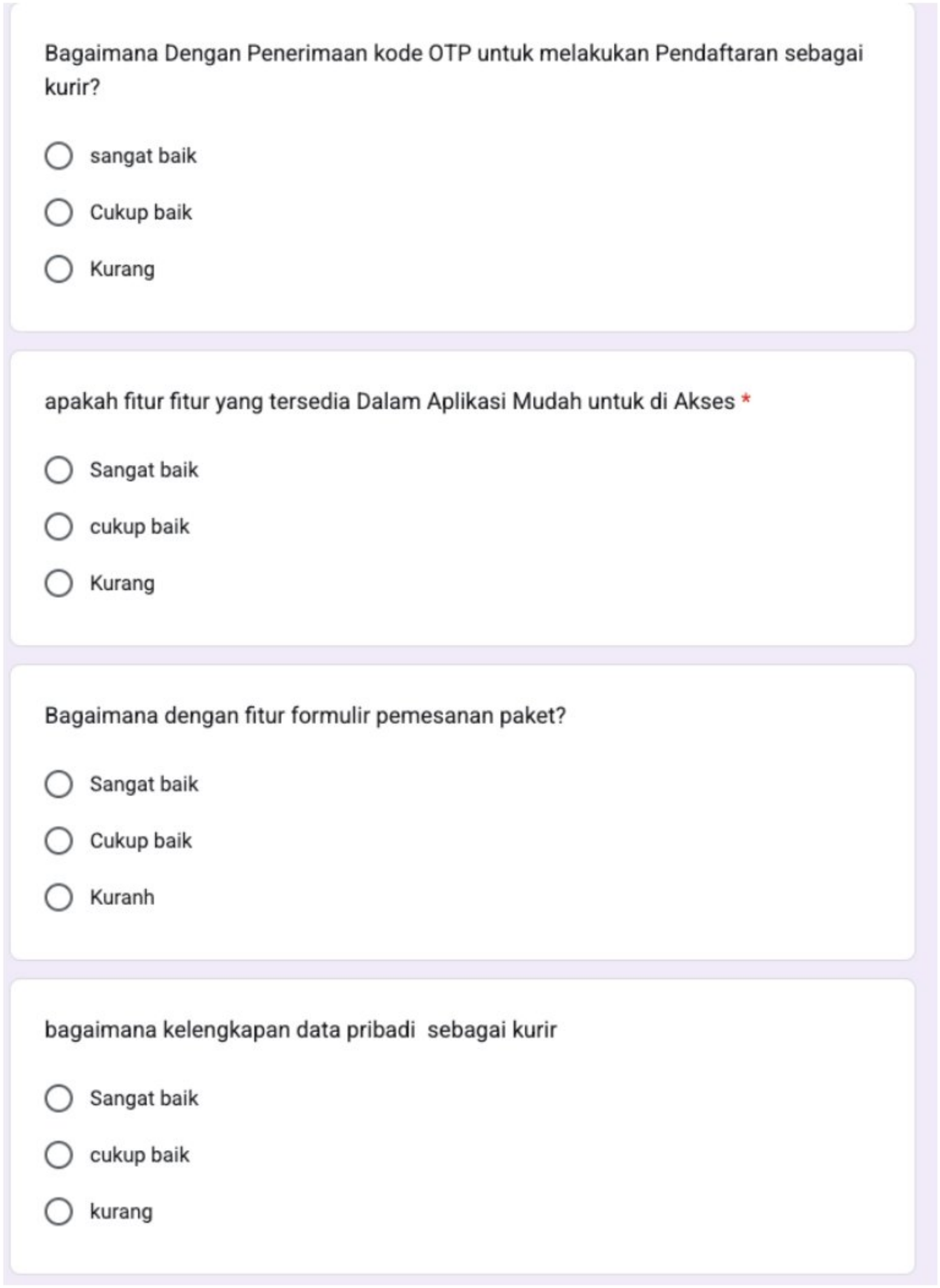


Gambar 4.31 Hasil Kuestioner Kurir

Berdasarkan dari hasil kuestioner dari 8 kurir yang bertugas di Kurir Shenior, terlihat bahwa 75 persen atau sekitar 6 orang memberikan tingkat

kepuasan sangat baik manakala 2 orang memberikan tingkat kepuasan baik. Peneliti menyimpulkan bahwa aplikasi memberikan tingkat kepuasan diatas 80 persen dan bahwa semua kurir memberikan tanggapan baik berkaitan aplikasi di dibuat.

2. Pengguna



Gambar 4.32 Pertanyaan Pengguna 1

Bagaimana Dengan fitur Log Penghantaran dalam aplikasi?

- ◯ Sangat baik
- ◯ cukup baik
- ◯ Kurang

Sebagai kurir apakah aplikasi Ini mudah untuk di pahami dan di mengerti terkait Kinerja aplikasi? *

- ◯ Sangat baik
- ◯ Cukup baik
- ◯ Kurang

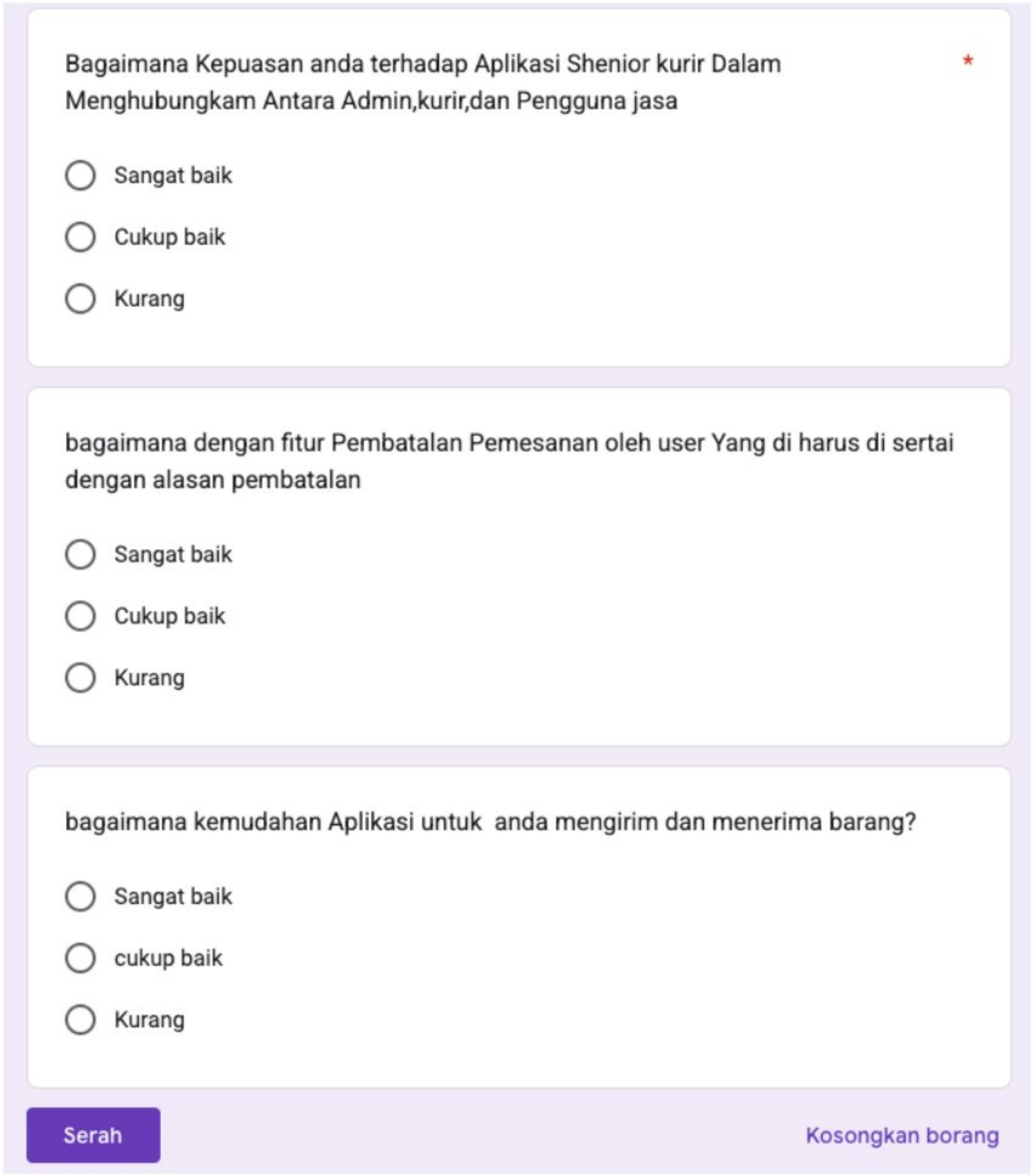
Bagaimana Detail Informasi Pemesanan barang Dari Pihak Pengguna jasa *

- ◯ Sangat baik
- ◯ Cukup baik
- ◯ kurang

Bagaimana pengalaman menggunakan aplikasi (user experience) *

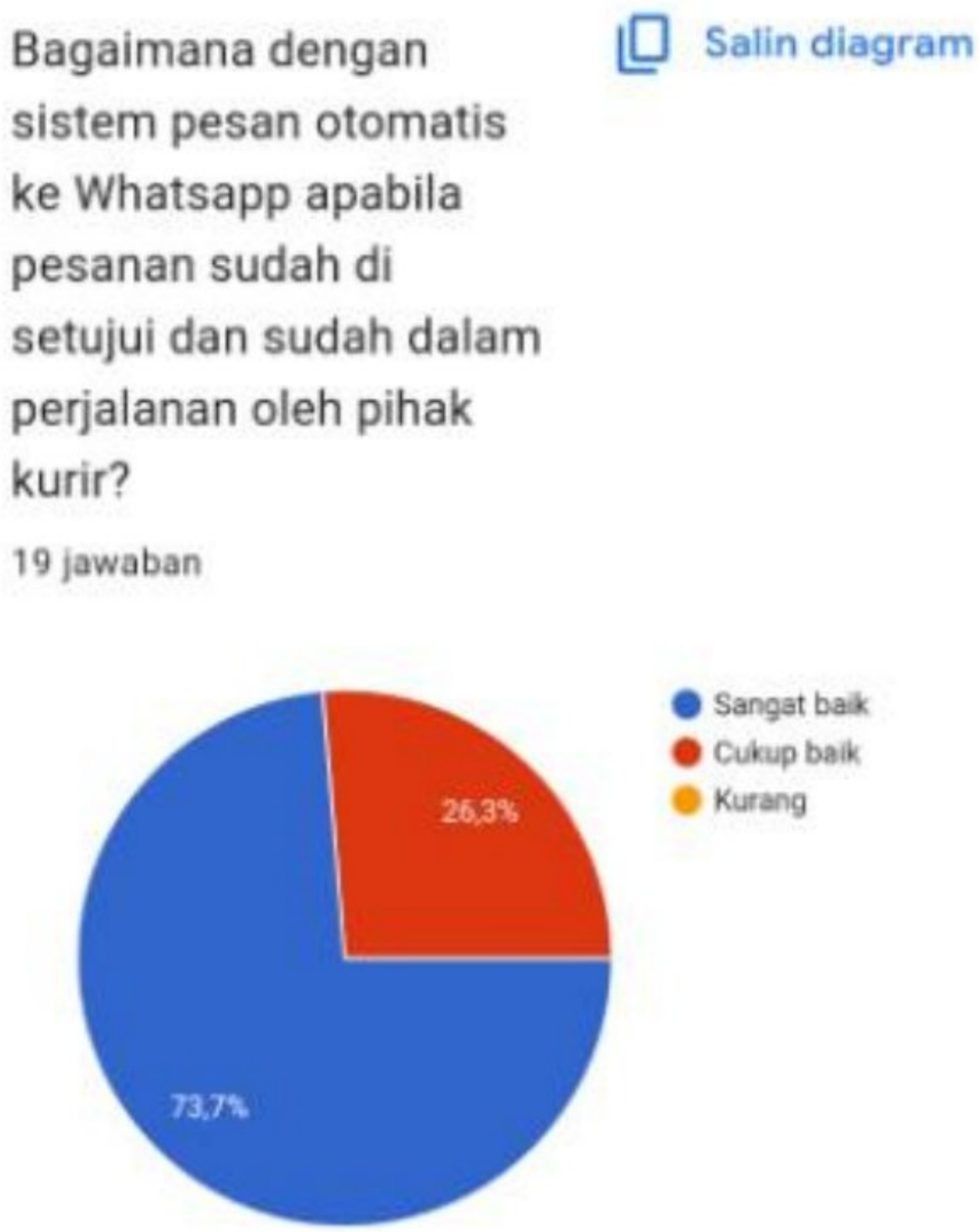
- ◯ Sangat baik
- ◯ Cukup baik
- ◯ Kurang

Gambar 4.33 Pertanyaan Pengguna 2



Gambar 4.34 Pertanyaan Pengguna 3

Diatas merupkan kuestioner *google form* yang diberikan untuk pengguna, disini terdapat sepuluh pertanyaan yang akan dijawab pengguna, tujuan pertanyaan ini adalah untuk menghasilkan tingkat kepuasan penggunaan aplikasi di kalangan pengguna, dan di bawa merupakan hasil dari kuestioner yang diberikan.



Gambar 4.35 Hasil Pertanyaan Pengguna

Berdasarkan dari hasil kuestioner dari 19 pengguna yang menggunakan aplikasi Kurir Shenior, terlihat bahwa 73.7 persen atau sekitar 14 orang memberikan tingkat kepuasan sangat baik manakala 5 orang memberikan tingkat kepuasan baik. Peneliti menyimpulkan bahwa aplikasi memberikan tingkat kepuasan diatas 80 persen dan bahwa semua pengguna memberikan tanggapan baik berkaitan aplikasi di dibuat.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkannya sebuah Aplikasi Shenior Kurir Jasa Antar Barang Berbasis PWA yang mana dapat disimpulkan sebagai berikut :

1. Penelitian ini berhasil merancang dan membangun sebuah Aplikasi Shenior Kurir Jasa Antar Barang Berbasis PWA.
2. Aplikasi ini dibuat untuk memudahkan sharing informasi baik untuk user, kurir dan admin.
3. Aplikasi ini dibuat agar log pengiriman paket yang telah berlaku dapat disimpan dan dilihat kembali baik untuk user, kurir dan admin
4. Aplikasi ini bersifat *user friendly* yang mana user, kurir dan admin dapat mengerti dan dengan mudah mengoperasikan aplikasi ini.
5. Berdasarkan kuestioner yang diberikan kepada kurir , sekitar 80 pesen kurir memberikan tanggapan yang sangat baik tentang layanan Kurir Shenior dan aplikasi.
6. Berdasarkan kuestioner yang diberikan kepada pengguna, sekitar 75 persen penggunakan memberikan tanggapan yang sangat baik tentang layanan Kurir Shenior dan aplikasi

B. Saran

Saran yang dapat diberikan penulis adalah seperti berikut yaitu aplikasi dapat dikembang ke arah lebih baik dengan menambah lebih banyak fitur-fitur berkaitan dengan Kurir Shenior.

DAFTAR PUSTAKA

Creswell, John W. (2015). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.

Denny Pribadi, R., et al. (2020*). Sistem Pendukung Keputusan Pemilihan Perumahan Menggunakan Metode Simple Additive Weighting (SAW) dan VIKOR.* Jurnal Teknologi Informasi dan Komputer, 10(1), 43-50.

Hadi, H., & Pratama, R. (2019*). Application of Simple Additive Weighting (SAW) and Additive Ratio Assessment (ARAS) Methods in Decision Support System for Determining Strategic Land Locations for Housing Development*. Journal of Engineering and Management in Industrial System, 7(1), 1-8.

Kurniawan, A., & Sari, D. K. (2021). *Analisis Multi Kriteria untuk Penentuan Lokasi Pembangunan Pusat Perbelanjaan Menggunakan Metode AHP dan SAW*. Jurnal Manajemen Teknologi, 10(3), 89-98.

Nugroho, L. E., & Suryana, N. (2020). *Decision Support System for Strategic Land Location Determination for Housing Development Using Simple Additive Weighting (SAW) and Additive Ratio Assessment (ARAS) Methods*. Journal of Housing and Settlements, 7(2), 87-96.

Putra, A. D., & Ramadhan, R. (2020). *Sistem Pendukung Keputusan Pemilihan Lokasi Ritel Menggunakan Metode SAW.* Jurnal Teknik Informatika, 6(1), 45-55.

Rahman, A., & Saudin, D. (2022). *Sistem Pendukung Keputusan (1st ed.)*. UIN Maulana Malik Ibrahim Malang.

Santoso, H. B., & Wijaya, A. B. (2021*). A Comparative Study of Simple Additive Weighting (SAW) and Additive Ratio Assessment (ARAS) Methods in Decision Making for Strategic Land Location Determination for Housing Development*. International Journal of Housing and Urban Development, 14(3), 45-52.

Setiyaningsih, D. (2017). *Sistem Pendukung Keputusan (2nd ed.)*. Universitas Negeri Semarang.

Sugiyono, Sugiyono. (2017). *Metode Penelitian Kuantitatif, Kualitatif, dan Kombinasi.* Alfabeta.

Susanto, A. (2017). *Penerapan Metode ARAS (Additive Ratio Assessment) untuk Menentukan Peringkat Alternatif pada Sistem Pendukung*

Keputusan Pemilihan Perumahan. Jurnal Sains dan Teknologi Universitas Muhammadiyah Surakarta, 10(1), 1-12.

Sugiyono, B., & Dewi, A. P. (2022). *Implementasi Metode SAW dan ARAS dalam Sistem Pendukung Keputusan untuk Penentuan Lokasi Pembangunan Restoran*. Jurnal Teknologi Informasi, 8(1), 100-110.

Wijaya, M. A., & Hartono, B. (2019). *Penerapan Metode ARAS dalam Penentuan Lokasi Gudang Logistik.* Jurnal Sistem Informasi, 15(2), 123-133.

Yin, Robert K. (2016). *Case Study Research: Design and Methods*. Sage Publications.

Yusriadi, A., et al. (2019). *Sistem Pendukung Keputusan Dalam Penyeleksian Anggota Paskibraka Tingkat Kabupaten Dengan Menggunakan Simple Additive Weighting (SAW) Dan Metode TOPSIS*. Jurnal Manajemen Teknologi Informasi, 1(1), 1-10.