

# PENGEMBANGAN BACKEND APLIKASI LAYANAN INTEGRASI LOGISTIK BERBASIS WEB MENGGUNAKAN LARAVEL

1<sup>st</sup> Muhammad Raga Devara Kusumo Widiyanto  
Fakultas Ilmu Terapan  
Universitas Telkom  
Bandung, Indonesia  
ragadevara@student.telkomuniversity.ac.id

2<sup>nd</sup> Muhammad Rizqy Alfarisi  
Fakultas Ilmu Terapan  
Universitas Telkom  
Bandung, Indonesia  
mrizkyalfarisi@telkomuniversity.ac.id

**Abstrak**— Dalam industri pengiriman barang, efisiensi dan efektifitas layanan logistik yang baik menjadi faktor kunci untuk mencapai keberhasilan dan kepuasan pelanggan. Dalam era digital saat ini, layanan integrasi menjadi solusi yang signifikan untuk mengatasi kekurangan dalam mengintegrasikan berbagai layanan logistik untuk meningkatkan efisiensi dan efektifitas pengiriman barang. Sehingga dikembangkanlah sistem berbasis website yang dirancang khusus untuk mengintegrasikan layanan logistik yang disediakan penyedia layanan.

Website yang dikembangkan memiliki berbagai fitur yang memudahkan perusahaan shipping untuk mengatur pengiriman barang, melacak lokasi barang, mengatur biaya operasional. Website ini juga mengintegrasikan berbagai perusahaan sehingga perusahaan – perusahaan tersebut dapat saling bekerjasama untuk mengelola operasi logistik mereka.

Penelitian ini bertujuan untuk mengembangkan website yang dapat dipakai untuk mengintegrasikan layanan logistik bagi perusahaan pengiriman barang dengan fokus pada pengoptimalan proses pengelolaan pengiriman. Website yang di buat juga telah dilakukan pengujian untuk memastikan bahwa fitur – fitur dan berbagai fungsi dapat bekerja dengan baik dan benar. Dari hasil pengujian yang telah dilakukan dapat diketahui bahwa website yang dibuat dapat membantu perusahaan shipping atau pengiriman dalam melakukan integrasi layanan logistik untuk mengoptimalkan kinerja pengiriman.

**Kata kunci:** Logistic Service Integrator, Optimalisasi pengiriman, Layanan logistik, Laravel.

## I. PENDAHULUAN

### A. Latar Belakang

Tidak dapat dipungkiri perkembangan teknologi saat ini sudah sangat pesat dan telah digunakan dalam kehidupan kita sehari-hari. Sehingga dapat kita ketahui bahwa teknologi berperan sangat penting untuk menunjang kehidupan, salah satunya pada bidang bisnis. Teknologi dapat berperan untuk meningkatkan indikator penting dalam dunia bisnis yaitu efektifitas dan produktifitas, sehingga sudah banyak perusahaan yang telah menggunakan teknologi yang tepat untuk menunjang ataupun meningkatkan efektifitas dan produktifitas mereka.

Berkaitan dengan hal ini teknologi juga sudah mulai diterapkan pada bisnis *Shipping Company* yang merupakan perusahaan yang bergerak dalam bidang jasa kirim dan ekspedisi barang dimana bisnis ekspedisi barang ini mulai mengalami kemajuan yang cukup signifikan. Kemajuan ini dapat diketahui salah satunya karena dampak dari pandemi yang telah mengubah kebiasaan belanja konsumen. Tercatat bahwa peningkatan konsumen pada e-commerce selama masa pandemi meningkat 51% namun proses pengiriman atau distribusi barang mengalami keterlambatan sehingga hal ini tentu saja membuat perusahaan jasa kirim untuk meningkatkan operasional jasa kirimnya dan melakukan otomatisasi sehingga lebih praktis dan dapat meningkatkan efektifitas dari pengirimannya

Banyak perusahaan yang bergerak dalam bidang pengiriman (*shipping*) masih memakai cara manual untuk

melakukan operasional pengiriman sehingga cukup menguras waktu dan kurang efektif, lalu di masa sekarang ini juga masih dalam masa covid-19 sehingga membatasi kontak langsung antar *shipment*, *agregator*, dan *vendor*. Karena itulah sangat penting dan dibutuhkan sistem untuk menghubungkan antara berbagai perusahaan secara praktis dan efektif. Sehingga penerapan teknologi pada bisnis *Shipping Company* ini dipakai untuk mengintegrasikan antara *Shipping Company*, *Aggregator*, dan jasa pengiriman. Dengan peranan teknologi untuk saling terintegrasi inilah yang dapat memudahkan dan meningkatkan kinerja dari perusahaan untuk melakukan jasa kirim dan ekspedisi barang.

Berdasarkan hal tersebut dikembangkanlah suatu sistem LSI (*Logistic Service Integrator*) yaitu sistem berbasis website dengan tujuan untuk mengintegrasikan *Shipment Company* yang berperan sebagai perusahaan untuk produksi barang dengan *Logistic Shipment Agregator* dimana akan berperan sebagai distributor barang yang nantinya akan bekerjasama dengan vendor sebagai penyedia kendaraan dan driver untuk melakukan jasa kirim. Lalu sistem LSI ini juga memungkinkan *vendor* untuk melakukan *realtime tracking* pada kendaraan mereka sehingga kendaraan dapat terpantau secara aman. Dengan sistem yang dikembangkan ini perusahaan *Shipping* dapat dipermudah dalam melakukan keperluan bisnisnya dalam bidang jasa pengiriman.

### B. Rumusan Masalah dan Solusi

Dalam pengembangan aplikasi *Logistic Service Integrator* yang berfokus pada layanan logistik untuk meningkatkan efisiensi pada jasa pengiriman barang didapatkan rumusan masalah, yaitu:

1. Bagaimana penerapan teknologi untuk mengoptimalkan produktifitas dan efektifitas pada operasional bisnis *Shipping Company*?

2. Bagaimana cara agar *Shipment Company*, *Logistic Shipment Agregator*, dan *Vendor* dapat terintegrasi?

Untuk mengatasi masalah-masalah di atas, maka dikembangkanlah aplikasi *Logistic Service Integrator* sehingga berikut adalah solusi dari rumusan masalah yang diidentifikasi:

- produktifitas dan efektifitas dari operasional bisnis *Shipping Company* dapat ditingkatkan dengan teknologi yang dapat menghubungkan antara pelaku bisnis yaitu *Shipment Company*, *Logistic Shipment Agregator*, dan *Vendor* secara efisien menggunakan aplikasi dimana dalam konteks penelitian ini yaitu menggunakan aplikasi *Logistic Service Integrator*.
- Shipment Company*, *Logistic Shipment Agregator*, dan *Vendor* dapat saling terintegrasi dengan saling bekerja sama menggunakan aplikasi yang dikembangkan yang juga merupakan fitur dari aplikasi *Logistic Service Integrator*.

Penelitian ini bertujuan untuk mencapai beberapa tujuan utama sebagai berikut:

1. Membangun sistem yang efektif berbasis website yang dapat mengintegrasikan antara *Shipping Company*, *Aggregator*, dan *Vendor*.
2. Dapat membangun sistem yang dapat mempermudah *Shipping Company* dalam operasionalnya untuk melakukan jasa kirim dan juga menampilkan informasi lokasi realtime vehicle dalam proses pengiriman.
3. Menyediakan transparansi dalam proses pengiriman barang yang dimulai dari permintaan order shipment sampai proses selesai pengiriman yang berguna bagi Perusahaan untuk dapat mengetahui kondisi proses pengiriman.

Dengan tujuan-tujuan yang telah dijabarkan, diharapkan bahwa penelitian yang dilakukan dapat memberi dampak positif bagi industri shipping dan logistic sehingga dapat meningkatkan operasional dan efisiensi industri logistik yaitu dalam proses bisnis shipping.

#### D. Kontribusi Penelitian

Penelitian ini berperan signifikan dalam kemajuan industri logistik yang terkait dengan bisnis pengiriman melalui pengembangan dan penerapan *Logistic Service Integrator*. Berikut beberapa kontribusi utama dari penelitian ini:

1. Pengembangan sistem *Logistic Service Integrator* berbasis Web yang dapat meningkatkan efisiensi dan produktifitas operasional dalam bisnis shipping dan logistik.
2. Meningkatkan layanan logistik dengan memberikan estimasi waktu pengiriman yang lebih akurat, melacak status pengiriman dengan lebih baik, dan memberikan transparansi yang lebih besar.
3. Menyediakan transparansi terhadap seluruh proses yang berkaitan dengan logistic dan shipping sehingga prosesnya dapat terpantau dengan jelas dan aman.

## II. METODOLOGI

Metodologi Agile adalah pendekatan pengembangan perangkat lunak yang didasarkan pada proses pengerjaan berulang yang berfokus pada fleksibilitas, kolaborasi, dan responsif terhadap perubahan kebutuhan. Dalam konteks pembangunan website integrasi layanan logistik, penerapan metodologi Agile digunakan untuk memastikan pengembangan solusi yang sesuai dengan kebutuhan yang berubah-ubah.

penerapan metodologi Agile dalam pengembangan layanan integrasi berbasis web untuk shipping company yaitu dimulai dari pembentukan tim yang dibagi menjadi 3 tim yaitu UI/UX, Frontend, dan Backend untuk mengidentifikasi fitur, merancang solusi, dan mengembangkan web. Selanjutnya perencanaan iteratif yang dilakukan dalam iterasi singkat yang disebut sprint, dimana pada awal sprint ini tim merencanakan secara rinci tentang berbagai fitur yang akan dikembangkan dan tujuan akhir yang akan dicapai. Selanjutnya yaitu pemilihan fitur prioritas. Pada saat sprint maka tim akan memprioritaskan fitur mana saja yang memiliki nilai tertinggi yang didasarkan pada kebutuhan untuk diimplementasikan terlebih dulu. Selanjutnya adalah pengembangan dan pengujian fitur yang hasilnya akan dipakai oleh pengguna sehingga apabila terdapat kekurangan maka dapat dilakukan perbaikan dan penyesuaian atau bahkan juga

dapat dilakukan pembaruan pada fitur yang ada sesuai keinginan pengguna. Lalu yang terakhir adalah adaptasi dan fleksibilitas. Dengan metodologi Agile ini dapat memberikan fleksibilitas dalam menanggapi perubahan kebutuhan atau tantangan yang muncul selama pengembangan. Baik dari segi perubahan fitur, prioritas ataupun kebutuhan baru maka tim dapat menyesuaikan rencana dan mengubah arah pengembangan dengan cepat.

## III. ANALISIS SISTEM

### A. Gambaran Sistem Saat Ini

Sistem saat ini yang diterapkan pada *Shipping Company* atau *Shipment Company* dan beberapa perusahaan yang bekerjasama dengan *Shipment Company* tersebut masih memakai cara manual dengan alur kerjanya yaitu perusahaan *Shipment Company* membuat produk dan menerima order dari produk tersebut ,selanjutnya *Shipment Company* akan melanjutkan ke *Agregator* sebagai distributor yang akan mengatur pengiriman, selanjutnya akan dilanjutkan ke *Vendor* untuk mengatur kendaraan dan *Driver* untuk melakukan pengirimannya.

Dalam proses kerjasama antar perusahaan seperti *Shipment Company*, *Agregator*, *Vendor*, dan *Driver* juga masih dilakukan secara manual dengan cara saling menghubungi lewat telepon ataupun dengan menghubungi lewat pesan. Sehingga hal tersebut menimbulkan proses yang tidak efisien dalam proses logistik dan shipping.

Untuk sistem yang akan dikembangkan saat ini masih dalam tahap perancangan. Pada perancangan ini telah dibuat struktur untuk *database*, beberapa desain , data apa saja yang dibutuhkan, juga perancangan dari fitur apa saja yang nantinya akan dibuat dibutuhkan. Dari rancangan inilah yang merupakan awal dari sebuah sistem yang akan dikembangkan nantinya.

### B. Pengembangan Sistem

Pengembangan sistem dari sisi *Backend Developer* merupakan pengembangan atau pembuatan database untuk berbagai fitur dari rancangan yang ada. Beberapa fitur yang dikembangkan dengan tujuannya atau fungsinya, yaitu sebagai berikut:

#### 1. Auth User

Ditujukan untuk *user (Admin, Shipment Company, Logistic Shipment Agregator, Vendor, Driver)* agar dapat *login* untuk mengakses fitur lain.

#### 2. User Management

Merupakan fitur untuk Admin agar bisa memanajemen *usernya* yaitu melakukan *create, read, update, delete data user*.

#### 3. Vendor Management

Merupakan fitur untuk *Logistic Shipment Agregator* untuk melihat daftar *vendor* yang bekerjasama agar bisa melakukan aktivasi atau deaktivasi status Kerjasama.

#### 4. Shipment Company Management

Merupakan fitur untuk *Logistic Shipment Agregator* untuk melihat daftar *Shipment Company* yang terdaftar agar bisa melakukan pemblokiran pada user *Shipment Company* yang melanggar kontrak.

#### 5. Profile Management

Merupakan fitur untuk *Logistic Shipment Agregator* untuk mengubah *profile company* agar bisa memanfaatkan untuk keperluan fitur *white label*.

#### 6. Management Pick up & Drop off Point

Merupakan fitur untuk *Logistic Shipment Agregator* untuk menyimpan data lokasi *pickup* dan *drop* milik *Shipment Company* agar *Shipment Company* bisa memilih lokasi *drop point* dan *pickup point* pada saat mengisi formulir *shipment*.

Merupakan fitur untuk *Shipment Company* untuk menginput *product* agar bisa memilih *product* yang akan dikirim saat mengisi *form order*.

8. Form Order dan SubOrder

Merupakan fitur untuk *Shipment Company* untuk membuat *order* agar barang dapat diproses untuk dikirimkan oleh *Logistic Shipment Agregator*.

9. Form Shipment

Merupakan fitur untuk *Logistic Shipment Agregator* untuk mengelempokan *sub order* agar bisa mengarahkan *Vendor vehicle* ke lokasi pickup dan *drop point* yang sudah di tetapkan.

10. List Shipment

Merupakan fitur untuk *Vendor* untuk memberikan aksi pada *shipment* yang sudah di *generate* agar bisa menerima atau menolak *shipment* tersebut.

11. Vendor Account FMS Integration

Merupakan fitur untuk *Vendor* untuk melakukan *login* dengan akun FMS yang saya miliki agar bisa melakukan integrasi lokasi *vehicle* dengan aplikasi LSI.

12. Vehicle Management

Merupakan fitur untuk *Vendor* untuk melakukan penambahan, pengurangan, perubahan *vehicle* agar bisa menawarkan *vehiclenya* ke *Logistic Shipment Agregator*.

13. Driver Management

Merupakan fitur untuk *Vendor* untuk mendaftarkan, mengubah dan menghapus akun *driver* agar bisa memberikan *driver* kepada *vehicle* yang dimiliki *Vendor* tersebut.

14. Invoice

Merupakan fitur untuk *Vendor* dan *Logistic Shipment Agregator* untuk mengirimkan *invoice* kepada *LSP* atau *SC* agar bisa menagihkan biaya pada *Logistic Shipment Agregator* atau *Shipment Company*.

C. Kebutuhan Perangkat Kerja

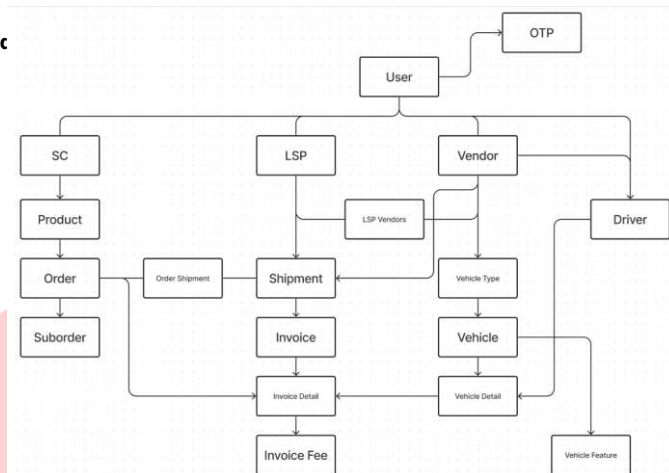
Dalam pembangunan aplikasi *Logistic Service Integrator* ada beberapa *software* (perangkat lunak) dan *hardware* (perangkat keras) yang digunakan yaitu; PhpStorm digunakan dalam penelitian ini untuk melakukan pengkodean dengan framework Laravel, Postman digunakan sebagai dokumentasi dan pengujian API yang telah dibuat, PostgreSQL merupakan database yang digunakan untuk mengolah data data yang dibutuhkan untuk membangun aplikasi atau sistem, Figma diperlukan untuk pembuatan dan implementasi desain, dan Gitlab merupakan aplikasi yang digunakan untuk menyimpan *source code*. Adapula perangkat keras yang digunakan pada penelitian ini program otomatisasi ini adalah Asus TUF A15 Ryzen 7 16GB .

IV. HASIL DAN PEMBAHASAN

A. Analisis hasil penelitian

Hasil dari penelitian ini adalah untuk menghasilkan aplikasi *Logistic Service Integrator* yang memiliki berbagai fitur didalamnya yang berguna untuk mempermudah proses kegiatan shipping dan logistik. Berikut hasil yang didapat dari penelitian ini beserta dengan penjelasan flow fitur dari aplikasi.

1. Pembuatan Database



Gambar 4. 1 struktur database

Hal pertama yang dibuat adalah database dari sistem *Logistic Service Integrator* sehingga nantinya dari database ini dipakai untuk menyimpan berbagai data yang akan dipakai untuk pembuatan berbagai fitur yang dibutuhkan.

Pada proyek LSI *role user* nya dibagi menjadi 5 *Shipment Company*, *Logistic Shipment Agregator/Provider*, *Vendor*, *Driver*, *Super Admin* dengan fitur yang dapat dilakukan dan tugasnya yaitu:

Role	Deskripsi
Super Admin	<ul style="list-style-type: none"> <li>• <i>Login</i></li> <li>• Membuat akun LSA/LSP beserta <i>user</i> lainnya</li> <li>• Membuat akun SC beserta <i>user</i> lainnya</li> <li>• Membuat akun Vendor beserta <i>user</i> lainnya</li> <li>• Membuat akun Driver beserta <i>user</i> lainnya</li> <li>• Mengubah status kerja yang dimiliki LSA/LSP dengan SC juga dengan <i>Vendor</i></li> <li>• Dapat melakukan semua hal terkait <i>Create Read Update Delete</i> data dengan sepengetahuan Perusahaan yang bersangkutan</li> </ul>
Shipment Company (SC)	<ul style="list-style-type: none"> <li>• <i>Login</i></li> <li>• Membuat, mengupdate dan menghapus daftar <i>user</i> akun <i>Shipment Company</i></li> <li>• Membuat daftar <i>Product</i></li> <li>• Membuat <i>order pickup</i> dan <i>dropoff</i> pengiriman</li> <li>• Melihat riwayat <i>order</i> dan pengiriman</li> <li>• Melihat tagihan/<i>invoice</i> dari <i>order</i></li> </ul>
LSA/LSP	<ul style="list-style-type: none"> <li>• <i>Login</i></li> <li>• Membuat, mengupdate dan menghapus daftar <i>user</i> akun LSP</li> <li>• Membuat link untuk registrasi akun SC</li> <li>• Merubah status kerjasama dengan SC</li> <li>• Melihat daftar <i>vendor</i> yang ingin bekerja sama</li> </ul>

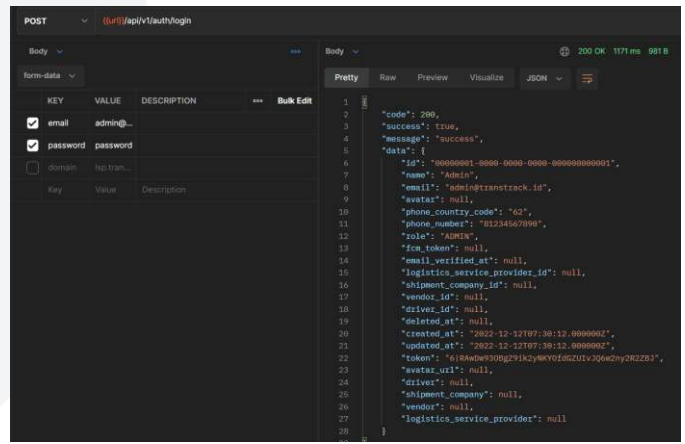
ISSN :	<ul style="list-style-type: none"> <li>• Merubah status kerjasama dengan <i>Vendor</i></li> <li>• Memilih dan mengelompokkan <i>order</i> yang diterima SC</li> <li>• Melihat detail kendaraan yang dimiliki <i>vendor</i></li> <li>• Mengkalkulasi muatan, menentukan kendaraan yang diperlukan, dan menentukan rute pengiriman</li> <li>• Melihat riwayat <i>order</i> dan pengiriman</li> <li>• Melihat tagihan atau <i>invoice</i> pengiriman dari kendaraan yang disewa dari <i>Vendor</i></li> <li>• Membuat tagihan atau <i>invoice</i> dari <i>order</i> SC</li> <li>• Mendaftarkan, mengupdate dan menghapus daftar lokasi yang dimiliki SC dalam perjanjian</li> </ul>
Vendor/Fleet Owner	<ul style="list-style-type: none"> <li>• <i>Login</i></li> <li>• Mendaftarkan, mengupdate dan menghapus akun <i>user</i> Vendor</li> <li>• Membuat, mengupdate, menghapus data kendaraan</li> <li>• Mengsinkronisasi kendaraan yang ada di FMS</li> <li>• Mendaftarkan, mengupdate dan menghapus akun <i>driver</i></li> <li>• Mengirim daftar permintaan kerja sama dengan LSP/LSA</li> <li>• Menerima atau menolak permintaan pengiriman (<i>shipment</i>)</li> <li>• Menentukan nomor plat dan <i>driver</i> dari kendaraan yang diminta oleh LSP/LSA</li> <li>• Membuat tagihan/<i>invoice</i> dari kendaraan yang diminta oleh LSA/LSP</li> <li>• Melihat riwayat pengiriman</li> </ul>
Driver	<ul style="list-style-type: none"> <li>• <i>Login</i> dengan OTP</li> <li>• Melakukan <i>update</i> akun</li> <li>• Melihat notifikasi untuk pengiriman</li> <li>• Menerima <i>Order Shipment</i></li> <li>• Melihat <i>history</i> pengiriman</li> </ul>

TABEL 4.1  
fitur dan tugas setiap role

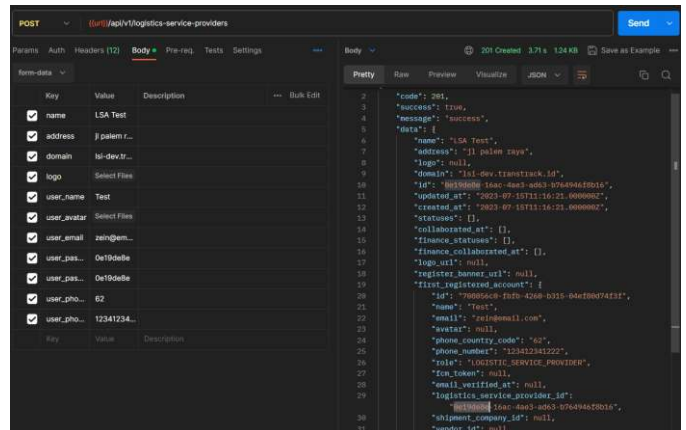
## 2. Fitur login dan Register Company

Sistem LSI ini dimulai dari *Admin* yang dimulai dari *login* lalu akan membuat akun LSP/LSA (*Logistic Shipment Provider/Aggregator*) beserta berbagai *user* lainnya yang diperlukan termasuk *Vendor*.

Tampilan pada API Postman:

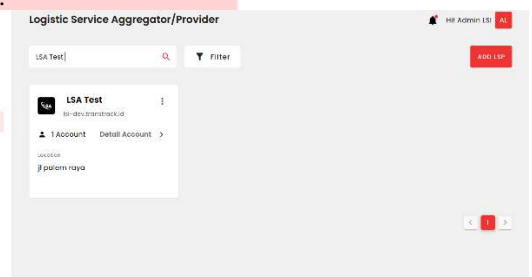


Gambar 4.2 login

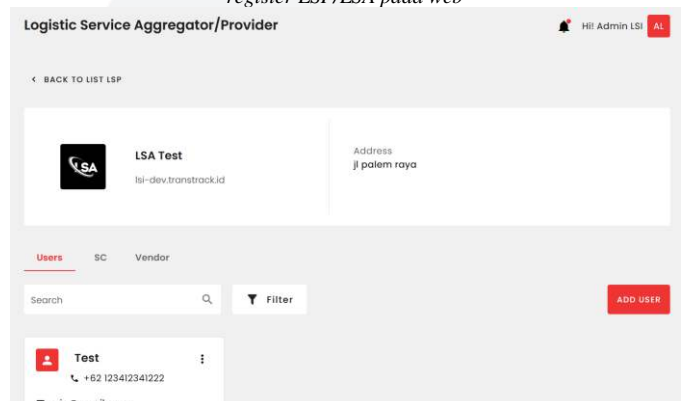


GAMBAR 4.3  
register user LSP/LSA

Dari data register yang berhasil dilakukan melalui Postman sebagai pengujian API maka berikut adalah outputnya pada website :



GAMBAR 4.4  
register LSP/LSA pada web

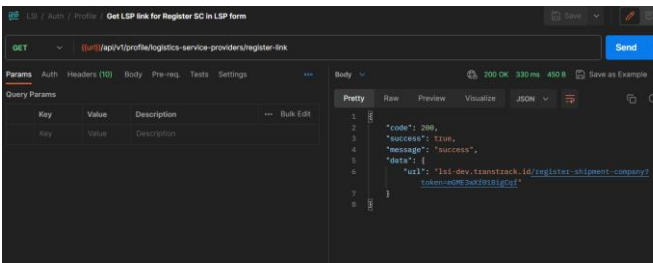


GAMBAR 4.5  
detail user yang telah di register

## 3. Registrasi Shipment Company

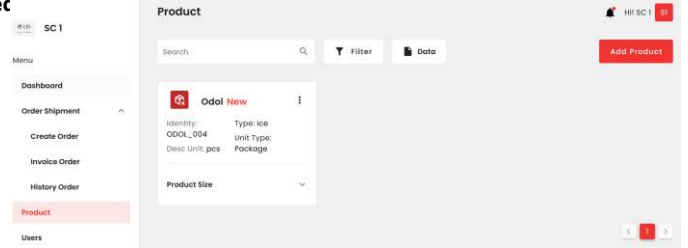
Selanjutnya LSP/LSA yang telah registrasi akan login memakai akun mereka dan akan membuat link registrasi untuk *Shipment Company* yang nantinya akan disalurkan ke *Shipment Company* agar dapat registrasi.

Tampilan pada API Postman:



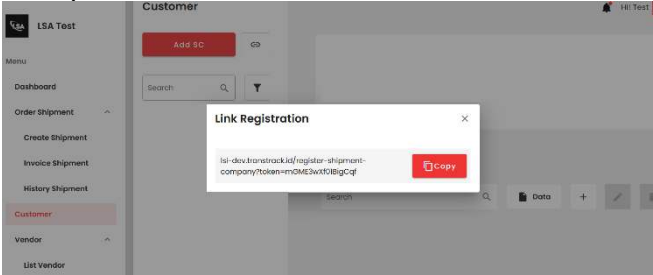
GAMBAR 4. 6  
LSP/LSA generate link registrasi SC

Tampilan web:



GAMBAR 4. 11  
tampilan web produk yang telah ditambahkan

Tampilan web:

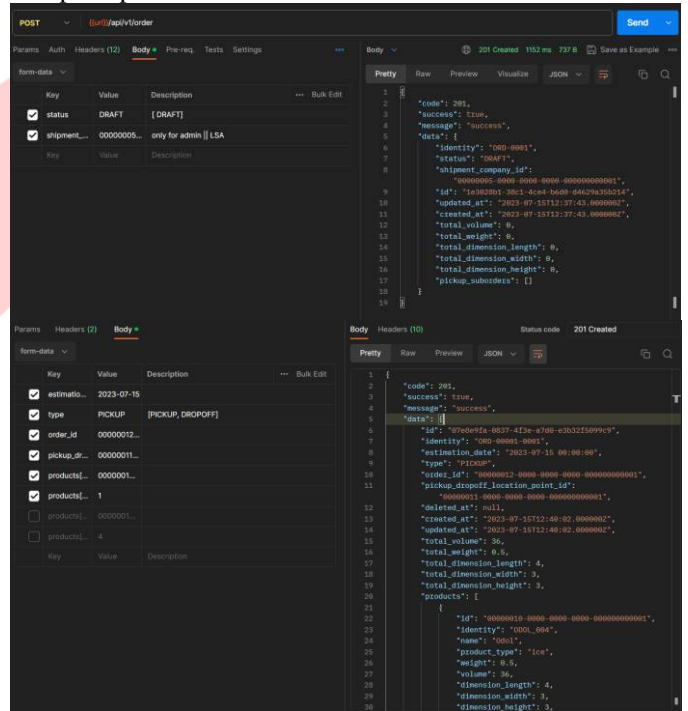


GAMBAR 4. 7  
Tampilan web untuk generate link registrasi SC

5. Membuat Order

Shipment Company dapat mulai melakukan manajemen order dari barang-barang yang telah didaftarkan pada aplikasi/sistem.

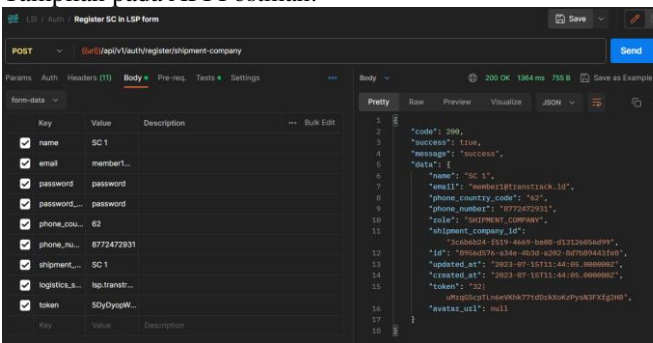
Tampilan pada API Postman:



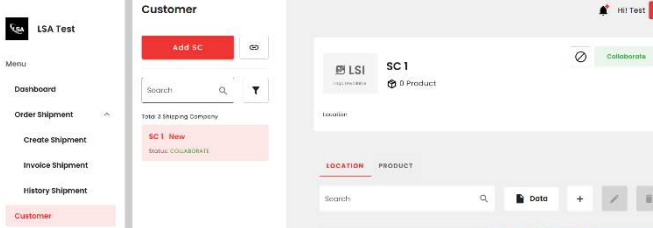
GAMBAR 4. 12  
SC membuat order yang dilanjutkan membuat suborder

Lalu apabila Shipment Company telah menerima link registrasi maka Shipment Company dapat melakukan registrasi dengan LSP/LSA tersebut.

Tampilan pada API Postman:



GAMBAR 4. 8  
register SC dengan link yang diberikan LSP/LSA

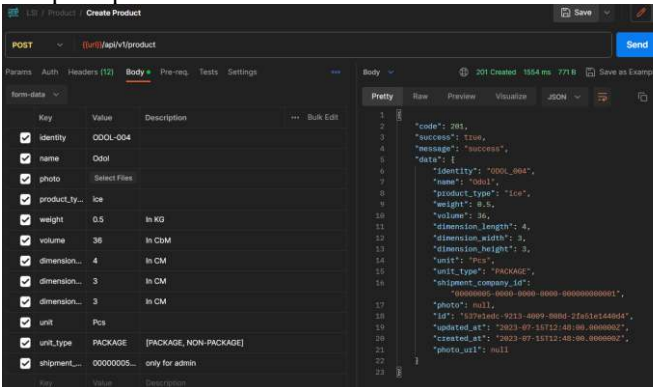


GAMBAR 4. 9  
tampilan web dari SC yang telah registrasi

4. Menambah data Produk

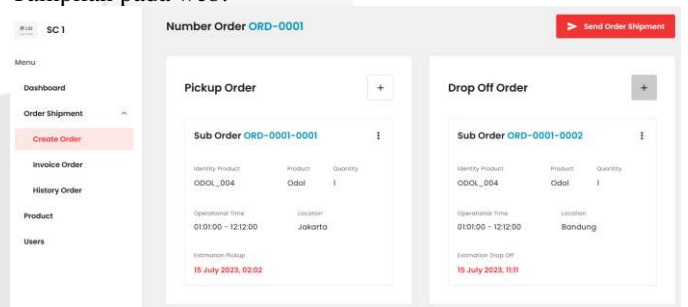
Shipment Company dapat mulai mendaftarkan produk-produk mereka yang selanjutnya produk-produk tersebut dapat dipakai pada proses selanjutnya manajemen order

Tampilan pada API Postman:



GAMBAR 4. 10  
menambahkan produk

Tampilan pada web:



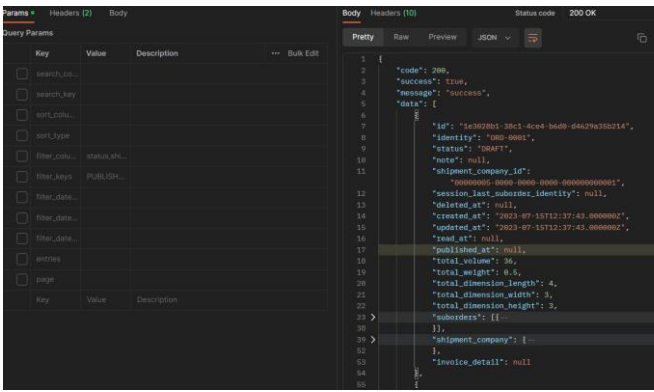
GAMBAR 4. 13  
tampilan web dari order dan suborder yang dibuat

Shipment Company membuat Suborder, Shipment Company juga memilih lokasi pickup dan dropoff untuk order product

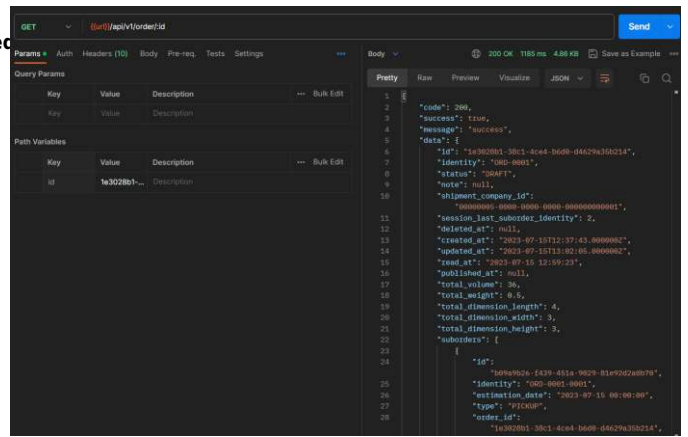
6. List dan Detail Order

LSP/LSA dapat melihat data order yang telah di input oleh Shipment Company. Dimana peran LSA/LSP ini adalah menghubungkan antara Shipment Company dan Vendor yang memiliki driver serta kendaraan yang akan dipakai untuk proses pengiriman.

Tampilan pada API Postman:

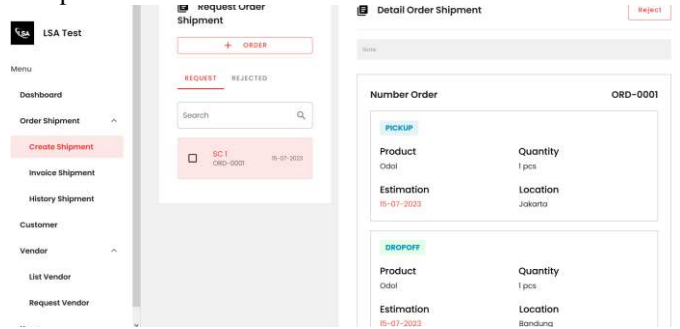


GAMBAR 4. 14  
list order



GAMBAR 4. 15  
detail order

Tampilan web:

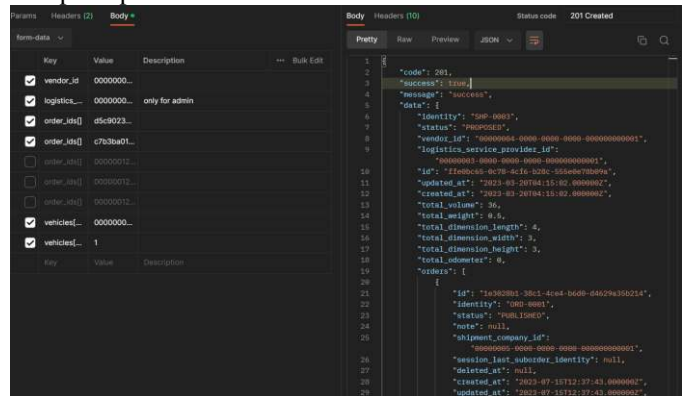


GAMBAR 4. 16  
tampilan web list dan detail order

## 7. Membuat Shipment

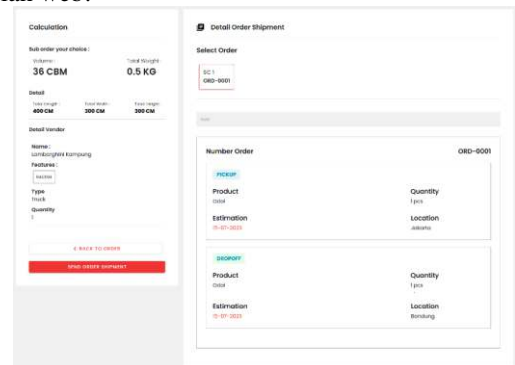
saat menerima *order* maka LSP/LSA dapat membuat *shipment* dari *order* yang telah dibuat oleh *Shipment Company*. Pada pembuatan *shipment* ini LSP/LSA dapat memilih jenis kendaraan dan jumlah kendaraan untuk pengiriman lalu juga dibuat data *track* secara otomatis yang berguna untuk tracking kendaraan.

Tampilan pada API Postman:



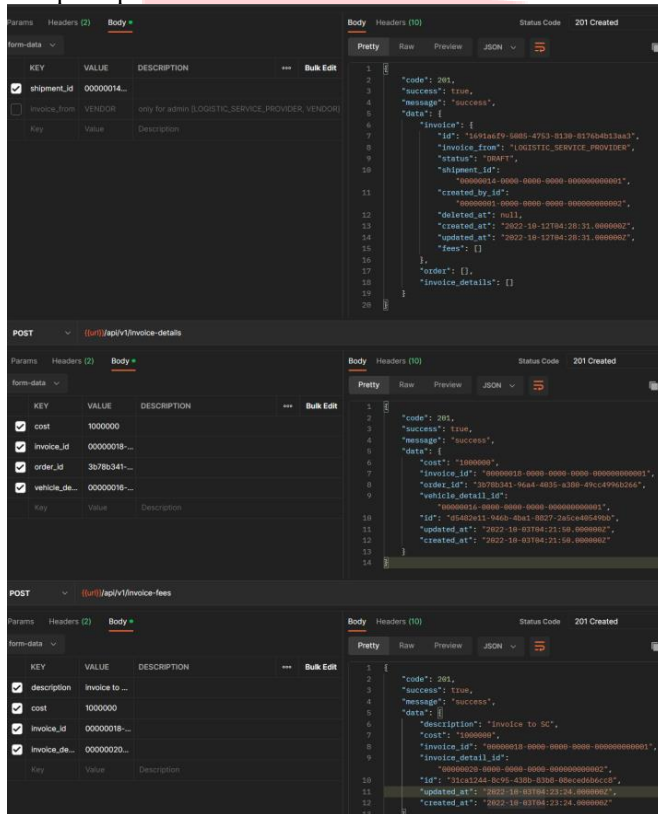
GAMBAR 4. 17  
pembuatan shipment

Tampilan web:



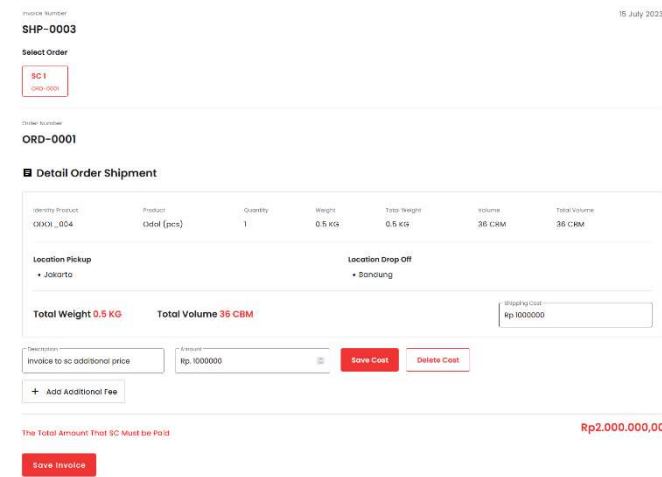
GAMBAR 4. 18  
tampilan web shipment yang dibuat

8. Membuat Invoice  
 ISSN : 2442-5826  
 selanjutnya dibuat *invoice* atau tagihan yang ditujukan ke *Shipment Company* untuk membayar tagihan yang dibutuhkan dalam proses pengiriman. Invoice juga bisa dibuat dari *Vendor* ke LSP/LSA  
 Tampilan pada API Postmant:



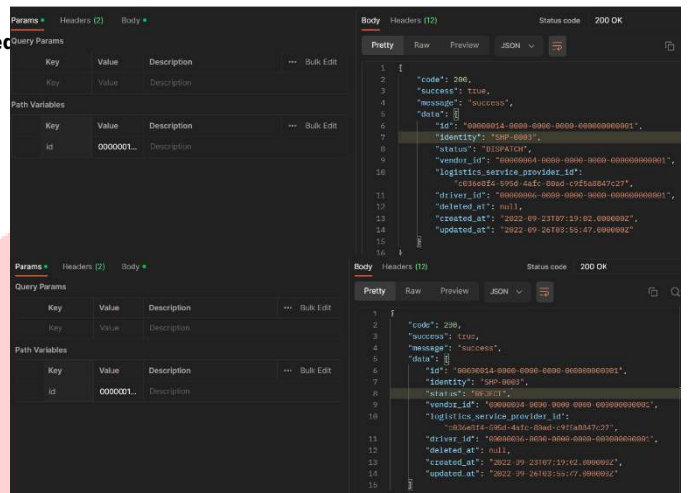
Gambar 4. 19 LSP/LSA membuat Invoice, invoice detail, invoice fee

Tampilan web:

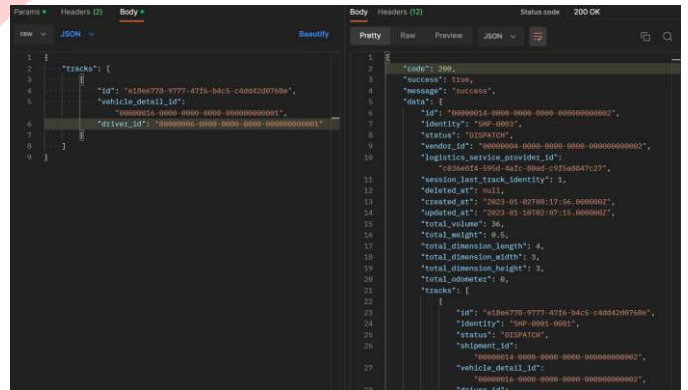


Gambar 4. 20 tampilan web dari invoice LSP ke SC

9. Accept/Reject Shipment  
 proses *shipment* akan diteruskan ke *Vendor* yang dipilih dan bekerja sama dengan LSP/LSA terkait. Sehingga *Vendor* nantinya menerima *shipment* dari LSP/LSA dan memutuskan untuk menerima *shipment* atau menolaknya.  
 Tampilan pada API Postmant:

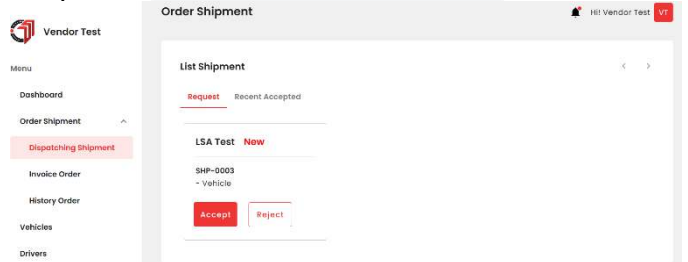


Gambar 4. 21 Vendor accept reject shipment

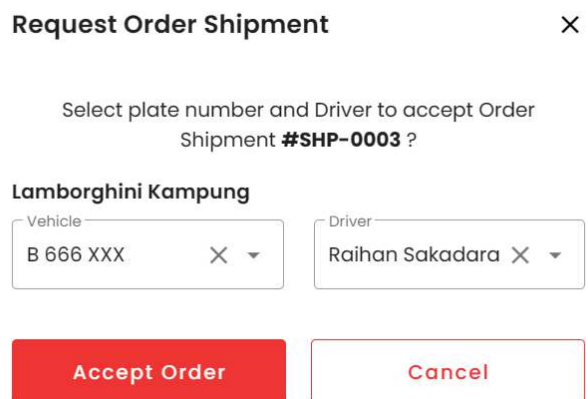


Gambar 4. 22 Vendor memilih driver untuk vehicle saat menerima shipment

Tampilan web:



Gambar 4. 23. tampilan web accept reject shipment



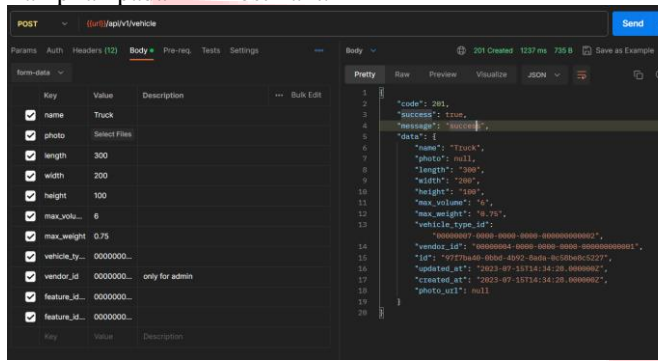
GAMBAR 4. 24 tampilan web memilih vehicle dan driver

apabila *shipment* ditolak biasanya diakibatkan karena syarat dari pihak LSP/LSA tidak terpenuhi sehingga perlu perbaikan *shipment* dari pihak LSP/LSA, sedangkan apabila *shipment* diterima *Vendor*, saat menerima *shipment* maka *Vendor* juga menentukan kendaraan dan juga *driver* untuk pengiriman. selanjutnya *Vendor* dapat membuat tagihan/*invoice* yang akan diserahkan ke pihak LSP/LSA.

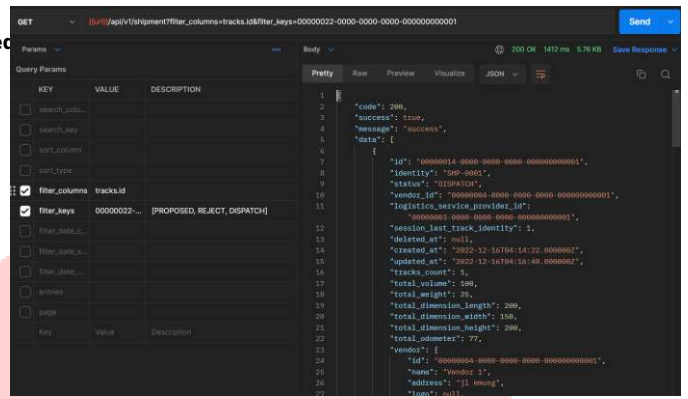
### 10. Manajemen Vehicle

Sebagai penyedia kendaraan Vendor juga dapat melakukan manajemen kendaraan seperti menambah daftar kendaraan mereka

Tampilan pada API Postman:



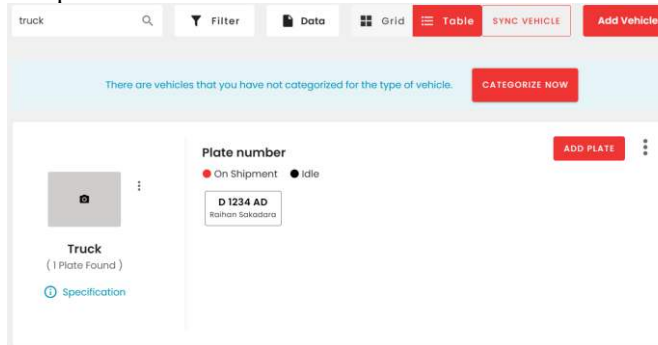
GAMBAR 4.25 Vendor menambah vehicle



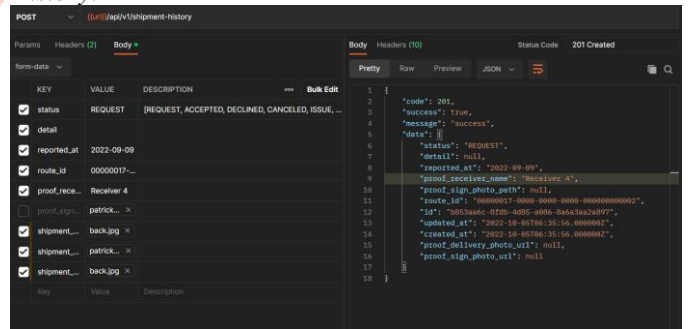
GAMBAR 4.29 Data list shipment yang di daftarkan ke Driver

Lalu setelah melihat list shipment maka Driver akan mengerjakan pengiriman dan apabila sudah selesai mengerjakan pengiriman maka dari shipment dibuat Shipment History.

Tampilan web:



GAMBAR 4.26 tampilan web dari vehicle yang telah dibuat

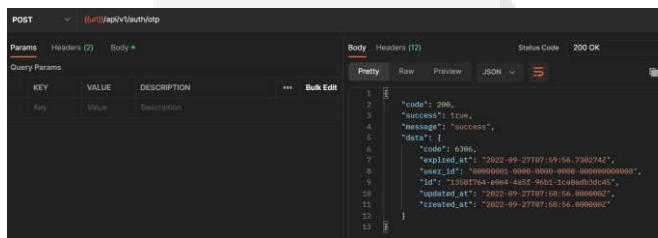


GAMBAR 4.30 Driver menginput data untuk shipment history

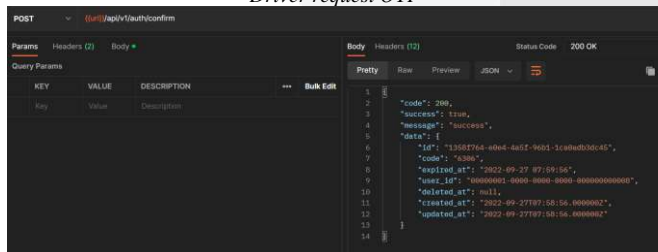
### 11. Driver OTP

Untuk Driver maka pertama akan login dengan OTP dan selanjutnya dapat melihat shipment yang telah ditugaskan kepada Driver.

Tampilan API Postman:



GAMBAR 4.27 Driver request OTP



GAMBAR 4.28 Driver login dengan OTP

### 12. Kuisioner tentang sistem yang dibuat

Dari hasil proyek yang dikerjakan maka selanjutnya disusunlah kuesioner untuk membuktikan bahwa sistem yang dikerjakan dapat berjalan dengan baik. Daftar pertanyaan dari kuesioner tersebut adalah sebagai berikut :

1. Apakah fitur Create (POST) dapat digunakan dengan mudah dan berjalan dengan baik / fitur mudah di implementasikan oleh Front End dan Mobile Developer ?
2. Apakah fitur Read (GET) dapat digunakan dengan mudah dan berjalan dengan baik / fitur mudah di implementasikan oleh Front End dan Mobile Developer ?
3. Apakah fitur Update (PUT) dapat digunakan dengan mudah dan berjalan dengan baik / fitur mudah di implementasikan oleh Front End dan Mobile Developer ?
4. fitur Delete (DELETE) dapat digunakan dengan mudah dan berjalan dengan baik / fitur mudah di implementasikan oleh Front End dan Mobile Developer ?
5. Apakah aplikasi dapat memudahkan integrasi antara SC, LSA/LSP, dan Vendor ?

Apakah aplikasi LSI dapat memudahkan dalam kegiatan bisnis shipping antara SC, LSA/LSP, dan Vendor?



## Formulir Kuesioner Logistic Service Integrator

Kuesioner ini dibuat dengan tujuan untuk mengetahui hasil dari sistem yang telah di buat baik dari sisi developer frontend, mobile, dan quality assurance

[Login ke Google](#) untuk menyimpan progres. [Pelajari lebih lanjut](#)

\* Menunjukkan pertanyaan yang wajib diisi

Nama \*

Jawaban Anda

1. Fitur Create (POST) dapat digunakan dengan mudah dan berjalan dengan baik / fitur mudah di implementasikan oleh Front End dan Mobile Developer

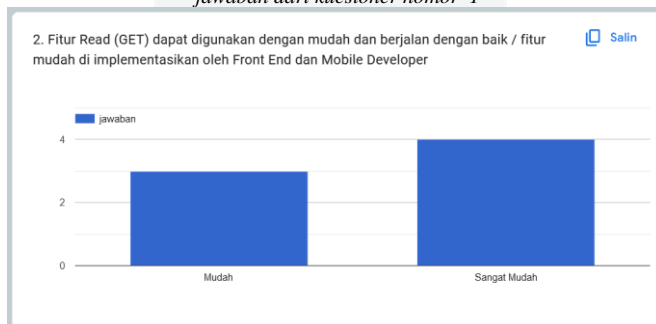
	jawaban
Tidak Mudah	<input type="radio"/>
Kurang Mudah	<input type="radio"/>
Mudah	<input type="radio"/>
Sangat Mudah	<input type="radio"/>

GAMBAR 4. 31  
Form Kuesioner

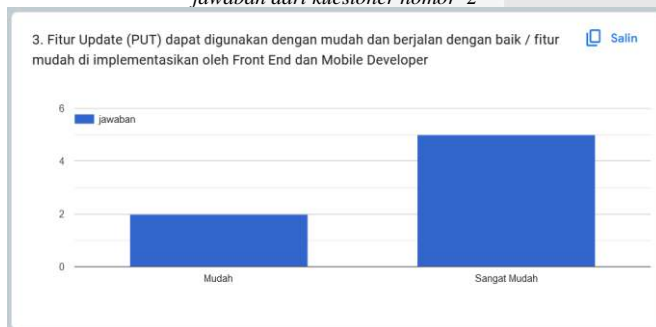
Dari form kuesioner yang dibuat didapatkan data jawaban dari 7 koresponden sebagai berikut :



GAMBAR 4. 32  
jawaban dari kuesioner nomor 1



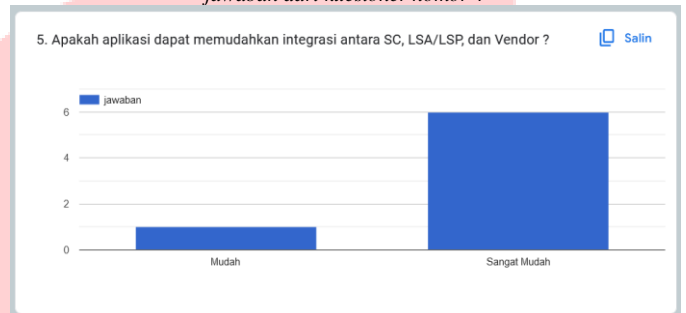
GAMBAR 4. 33  
jawaban dari kuesioner nomor 2



GAMBAR 4. 34  
jawaban dari kuesioner nomor 3



GAMBAR 4. 35  
jawaban dari kuesioner nomor 4



GAMBAR 4. 36  
jawaban dari kuesioner nomor 5



GAMBAR 4. 37  
jawaban dari kuesioner nomor 5

Dari hasil kuesioner tersebut dapat diketahui bahwa sistem yang di buat telah berhasil dan berjalan dengan baik sehingga dapat diimplementasikan selanjutnya oleh tim Front End dan Mobile Developer sehingga nantinya dari sistem ini dapat mengintegrasikan antara Shipment Company, Logistic Service Agregator, dan Vendor untuk memudahkan perusahaan – perusahaan tersebut dalam proses bisnis shipping dan logistik.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Dari penelitian yang disusun dapat diambil kesimpulan bahwa dengan *website* yang dibangun maka dapat mempermudah perusahaan *Shipping Company* untuk melakukan kegiatan operasional dalam jasa pengiriman barang dengan mengintegrasikan antara pelaku bisnis pengiriman yaitu dari *Shipping Company*, *Logistic shipment provider/aggregator*, *Vendor* dan *Driver*. Sehingga dapat meningkatkan efektifitas dan produktifitas dari bisnis *Shipping Company*.

### B. Saran

Dari project yang dikerjakan dengan menggunakan *Framework* yaitu *Laravel* sudah cukup baik, tetapi dapat ditingkatkan lagi performanya apabila memakai *Framework* lain yaitu *GO-Lang* karena dengan memakai *Framework GO-Lang* dapat melakukan eksekusi *logic* dan perhitungan lebih cepat dan pemanfaatan *core CPU* yang lebih efektif.

## REFERENCES

- [1] “Begini Perubahan Pola Belanja di Toko Online Selama Pandemi,” 2021. <https://www.cnbcindonesia.com/tech/20210329124334-37-233619/begini-perubahan-pola-belanja-di-toko-online-selama-pandemi> (diakses Jan 09, 2023).
- [2] D. Hernikawati, “Analisis Dampak Pandemi COVID-19 terhadap Jumlah Kunjungan pada Situs E-Commerce di Indonesia Menggunakan Uji T Berpasangan,” *Jurnal Studi Komunikasi dan Media*, vol. 25, no. 2, hlm. 191, Des 2021, doi: 10.31445/jskm.2021.4389.
- [3] “Fleet Operation Optimizer & Supply Chain Integrator,” 2022.
- [4] Dan Humdiana, “IMPLEMENTASI APLIKASI FULL STACK WEB DEVELOPMENT UNTUK DATA ADMIN PADA MEDIA SOSIAL BUZZBUDDIES,” 2019. [Daring]. Available: <http://www.kwikkiangie.ac.id>
- [5] R. Renaldo Prasena dan H. Sama, “STUDI KOMPARASI PENGEMBANGAN WEBSITE DENGAN FRAMEWORK CODEIGNITER DAN LARAVEL,” 2020. [Daring]. Available: <http://journal.uib.ac.id/index.php/cbssit>
- [6] R. Bangun Website Mencari Tukang Menggunakan Framework Laravel, R. Bangun Website Mencari Tukang Menggunakan, dan M. Rifqi Al Hazmi, “Framework Laravel,” 2018.
- [7] Sucipto, “Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar,” *Jurnal INTENSIF*, vol. 1, no. 1, 2017.
- [8] A. Dwi Praba dan M. Safitri, “STUDI PERBANDINGAN PERFORMANSI ANTARA MYSQL DAN POSTGRESQL,” vol. VIII, no. 2, 2020, [Daring]. Available: <https://www.adminer.org/>.
- [9] S. N. Yanti dan E. Rihyanti, “Penerapan Rest API untuk Sistem Informasi Film Secara Daring,” *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 1, hlm. 195, Mar 2021, doi: 10.32493/informatika.v6i1.10033.
- [10] W. Galindra Wardhana, I. Arwani, dan B. Rahayudi, “Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya),” 2020. [Daring]. Available: <http://j-ptiik.ub.ac.id>