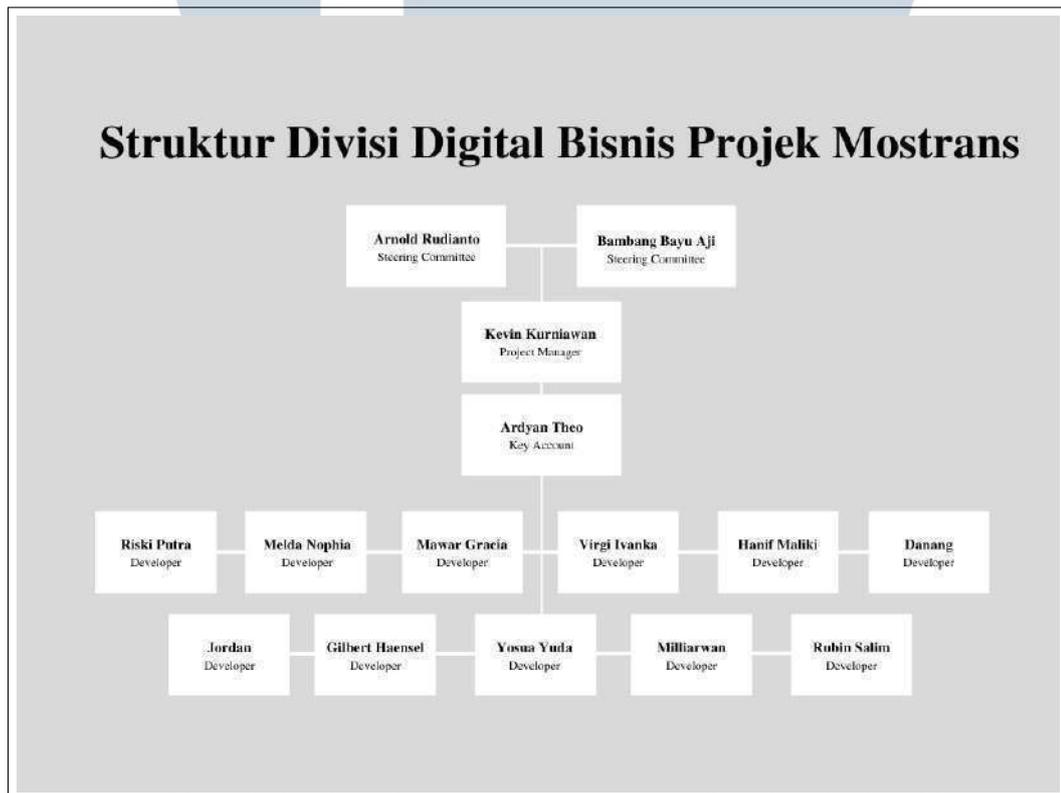


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Selama menjalankan proses praktik kerja magang, penulis mendapatkan kesempatan untuk bekerja di cabang pusat yaitu di Jalan Pulo Lentut No. 10, Jakarta Timur, DKI Jakarta 13920. Penulis ditempatkan pada divisi digital bisnis yang mengurus berbagai macam proyek eksternal yang dijual secara komersil. Penulis mendapat kesempatan untuk mengikuti proyek yaitu Mostrans yang merupakan *platform* digital transportasi sehingga dapat memudahkan akses dan *supply chain*.



MULTIMEDIA
NUSANTARA

Gambar 3.1. Struktur proyek Mostrans divisi digital bisnis

3.2 Tugas dan Uraian Kerja Magang

3.2.1 Tugas yang Dilakukan

Pelaksanaan praktik kerja magang di pada cabang pusat dimulai pada 15 Juli 2021 hingga 14 Januari 2022. Pada divisi digital bisnis, diperlukan *developer* yang dapat mengaplikasikan aplikasi *mobile* dan web. Terdapat beberapa pekerjaan yang dilakukan selama praktik kerja magang berlangsung. Selama pengerjaan berlangsung tugas yang diberikan merupakan proyek permintaan pengguna aplikasi, sehingga pekerjaan dibagi kepada setiap karyawan dan pelaksana kerja magang berdasarkan fitur-fitur yang diperlukan. Bahasa yang digunakan untuk aplikasi *mobile* adalah *java* dan untuk web menggunakan *javascript* dengan *framework React JS* untuk *frontend* serta *GraphQL* untuk *backend*. Berikut merupakan pekerjaan yang dikerjakan selama praktik kerja magang :

A. Aplikasi Web

A.1 Marketplace

Membuat fitur sehingga pengguna dapat mengunggah gambar dan *PDF file*. *File* yang diunggah oleh pengguna nantinya akan disimpan menggunakan *Firebase*. Setelah itu, *link* dari *Firebase* tersebut akan disimpan di *database*.

Membuat tampilan pada *help desk*. Beberapa tampilan yang dibuat mencakup tampilan awal yang dapat melihat keseluruhan status tiket, tampilan tiket yang dibuat oleh pengguna, dan formulir yang digunakan untuk membuat tiket

A.2 Software as a Service (SaaS)

Membuat fitur *lead time*. *Lead time* adalah waktu yang dibutuhkan antara inisialisasi sampai dengan penyelesaian suatu proses. Fitur ini dibuat agar pengguna dapat melakukan penambahan, melihat, dan edit data *lead time*. Pengguna dapat menambah data menggunakan formulir yang telah disediakan dan menggunakan *file* dengan ekstensi *xlsx*. Pengguna dapat melihat data menggunakan tabel dari web

dan menggunakan fitur *export file excel*. Pengguna dapat melakukan *edit* data dengan menggunakan tombol yang ada pada tabel. Fitur ini juga menggunakan integrasi sehingga setiap pengguna akan memiliki data yang berbeda sesuai dengan *id* dari pengguna.

Membuat fitur muat bongkar. Fitur ini dibuat agar pengguna dapat melakukan penambahan, melihat, dan edit data muat bongkar. Pengguna dapat menambah data menggunakan *form* yang telah disediakan dan menggunakan *file* dengan ekstensi *xlsx*. Pengguna dapat melihat data menggunakan tabel dari web dan menggunakan fitur *export file excel*. Pengguna dapat melakukan *edit* data dengan menggunakan tombol yang ada pada tabel. Fitur ini juga menggunakan integrasi sehingga setiap pengguna akan memiliki data yang berbeda sesuai dengan *id* dari pengguna.

Membuat fitur *shipper contact*. Fitur ini dibuat agar pengguna dapat menginput sebuah *file* dengan ekstensi *xlsx*. Setelah itu, data yang didapatkan dari *file* tersebut akan dimasukkan ke dalam sebuah tabel dengan menampilkan informasi-informasi yang dibutuhkan. Jika sudah sesuai pengguna dapat mengirim data-data tersebut sehingga disimpan menggunakan database. Fitur ini juga menggunakan integrasi sehingga setiap pengguna akan memiliki data yang berbeda sesuai dengan *id* dari pengguna.

Membuat fitur *listing claim* untuk *shipper*. Fitur ini dibuat agar *shipper* dapat melihat *claim* yang dibuat oleh pengguna. Data yang ada akan ditampilkan menggunakan tabel.

Membuat fitur penambahan *claim*. Fitur ini dibuat agar pengguna dapat membuat suatu *claim* yang digunakan untuk melakukan retur barang. Pengguna akan diminta untuk mengisi sebuah *form* serta mengunggah bukti berupa gambar. Selain itu, pengguna juga dapat melihat *claim* yang telah dibuat.

B. Aplikasi *Mobile*

B.1 *Marketplace*

Aplikasi tersebut dibuat agar *admin* dapat melakukan pengecekan pada vendor dan kendaraan dari dokumen-dokumen legalitas dan kelayakannya. Fitur yang dikerjakan adalah *dashboard* dan kendaraan. *Dashboard* digunakan agar pengguna dapat melihat secara sekilas tentang status dari vendor dan kendaraan. Untuk fitur kendaraan, pengguna dapat melihat status dari kendaraan tersebut. Setelah itu, pengguna dapat membuat sebuah tiket sehingga pengguna dapat melakukan pengecekan pada dokumen-dokumen yang diperlukan serta kelayakan dari suatu kendaraan. Pengguna dapat menyetujui dan menolak dokumen-dokumen dan kondisi kendaraan dengan menyertakan bukti berupa gambar ataupun *PDF file*. Jika kendaraan tersebut sedang digunakan, pengguna dapat melihat perjalanan serta pemesanan yang kendaraan tersebut sedang jalankan.

3.2.2 Uraian Kerja Magang

Proses pelaksanaan praktik kerja magang dimulai dengan melakukan pengiriman *curriculum vitae* (CV) ke yang membutuhkan *developer intern* melalui sebuah web atau *platform* rekrutmen. Setelah mengirimkan *curriculum vitae* (CV), pihak perusahaan memberikan kesempatan untuk datang langsung ke kantor untuk melakukan beberapa tes yang diperlukan. Setelah melakukan penilaian atas tes yang telah dikerjakan, pihak perusahaan menyatakan lulus dan dapat melanjutkan ke tahap berikutnya yaitu *live coding* bersama *user*. Beberapa hari kemudian, dari pihak perusahaan menyatakan bahwa *live coding* yang dilakukan beberapa hari sebelumnya dinyatakan lulus dan dapat melanjutkan ke tahap wawancara. Setelah melalui tahap wawancara, pihak perusahaan memberi kabar bahwa penulis lulus pada tahap wawancara dan dapat memulai praktik kerja magang setelah perjanjian kerja disetujui oleh penulis dan mengirimkan informasi-informasi yang dibutuhkan terkait oleh penulis.

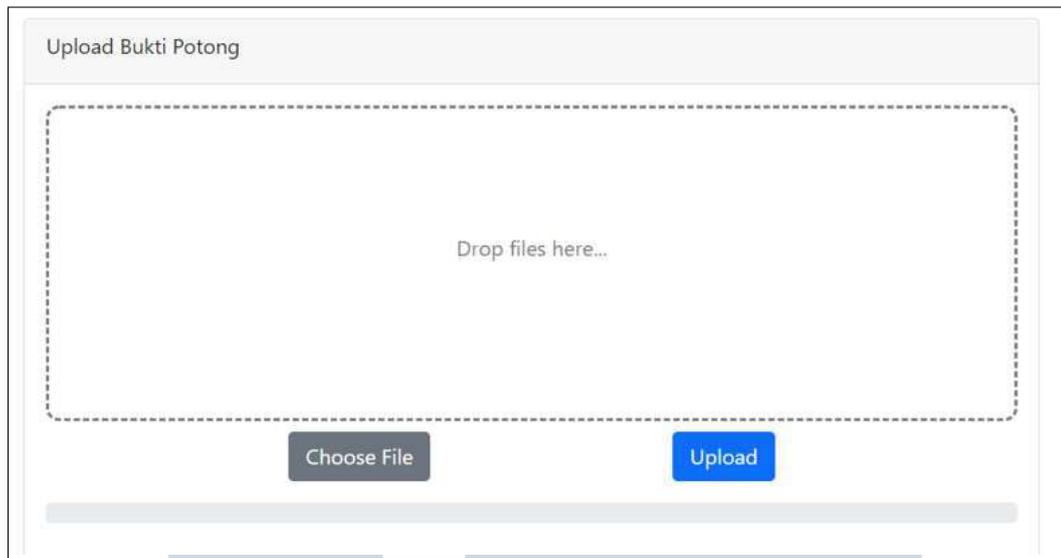
Jangka waktu praktik kerja magang terhitung sejak tanggal 15 Juli

2021 sampai dengan 14 Januari 2022. Waktu kerja adalah selama 8 (delapan) jam dimulai dari pukul 08.00 hingga 17.00, setiap hari senin sampai dengan jumat. Praktik kerja magang dilakukan secara *online* dan *offline*. Ketika bekerja dirumah atau secara *online* penulis diminta untuk masuk kedalam web *kumospace* sehingga dapat mempermudah jalur komunikasi dan absensi menggunakan web yang telah disediakan. Jika bekerja di kantor atau secara *offline* penulis diminta untuk datang ke cabang pusat yang berlokasi di Jalan Pulo Lentut No. 10, Jakarta Timur, DKI Jakarta 13920.

Pada hari pertama, praktik kerja magang dilakukan dirumah melalui web *kumospace*. Penulis mendapat informasi-informasi terkait divisi dan projek yang sedang dikerjakan. Penulis juga mendapatkan kesempatan untuk bertanya dan mempelajari kembali pengetahuan yang telah didapatkan selama perkuliahan. Pemimbing atau mentor meminta agar penulis dapat mempelajari juga beberapa bahasa atau *framework* pemrograman yang digunakan pada saat itu.

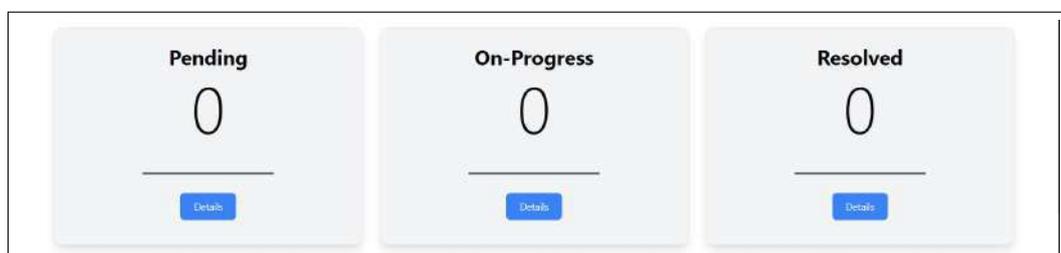
Penulis mendapat tugas pertama yaitu membuat fitur unggah gambar atau *PDF file* menggunakan *HTML*, *CSS*, dan *JavaScript*. Selanjutnya jika sudah berhasil membuat fitur tersebut penulis diminta untuk menambahkan fitur agar *file* yang diunggah tersebut dapat disimpan menggunakan *Fire-base*. Terdapat beberapa revisi pada fitur unggah tersebut yaitu pemimbing meminta untuk menambahkan fitur *drag and drop*. Setelah itu, melakukan implementasi pada web terkait. Fungsi ini bertujuan agar pengguna dapat mengunggah bukti pembayaran.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.2. Fitur unggah gambar dan *file pdf*

Pada penugasan berikutnya penulis diberi kesempatan untuk mempelajari *framework* yaitu *React JS*. *framework* tersebut nantinya akan digunakan untuk membuat suatu web yaitu *help desk*. Web tersebut digunakan untuk menangani manajemen tiket terkait permasalahan-permasalahan yang ada. Tujuan dari implementasi web tersebut adalah laporan dari suatu permasalahan dapat dikirim ke pihak IT untuk dapat digunakan sebagai bahan evaluasi sistem agar proses pengendalian aplikasi berjalan lebih efektif dan membantu pengambilan keputusan. Penulis pada pembuatan web tersebut mendapatkan tugas yaitu membuat tampilan awal atau *home page*, tiket, dan formulir yang digunakan untuk melaporkan suatu permasalahan. Selain menggunakan *framework JavaScript* yaitu *React JS*, pembuatan web ini juga menggunakan *framework CSS* yaitu *Tailwind*.



Gambar 3.3. Tampilan awal *help desk*



Gambar 3.4. Tampilan *ticket*

Ticket Form

Full name:

Email address:

What type of event is it?:

What type of event is it?:

What type of event is it?:

ID	Current Price	New Price	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add Price"/> <input type="button" value="Delete Price"/>

Additional details:

Gambar 3.5. Tampilan formulir

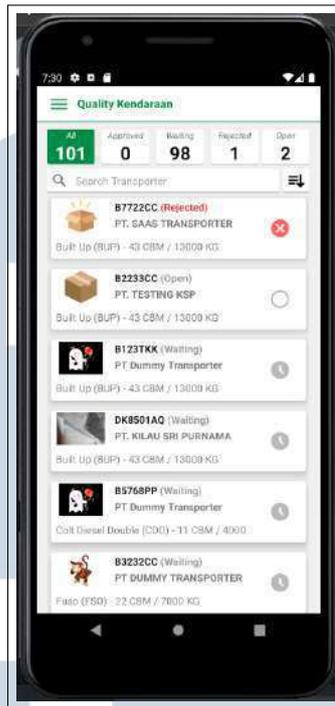
Setelah menyelesaikan pembuatan tampilan web *help desk*, penulis diberi kesempatan untuk mengerjakan aplikasi *mobile* yaitu *quality assurance*. Pengerjaan tugas tersebut menggunakan bahasa *Java*. Aplikasi tersebut akan digunakan untuk melakukan pengecekan pada dokumen-dokumen vendor maupun kendaraan. Penulis mendapatkan kesempatan untuk membuat tampilan atau *frontend* pada *dashboard* dan halaman kendaraan. Untuk membuat tampilan *dashboard*, penulis menggunakan representasi data menggunakan 2 tipe *chart* yaitu *bar chart* dan *pie chart* yang diharapkan dapat memudahkan pengguna saat melihat tampilan tersebut. Pembuatan *chart* tersebut menggunakan *library* yaitu *MPAndroidChart* sehingga memudahkan dalam pembuatan tampilan. Halaman kendaraan dibuat dengan menampilkan data-data berbentuk seperti kartu. Ketika kartu kendaraan ditekan maka pengguna akan dialihkan ke halaman baru yang menampilkan

data kendaraan secara lebih detil serta dokumen-dokumen yang perlu dilakukan pengecekan. Selain itu, pengguna juga dapat melihat perjalanan yang sedang kendaraan lakukan. Setelah membuat tampilan, aplikasi ini ditunda terlebih dahulu karena penulis dipindahkan ke implementasi fitur web.

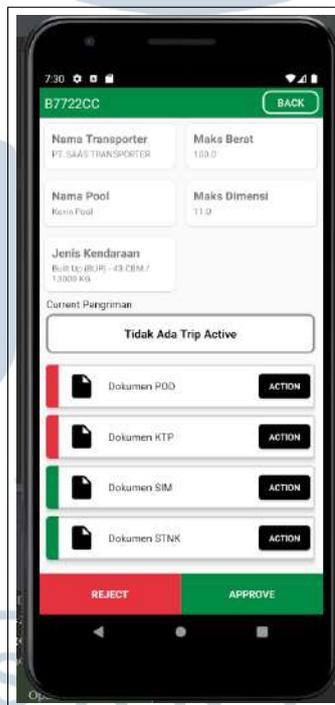


Gambar 3.6. Tampilan dashboard

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Tampilan kendaraan



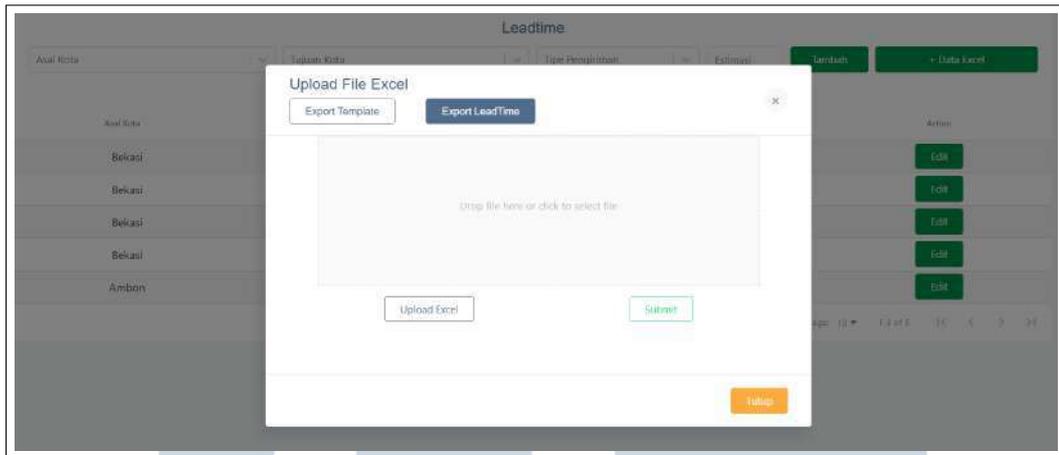
Gambar 3.8. Tampilan detail kendaraan

Penulis diminta untuk membuat fitur yaitu *lead time* yang bertujuan

untuk menampilkan estimasi pengantaran dari kota ke kota. Pengerjaan fitur ini menggunakan penghubung atau *middleware* yaitu *GraphQL* untuk mendapatkan data melalui API. Pada pengerjaannya diperlukan juga *library React JS* yaitu *Apollo*. Pengerjaan diawali dengan membuat tampilan tabel menggunakan *library React* yaitu *React data table component* dan mengambil data dari *GraphQL*. Pada tabel tersebut, pengguna dapat melakukan perubahan estimasi *lead time* yang sudah ada. Pengguna juga dapat input data *lead time* yang baru dengan menginput kota asal, kota tujuan, tipe pengiriman dan juga estimasi waktu pada formulir yang telah disediakan pada web. Formulir yang digunakan untuk menambah data juga dapat melakukan *filter* data pada tabel. Selain itu, pengguna juga dapat memasukan data berupa *file* dengan ekstensi *xlsx*. Terdapat 2 cara pengguna dapat memasukkan *file* tersebut yang pertama menggunakan tombol yang disediakan dan *drag and drop*. *drag and drop* pada web menggunakan *library react* yaitu *react drop-zone*. Ketika pengguna memasukkan *file* maka akan dicek terlebih dahulu ekstensinya jika sudah sesuai akan diterima dan jika tidak sesuai maka *file* tidak akan diterima. Setelah itu, data yang didapatkan dari *file* akan ditampilkan dengan menggunakan tabel jika terdapat data yang sudah pernah diinput maka baris tabel akan berwarna merah. Ketika pengguna mengirim data menggunakan *file* tersebut maka *file* tersebut akan disimpan menggunakan *Firebase* dan *link* akan disimpan di database. Selain menerima input *file*, pengguna juga dapat melakukan ekspor data tabel kedalam bentuk *file* dengan ekstensi *xlsx*. Fitur ini dapat digunakan oleh berbagai macam pengguna maka diperlukan suatu implementasi agar dapat membedakan antar pengguna. Implementasi tersebut menggunakan *parameter link* yang akan di encode terlebih dahulu.

Asal Kota	Tujuan Kota	Tipe Pengiriman	Estimasi (hr/d)	Action
Bekasi	Bekasi	Direct	4	Edit
Bekasi	Tangerang	Direct	1	Edit
Bekasi	Bandung	Direct	1	Edit
Bekasi	Bandung	Transit	4	Edit
Ambon	Amurang	Direct	2	Edit

Gambar 3.9. Tampilan awal *lead time*

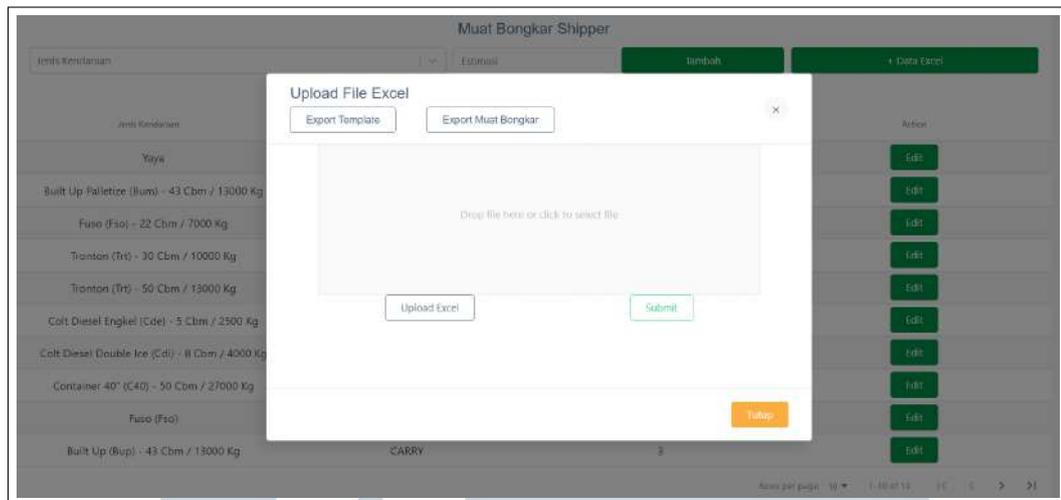


Gambar 3.10. Tampilan unggah *file* ekstensi *xlsx*

Fitur yang dibuat berikutnya adalah fitur muat bongkar yang akan digunakan untuk membuat estimasi muat bongkar. Pada tampilan awal, terdapat formulir yang digunakan untuk input data serta melakukan *filter* pada tabel. Lalu terdapat tombol untuk input data menggunakan *file* dengan ekstensi *xlsx*. Data ditampilkan menggunakan tabel. Implementasi fitur-fitur dan *library* yang digunakan sama seperti fitur *lead time* yang dijelaskan sebelumnya. Terdapat perbedaan pada pengecekan data dari input *file* yaitu perlu dibedakan data yang sudah memiliki eksternal dan yang belum. Ketika terdapat eksternal *id* yang belum pernah ada atau dibuat maka akan muncul pemberitahuan bahwa eksternal *id* tersebut belum terdaftar.

Jenis Sandarman	Estimasi	Aksi	
Yaya	2	Edit	
Built Up Palletize (Bum) - 43 Cbm / 13000 Kg	BUMN	2	Edit
Fuso (Fso) - 22 Cbm / 7000 Kg	FUSO	3	Edit
Tronton (Trt) - 30 Cbm / 10000 Kg	TRT	6	Edit
Tronton (Trt) - 50 Cbm / 13000 Kg	TRT50	2	Edit
Colt Diesel Engkel (Cde) - 5 Cbm / 2500 Kg	CCDE	1	Edit
Colt Diesel Double Ice (Cdi) - 8 Cbm / 4000 Kg	CDI	182	Edit
Container 40" (C40) - 50 Cbm / 27000 Kg	C501	11	Edit
Fuso (Fso)	-	1	Edit
Built Up (Bup) - 43 Cbm / 13000 Kg	CARRY	3	Edit

Gambar 3.11. Tampilan awal muat bongkar



Gambar 3.12. Tampilan unggah *file* ekstensi *xlsx*

Fitur selanjutnya yang dibuat adalah fitur mengunggah *shipper contact* yang bertujuan untuk input data-data *shipper*. Pengguna dapat input data menggunakan *file* dengan ekstensi *xlsx*. Selain itu, Pengguna juga dapat input *file* dengan 2 cara yaitu menekan tombol atau dengan *drag and drop*. Ketika *file* yang diinput berbeda dengan yang sudah ditetapkan maka *file* tidak akan diterima. Jika berhasil diterima, maka data akan ditampilkan menggunakan tabel. Pada tampilan tabel, pengguna dapat melihat 2 hal penting yang perlu pengguna input pada *file* tersebut yaitu koordinat dan kota. jika koordinat tidak benar atau tidak ada maka tampilan tabel pada baris tersebut akan berwarna merah serta keterangan bahwa koordinat belum diinput. Kota juga harus dicek terlebih dahulu apakah sebelumnya sudah ada pada database. Jika belum maka baris pada tabel tersebut akan berwarna merah dengan keterangan bahwa kota belum terdaftar. Pada tabel, pengguna juga dapat melakukan *filter* data untuk melihat data mana yang akan diinput dan data mana yang tidak akan diinput karena ada kesalahan pada koordinat dan atau kota. Jika baris berwarna merah artinya data tersebut tidak akan diinput ke dalam database. Melakukan unggah data tersebut menggunakan API dari *GraphQL*. Sama seperti pengerjaan fitur lainnya, fitur ini juga dapat memiliki pengguna yang berbeda-beda sehingga perlu implementasi *parameter link* yang diencode sehingga dapat dibedakan antar pengguna.

Internal ID	Nama Perusahaan	Kota	Status	Alamat	Keterangan
62004344	PT. MADISARU EFUTE ALINDO	SURABAYA	Active	KOMPLEK KANTOR MANGGA DUA, BL...	Data valid
62001129	PT SAKA AGUNG KARVA ABADI	SIDGARJO	Active	Jl. Raya Lingkar Timur No 9 RT 002/03...	Data valid
62001238	PT. SUMBER ALFARIA TRIJAWA	MALANG	Active	Jl. Raya Singoani No.1, Sempol, Andl...	Data valid
62001300	PT. Sinar Kencana Elektrik	SURABAYA	Active	Jl. KERTAJAYA NO.135-A GUBENG SU...	Data valid
62001519	PT. MADISARU EFUTE ALINDO	SURABAYA	Active	Jl. GEMBONG NO.2-16 BLOK C-4, RT...	Data valid
62001547	PT. SUMBER ALFARIA TRIJAWA	SIDGARJO	Active	Jl. Raya Mawigeni Wisesa - Sokodono...	Data valid
62002208	Mitra TO Wiyung	SURABAYA	Active	Jl. Raya Kembang Babatan No.477, Ba...	Data valid
62003393	PT. SUMBER ALFARIA TRIJAWA DC IO...	LIMBOKE	Active	Jalan Tgh. Seleh Hamba'i No.20, Desa...	Data valid
62001486	PT. ESPERINDO	SURABAYA	Active	Ngagep Jaya Selatan RW Blok K-19 Ba...	Data valid
62003712	PT MEGADEPO INDONESIA - MALANG	MALANG	Active	Jl. Raya Karanglo No.89, Karanglo, Ba...	Data valid

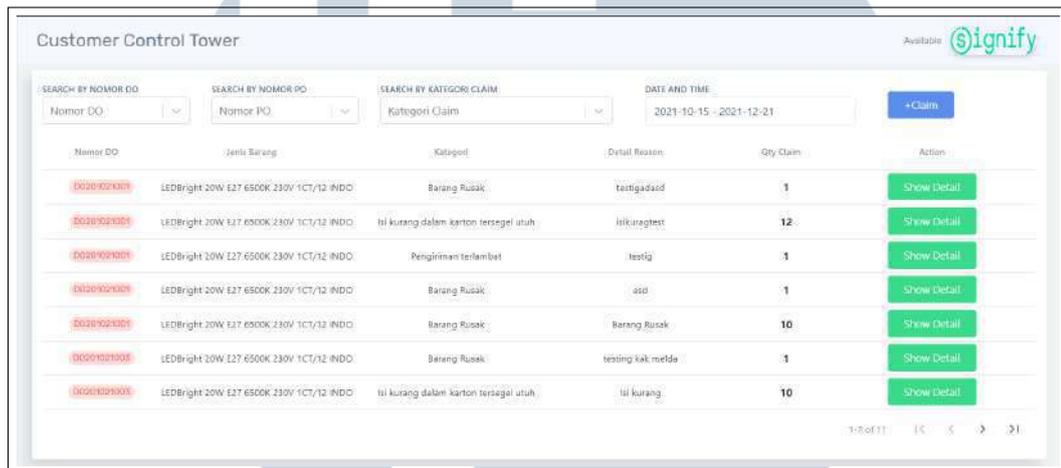
Gambar 3.13. Tampilan awal unggah *shipper contact*

Internal ID	Nama Perusahaan	Kota	Status	Alamat	Keterangan
72006507	PT Menara Nusantara Persada	TANJUNGPANDAN	Active	PT Menara N...	Koordinat
62001647	SOGO - GRAND INDONESIA	JAKARTA	Active	Jl. MH. Tham...	Koordinat
62001649	SOGO - PLAZA INDONESIA	JAKARTA	Active	Jl. MH Thamrin Kav. 28-30 10250 JAK...	Koordinat belum diinput
62002167	OC INDOMARET SENTUL	BOGOR	Active	OC Indomaret Sentul J.KP.Jembatan ...	Koordinat belum diinput
72000669	DC INDOMARCO-JAKARTA	JAKARTA	Active	Jl. Ancol Barat VII No.2	Koordinat belum diinput
62002783	PT. Panah Lestari Internusa	JAKARTA	Active	Unit G 17-18, Mega Kuningan Kav E4 ...	Koordinat belum diinput
62003736	PT. Panah Lestari Internusa / Soga Fo...	JAKARTA	Active	PK Lifestyle Xliner Jl.Jend Sudirman PK...	Koordinat belum diinput
62002991	PT. Panah Lestari Internusa	TANGKARANG	Active	Alam Sutera Mall Lt.LG Jl.Jalir Sutera ...	Koordinat belum diinput
62003100	PT. Titan Articlindo	JAKARTA	Active	Komplek Rukan Puri Blok D Kav 17 Ke...	Koordinat belum diinput
62003285	PT. Titan Articlindo BRC	JAKARTA	Active	JL.R.S Fatmawati Raya No. 33H/8 Cila...	Koordinat belum diinput

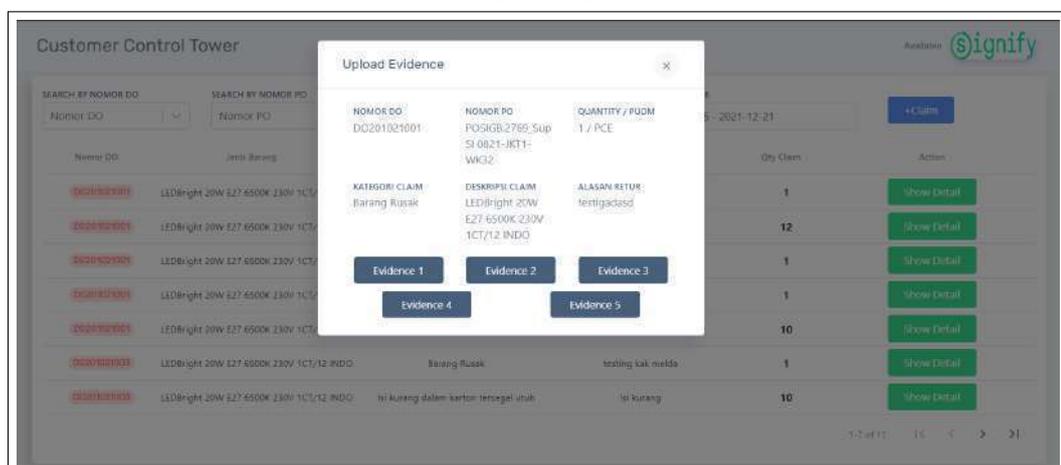
Gambar 3.14. Tampilan *shipper contact* dengan filter

Ketika barang dikirim mungkin terjadi kerusakan atau hal-hal lain yang tidak diinginkan. Maka dari itu, diperlukan suatu fitur agar pembeli dapat memberikan keluhan untuk retur barang dengan alasan-alasan tertentu. Penulis diberi kesempatan untuk membuat fitur *add claim* yang akan menerima keluhan dari pembeli. Untuk membuat *claim* pengguna perlu klik tombol "+Claim" setelah itu memasukkan informasi-informasi yang dibutuhkan pada formulir. Pengguna juga diminta untuk memberikan bukti menggunakan gambar. Terdapat sampai dengan 5 gambar bukti yang pengguna dapat kirimkan. Gambar yang diinput oleh pengguna akan disimpan melalui *Firebase* lalu *link* yang dikembalikan oleh *Firebase* akan disimpan di *database*. Ketika sudah selesai menginput informasi pada formulir, pengguna perlu klik tombol submit untuk mengirim data agar dapat disimpan di *database*. Penyimpanan data dilakukan dengan menggunakan *API GraphQL*. Pengguna juga dapat melihat *claim* yang dibuat se-

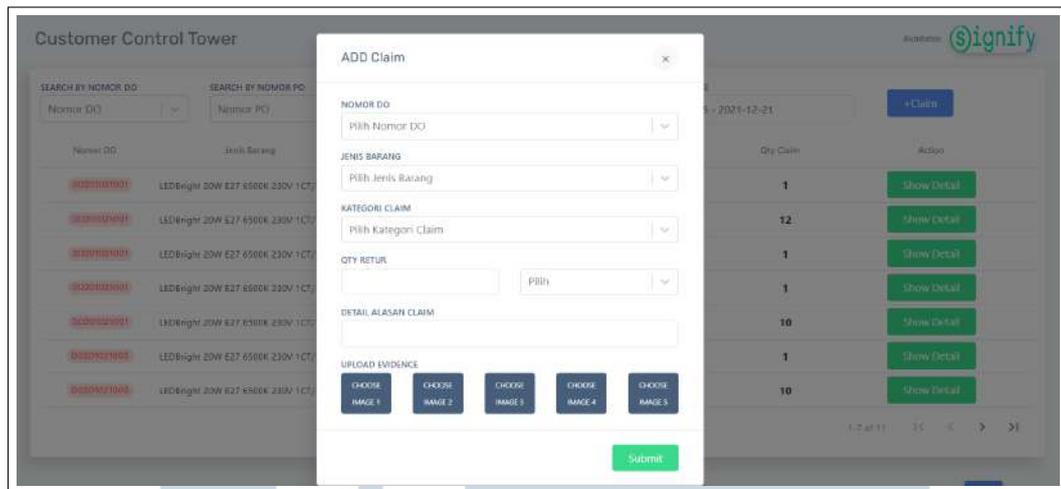
belumnya melalui tabel. Data diambil menggunakan API *GraphQL* tabel terdapat tombol untuk memunculkan tampilan data secara lebih detail serta gambar yang diupload. Selain itu pengguna juga dapat melakukan *sorting* berdasarkan kolom tabel dan *filter* data berdasarkan nomor DO, nomor PO, kategori *claim*, dan tanggal serta waktu.



Gambar 3.15. Tampilan awal *claim*

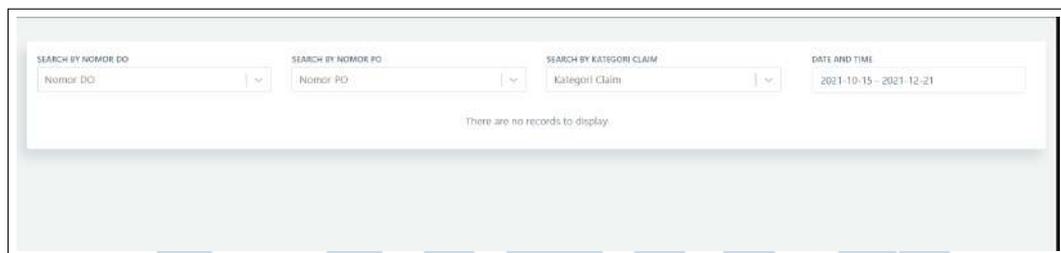


Gambar 3.16. Tampilan detail *claim*



Gambar 3.17. Tampilan formulir penambahan *claim*

Pembeli dapat membuat *claim* maka *shipper* atau pemilik barang perlu mengetahui *claim* yang telah dibuat oleh pembeli. Disini penulis juga membuat tampilan data *claim* berdasarkan *shipper* yang diambil menggunakan API *GraphQL*. Data akan direpresentasikan menggunakan tabel dengan fitur *sorting* berdasarkan kolom tabel dan *filter* data berdasarkan nomor DO, nomor PO, kategori *claim*, dan tanggal serta waktu.



Gambar 3.18. Tampilan awal *list claim*

Pengerjaan fitur-fitur web sudah selesai dan sekarang melanjutkan kembali pekerjaan aplikasi *mobile* yaitu *quality assurance*. Selama pengerjaan tugas-tugas penulis berfokus pada *front end* membuat tampilan dan menghubungkan API dengan *front end*. Sekarang penulis diberi kesempatan untuk membuat API menggunakan *GraphQL*. API ini akan digunakan untuk mendapatkan data-data kendaraan yang dibutuhkan untuk aplikasi *mobile* tersebut. Perancangan API dan relasi antar database dibantu langsung oleh pemimbing dengan membicarakan hal-hal apa saja yang diperlukan dalam API dan database yang diperlukan. Terdapat 6 API yang diperlukan yaitu :

1. API untuk daftar kendaraan
API ini digunakan untuk mengambil seluruh daftar kendaraan berdasarkan tahun. Input pada *parameter* berupa *string* yaitu tahun dan *output* akan berupa *array of object*. Data kendaraan ini juga sudah dihubungkan dengan data pada tabel lain. Dari kendaraan, dapat dilihat tiket *quality assurance*, *transporter*, *pool*, dan jenis kendaraan.
2. API untuk detail kendaraan API ini digunakan untuk mengambil detail kendaraan berdasarkan *id* kendaraan. Input pada *parameter* berupa *string* yaitu *id* dari kendaraan dan *output* berupa *object*. Data detail kendaraan ini terhubung dengan berbagai macam tabel pada database. Tabel yang digunakan untuk membuat API ini dari tabel kendaraan, tiket, detail tiket, *pool*, *transporter*, *shipper*, *trip*, *trip order transaction*, *order*, *order detail*, kota asal, kota tujuan.
3. API untuk dokumen API ini digunakan untuk mengambil data dokumen yang diperlukan berdasarkan tahun. Input pada *parameter* berupa *string* yaitu tahun dan *output* akan berupa *array of object*. Data tiket ini hanya diambil menggunakan satu tabel.
4. API untuk menambah tiket API ini digunakan untuk membuat tiket *quality assurance*. Ketika pengguna menekan salah satu daftar dari kendaraan maka akan dibuatkan tiket.
5. API untuk memperbarui status dokumen Ketika pengguna memilih salah satu dokumen untuk dilakukan pengecekan maka pengguna dapat memilih *reject* atau *approve* pada dokumen tersebut setelah itu memasukkan gambar atau *file* dengan ekstensi pdf sebagai bukti.
6. API untuk memperbarui status terakhir API ini digunakan untuk memperbarui status terakhir pada tiket *quality assurance*. Pengguna dapat memilih *reject* atau *approve* pada kendaraan tersebut. Jika *reject* maka kendaraan tersebut tidak dapat digunakan. Jika *approve* maka kendaraan dapat digunakan.

Fitur Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Bulan Ke -	Pekerjaan yang dilakukan
1	Mempelajari kembali <i>JavaScript</i> dan <i>React JS</i> serta membuat fitur unggah gambar atau <i>file pdf</i> dan tampilan pada web <i>help desk</i>
2	Mempelajari kembali <i>Java</i> untuk aplikasi <i>mobile</i> dan membuat tampilan pada aplikasi <i>quality assurance</i>
3	Membuat fitur web yaitu <i>lead time</i> , muat bongkar dan unggah <i>shipper contact</i>
4	Membuat fitur web yaitu menambah <i>claim</i> dan daftar <i>claim shipper</i> , lalu mempelajari penggunaan <i>GraphQL</i>
5	Membuat API untuk aplikasi <i>mobile quality assurance</i> dan menghubungkan API agar dapat dilihat di dalam aplikasi
6	Melanjutkan pembuatan aplikasi <i>mobile quality assurance</i>

Setelah melakukan implementasi pada pembuatan setiap fitur, maka akan dilakukan *testing*. Pekerjaan yang dibuat adalah salah satu fitur dari berbagai macam fitur sehingga untuk *testing* dimulai dengan metode *white box testing*. *testing* dibantu oleh penanggung jawab dan dimulai dengan menguji setiap *functional* komponen yang ada didalam fitur tersebut. Jika sudah berjalan dengan baik komponen tersebut maka akan lanjut ke komponen berikutnya. Jika tidak sesuai dengan yang diinginkan misalkan terjadi *error* atau komponen dilihat memerlukan waktu yang lama untuk menjalankan fungsi tersebut maka akan dilakukan revisi dan diberi masukan bagaimana menyelesaikan masalah yang ada. Jika fungsionalitas sudah berjalan dengan baik maka sekarang akan beralih ke *black box testing* yang dimulai dengan melihat tampilan atau *user interface*. Jika tampilan dan komponen yang dibutuhkan sudah sesuai maka akan berlanjut ke komponen berikutnya jika belum sesuai maka akan dilakukan revisi dan diberi masukan dalam desain tampilannya.

Ketika *testing* sudah selesai dari segi tampilan dan juga fungsionalitas maka kodingan yang telah dibuat akan dipush ke *branch* milik *developer*. Selanjutnya penanggung jawab akan melakukan implementasi hasil dari yang telah dibuat oleh masing-masing *developer* lalu akan di *embed* ke *main page* sesuai dengan fitur yang dibutuhkan. Setelah melakukan *embed* maka dilanjutkan dengan *integration testing* yang dilakukan sendiri oleh

penanggung jawab.

3.2.3 Kendala yang Ditemukan

Berikut merupakan beberapa kendala yang dihadapi penulis selama praktik kerja magang berlangsung antara lain :

1. Praktik kerja magang yang dilakukan dirumah atau *online* membuat jalur komunikasi terhambat
2. Pemahaman akan proses digital bisnis yang masih sangat minim
3. Minimnya pengetahuan akan beberapa *framework* yang digunakan pada perusahaan

3.2.4 Solusi atas Kendala yang ditemukan

Solusi atas kendala yang dihadapi selama praktik kerja magang berlangsung antara lain :

1. Melakukan koordinasi melalui telepon atau *meeting*
2. Bertanya kepada pemimbing maupun rekan kerja lainnya
3. Mempelajari *framework* melalui internet dan bertanya kepada pemimbing maupun rekan kerja lainnya

