

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Program kerja magang dilakukan di PT. Intersoft yang ditempatkan pada divisi *Management Information Systems* (MIS) yang dipimpin langsung oleh seorang *team leader* dan seorang Pembimbing Lapangan. Bersama dengan anggota tim *developer* yang lain, mahasiswa magang diminta untuk membantu perancangan dan pengembangan aplikasi berbasis Android yang diberi nama Galva Apps.



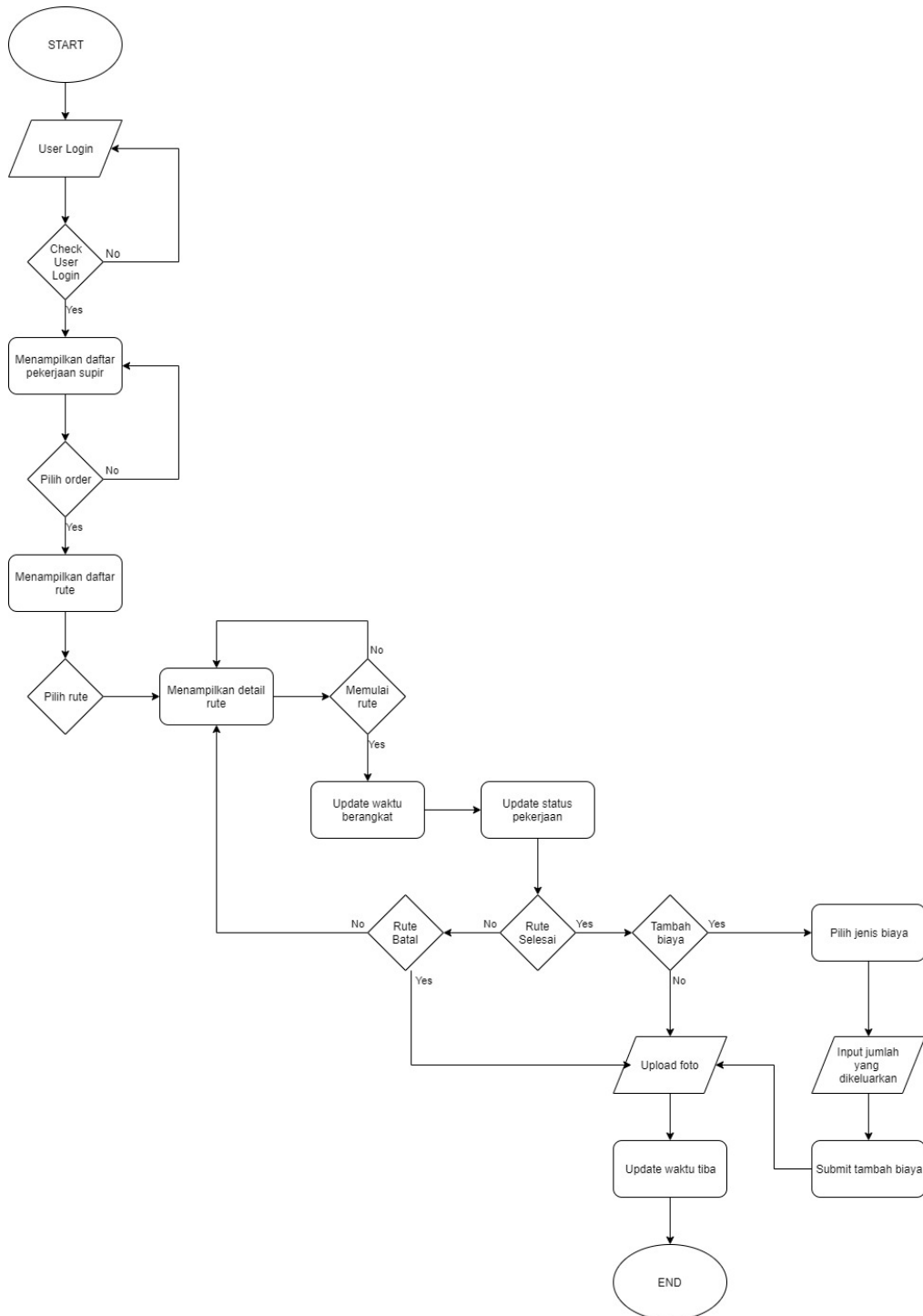
Gambar 3.1. Logo dari aplikasi Galva Apps

3.2. Tugas yang Dilakukan

Pada program kerja magang ini, sebagai *back-end programmer* memiliki tugas di PT. Intersoft yaitu membuat sebuah aplikasi berbasis Android, dimana aplikasi ini akan menampilkan daftar tugas yang harus dikerjakan oleh tiap supir setiap harinya. Tugas akan diberikan oleh admin melalui *web server* yang kemudian akan terkirim ke telepon genggam masing-masing supir. Aplikasi ini akan digunakan untuk operasional supir perusahaan, baik kantor pusat maupun kantor perwakilan.

Galva Trans merupakan salah satu dari beberapa aplikasi lain yang berada di dalam aplikasi *Galva Apps*, dimana aplikasi ini akan digunakan untuk menunjang operasional perusahaan sehari-hari.

Berikut merupakan *flowchart diagram* dari alur aplikasi Galva Trans.



Gambar 3.2. Flowchart Diagram Aplikasi Galva Trans

3.3. Uraian Pelaksanaan Kerja Magang

Berikut merupakan *timeline* dan uraian pelaksanaan kerja selama menjalani Program Kerja Magang sebagai *back end programmer*.

Tabel 3.1. Timeline Uraian Kerja Magang

Minggu ke-	Kegiatan Yang Dilakukan
1	<ol style="list-style-type: none">Membuat <i>layout</i> atau tampilan untuk aplikasi Galva Trans.Belajar mandiri tentang penggunaan <i>RecyclerView</i> dan <i>CardView</i> dalam Android Studio.
2	<ol style="list-style-type: none">Mempelajari cara penulisan dengan arsitektur MVP pada Android Studio.Melakukan revisi terhadap <i>layout</i> yang sudah dibuat (sesuai dengan permintaan perusahaan).Mempelajari <i>flow</i> aplikasi Galva Trans secara keseluruhan.
3	<ol style="list-style-type: none">Mempelajari dan menggunakan Retrofit untuk menampilkan data pada aplikasi.Melakukan <i>request</i> API untuk menampilkan daftar pekerjaan pada aplikasi Galva Trans.
4	<ol style="list-style-type: none">Membuat <i>coding</i> untuk menampilkan data melalui API yang sudah diberikan.Membuat <i>coding</i> aplikasi sesuai dengan <i>flow</i> yang sudah direncanakan.Membuat halaman <i>login</i> aplikasi Galva Trans.
5	<ol style="list-style-type: none">Melakukan <i>coding</i> aplikasi untuk Galva Trans hingga selesai.Melakukan penambahan fitur tambah biaya pada aplikasi Galva Trans.Melakukan penambahan fitur <i>upload</i> foto pada aplikasi Galva Trans.
6	<ol style="list-style-type: none">Melakukan <i>testing</i> terhadap aplikasi Galva Trans.Membuat <i>layout</i> atau tampilan <i>dashboard</i> (menu utama) untuk Galva Apps.Membuat <i>layout</i> untuk fitur gaji dan cuti pada aplikasi Galva Apps.
7	<ol style="list-style-type: none">Melakukan <i>request</i> API untuk menampilkan gaji, serta mengajukan dan menampilkan daftar cuti.

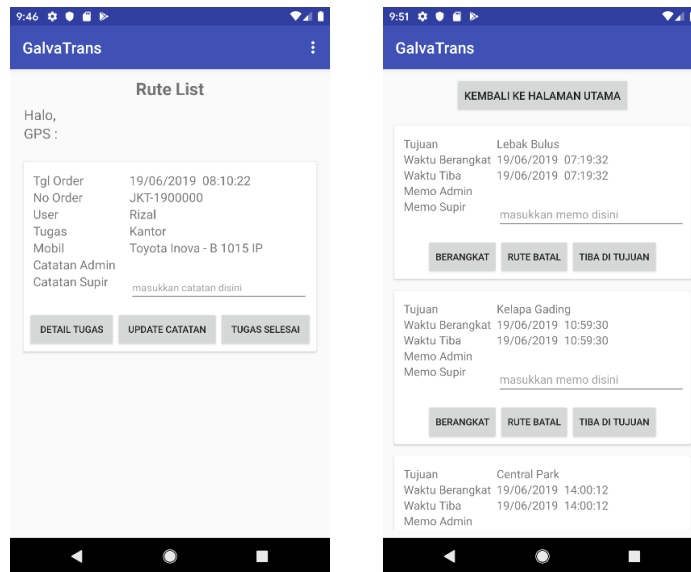
	<ul style="list-style-type: none"> b. Membuat <i>coding</i> untuk menampilkan data pada fitur gaji. c. Membuat <i>coding</i> untuk menambah dan menampilkan data pada fitur cuti.
8	<ul style="list-style-type: none"> a. Membuat <i>layout</i> untuk fitur klaim. b. Membuat <i>coding</i> untuk menambah dan menampilkan data pada fitur klaim.

3.3.1. Minggu Pertama

Pada minggu pertama, kegiatan yang dilakukan adalah membuat tampilan dari aplikasi Galva Trans dan mempelajari cara menggunakan *RecyclerView* dan *CardView* pada Android Studio. *RecyclerView* merupakan salah satu elemen dalam *Android Material Design* yang digunakan untuk menampilkan informasi atau data dalam jumlah besar dalam layar yang terbatas. *RecyclerView* menjadi salah satu cara yang efisien karena keunggulannya sendiri yaitu dapat menampilkan data dalam jumlah besar secara rapi dan terstruktur, serta performanya lebih baik dibandingkan dengan *ListView*.

CardView juga merupakan salah satu elemen dalam *Android Material Design* yang biasa digunakan sebagai container untuk setiap *item* dalam *RecyclerView* dan untuk membungkus suatu *layout* dalam aplikasi. Keunggulan yang dimiliki oleh *CardView* adalah sudah memiliki *elevation* dan *radius* sudut bawaan agar menghasilkan *layout* yang konsisten.

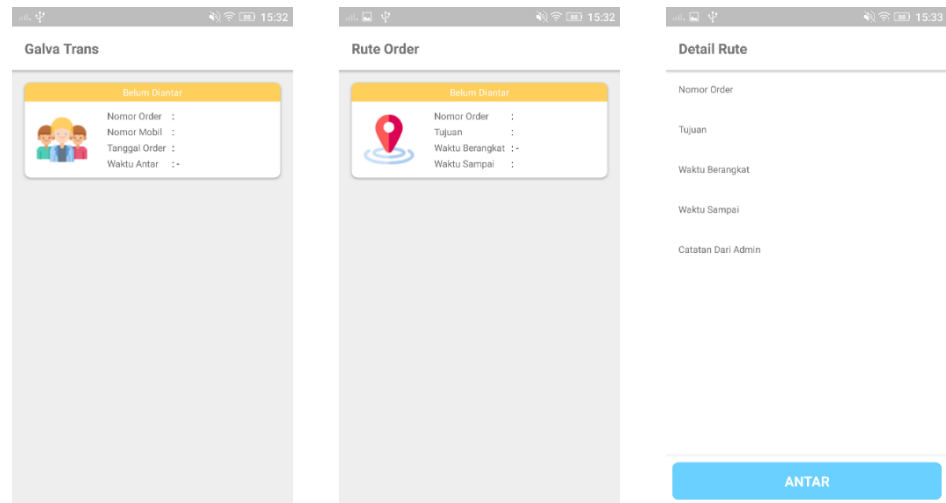
Kedua elemen tersebut digunakan dalam pembuatan *layout* pada aplikasi Galva Trans seperti yang terlihat pada Gambar 3.2. untuk menampilkan daftar pekerjaan yang diberikan kepada supir.



Gambar 3.3. Layout awal dari Galva Trans

3.3.2. Minggu Kedua

Di minggu kedua, kegiatan yang dilakukan adalah melakukan revisi terhadap *layout* yang sudah dibuat sesuai dengan permintaan dari perusahaan. Perusahaan menginginkan tampilan yang lebih sederhana dan lebih konsep dalam segi penerapan warna, *font*, dan format desain supaya aplikasi yang dikembangkan dapat sesuai dengan kebutuhan sehingga perubahan yang dilakukan cukup signifikan. Tampilan yang lebih sederhana disini diharapkan dapat memudahkan para *user* dalam memahami alur aplikasi yang akan digunakan. Perubahan tampilan *layout* dapat dilihat pada gambar 3.3..



Gambar 3.4. Layout setelah direvisi

Kegiatan lain yang dilakukan adalah mempelajari cara penulisan menggunakan arsitektur MVP atau *Model-View-Presenter* pada Android Studio. MVP merupakan sebuah konsep arsitektur pengembangan aplikasi yang memisahkan antara tampilan aplikasi dengan proses bisnis yang bekerja pada aplikasi. Pengembangan aplikasi Galva Trans menggunakan arsitektur MVP agar lebih terstruktur, mudah dilakukan *testing*, dan juga mudah untuk melakukan *maintenance*. MVP terbagi menjadi tiga *layer* yaitu:

1. *Model* merupakan *layer* yang menunjuk kepada objek dan data yang ada pada aplikasi,
2. *View* merupakan *layer* untuk menampilkan data dan interaksi langsung dengan *user*, biasanya berupa *activity*, *fragment*, atau *dialog* pada Android.
3. *Presenter* merupakan *layer* yang menghubungkan komunikasi antara *Model* dan *View*. Setiap interaksi yang

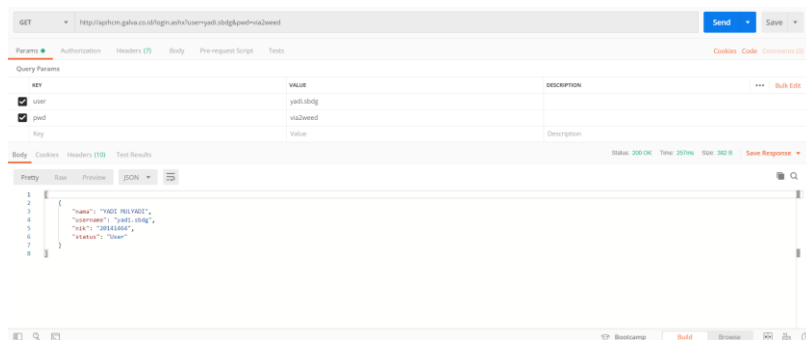
dilakukan oleh *user* akan memanggil *Presenter* untuk memproses dan mengakses *Model* lalu mengembalikan responsnya kembali kepada *View*.

3.3.3. Minggu Ketiga

Pada minggu ketiga, kegiatan yang dilakukan adalah mempelajari tentang penggunaan Retrofit untuk menampilkan data pada sistem atau aplikasi yang dituju. Retrofit merupakan *client* HTTP yang aman untuk Android dan Java. Retrofit memudahkan untuk terhubung dengan layanan web REST dengan menerjemahkan API ke dalam *interface* Java. Retrofit mengurai data JSON atau XML menjadi *Plain Old Java Objects* atau disebut juga POJOs. Segala permintaan GET, POST, PUT, PATCH, dan DELETE dapat dieksekusi menggunakan Retrofit. Di sisi lain, Retrofit merupakan *open-source library* yang sudah terdokumentasi dan teruji sehingga akan menghemat banyak waktu dan tenaga, khususnya dalam menangani permintaan jaringan.

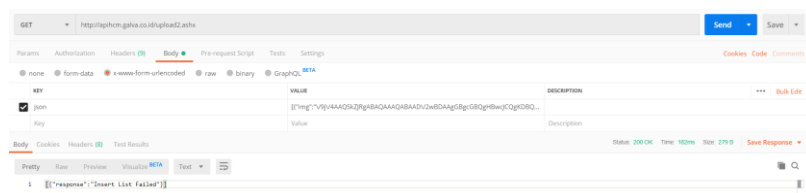
Kegiatan lain yang dilakukan adalah melakukan request API kepada anggota tim yang lain untuk menampilkan data yang diperlukan di dalam aplikasi. Beberapa API yang diminta digunakan untuk memeriksa apakah *user* terdaftar sebagai supir atau tidak, menampilkan daftar pekerjaan yang harus dikerjakan oleh supir, serta menyimpan data yang dimasukkan oleh *user* melalui aplikasi seperti biaya dan foto.

Pada gambar 3.4. dapat dilihat contoh API yang melakukan *request* validasi *user*, apakah *user* terdaftar sebagai supir atau tidak.



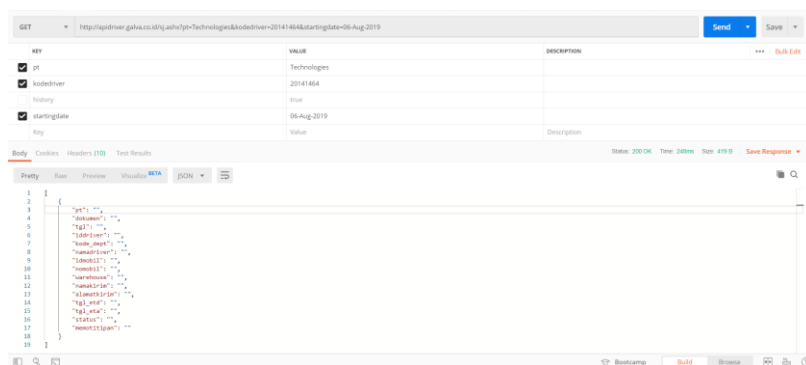
Gambar 3.5. Contoh API validasi user

Contoh API pada gambar 3.5. memperlihatkan proses *request* untuk *upload* foto, apakah foto sudah berhasil ter-*upload* atau tidak.



Gambar 3.6. Contoh API upload foto

Pada gambar 3.6. dapat dilihat contoh API yang melakukan *request* untuk menampilkan daftar pekerjaan supir.



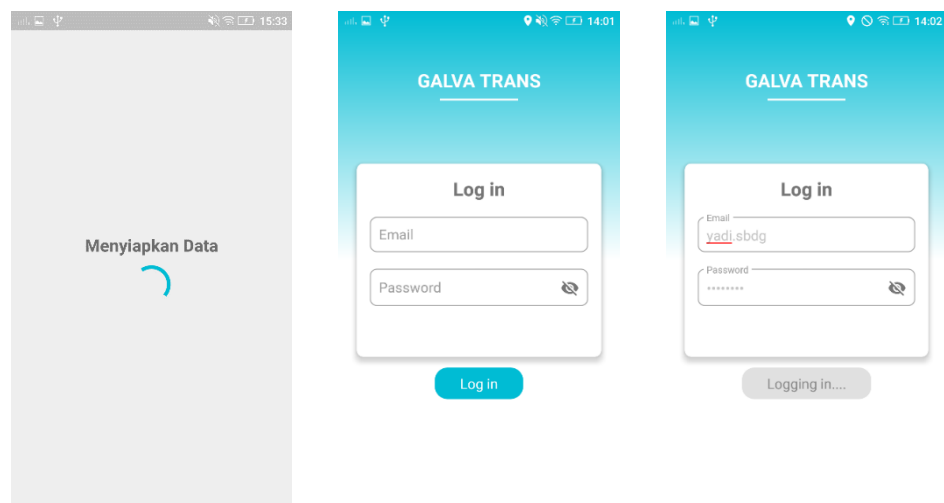
Gambar 3.7. Contoh API daftar pekerjaan supir

3.3.4. Minggu Keempat

Di minggu keempat, kegiatan yang dilakukan adalah membuat *layout* untuk *login* ke dalam aplikasi Galva Trans. Di halaman *login* ini,

user harus memasukkan *username* dan *password* yang sudah diberikan. *User* atau supir yang tidak memiliki *username* dan *password* tidak dapat memasuki aplikasi Galva Trans. Oleh karena itu, setiap supir atau *user* yang akan menggunakan aplikasi Galva Trans akan dibuatkan sebuah *username* dan *password* untuk *login*.

Sebelum masuk ke halaman *login*, aplikasi akan menampilkan *splash screen* dengan tulisan “Menyiapkan Data”, dimana hal ini menandakan bahwa aplikasi sedang menyiapkan data dalam aplikasi. Apabila *user* sudah *login* sebelumnya, maka setelah muncul *splash screen*, *user* akan diarahkan langsung ke halaman utama dari Galva Trans. Langkah ini dapat dilihat pada Gambar 3.7..



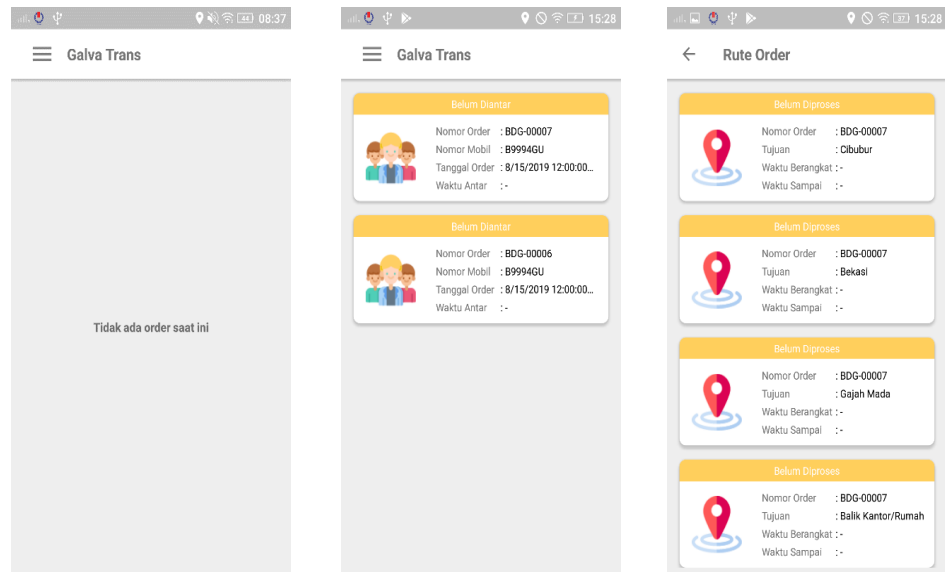
Gambar 3.8. Splash dan login screen Galva Trans

Kegiatan lain yang dilakukan adalah membuat *coding* terhadap aplikasi Galva Trans sesuai dengan alur yang sudah direncanakan dan membuat *coding* untuk menampilkan data menggunakan Retrofit dan API yang sudah diberikan.

3.3.5. Minggu Kelima

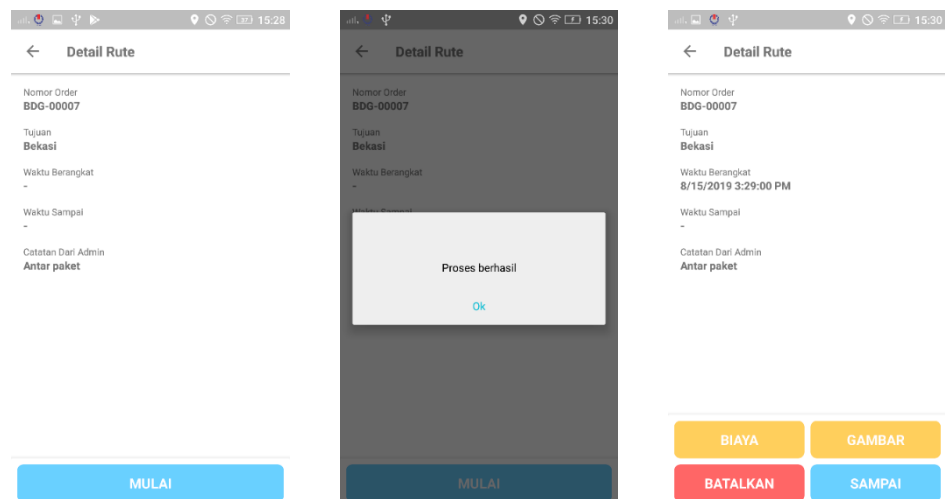
Pada minggu berikutnya, kegiatan yang dilakukan adalah menyelesaikan *coding* untuk pembuatan aplikasi Galva Trans dan menambahkan beberapa fitur pada aplikasi, seperti *upload* foto dan *input* biaya tambahan. Fitur ini ditambahkan untuk mempermudah pemeriksaan pekerjaan yang dilakukan oleh supir.

Berikut merupakan tampilan dan alur aplikasi yang sudah selesai. Pertama, setelah supir melakukan *login* dengan *username* dan *password* yang sudah diberikan, aplikasi Galva Trans akan menampilkan daftar pekerjaan yang harus diselesaikan. Ketika salah satu *order* dibuka, akan memunculkan daftar rute yang harus dituju. Rute tersebut dapat diselesaikan tanpa sesuai urutan, namun untuk rute “Balik Kantor/Rumah” harus diselesaikan terakhir setelah seluruh rute yang lain diselesaikan. Jika supir tersebut tidak mendapatkan daftar pekerjaan, maka aplikasi akan menampilkan tulisan “Tidak ada order saat ini”. Tampilan ini dapat dilihat pada Gambar 3.8..



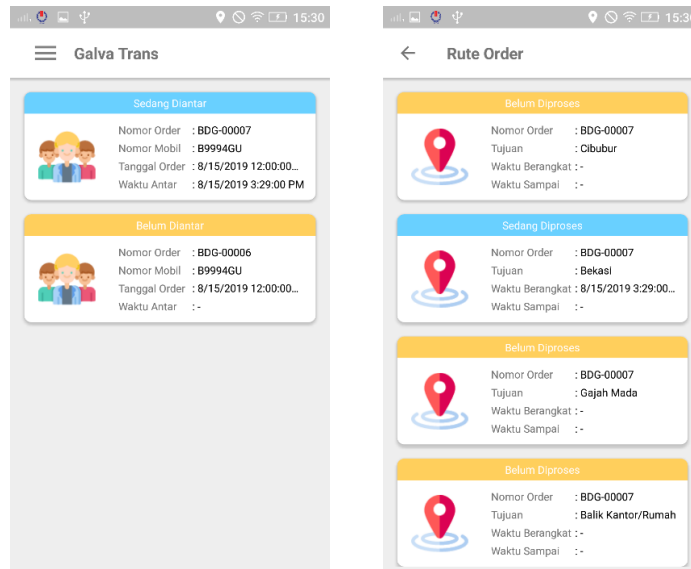
Gambar 3.9. Tampilan daftar pekerjaan dan rute order

Kemudian ketika salah satu rute dibuka, aplikasi akan menampilkan informasi mengenai detail rute dan tombol untuk memulai pekerjaan. Saat supir akan berangkat menuju rute tersebut, tombol “Mulai” harus ditekan untuk meng-*update* waktu berangkat. Jika proses “Mulai” berhasil, tampilan akan berubah menjadi seperti Gambar 3.9 paling kanan.



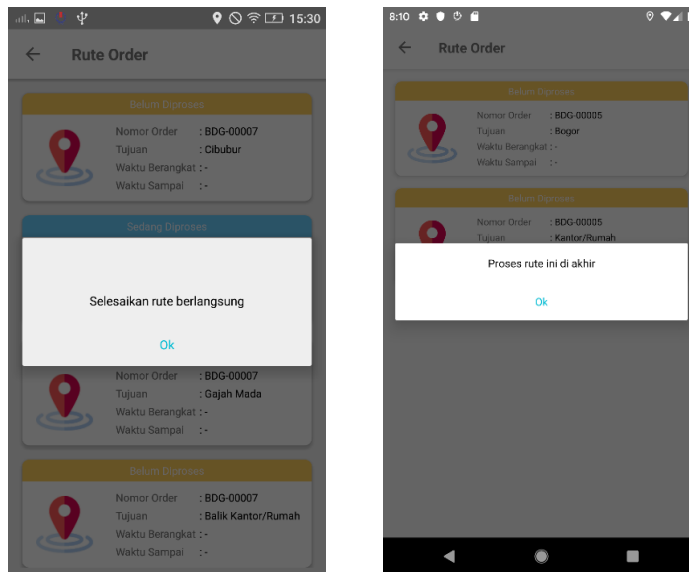
Gambar 3.10 Tampilan detail rute (Start)

Ketika supir sudah memulai perjalanan, status pekerjaan akan berubah menjadi “Sedang Diantar”, sedangkan status dari rute yang sedang dikerjakan akan berubah menjadi “Sedang Diproses”, seperti yang terlihat pada Gambar 3.10..



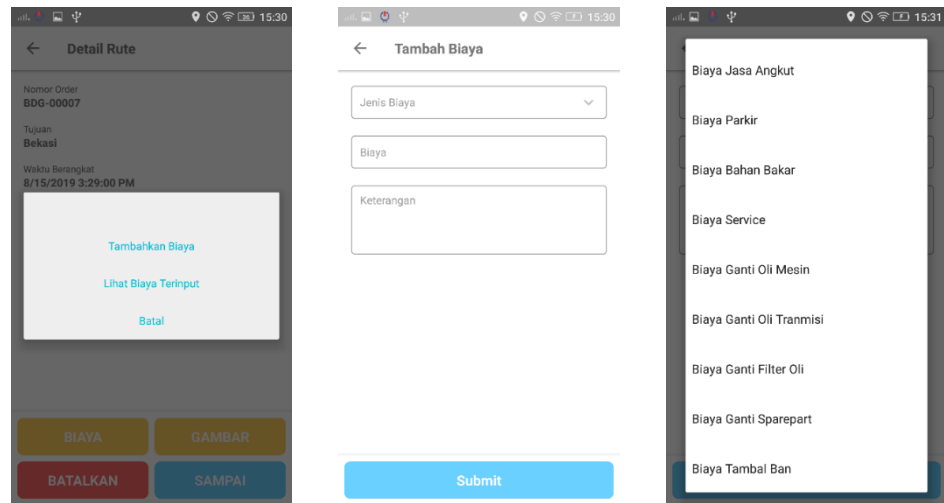
Gambar 3.11. Status pada daftar pekerjaan akan berubah

Supir tidak dapat membuka informasi rute lain selama ada rute yang sedang diproses. Jika supir mencoba untuk membuka rute, akan muncul sebuah *alert box* dengan tulisan “Selesaikan rute berlangsung”. Ketika supir ingin kembali ke kantor atau rumah ketika rute atau pekerjaan belum selesai, maka akan muncul *alert box* yang lain dengan tulisan “Proses rute ini di akhir”. Tampilan *alert box* bisa dilihat pada Gambar 3.11..



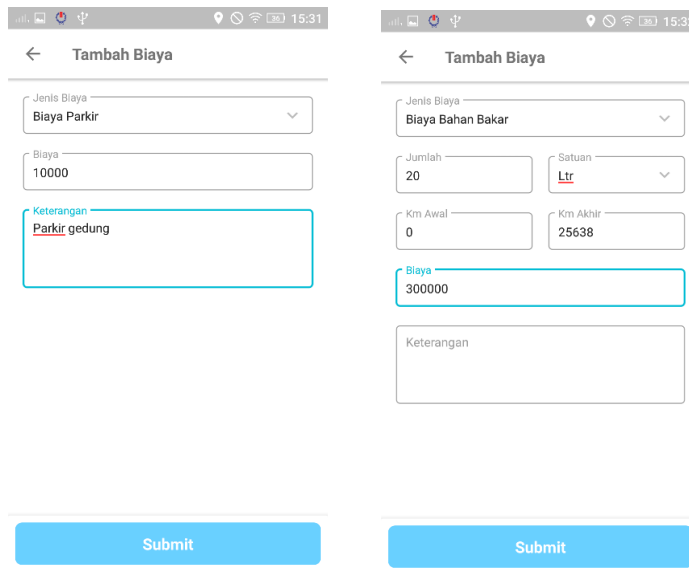
Gambar 3.12. Dialog box ketika supir membuka rute lain

Selama menyelesaikan pekerjaan atau rute, supir dapat menambahkan atau memasukkan biaya yang dikeluarkan seperti yang ditampilkan pada Gambar 3.12.. Tombol “Biaya” akan memunculkan sebuah *dialog box* dengan tiga pilihan yaitu, “Tambahkan biaya”, “Lihat Biaya Terinput”, dan “Batal”. Jika supir ingin memasukkan atau menambahkan biaya, supir dapat memilih pilihan pertama yaitu “Tambahkan Biaya”. Kemudian aplikasi akan mengarahkan ke halaman untuk menambahkan biaya. Supir dapat memilih jenis biaya yang dikeluarkan, dengan menekan bagian “Jenis Biaya”, kemudian akan muncul beberapa pilihan untuk jenis biaya.



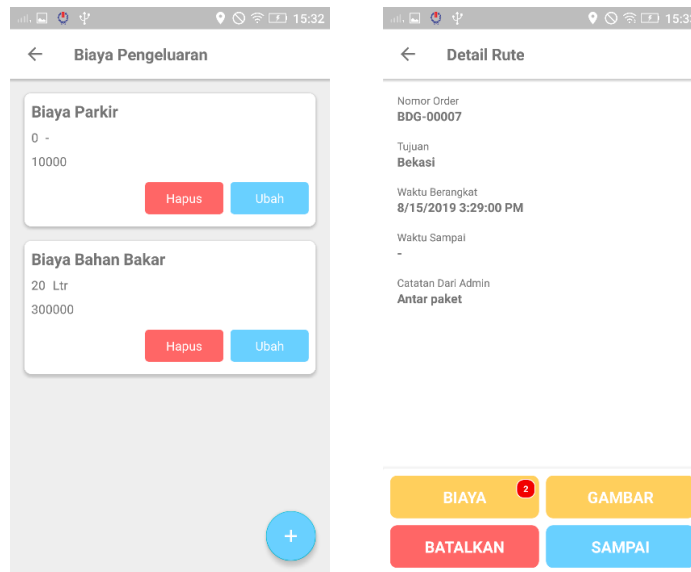
Gambar 3.13. Tampilan untuk menambahkan biaya

Sebagai contoh pada Gambarr 3.13. sebelah kiri, jika jenis biaya yang dipilih adalah “Biaya Parkir”, maka supir hanya akan memasukkan jumlah biaya yang dikeluarkan dan keterangannya, tetapi jika jenis biaya yang dipilih adalah “Biaya Bahan Bakar” seperti pada Gambar 3.13. sebelah kanan, maka supir harus memasukkan jumlah liter yang didapat dengan angka bulat, KM akhir ketika kendaraan mengisi bahan bakar, biaya yang dikeluarkan, dan keterangan berupa jumlah liter yang didapat secara lengkap. Setelah supir mengisi seluruh form, maka supir dapat melanjutkan proses dengan menekan tombol “Submit”.



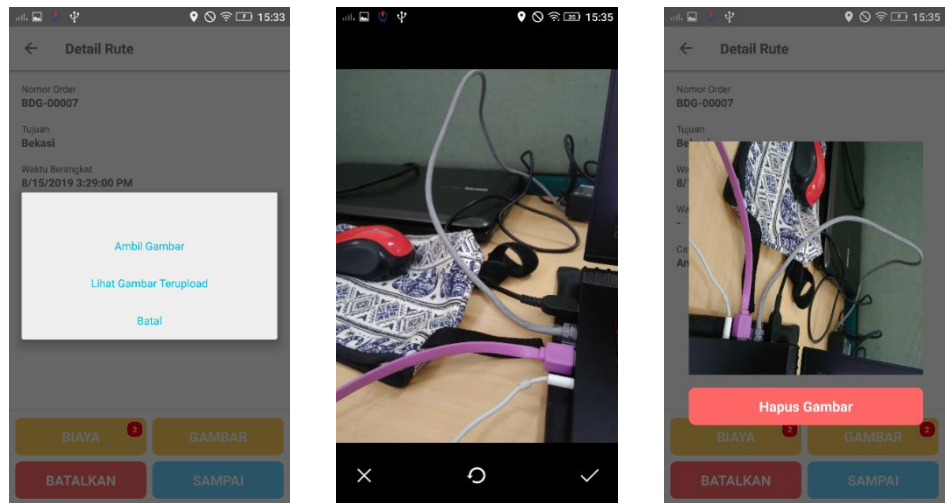
Gambar 3.14. Tampilan penambahan Biaya Parkir dan Bahan Bakar

Ketika biaya yang sudah dimasukkan berhasil diproses, supir dapat melihat biaya yang sudah dimasukkan di dalam sebuah daftar (seperti yang terlihat pada Gambar 3.14. sebelah kiri) melalui pilihan “Lihat Biaya Terinput” pada *dialog box*. Supir dapat melihat berapa jenis biaya yang dimasukkan melalui *badge* yang muncul pada tombol “Biaya”. Sebagai contoh pada Gambar 3.14. sebelah kanan, ketika supir memasukkan Biaya Parkir dan Biaya Bahan Bakar, maka *badge* akan menunjukkan angka dua (2).



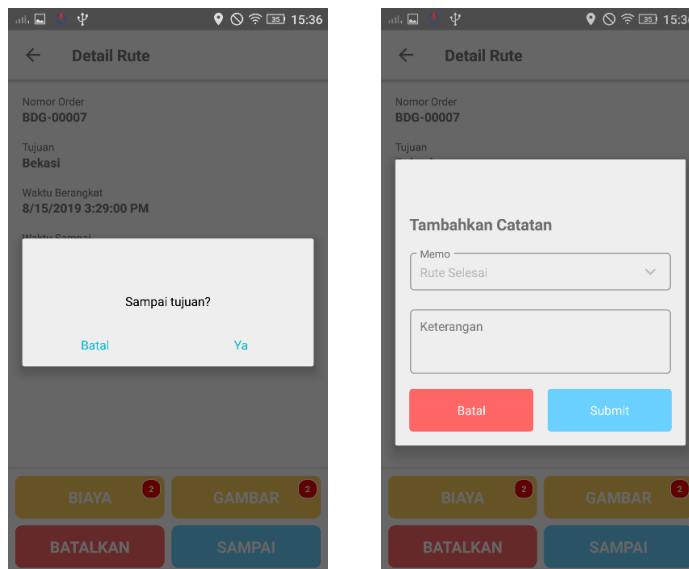
Gambar 3.15. Tampilan daftar biaya yang sudah dimasukkan

Ketika supir akan menyelesaikan sebuah rute, supir harus meng-*upload* foto sebagai bukti bahwa rute sudah diselesaikan. Supir dapat melakukan proses *upload* foto dengan menekan tombol “Gambar”. Tombol tersebut akan memunculkan sebuah *dialog box* dengan tiga pilihan, yaitu “Ambil Gambar”, “Lihat Gambar Terupload”, dan “Batal”. Foto tidak dapat di-*upload* melalui foto yang sudah tersimpan di galeri, oleh karena itu foto harus diambil ketika pekerjaan sudah diselesaikan. Untuk mengambil foto, supir akan memilih “Ambil Gambar” dan aplikasi akan membuka kamera *handphone*. Supir dapat melihat gambar yang sudah diambil dengan memilih “Lihat Gambar Terupload”. Jumlah maksimal foto yang dapat diambil adalah tiga foto. Fitur *upload* foto ini dapat dilihat pada Gambar 3.15..



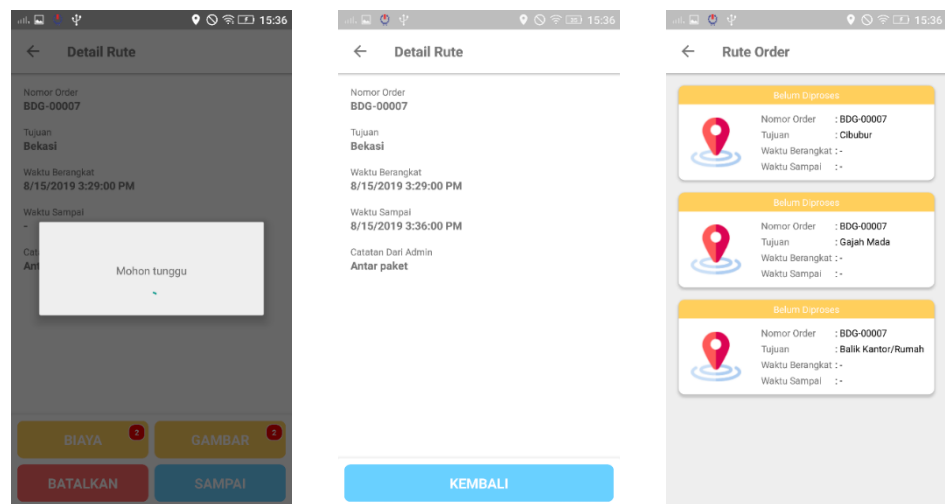
Gambar 3.16. Tampilan mengambil dan melihat gambar yang sudah diambil

Jika supir sudah meng-*upload* foto, maka supir dapat menyelesaikan rute dengan menekan tombol “Sampai” dan menambahkan catatan jika ada. Tampilan ketika supir sudah menekan tombol “Sampai” dapat dilihat pada Gambar 3.16..



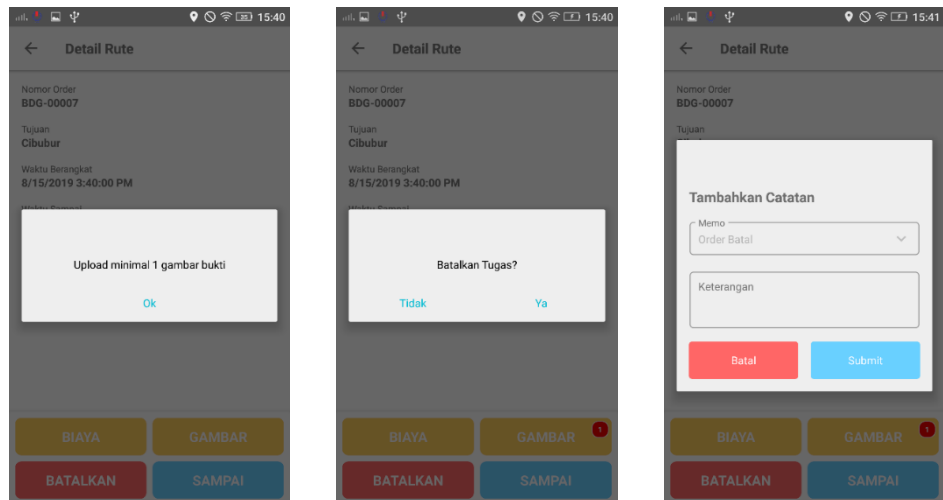
Gambar 3.17. Tampilan ketika supir akan menyelesaikan rute

Dapat dilihat pada Gambar 3.17., aplikasi akan meng-*update* waktu sampai dan mengubah status pekerjaan ketika proses berhasil. Rute yang sudah diselesaikan akan hilang dari daftar rute.



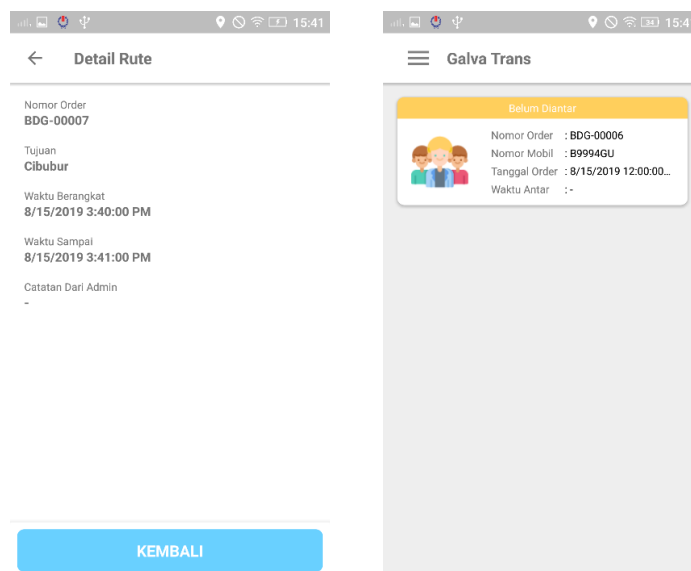
Gambar 3.18. Tampilan setelah supir berhasil menyelesaikan rute

Jika supir ingin membatalkan sebuah rute atau pekerjaan, supir diharuskan meng-*upload* foto atau gambar sebagai bukti bahwa rute dibatalkan. Setelah supir meng-*upload* foto, maka supir dapat membatalkan rute dan menambahkan catatan berupa alasan rute tersebut dibatalkan, seperti pada Gambar 3.18. paling kanan.



Gambar 3.19. Tampilan ketika supir akan membatalkan rute

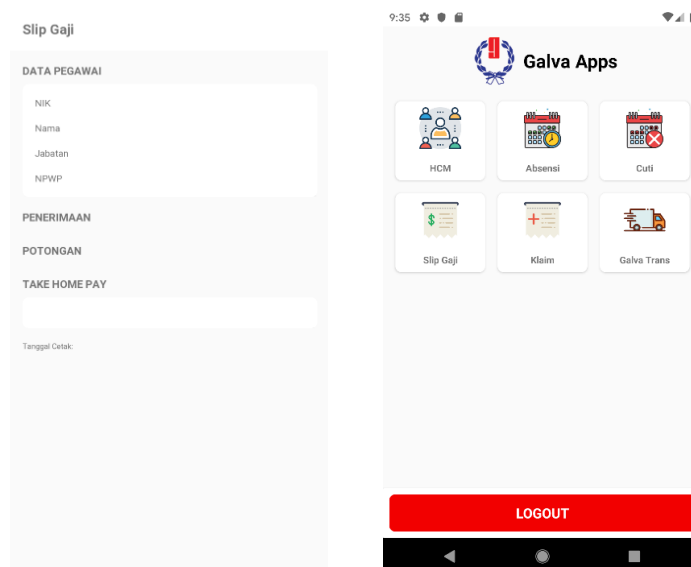
Ketika sebuah rute dibatalkan, maka seluruh rute lain yang belum diselesaikan pada pekerjaan tersebut juga akan ikut dibatalkan dan hilang dari daftar pekerjaan yang harus diselesaikan. Proses ini dapat dilihat pada Gambar 3.19..



Gambar 3.20. Tampilan ketika supir berhasil membatalkan rute

3.3.6. Minggu Keenam

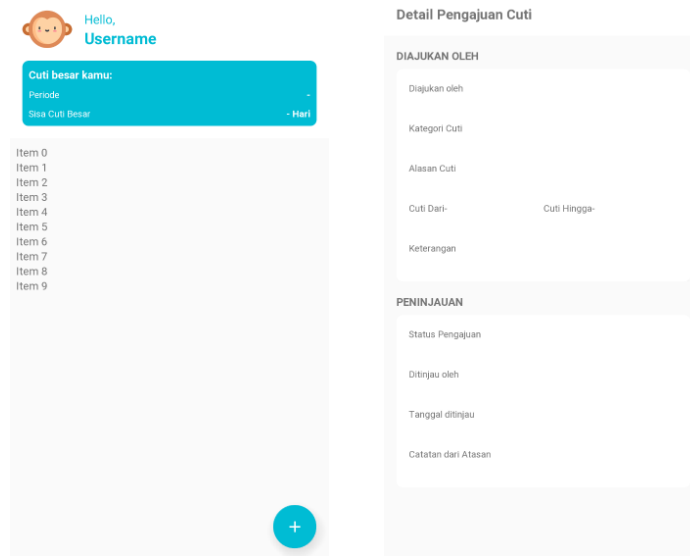
Di minggu keenam, kegiatan yang dilakukan adalah membuat *layout* untuk menampilkan laporan gaji milik karyawan dan *layout* untuk *dashboard* atau menu utama dari Galva Apps. Seperti yang terlihat pada Gambar 3.20. sebelah kiri, laporan yang ditampilkan berupa jumlah penerimaan dan potongan yang diperoleh oleh karyawan tersebut. Informasi lain yang perlu ditampilkan adalah NIK, nama karyawan, jabatan, dan NPWP karyawan tersebut. Sedangkan, untuk *layout* menu utama Galva Apps, beberapa fitur yang ditampilkan adalah absensi, cuti, slip gaji, klaim, HCM, dan Galva Trans. Tampilan menu utama Galva Apps dapat dilihat pada Gambar 3.20. sebelah kanan.



Gambar 3.21. Tampilan laporan gaji dan *layout dashboard* Galva Apps

Kegiatan lain yang dilakukan adalah membuat *layout* untuk menampilkan daftar cuti dan detail pengajuan cuti seperti terlihat pada Gambar 3.21.. Daftar cuti yang ditampilkan juga memberikan informasi lain

yaitu jumlah jatah cuti besar yang dimiliki oleh karyawan dalam periode tertentu.



Gambar 3.22. Tampilan daftar cuti dan detail pengajuan cuti

Selain itu, dilakukan juga *testing* terhadap aplikasi Galva Trans untuk menghindari terjadinya *error* dan memastikan bahwa aplikasi sudah dapat digunakan oleh *user*. *Testing* dilakukan mulai dari halaman *login* hingga proses penyelesaian rute order dalam daftar pekerjaan.

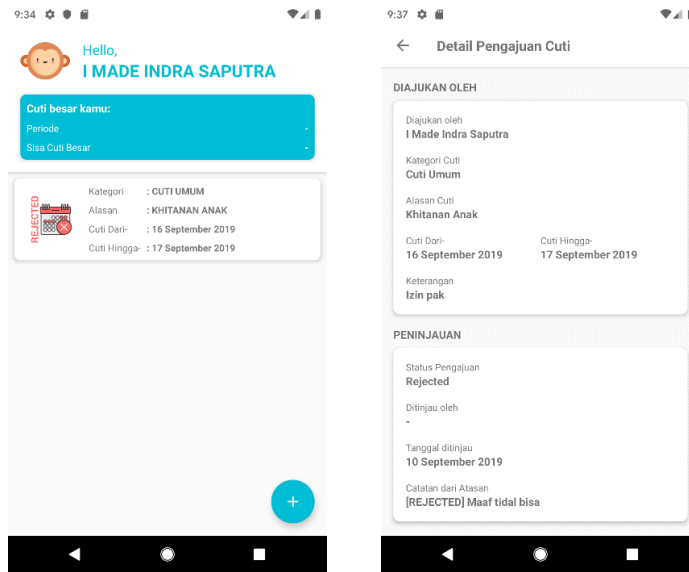
3.3.7. Minggu Ketujuh

Pada minggu ketujuh, kegiatan yang dilakukan adalah menampilkan data untuk laporan gaji karyawan yang menunjukkan secara detail penerimaan dan potongan yang diperoleh oleh karyawan. *Take Home Pay* merupakan total gaji yang diperoleh setelah total penerimaan dikurangi dengan total potongan. Laporan ini hanya menampilkan gaji karyawan per bulan. Tampilan laporan gaji dapat dilihat pada Gambar 3.22..



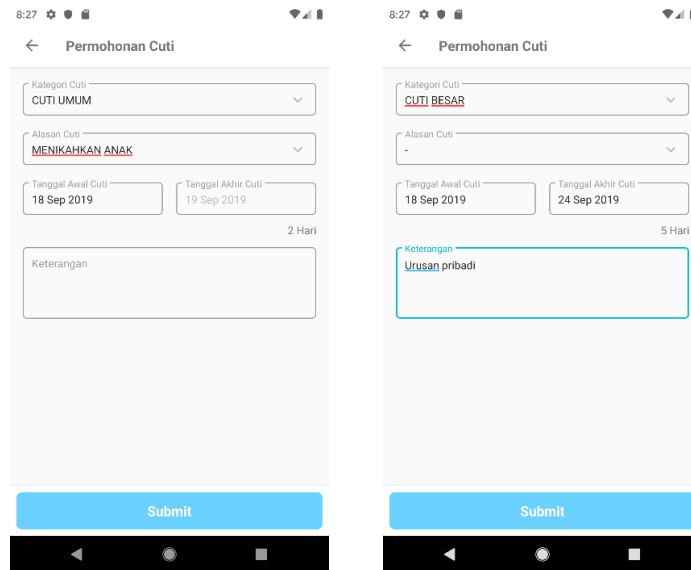
Gambar 3.23. Tampilan laporan gaji karyawan

Kegiatan lain yang dilakukan adalah membuat *coding* untuk mengajukan cuti, menampilkan jatah cuti yang dimiliki, dan menampilkan daftar cuti beserta detail pengajuannya. Pada daftar cuti akan ditampilkan cuti yang sudah diajukan oleh karyawan beserta dengan statusnya. Ketika karyawan baru mengajukan cuti, maka statusnya akan “*Pending*”. Jika pengajuan cuti karyawan disetujui oleh atasannya, maka statusnya akan “*Approved*”, tetapi jika pengajuan cutinya ditolak oleh atasan, maka statusnya akan “*Rejected*”. Halaman ini juga akan menampilkan jatah cuti yang dimiliki oleh karyawan dalam periode tertentu. Umumnya, setiap karyawan memiliki jatah cuti sebanyak 30 (tiga puluh) hari dalam kurun periode 5 (lima) tahun. Pada Gambar 3.23. dapat dilihat daftar dan detail pengajuan cuti yang diajukan oleh karyawan.



Gambar 3.24. Tampilan daftar cuti dan detail pengajuan cuti

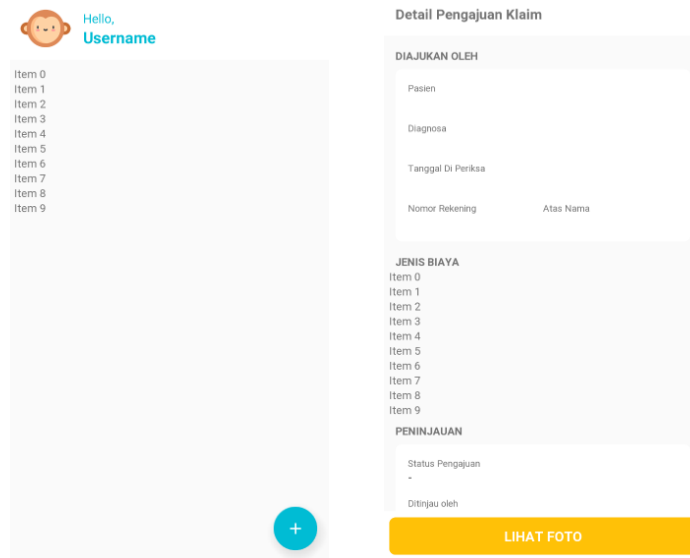
Untuk permohonan cuti, kategori cuti sendiri terbagi menjadi tiga, yaitu Cuti Umum, Cuti Khusus, dan Cuti Besar. Untuk Cuti Umum dan Cuti Khusus, alasan cuti dapat dipilih langsung karena sudah ditampilkan menggunakan API. Sedangkan untuk Cuti Besar, karyawan harus memasukkan keterangannya sendiri. Perhitungan jumlah hari cuti yang diajukan dilakukan secara otomatis ketika karyawan sudah memilih tanggal awal dan tanggal akhir cuti. Setelah mengisi seluruh form untuk permohonan cuti, karyawan dapat menekan tombol “Submit” untuk melanjutkan proses. Proses permohonan cuti karyawan dapat dilihat pada Gambar 3.24..



Gambar 3.25. Tampilan permohonan cuti

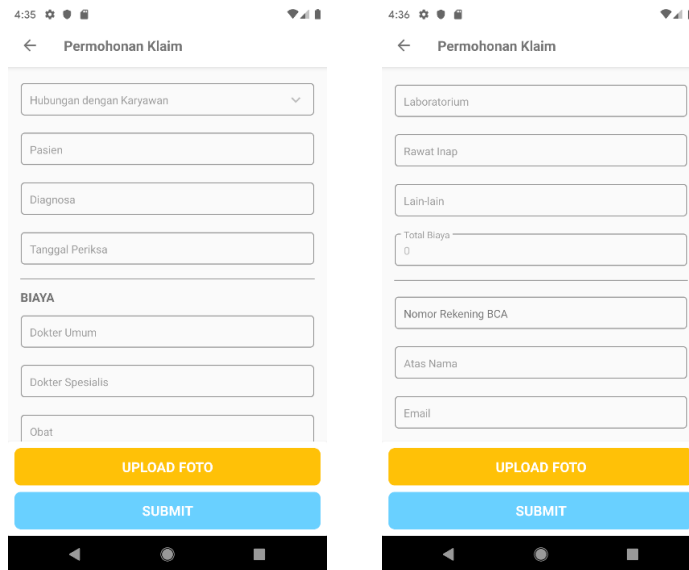
3.3.8. Minggu Kedelapan

Di minggu terakhir, kegiatan yang dilakukan adalah membuat *layout* untuk mengajukan klaim dan menampilkan daftar klaim yang diajukan beserta dengan detail pengajuannya. Seperti terlihat pada Gambar 3.25. detail pengajuan klaim akan menampilkan nama pasien, diagnosa, tanggal periksa, nomor rekening, nama pemilik rekening, dan jenis biaya yang diajukan. Jenis biaya yang bisa diajukan antara lain Dokter Umum, Dokter Spesialis, Obat-obatan, Rawat Inap, Laboratorium, dan Lain-lain.



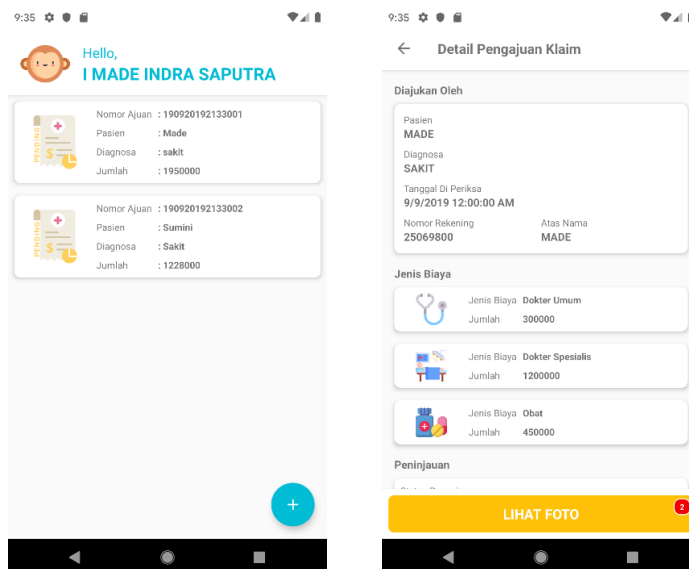
Gambar 3.26. Tampilan daftar klaim dan detail pengajuan klaim

Ketika karyawan ingin mengajukan permohonan klaim, karyawan dapat menekan tombol “+” pada bagian pojok kanan bawah aplikasi. Aplikasi akan mengarahkan langsung ke halaman untuk pengajuan klaim. Halaman ini dapat dilihat pada Gambar 3.26.. Karyawan diwajibkan untuk mengisi data-data yang diperlukan dan foto bukti untuk melakukan proses klaim. Foto bukti yang di-*upload* merupakan kuitansi asli dari dokter dan kuitansi asli untuk pembelian obat dengan mencantumkan harga obat secara terperinci.



Gambar 3.27. Tampilan permohonan klaim

Karyawan juga dapat melihat daftar klaim yang sudah diajukan beserta dengan detail pengajuannya jika menekan salah satu ajuan klaim. Proses ini dapat dilihat pada Gambar 3.27..



Gambar 3.28. Tampilan daftar klaim dan detail pengajuan klaim

3.4. Kendala yang Dihadapi

Adapun kendala yang dihadapi selama menjalankan program kerja magang di PT. Intersoft, sebagai berikut:

1. Pengembangan aplikasi Android yang menggunakan arsitektur MVP (*Model-View-Presenter*) sehingga ada kesulitan dalam pengerjaan di awal karena belum pernah menggunakan arsitektur MVP sebelumnya.
2. Terdapat beberapa istilah baru yang belum dimengerti, sehingga saat mengerjakan tugas masih membutuhkan waktu lebih untuk memahami istilah tersebut.

3.5. Solusi atas Kendala

Berikut solusi yang dapat dilakukan dari kendala yang dihadapi selama menjalankan program kerja magang:

1. Dijelaskan oleh *supervisor* tentang arsitektur MVP di awal pengerjaan dan dilanjutkan dengan melakukan belajar mandiri.
2. Mencari tahu sendiri terlebih dahulu dan bertanya pada *supervisor* apabila dirasa masih kurang mengerti.