



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kerja praktek magang dilakukan pada Divisi Mobile pada CV Surya Tirta Kencana Putra sebagai Junior Android Programmer dalam pengawasan Bapak Adam Kartanegara selaku Kepala Divisi Mobile. Beliau memberikan informasi dan *user requirement* secara garis besar, serta menjabarkan kendala – kendala yang ada pada proses jasa antar jenazah yang sudah ada sebelumnya. Dari informasi dan *user requirement* yang didapat kemudian dikembangkan sebuah aplikasi berbasis *mobile* untuk mempermudah sopir pada jasa antar jenazah untuk mencatat *order* yang sudah di lakukannya.

3.2 Tugas yang Dilakukan

Selama Periode magang selama 4 bulan, tugas yang dilakukan adalah membangun modul sistem pencatatan (Order Tracking) untuk dipakai oleh admin perusahaan CV Surya Tirta Kencana Putra guna men track orderan pada jasa antar jenazah pada CV Surya Tirta Kencana Putra dengan menggunakan bahasa JAVA dengan menggunakan aplikasi Android Studio. Modul yang akan dikerjakan adalah *login*, *searching*, *sorting*, dan *order management*. Model *iterative* digunakan sebagai *system development life cycle* dalam merancang dan membangun modul admin jasa antar jenazah ini. Model *iterative* digunakan karena lebih fleksibel dalam mengimplementasikan jika adanya perubahan yang

terjadi dalam proses pengembangan aplikasi, dibandingkan dengan model lainnya seperti model waterfall yang harus mengulang dari tahap awal ketika terjadi suatu perubahan dalam pertengahan proses pengembangan sistem. Selain itu model *iterative* digunakan karena *developer* belum mendapatkan / menemukan requirement keseluruhan yang diminta oleh client sehingga developer mengembangkan sistem sampai checkpoint tertentu dan selanjutnya akan dipresentasikan kepada *client* untuk mendapatkan *feedback* dan revisi dari *client* sehingga requirements semakin jelas dan dapat dilanjutkan ke *checkpoint* pengembangan sistem selanjutnya. Model *iterative* akan mengulang tahap dari analisis kebutuhan sistem sampai presentasi dan *feedback* dari *client* hingga sistem yang sudah dikira baik oleh *client* terbentuk. Tabel 2.1 merupakan tabel dari jadwal kegiatan kerja magang yang dilaksanakan sesuai dengan proses model *iterative*.

Tabel 2.1 Waktu Kerja magang

No	Kegiatan	juli			Agustus				September				Oktober		
		2	3	4	1	2	3	4	1	2	3	4	1	2	3
1	Penelitian terdahulu	■	■	■	■	■	■	■	■	■	■	■	■		
2	Analisis kebutuhan	■	■			■	■			■	■				
3	pembuatan sistem		■	■			■	■			■	■			
4	Pengujian sistem			■	■			■	■			■	■		
5	Presentasi project				■				■				■		

3.2.1 Penelitian terdahulu

Dalam proses pembuatan modul antar jenazah pada CV Surya Tirta Kencana Putra dibutuhkan *brainstorming* untuk menemui fitur-fitur yang diinginkan dalam aplikasi *mobile* ini. Setelah mendapatkan dan mempelajari kebutuhan *client*, maka *developer* memutuskan untuk memakai IDE Android Studio dan menggunakan Bahasa JAVA untuk membuat aplikasi *mobile* ini. Android Studio merupakan IDE yang sangat *update* dan jika adanya terjadi *update* maka *developer* harus menggali informasi tentang *update* tersebut karena bisa saja fitur yang lama dan biasa *developer* digantikan dengan fitur baru. Beberapa pelajaran penting yang berguna adalah mata kuliah seperti Pemrograman Aplikasi Mobile pada semester 6 untuk mempelajari ulang mengenai JAVA dan sisanya yang tidak diajarkan *developer* mempelajarinya dari *tutorial online* yang disediakan pada *website* youtube atau penyedia *framework* tersebut.

3.2.2 Analisi Kebutuhan

Pada tahapan berikutnya adalah analisis *kebutuhan* dari sistem aplikasi *mobile* tersebut. Tahap ini dimulai dengan analisis mengenai kebutuhan-kebutuhan yang ada untuk diterapkan ke aplikasi *mobile* yang akan dibangun. Kebutuhan aplikasi yang utama adalah *order management*.

Order Management berguna untuk melihat dan menghapus data dari tabel-tabel yang telah dibuat dalam *database*. *User* dengan otoritas *admin* saja yang dapat mengakses modul *order management*. Hal ini diperlukan untuk melakukan membantu proses pembukuan pada sistem sebelumnya yang hanya dicatat oleh sopir dengan kertas dan pulpen saja dan jika kertas tersebut itu hilang sudah tidak

ada *backup* lagi dan akan mempersulit proses pembukuan. *User* admin juga bisa melakukan penghapusan tabel pada *database* jika penghapusan perlu dilakukan.

Login berguna untuk mencegah terjadinya kebocoran/pencurian data oleh orang-orang yang tidak bertanggung jawab, karena data perusahaan terdapat beberapa *privacy* yang tidak boleh disebar luaskan. *Login* juga mencegah terjadinya perusakan terhadap *database*, karena bisa saja ada orang yang masuk dan tidak sengaja menghapus semua isi *dababase*.

Searching berguna untuk melakukan pencarian sejumlah data pada sistem. *Usser* dengan otoritas admin mampu melakukan pencarian pada semua tabel yang ada dalam *database*, sehingga *admin* dapat dengan mudah mencari *data* yang ingin dilihatnya. *Sorting* berguna untuk melakukan *sorting* data berdasarkan bulan sehingga *admin* dapat dengan mudah mencari *data* yang ingin dilihatnya.

3.2.3 Pembuatan Sistem

Dalam pembuatan sistem, diperlukan beberapa hal, yaitu *software* dan *hardware* untuk menunjang kerja magang ini. Perangkat lunak atau *software* yang digunakan adalah sebagai berikut.

1. Sistem operasi Windows 10 64 bit
2. Browser ini digunakan untuk mencari informasi yang berkaitan dengan requirement yang membantu pembuatan dari aplikasi ini.
3. Android Studio digunakan sebagai IDE untuk menulis script program.
4. NOX digunakan sebagai emulator untuk mengetest aplikasi agar responsive di berbagai ukuran smartphone

Adapun beberapa perangkat keras yang dibutuhkan untuk membantu pembuatan sistem adalah sebagai berikut.

1. Laptop MSI GL62M 7REX, dengan spesifikasi sebagai berikut.
 - a. Processor Intel i7-770HQ 2.80GHz
 - b. VGA Nvidia GeForce 1050 TI 4GB
 - c. Hard disk dengan kapasitas total 1TB
 - d. RAM 8 GB
2. Smartphone Samsung Galaxy S7 Edge

3.2.4 Presentasi Project

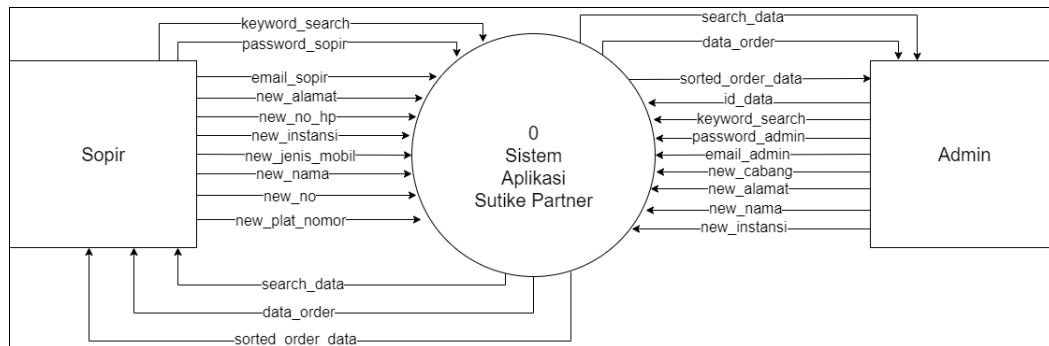
Presentasi hasil kerja dilakukan setiap 1 bulan atau saat sekiranya aplikasi sudah dalam tahap testing dan siap dilakukan revisi. Presentasi dilakukan kepada HRD Manager yaitu sebagai client dan memberikan feedback yang akan digunakan untuk memenuhi requirements sebagai pengembangan sistem selanjutnya.

3.3 Uraian Pelaksanaan Kerja Magang

3.3.1 Rancangan Aplikasi

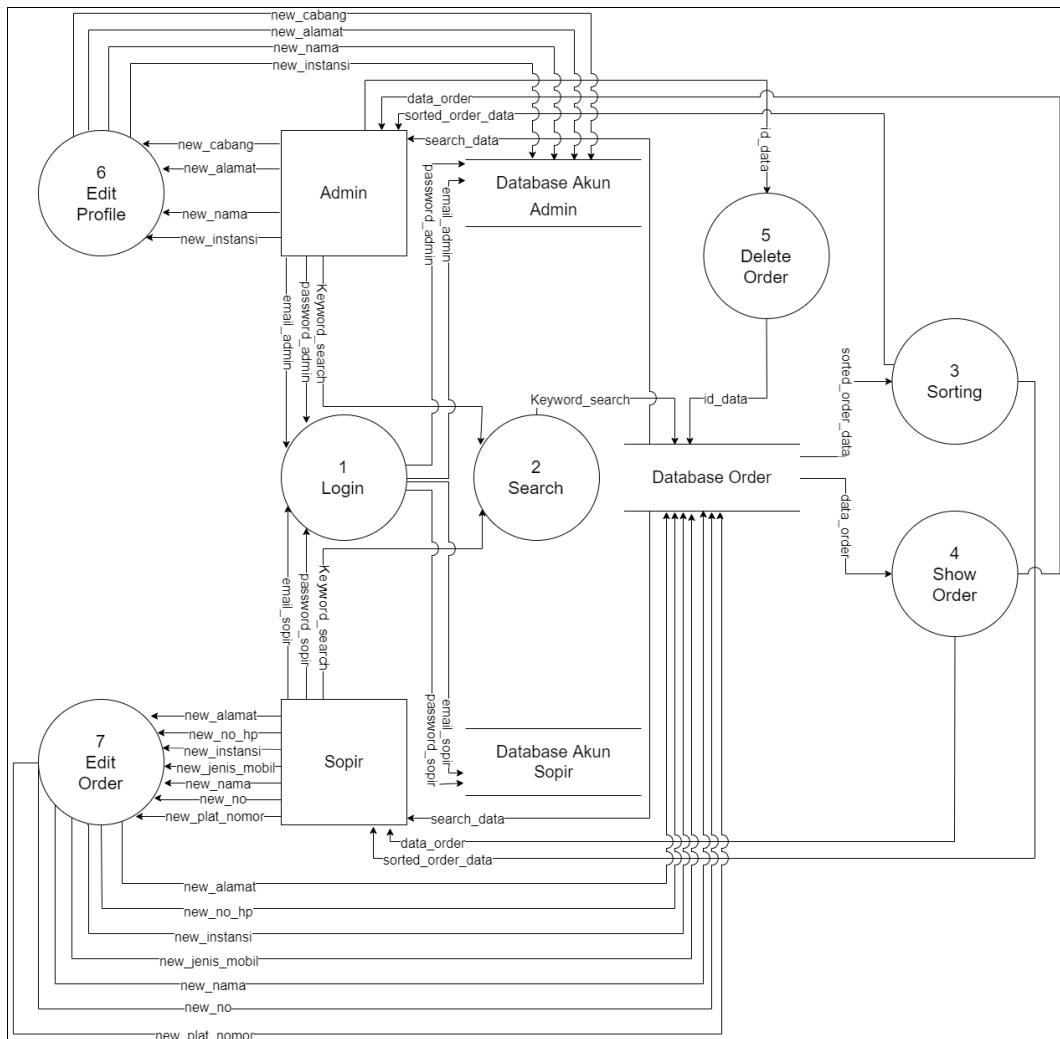
Aplikasi yang baik harus dikerjakan sesuai dengan perencanaan dan perancangan yang sudah dilakukan pada proses user requirement. Perancangan suatu aplikasi dapat ditunjukkan dengan menggunakan diagram-diagram, dimana diagram yang digunakan dalam aplikasi ini adalah DFD, flowchart dan *database schema*.

A. DFD (Data Flow Diagram)



Gambar 3.1 *Context Diagram* sistem aplikasi sutike partner

Context diagram pada gambar 3.1 memiliki 2 entitas yaitu admin dan Sopir. Entitas admin mengirim data kepada sistem berupa id_data, keyboard_search, password_admin, email_admin, new_cabang, new_alamat dan new_nama, new_instansi. Entitas admin juga menerima data dari sistem berupa search_data, data_order dan sorted_data_order. Sedangkan entitas sopir mengirim data kepada sistem berupa keyword_search, password_sopir, _email_sopir, new_alamat, new_no_hp, new_instansi, new_jenis_mobil, new_nama, new_no dan new_plat_nomor. Entitas sopir juga menerima data dari sistem berupa search_data, data_order dan sorted_order_data.



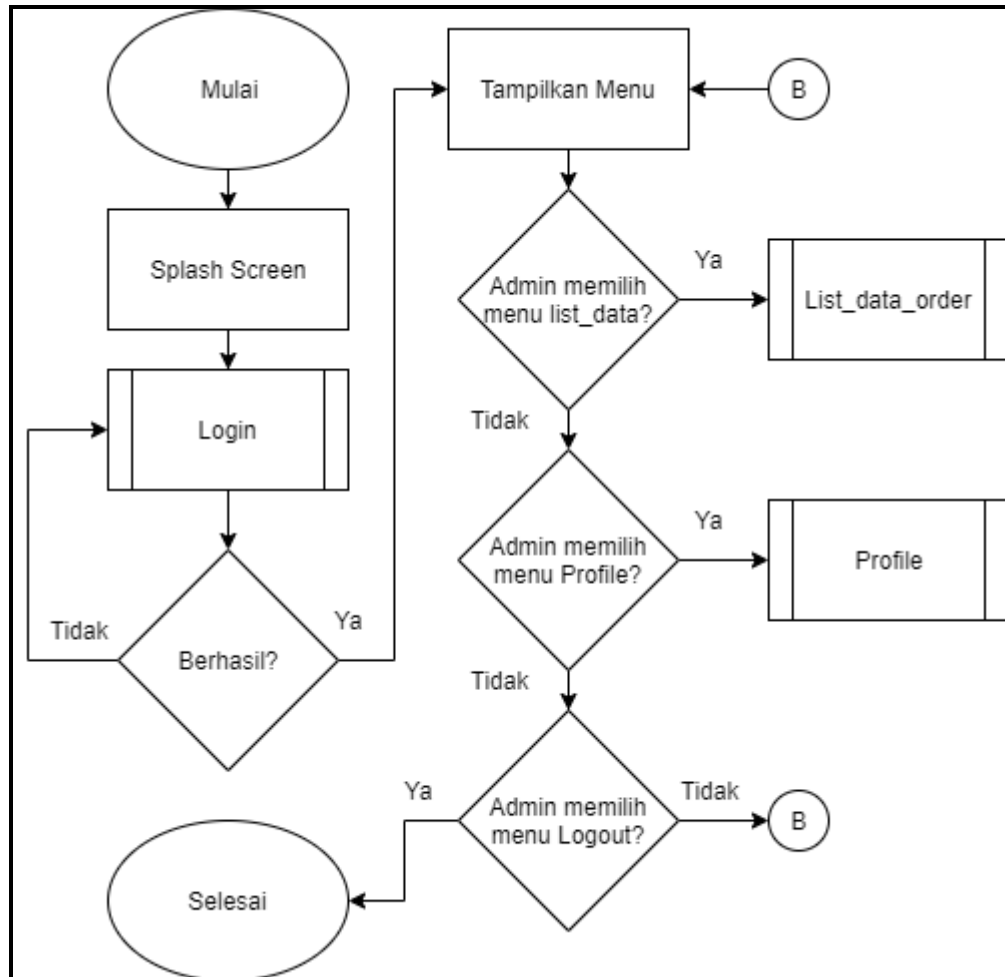
Gambar 3.2 DFD level 1 sistem aplikasi sutike partner

Pada gambar 3.2 terdapat tujuh proses yang mengirim dan menerima data. Proses login menerima data email_admin dan password_admin dari entitas admin, menerima data email_sopir dan password_sopir dari entitas sopir, proses ini juga mengirimkan email_admin dan password_admin ke dalam database akun admin dan mengirimkan data email_sopir dan password_sopir kedalam database akun sopir.

Proses search menerima data keyword_search dari entitas admin dan sopir, dan meneruskan data keyword_search ke database order. Proses sorting menerima

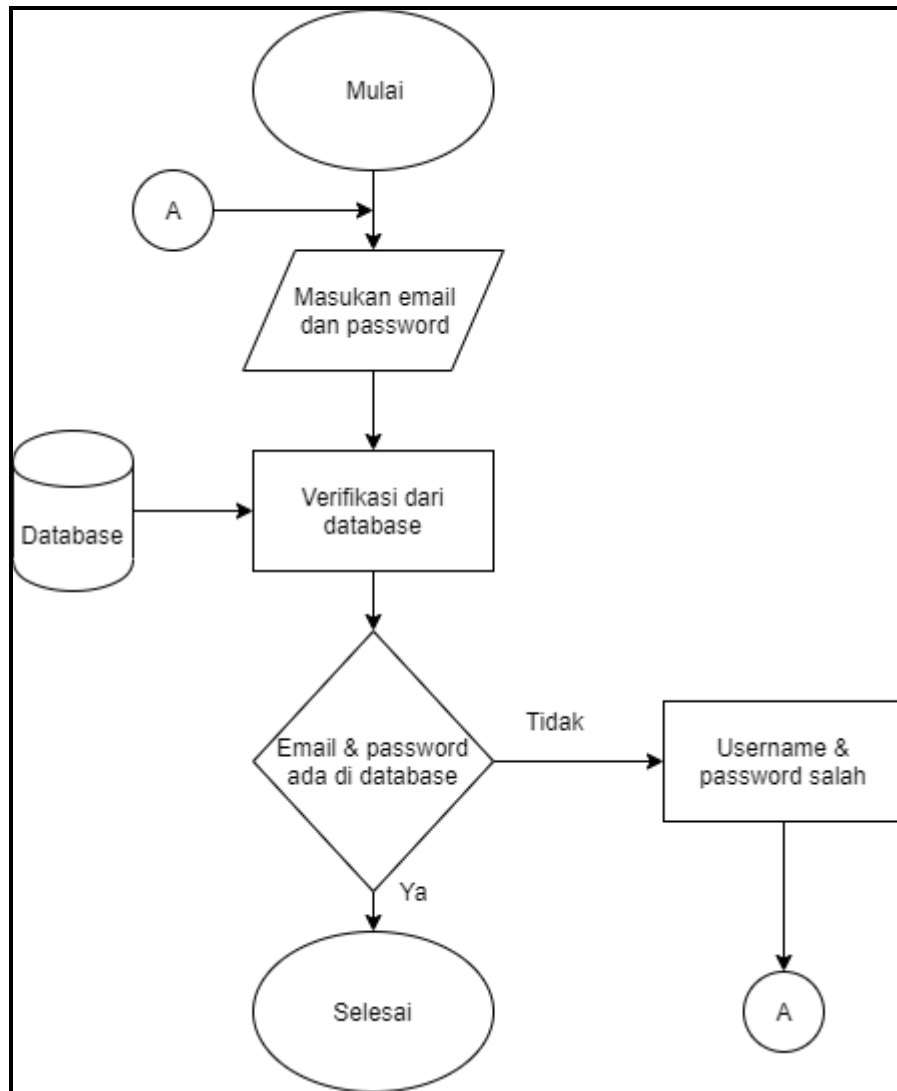
data sorted_order_data dari database order dan mengirimkan sorted_order_data ke entitas admin dan entitas sopir. Proses show order menerima data data_order dari database order dan meneruskan data data_order ke dalam entitas admin dan entitas sopir. Proses delete order menerima data id_data dari entitas admin dan meneruskan id_data ke dalam database order. Proses edit profile menerima data new_cabang, new_alamat, new_nama dan new_instansi dari entitas admin dan meneruskan data new_cabang, new_alamat, new_nama, new_instansi ke database akun admin. Proses edit order menerima data new_alamat, new_no_hp, new_nama, new_instansi, new_plat_nomor, new_jenis_mobil dan new_no dari entitas sopir dan meneruskan data new_alamat, new_no_hp, new_nama, new_instansi, new_plat_nomor, new_jenis_mobil dan new_no ke dalam database order.

B. Flowchart diagram



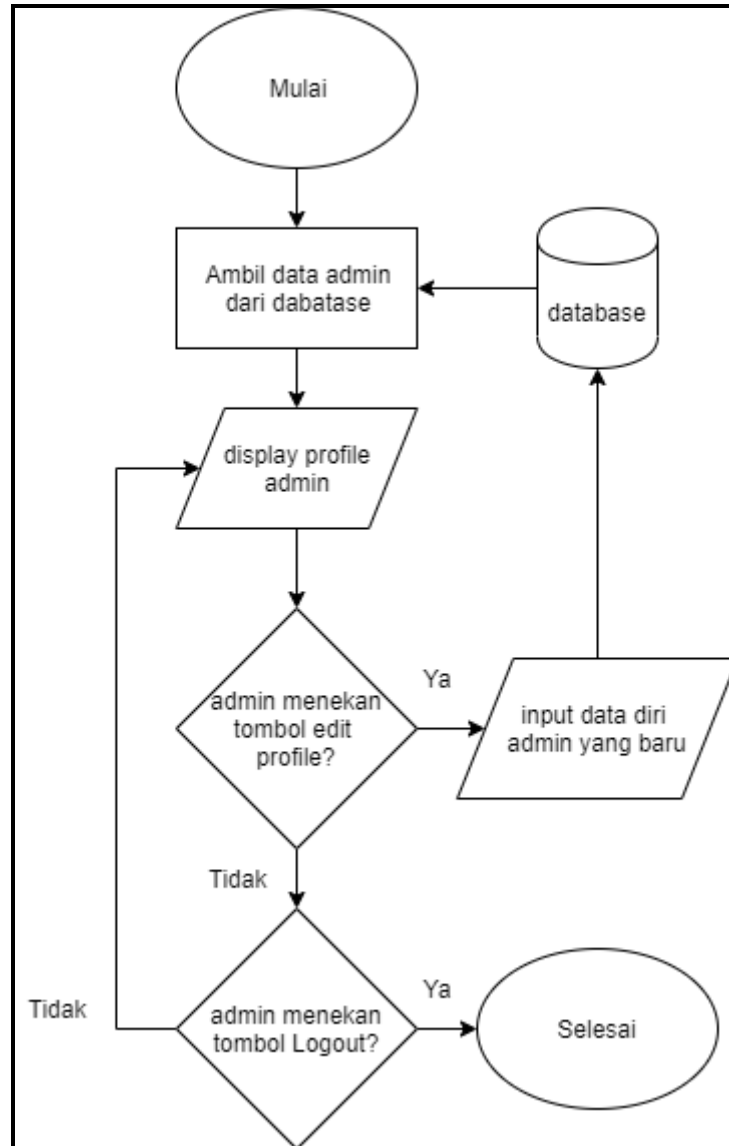
Gambar 3.3 Flowchart Main Menu pada modul admin

Dari keseluruhan sistem terbentuk lima subproses. Lima subproses yang dikerjakan, yaitu MainMenu, login, list_data_order, profile, dan display_details_order.



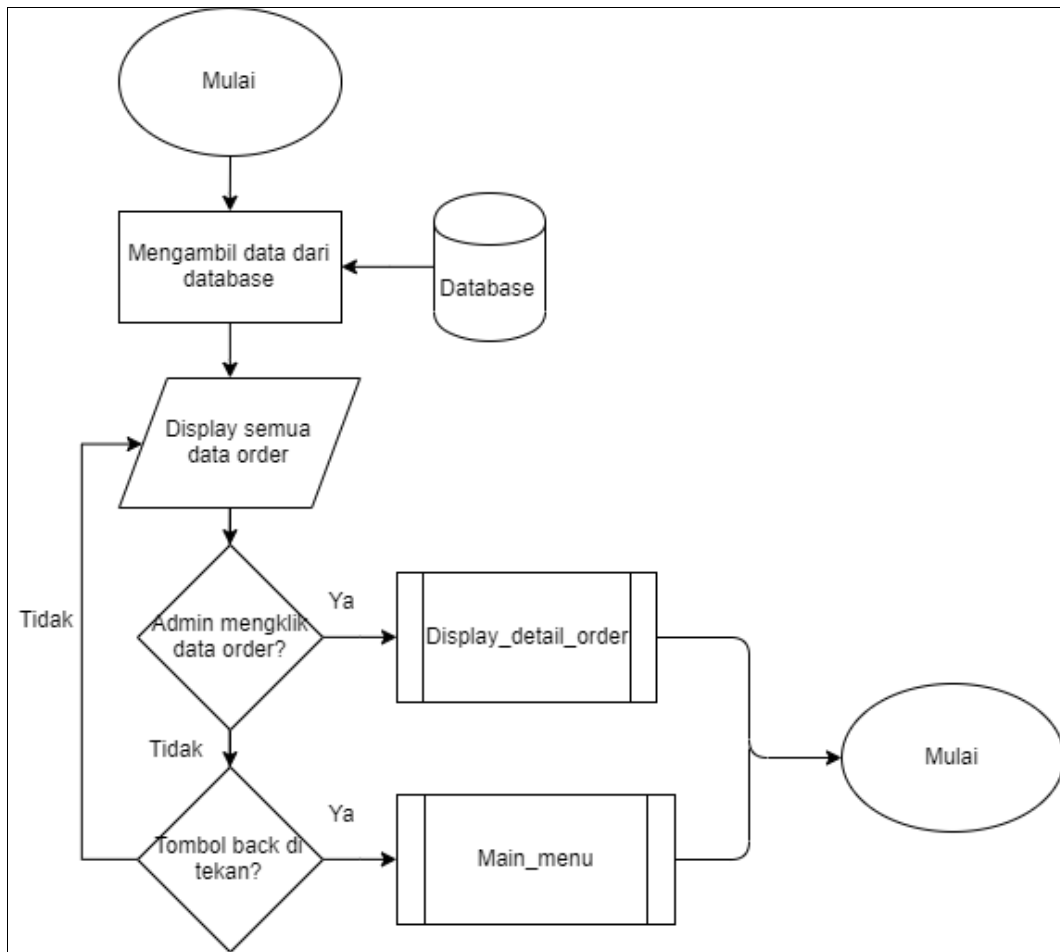
Gambar 3.4 Flowchart Subproses Login

Proses login dilakukan dengan SDK Firebase yang sudah built in dari android studio. Dengan memakai fungsi authentication dari firebase developer tinggal mendaftarkan email dan password admin kedalam database yang sudah disediakan firebase dan tinggal menarik data tersebut dan mencocokkannya jika ingin melakukan login.



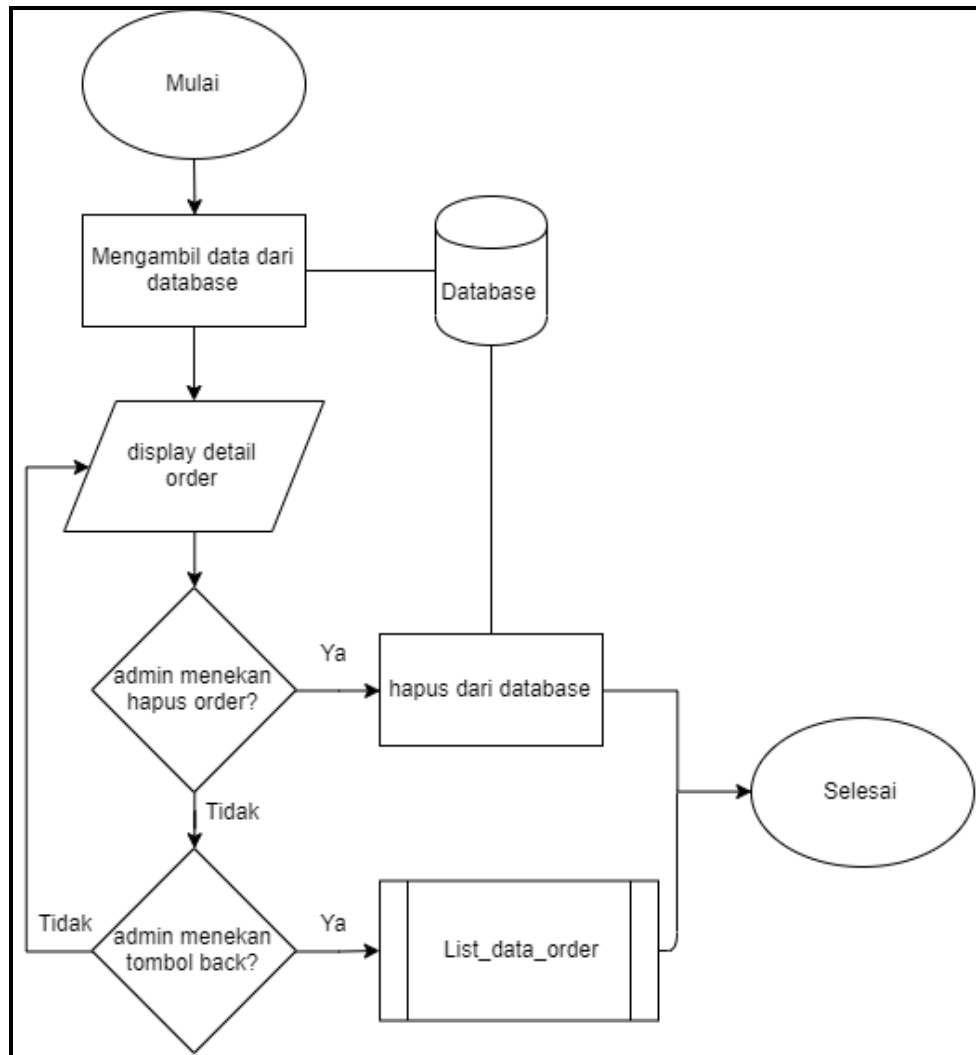
Gambar 3.5 Flowchart subproses profile

Subproses profile digunakan untuk admin melihat profile admin dan juga untuk melakukan logout. Ada tombol ubah profile pada subproses admin yang berguna untuk mengganti data diri secara dinamis jika ada terjadi kesalahan input pada data diri admin.



Gambar 3.6 Flowchart subproses list_data_order

Pada mainmenu jika admin menekan tombol listdata maka akan mempopulate data yang ditarik dari database realtime firebase, dan akan menampilkan card card yang berisi nama , tanggal order dan status order.



Gambar 3.7 Flowchart subprocess display_detail_order

Jika admin ingin mendapatkan data yang lebih detail lagi admin bisa menekan card pada menu list data untuk melihat data yang lebih spesifik lagi. Pada bagian details admin juga dapat melakukan penghapusan data jika diperlukan, dan sistem akan melakukan penghapusan secara dinamik kedalam database.

C. Struktur tabel

1. Tabel order

Table 3.1 Tabel Order

Nama kolom	Tipe Data	Keterangan
id	String	ID sopir
alamat	String	Alamat sopir mengantar atau menjemput almarhum
hp	String	Nomor telepon sopir
instansi	String	Nama instansi sopir
jenismobil	String	Jenis mobil yang dikendarai sopir
nama	String	Nama Almarhum yang diantar oleh sopir
no	String	Nomor sopir
platnomor	String	Plat nomor kendaraan sopir

Tabel order dijelaskan pada Tabel 3.1 memiliki delapan kolom, yaitu kolom id untuk membedakan tiap order, sisanya adalah kolom yang berisikan tentang deskripsi dari order, pelayanan dan identitas dari sopir yang melakukan order tersebut.

2. Tabel profile admin

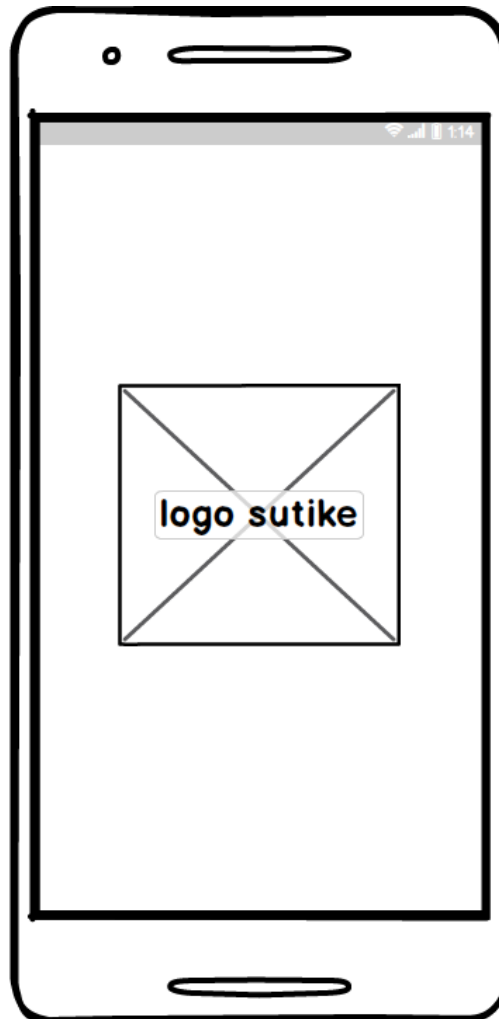
Table 3.2 Tabel profile admin

Nama kolom	Tipe Data	Keterangan
id	String	Id admin
alamat	String	Alamat admin
instansi	String	Instansi admin
nama	String	Nama admin
cabang	String	Lokasi cabang admin

Tabel profile admin dijelaskan pada Tabel 3.2 memiliki lima kolom, yaitu kolom id untuk membedakan tiap admin yang terdaftar pada sistem, sisanya adalah kolom yang berisikan tentang deskripsi profile admin tersebut.

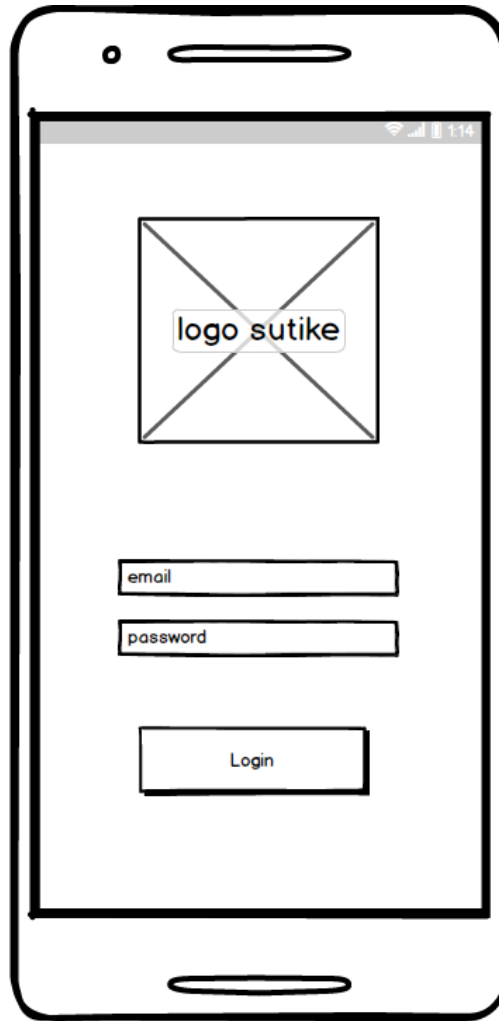
D. Desain *Mockup*

Dari rancang bangun DFD dapat dibentuk model mockup design yang dibuat dengan aplikasi balsamiq mockup 3. Hasil dari mockup tidak sama seutuhnya dengan hasil akhir aplikasi, dikarenakan adanya beberapa revisi dari supervisor untuk menyesuaikan dengan kebutuhan akhir dari modul admin pada aplikasi sutikepartner. Modul admin aplikasi sutikepartner memiliki user interface yang terdiri dari splash screen, login, main menu, daftar transaksi, details, profile dan edit profile.



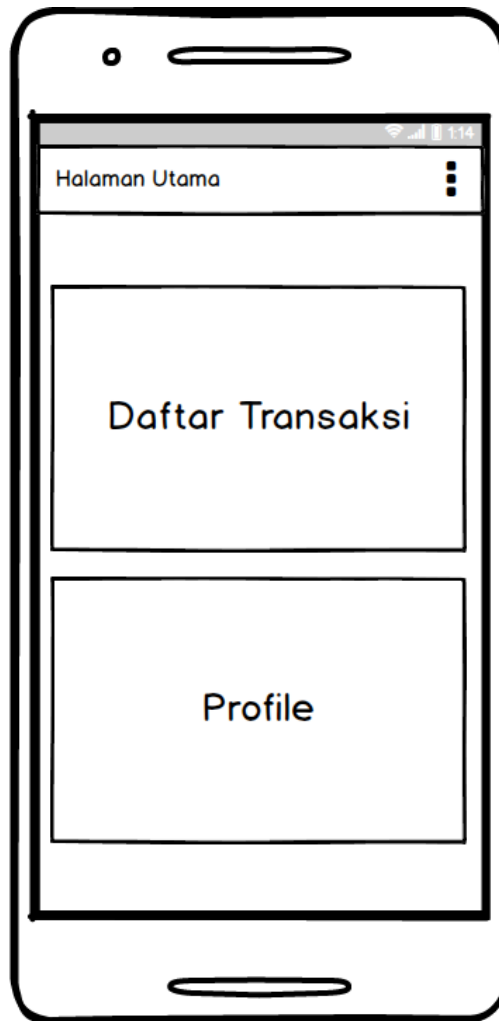
Gambar 3.8 *Mockup Splash Screen*

Gambar 3.8 adalah desain antarmuka yang akan ditampilkan saat aplikasi dibuka, aplikasi akan menampilkan logo dari perusahaan surya tirta kencana putra selama beberapa detik dan akan menampilkan menu selanjutnya jika sistem sudah siap untuk digunakan.



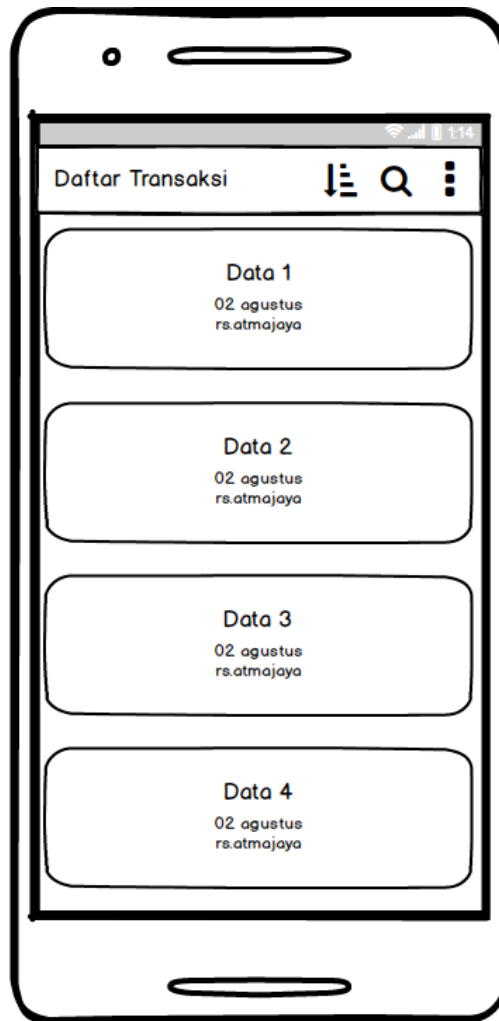
Gambar 3.9 *Mockup Login*

Pada Gambar 3.9 menunjukkan aplikasi akan menampilkan desain antarmuka yang akan dikeluarkan secara otomatis oleh sistem sehabis splash screen selesai ditampilkan. Tampilan login memiliki logo sutike yang terletak di tengah atas, kolom untuk mengisi email, kolom untuk mengisi password dan satu tombol untuk melakukan login.



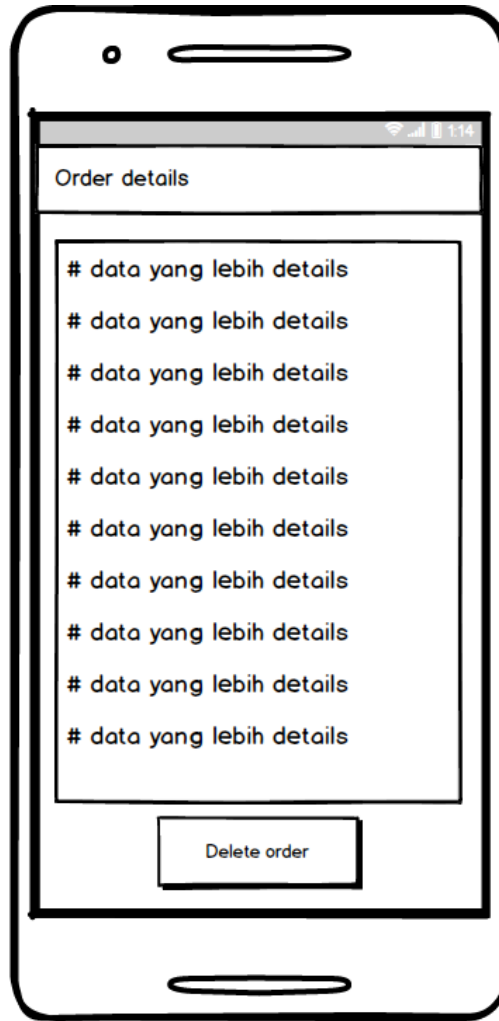
Gambar 3.10 *Mockup Main Menu*

Pada Gambar 3.10 menunjukkan aplikasi akan menampilkan dua tombol yaitu tombol daftar transaksi yang berfungsi untuk berpindah ke halaman antarmuka daftar transaksi dan tombol profile yang digunakan untuk berpindah ke halaman antarmuka profile dari admin yang sedang login. Pada header aplikasi terdapat tombol action bar yang dapat digunakan untuk melakukan logout.



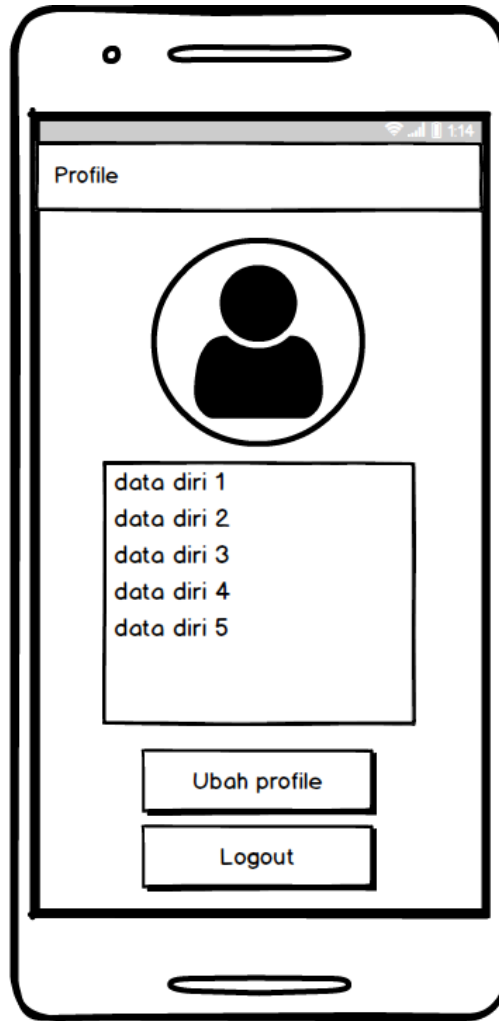
Gambar 3.11 *Mockup* Daftar Transaksi

Gambar 3.11 menunjukkan aplikasi akan menampilkan card view dan recycle view yang berisi data order. Jika pengguna mengklik salah satu card pada halaman daftar transaksi maka akan membuka halaman detail. Pada header android terdapat icon sort yang digunakan pengguna untuk melakukan sorting, icon search yang dapat digunakan pengguna untuk melakukan searching dan juga terdapat tombol action bar yang dapat digunakan untuk melakukan logout.



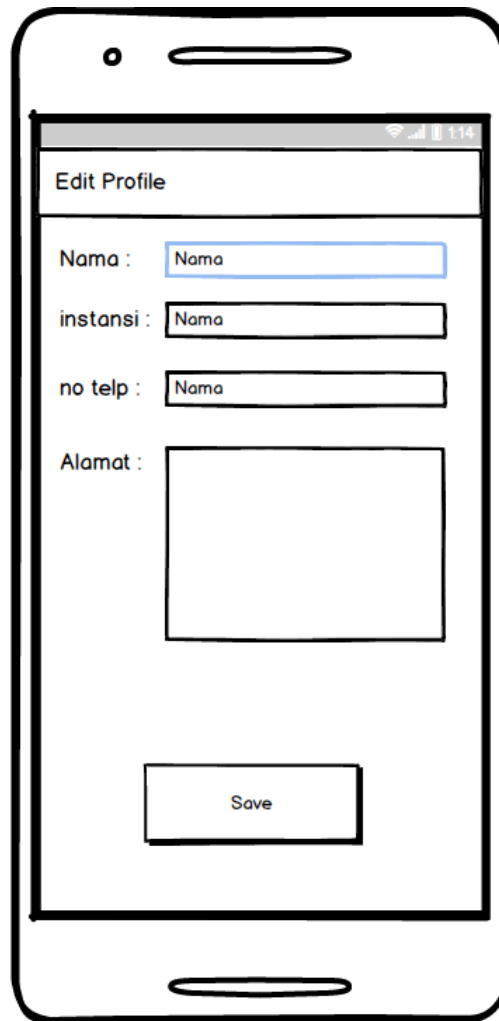
Gambar 3.12 *Mockup Detail*

Pada Gambar 3.12 akan ditampilkan list data yang lebih lengkap dari data pada daftar transaksi. Pada halaman details juga ada satu button yang terletak pada bawah tengah layer yang dapat digunakan untuk melakukan penghapusan data.



Gambar 3.13 *Mockup Profile*

Pada Gambar 3.13 akan ada beberapa data yang akan ditampilkan pada antarmuka aplikasi yaitu profile picture dari admin, data diri admin dan ada dua tombol pada bagian bawah untuk melakukan ubah profile dan logout.



Gambar 3.14 *Mockup Edit Profile*

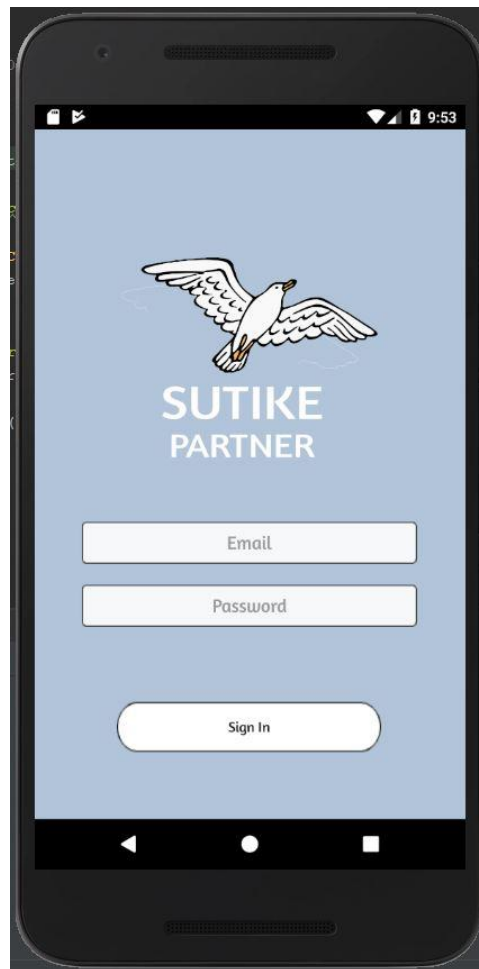
Pada Gambar 3.14 terdapat empat form yaitu nama, instansi, no telp, alamat dan 1 button pada desain antarmuka edit profile. Untuk melakukan perubahan data diri admin harus mengisi ke-empat form tersebut dan menekan tombol save pada bagian tengah bawah layar aplikasi.

E. Implementasi



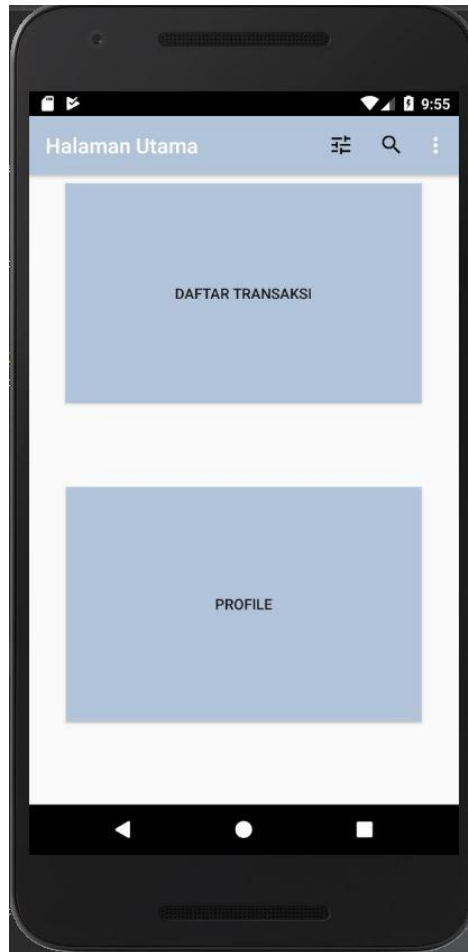
Gambar 3.15 Tampilan *Splash Screen*

Gambar 3.15 menunjukkan aplikasi saat pertama kali dijalankan. Setelah beberapa saat maka aplikasi akan menampilkan halaman login page.



Gambar 3.16 Tampilan *Login*

Gambar 3.16 menunjukkan login page aplikasi sutike partner. User akan diminta untuk menginput email dan password untuk melakukan login. Developer menggunakan salah satu fungsi dari firebase untuk melakukan proses autentikasi.



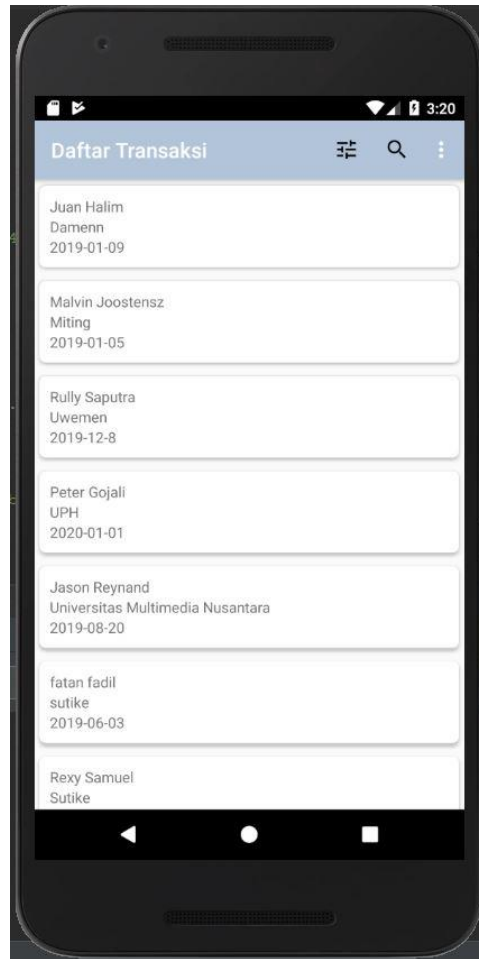
Gambar 3.17 Tampilan *Main Menu*

Saat berhasil login, maka akan muncul tampilan seperti Gambar 3.17. Halaman ini digunakan sebagai main menu untuk user bernavigasi ke halaman selanjutnya. Pada halaman ini terdapat dua tombol yang berfungsi untuk berpindah ke halaman daftar transaksi dan halaman profile admin.



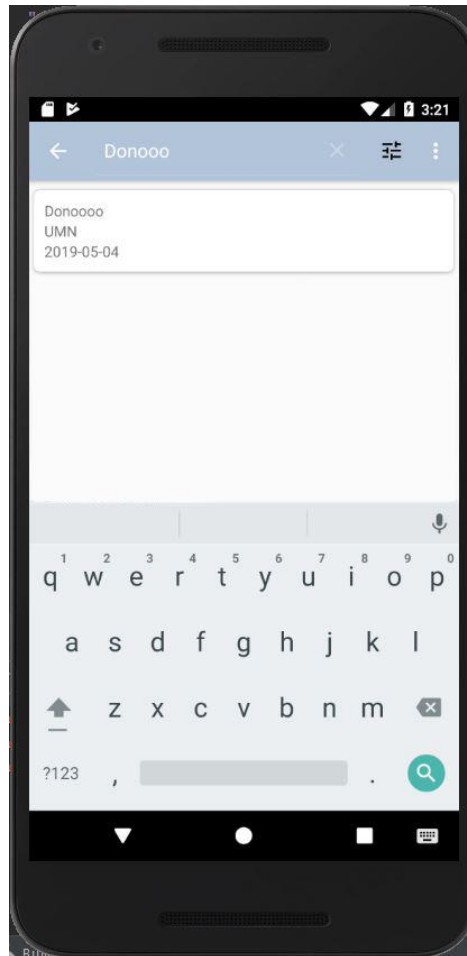
Gambar 3.18 Tampilan *Action Bar*

Pada Gambar 3.18 saat user menekan action bar yang terdapat pada atas kanan tampilan maka akan menampilkan isi dari action bar. Action bar memiliki satu opsi yaitu logout yang digunakan untuk berpindah ke login page.



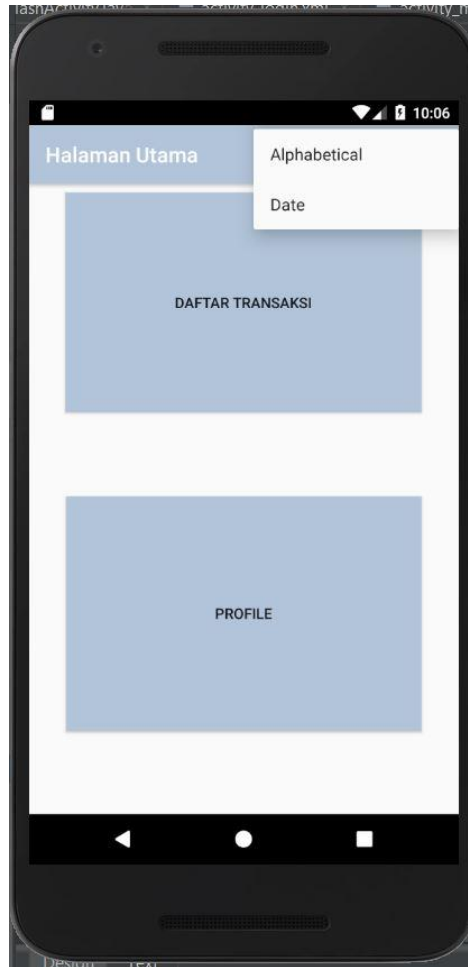
Gambar 3.19 Tampilan Daftar Transaksi

Jika tombol daftar transaksi pada main menu ditekan maka sistem akan menampilkan tampilan seperti pada Gambar 3.19 yang berisi list – list order yang sudah pernah dilakukan oleh sopir.



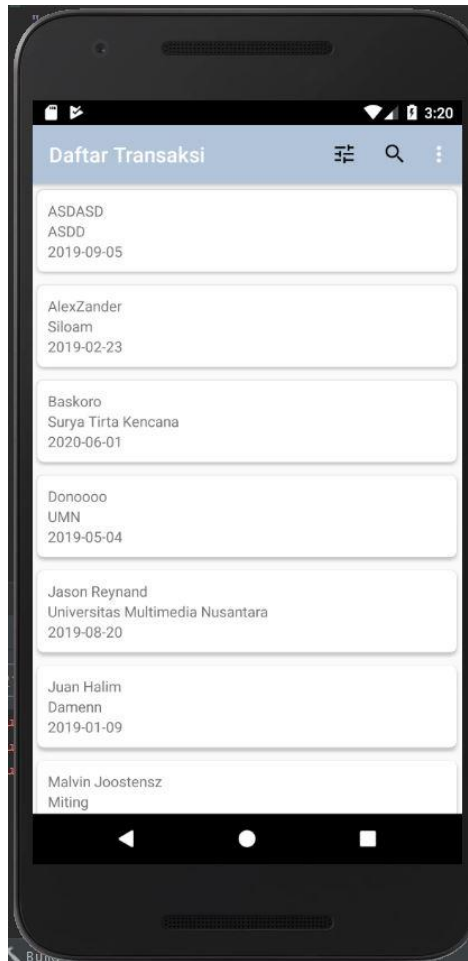
Gambar 3.20 Tampilan Search Daftar Transaksi

Jika icon kaca pembesar pada header aplikasi ditekan maka akan menampilkan text field dan keyboard untuk menginput data yang ingin dicari seperti pada Gambar 3.20.



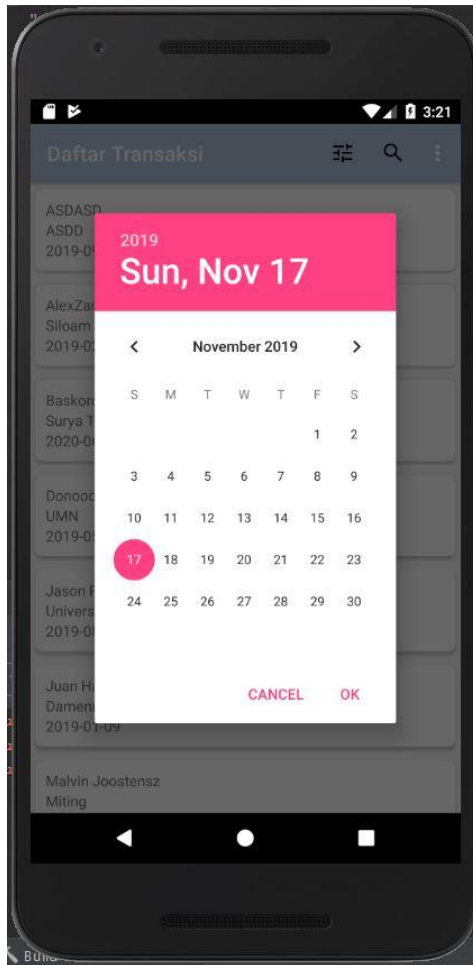
Gambar 3.21 Tampilan *Sort* Daftar Transaksi

Jika icon sort pada header aplikasi di tekan maka akan ditampilkan tampilan seperti Gambar 3.21. Fitur sort mempunyai dua buah opsi yaitu Alphabetical yang berguna untuk mensort data sesuai urutan alphabet dan Date yang berguna untuk mensort data sesuai dengan tanggal yang di input.



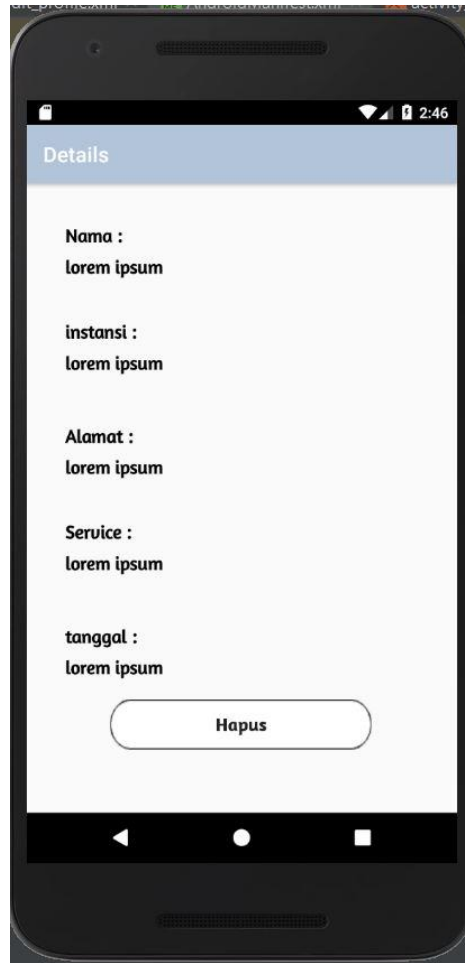
Gambar 3.22 Tampilan *Sort Alphabet* Daftar Transaksi

Gambar 3.22 menunjukkan data yang sudah di sort berdasarkan alphabetical.



Gambar 3.23 Tampilan *Sort Date* Daftar Transaksi

Pada Gambar 3.23 adalah tampilan dari calendar picker yang muncul jika user ingin melakukan sorting berdasarkan tanggal.



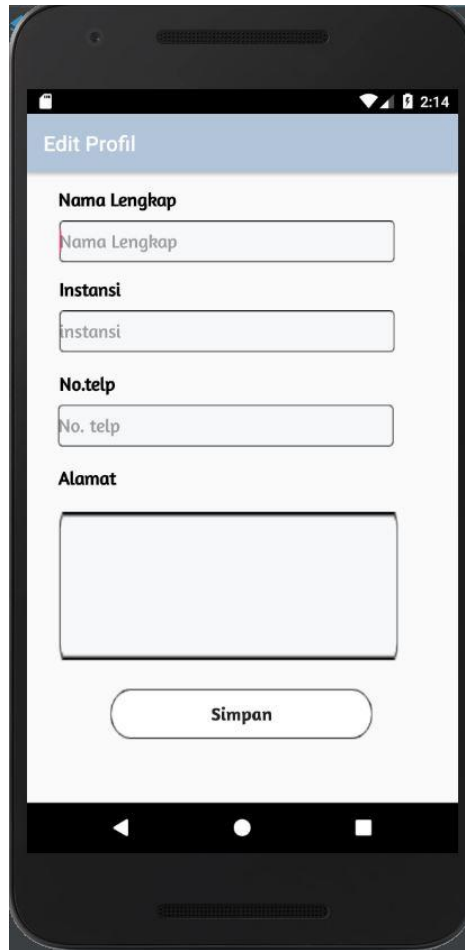
Gambar 3.24 Tampilan *Detail*

Pada saat card pada page daftar transaksi di tekan maka akan tampilan seperti pada Gambar 3.22 yang berisi data yang lebih lengkap dari data yang ditampilkan di page daftar transaksi. Pada page details user juga dapat melakukan penghapusan order dengan menekan button hapus pada bawah layar.



Gambar 3.25 Tampilan *Profile*

Pada saat tombol profile pada page main menu ditekan maka akan keluar tampilan seperti Gambar 3.22. Pada page profile terdapat data diri admin berupa photo profile, cabang admi ntersebut di tempatkan, instansi, nomor telpon, alamat, serta terdapat satu button yang dapat digunakan untuk mengganti data diri admin tersebut.



Gambar 3.26 Tampilan Edit Profile

Pada saat tombol ubah profile pada page profile ditekan maka akan keluar tampilan seperti Gambar 3.23. Untuk mengubah data diri User diwajibkan untuk mengisi ke-empat text field form dan menekan tombol simpan.

3.3.2 Pengujian Sistem

Tahap pengujian sistem merupakan tahap dimana melihat apakah sistem yang telah dibangun berjalan dengan baik dan sesuai dengan requirements dan keinginan yang diharapkan client. Pengujian dilakukan dengan cara menjalankan sistem di beberapa ukuran *smartphone* yang berbeda dengan menggunakan emulator NOX. Pengujian menggunakan data dummy untuk memastikan apakah aliran data berjalan dengan baik atau tidak. Pengujian sistem juga dilakukan untuk mencari celah dari sistem atau bug-bug yang dapat terjadi pada sistem dan segera memperbaikinya.

Beberapa bug dan cara memperbaiki bug yang ditemukan dijelaskan pada poin – poin dibawah ini:

1. Fitur sorting yang salah mensort data order yang berawalan huruf kapital dan tidak huruf kapital, solusi yang dilakukan untuk memperbaiki bug ini adalah menggunakan fitur `toLowerCase()`; sebelum mengquery data sorting.
2. Button ubah profile pada tampilan profile admin yang kadang terpotong setengah atau hilang karena resolusi layar *smartphone*, solusi yang dilakukan untuk memperbaiki bug ini adalah mengganti layout dari linear layout ke constrain layout dan menyambungkan constraint dari bawah layar ke bawah button agar dapat mengikuti layar *smartphone* berbagai ukuran.
3. Fitur show order yang tiba – tiba error saat dibuka, solusi yang dilakukan untuk memperbaiki bug ini adalah pertama-tama melihat dahulu error apa yang ditampilkan pada logcat dan ternyata error terjadi karena *constructor* untuk menarik data dari *database* mengambil data dengan format yang salah. Kemudian dilakukan pengecekan data di *database* dan ternyata ada data yang

memiliki format salah diinput oleh divisi yang mengerjakan modul sopir di saat sedang melakukan pengetesan modul sopir.

4. Fitur *sorting by date* yang salah mengquery data tanggal pada *database*, solusi yang dilakukan untuk memperbaiki bug ini adalah mendefinisikan *simple date* format dengan *syntax* `dateFormatter = new SimpleDateFormat("yyyy-MM-dd", Locale.US)` agar format penulisan tanggal sama dengan format pada *database*.

Adapun rekomendasi dari perangkat platform untuk menjalankan aplikasi ini. Diharapkan menggunakan smartphone dengan spesifikasi minimal sebagai berikut.

1. Android 7.1 Nougat (API 25).
2. RAM sebesar 512 MB.

3.4 Kendala yang Ditemukan

Selama periode pengerjaan magang, ditemukan beberapa kendala, antara lain:

1. *Miss communication*

Pada saat penjelasan tentang *user requirement* untuk aplikasi, penjelasan yang di dapat terkadang berbeda dengan apa yang dipahami oleh *developer*. *Miss communication* yang cukup fatal terjadi saat minggu pertama di saat pembagian pekerjaan oleh supervisor, seharusnya *developer* didapatkan mengerjakan modul sopir tetapi kelompok sebelah yang seharusnya mengerjakan modul admin malah mengerjakan modul sopir juga sehingga menyebabkan *developer* harus berpindah ke *project* modul admin yang membuat pekerjaan cukup terhambat karena harus memulai dari awal lagi.

2. Laptop mengalami *freeze*

Pada saat melakukan pengujian sistem terkadang laptop mengalami *freeze* karena beratnya aplikasi android studio dan emulator yang digunakan untuk mengetes aplikasi tersebut.

3. Pemakaian *database firebase* yang berformat *nosql*

Developer diperharuskan menggunakan database firebase yang tidak memiliki tabel pada databasanya atau biasa disebut *nosql* kurang dipahami oleh developer.

4. Migrasi ke androidX

Disaat mengimplementasikan database firebase pada sistem, ternyata terjadi error dan harus melakukan migrasi support library dari Android AppCompat ke AndroidX yang cukup memakan waktu.

3.5 Solusi atas Kendala yang Ditemukan

Solusi atas beberapa kendala yang ditemukan selama pelaksanaan kerja magang adalah sebagai berikut.

1. *Miss Communication*

Untuk menghindari terjadinya *miss communication developer* mencatat setiap *user requirement* yang disebutkan dan bertanya hingga mendapat kejelasan atas keterangan yang diberikan kepada *developer*. Karena adanya kesalahpahaman di pembagian pekerjaan maka *developer* harus mengumpulkan sistem *requirement* dan membuat *mockup* untuk sistem admin lagi.

2. Laptop mengalami *freeze*

Untuk menghindari terjadinya *freeze* pada laptop yang sangat menghambat pekerjaan *developer*, *developer* menggunakan gadget untuk melakukan *debugging* aplikasi. Dan menutup *android studio* jika ingin *mendebug* menggunakan emulator.

3. Pemakaian *database firebase* yang berformat *nosql*

Untuk masalah pemakaian *database* berformat *no sql* *developer* membaca dokumentasi dokumentasi pada *website firebase* dan mencari tata struktur yang cocok untuk aplikasi *sutike partner* di internet.

4. Migrasi ke *androidX*

Untuk melakukan migrasi ke *androdX* *developer* mencari tutorial dari *stackoverflow* dan *website android.com*. Serta membackup *project* sebelum melakukan migrasi.