



KERJA PRAKTIK - IF184801

Implementasi Aplikasi Android Vehicle Control and Safety System (VCSS)

PT. Esurya Datapedia Semesta
East Coast Park Blok R5 No. 31, Kota Surabaya, Jawa Timur
Periode: 26 Januari 2023 – 26 April 2023

Oleh:

Daanii Nabil Ghinannafsi Kusananta 05111940000163

Pembimbing Jurusan

Agus Budi Raharjo, S.Kom., M.Kom., Ph.D.

Pembimbing Lapangan

Faturochman Pranacahya Andrianto

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2023



KERJA PRAKTIK - IF184801

Implementasi Aplikasi Android Vehicle Control and Safety System (VCSS)

PT. Esurya Datapedia Semesta
East Coast Park Blok R5 No. 31, Kota Surabaya, Jawa Timur
Periode: 26 Januari 2023 – 26 April 2023

Oleh:

Daanii Nabil Ghinannafsi Kusnanta 05111940000163

Pembimbing Jurusan

Agus Budi Raharjo, S.Kom., M.Kom., Ph.D.

Pembimbing Lapangan

Faturochman Pranacahya Andrianto

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2023

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI.....	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
DAFTAR KODE SUMBER	xii
LEMBAR PENGESAHAN KERJA PRAKTIK.....	xiv
KATA PENGANTAR.....	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Manfaat.....	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik	2
1.6. Metodologi Kerja Praktik.....	3
1.6.1. Perumusan Masalah.....	3
1.6.2. Studi Literatur.....	3
1.6.3. Analisis dan Perancangan Sistem	3
1.6.4. Implementasi Sistem	3
1.6.5. Pengujian dan Evaluasi.....	4
1.6.6. Kesimpulan dan Saran.....	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan.....	4
1.7.2. Bab II Profil Perusahaan.....	4
1.7.3. Bab III Tinjauan Pustaka	5
1.7.4. Bab IV Analisis dan Perancangan Sistem	5

1.7.5.	Bab V Implementasi Sistem	5
1.7.6.	Bab VI Pengujian dan Evaluasi	5
1.7.7.	Bab VII Kesimpulan dan Saran	6
BAB II PROFIL PERUSAHAAN		8
2.1.	Profil PT. Esurya Datapedia Semesta.....	8
2.2.	Lokasi Perusahaan	8
BAB III TINJAUAN PUSTAKA.....		10
3.1.	Pemrograman Aplikasi Android.....	10
3.2.	Android Studio	10
3.3.	Kotlin.....	10
3.4.	Firebase	11
3.5.	Google Maps Platform	11
3.6.	Git.....	12
BAB IV ANALISIS DAN PERANCANGAN SISTEM		14
4.1.	Analisis Kebutuhan Sistem.....	14
4.2.	Rancangan Antarmuka Sistem	14
BAB V IMPLEMENTASI SISTEM.....		19
5.1.	Implementasi Fungsi	19
5.1.1.	Mendapatkan Informasi Fasilitas Umum	19
5.1.2.	Implementasi Mengganti Foto Akun.....	24
5.1.3.	Implementasi Mengganti Nama Akun.....	27
5.1.4.	Mengganti Email Akun	28
5.1.5.	Melakukan Logout Akun.....	31
5.2.	Implementasi Antarmuka	31
5.2.1.	Implementasi Halaman Autentikasi.....	31

5.2.2.	Implementasi Halaman Dalam Navigasi	33
5.2.3.	Implementasi Halaman Navigasi.....	34
5.2.4.	Implementasi Overlay Menu	35
5.2.5.	Implementasi Halaman Fasilitas Umum.....	36
5.2.6.	Implementasi Halaman Transportasi Umum.....	37
5.2.7.	Implementasi Halaman Informasi Lalu Lintas	38
BAB VI PENGUJIAN DAN EVALUASI		41
6.1.	Tujuan Pengujian.....	41
6.2.	Kriteria Pengujian.....	41
6.3.	Skenario Pengujian.....	41
6.4.	Evaluasi Pengujian	42
BAB VII KESIMPULAN DAN SARAN		44
7.1.	Kesimpulan.....	44
7.2.	Saran.....	44
DAFTAR PUSTAKA.....		46
BIODATA PENULIS.....		48

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Rancangan Antarmuka Halaman Autentikasi.....	15
Gambar 4.2 Rancangan Antarmuka Halaman Dalam Navigasi ..	15
Gambar 4.3 Rancangan Halaman Navigasi.....	16
Gambar 4.4 Rancangan Antarmuka Overlay Menu	16
Gambar 4.5 Rancangan Antarmuka Halaman Fasilitas Umum...	16
Gambar 4.6 Rancangan Antarmuka Halaman Transportasi Umum	17
Gambar 4.7 Rancangan Antarmuka Halaman	17
Gambar 5.1 Halaman Autentikasi	32
Gambar 5.2 Halaman Autentikasi Menunjukkan Peringatan Input Tidak Valid.....	33
Gambar 5.3 Halaman Dalam Navigasi.....	34
Gambar 5.4 Halaman Navigasi	35
Gambar 5.5 Overlay Menu pada Halaman Utama	36
Gambar 5.6 Halaman Fasilitas Umum	37
Gambar 5.7 Halaman Transportasi Umum.....	38
Gambar 5.8 Halaman Informasi Lalu Lintas	39

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Kebutuhan Fungsional Sistem	14
Tabel 6.1 Hasil Evaluasi Pengujian.....	42

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 5.1 Mengambil Lokasi Perangkat Terkini	20
Kode Sumber 5.2 Meminta Izin Lokasi Perangkat.....	21
Kode Sumber 5.3 Mendapatkan dan Memproses Data Tempat- Tempat Terdekat.....	24
Kode Sumber 5.4 Memeriksa Izin Akses Penyimpanan Perangkat	25
Kode Sumber 5.5 Meng-upload Foto Akun Baru	27
Kode Sumber 5.6 Mengganti Nama Akun	28
Kode Sumber 5.7 Autentikasi Ulang Kredensial	29
Kode Sumber 5.8 Mengganti Email Akun	30
Kode Sumber 5.9 Melakukan Logout dari Akun	31

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN KERJA PRAKTIK

Implementasi Aplikasi Android Vehicle Control and Safety System (VCSS)

Oleh:

Daanii Nabil Ghinannafsi Kusnanta 05111940000163

Disetujui oleh Pembimbing Kerja Praktik:

1. Agus Budi Raharjo, S.Kom.,
M.Kom., Ph.D.
NIP. 1990202011022



(Pembimbing Departemen)

2. Faturochman Pranacahya
Andrianto



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Implementasi Aplikasi Android Vehicle Control and Safety System (VCSS)

Nama Mahasiswa : Daanii Nabil Ghinannafsi Kusnanta
NRP : 0511194000163
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Agus Budi Raharjo, S.Kom.,
M.Kom., Ph.D.
Pembimbing Lapangan : Faturochman Pranacahya Andrianto

ABSTRAK

Aplikasi VCSS dikembangkan sebagai solusi atas masalah transportasi umum dan mobilitas di Kota Surabaya. Aplikasi ini mengumpulkan informasi lalu lintas, fasilitas umum, transportasi umum, dan navigasi dalam satu tempat. Aplikasi juga menjadi sistem penjamin keamanan bagi pengemudi transportasi umum dan masyarakat.

Implementasi dilakukan pada fungsionalitas aplikasi dan antarmuka yang telah dirancang sebelumnya. Implementasi awal ini diharapkan dapat dilanjutkan hingga aplikasi siap untuk dirilis ke publik.

Aplikasi ini adalah aplikasi Android yang dikembangkan dengan bahasa pemrograman Kotlin. Aplikasi juga menggunakan layanan Firebase untuk autentikasi dan basis data serta memanfaatkan data dari Google Maps menggunakan layanan API dan SDK dari Google Maps Platform.

Kata Kunci : Aplikasi Android, Firebase, Google Maps Platform, Kotlin

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah Subhanahu Wa Taala yang karena berkat rahmat-Nya penulis dapat melaksanakan Kerja Praktik yang menjadi kewajiban sebagai mahasiswa Departemen Teknik Informatika ITS dengan judul

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun dalam penyusunan buku laporan kerja praktik ini. Namun, penulis berharap buku laporan ini dapat digunakan sebagai bahan bacaan yang dapat menambah wawasan pembaca dan menjadi sumber referensi. Penulis juga mengharapkan kritik dan saran yang membangun untuk meningkatkan kesempurnaan buku ini.

Melalui buku ini pula, penulis ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan kerja praktik hingga penyusunan buku laporan ini. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis serta keluarga yang telah memberikan dukungan selama pelaksanaan kerja praktik.
2. Bapak Ary Mazharuddin S.Kom., M.Comp., Ph.D., selaku Koordinator Kerja Praktik.
3. Bapak Agus Budi Raharjo, S.Kom., M.Kom., Ph.D., selaku dosen pembimbing kerja praktik.
4. Bapak Faturochman Pranacahya Andrianto, selaku pembimbing lapangan kerja praktik.
5. Ibu Prof Erma Suryani, S.T., M.T., Ph.D., selaku pembimbing serta manajer proyek kerja praktik.
6. Serta teman-teman penulis yang juga memberikan dukungan selama pelaksanaan kerja praktik hingga penyusunan buku laporan ini.

Surabaya, 5 Mei 2023

Daanii Nabil Ghinannafsi Kusnanta

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Mobilitas menjadi hal yang sangat diperhatikan pada abad ke-21 ini. Terutama di daerah-daerah padat penduduk yang tiap harinya mengalami perpindahan manusia yang sangat banyak. Pemerintah kota-kota besar seperti Jakarta, Semarang, Solo, Bandung, dan Surabaya sudah mulai menganggap mobilitas dan transportasi umum sebagai aspek penting pada suatu kota.

Pada April 2018, Pemerintah Kota Surabaya meluncurkan layanan transportasi bus bernama Suroboyo Bus. Hingga Januari 2022, layanan ini melayani lima koridor. Kemudian pada Desember 2021, Kementerian Perhubungan RI meluncurkan layanan Trans Semanggi Suroboyo yang akan mengambil alih beberapa koridor Suroboyo Bus. Dua koridor Trans Semanggi Suroboyo mulai beroperasi pada 2022. Dengan begitu, saat ini Kota Surabaya memiliki dua operator bus kota.

Walaupun telah dalam tahap pengembangan, transportasi umum dan mobilitas di Kota Surabaya masih jauh dari sempurna. Masyarakat masih dibingungkan dengan adanya dua operator dengan dua aplikasi *tracking* berbeda. Selain itu, keamanan bagi pengemudi dan masyarakat dalam menggunakan transportasi umum juga belum terjamin di tengah kondisi jalanan Kota Surabaya yang macet.

Oleh karena itu, dikembangkan aplikasi VCSS (Vehicle Control and Safety System) yang diharapkan dapat menjadi solusi terhadap masalah-masalah tersebut. Aplikasi ini mengumpulkan informasi transportasi umum, fasilitas umum, dan lalu lintas dalam satu tempat. Selain itu, aplikasi juga dapat menjadi sistem penjamin keamanan bagi pengemudi dan masyarakat dengan sistem peringatan kecepatan tinggi, tombol panggilan darurat, dan pengingat sabuk pengaman dan helm.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai mata kuliah Kerja Praktik sebesar 2 SKS dan membantu pengembangan dan implementasi aplikasi VCSS. Pada kerja praktik ini penulis akan mengerjakan beberapa tugas pengembangan seperti implementasi dan pengujian. Selain itu, kerja praktik ini juga diharapkan dapat mempercepat proses implementasi agar aplikasi dapat segera dirilis.

1.3. Manfaat

Dengan proses implementasi yang akan dikerjakan, aplikasi nantinya dapat segera diuji ke pengguna sehingga pada akhirnya dapat dirilis untuk umum. Sehingga, aplikasi dapat segera dirasakan manfaatnya bagi masyarakat umum. Selain itu, kerja praktik ini juga memberikan manfaat berupa pengalaman kerja pada proyek riil bagi penulis.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana melakukan implementasi rancangan fungsionalitas yang telah ada pada aplikasi VCSS?
2. Bagaimana melakukan implementasi rancangan antarmuka pada aplikasi VCSS?

1.5. Lokasi dan Waktu Kerja Praktik

Pengerjaan kerja praktik dilakukan secara *remote*. Selama kerja praktik juga akan ada beberapa pertemuan daring untuk membahas kemajuan pekerjaan. Adapun kerja praktik dimulai pada tanggal 26 Januari 2023 hingga 26 April 2023.

1.6. Metodologi Kerja Praktik

1.6.1. Perumusan Masalah

Dalam tahap ini, penulis mengadakan pertemuan dengan tim pengembang yang telah ada untuk membahas kebutuhan dan rancangan yang telah ada. Pada pertemuan tersebut dibahas mengenai status proyek terkini dan rancangan yang akan diimplementasi. Selain itu, juga dibahas rencana implementasi yang meliputi *framework* dan bahasa yang akan digunakan.

1.6.2. Studi Literatur

Tahapan ini dilakukan untuk membahas kebutuhan teknologi yang telah ditentukan. Aplikasi akan menggunakan Kotlin dan Android Studio untuk pengembangannya. Untuk *back-end* akan digunakan Firebase. Aplikasi juga akan menggunakan beberapa API pihak ketiga pada Google Maps Platform. Pada tahap ini dilakukan pendalaman dan pemahaman terhadap literatur mengenai teknologi terkait yang akan membantu dalam proses implementasi.

1.6.3. Analisis dan Perancangan Sistem

Tahap ini meliputi penjelasan mengenai hasil dari studi literatur yang dilakukan. Dari beberapa metode yang ditemukan saat literatur, akan dilakukan analisis terkait metode mana yang paling tepat dan efektif untuk digunakan untuk menyelesaikan permasalahan. Hasil analisis kemudian akan digunakan sebagai pertimbangan dalam membangun kesepakatan terkait metode pengembangan.

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini dibagi menjadi dua pekerjaan utama, yaitu implementasi *back-end* dan implementasi *front-end*. Implementasi

back-end akan membahas mengenai basis data dan penggunaan API pihak ketiga. Sementara implementasi *front-end* akan membahas implementasi antarmuka.

1.6.5. Pengujian dan Evaluasi

Pengujian dilakukan terhadap fitur-fitur yang telah diimplementasi. Aplikasi akan diuji oleh pembimbing kerja praktik dan *owner* dari proyek ini. Hal ini dilakukan untuk menilai apakah implementasi sudah sesuai dengan rancangan serta tujuan dan kebutuhan dari aplikasi.

1.6.6. Kesimpulan dan Saran

Dilakukan penarikan kesimpulan dan saran dari pengujian dan evaluasi yang telah dilakukan. Kesimpulan juga akan menggambarkan hasil dari kerja praktik ini secara menyeluruh. Saran yang diambil dapat menjadi pertimbangan bagi pengembangan lanjutan dari proyek ini atau pengembangan sistem lain yang serupa.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini akan menjelaskan latar belakang dari dilakukannya proyek dalam kerja praktik ini. Selain itu, juga akan disampaikan tujuan, rumusan masalah, dan manfaat yang diharapkan dapat didapat setelah pengerjaan. Pada bab ini juga akan dijelaskan detail pengerjaan seperti lokasi dan waktu kerja praktik. Terakhir, pada bab ini akan dijelaskan metodologi dan sistematika laporan dari kerja praktik yang akan dilaksanakan

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum PT. Esurya Datapedia Semesta yang dijelaskan melalui profil dan lokasi perusahaan.

Profil perusahaan akan membahas bidang pekerjaan yang dilakukan perusahaan. Profil juga akan membahas beberapa produk yang pernah dikerjakan perusahaan.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik. Beberapa dasar teori yang akan dibahas adalah terkait pengembangan aplikasi Android seperti penjelasan soal pemrograman aplikasi Android serta IDE dan bahasa pemrograman yang digunakan. Selibuhnya juga akan dijelaskan mengenai layanan dan API pihak ketiga seperti Firebase dan Google Maps Platform.

1.7.4. Bab IV Analisis dan Perancangan Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik. Analisis sistem akan menjelaskan kebutuhan apa saja yang akan dipenuhi oleh sistem. Selain itu juga akan dijelaskan rancangan antarmuka sistem yang telah dirancang sebelumnya.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi. Penjelasan implementasi akan dibagi menjadi dua bagian. Bagian pertama akan menjelaskan implementasi fungsi sesuai analisis kebutuhan. Bagian berikutnya akan menjelaskan implementasi antarmuka sesuai rancangan.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik. Sebelum membahas soal hasil pengujian, akan dijelaskan mengenai tujuan, kriteria, dan skenario dari pengujian yang akan dilakukan. Setelah

itu, akan dibahas hasil berdasarkan skenario yang telah dijelaskan sebelumnya.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik. Pada bagian kesimpulan akan dijelaskan ringkasan dan poin-poin penting yang didapat setelah melakukan kerja praktik ini. Bagian saran akan menjelaskan rekomendasi bagi pengembangan selanjutnya dari hasil yang didapat dari proses pengerjaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil PT. Esurya Datapedia Semesta

PT. Esurya Datapedia Semesta merupakan perusahaan yang bergerak dalam bidang informasi dan komunikasi serta aktivitas profesional ilmiah dan teknis. Perusahaan ini telah banyak melakukan kerja sama dengan peneliti terkait proyek riset peneliti tersebut. Kerja sama yang dimaksud adalah bantuan pengembangan perangkat lunak dari riset terkait.

Perusahaan ini telah banyak merilis perangkat lunak yang juga telah terdaftar sebagai hak kekayaan intelektual. Beberapa perangkat lunak yang telah dirilis adalah logis.co.id dan Buperdas. logis.co.id adalah platform jual beli produk pertanian berbasis web. Sementara Buperdas adalah platform yang menyediakan informasi pengembangan budidaya pertanian untuk berbagai komoditas pertanian seperti jagung dan padi.

2.2. Lokasi Perusahaan

Meski kerja praktik dilakukan secara *remote*, perusahaan yang menjadi lokasi kerja praktik tetap memiliki kantor fisik. Kantor perusahaan ini berbasis di Kota Surabaya, Jawa Timur. Alamat lengkap dari perusahaan ini adalah East Coast Park Blok R5 No. 31, Kota Surabaya, Jawa Timur.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Pemrograman Aplikasi Android

Android adalah sistem operasi *mobile* yang dikembangkan oleh Google. Pengembang dapat mengembangkan aplikasi yang kemudian dapat digunakan pada perangkat *mobile* dengan sistem operasi tersebut. Aplikasi Android dapat dibangun dengan bahasa pemrograman Kotlin, Java, atau C++. Aplikasi kemudian dapat dikompilasi dengan Android SDK menjadi APK atau Android App Bundle yang kemudian dapat dipasang pada perangkat Android. (Horton, 2019)

3.2. Android Studio

Android Studio adalah *intergrated development environment* (IDE) resmi dari Google untuk pengembangan aplikasi Android. Android Studio dibangun berdasarkan kode editor dari JetBrains, IntelliJ IDEA dengan beberapa peningkatan fitur khusus untuk pengembangan aplikasi Android. Pengembangan aplikasi dalam kerja praktik ini menggunakan Android Studio sebagai IDE. (Horton, 2019)

3.3. Kotlin

Kotlin adalah bahasa pemrograman *cross-platform* dan *multi-purpose*. Bahasa ini dikembangkan oleh JetBrains sejak 2010. Kotlin banyak digunakan pada proyek *mobile* (Android dan iOS), server, aplikasi web, dan sains data. (Horton, 2019)

Pada 2017, Google mengumumkan bahwa Kotlin akan mulai didukung sebagai bahasa pemrograman untuk pengembangan aplikasi Android. Google juga mendukung dan mensponsori pengembangan Kotlin. Pada 2019, Google mengumumkan bahwa Kotlin adalah bahasa pemrograman pilihan untuk pengembangan aplikasi Android dan menyarankan semua proyek Android baru untuk menggunakan Kotlin. Proyek aplikasi

pada kerja praktik ini juga akan menggunakan Kotlin sebagai bahasa pemrograman pilihan. (Horton, 2019)

3.4. Firebase

Firebase adalah layanan *back-end* (*Backend as a Service*) dari Google. Layanan ini bertujuan untuk menjadi solusi pengembangan aplikasi dengan cepat dan mudah. Beberapa layanan yang disediakan Firebase adalah layanan Firebase Authentication, Firebase Cloud Firestore, Firebase Realtime Database dan lainnya. (Kumar S, 2018)

Firebase yang juga dikembangkan oleh Google mendapat banyak dukungan terkait integrasinya dengan Android. Oleh karena itu, layanan ini sering menjadi pilihan sebagai layanan *back-end* dalam pengembangan aplikasi Android karena integrasinya yang sangat mudah. Aplikasi pada kerja praktik ini juga menggunakan beberapa layanan Firebase seperti Firebase Authentication.

3.5. Google Maps Platform

Google Maps Platform adalah sekumpulan API (Application Programming Interface) dan SDK (Software Development Kit) yang disediakan Google bagi pengembang. Platform ini memungkinkan mereka untuk menggunakan Google Maps dalam sistem mereka atau mengambil data dari Google Maps. Pengembang dapat menggunakan API dan SDK yang tersedia dengan membayar sejumlah biaya sesuai dengan jumlah *request* yang digunakan.

Terdapat beberapa API dan SDK yang dapat digunakan oleh pengembang seperti Places API untuk mengambil data tempat, Maps SDK for Android untuk menggunakan Google Maps pada aplikasi Android, Directions API untuk mendapatkan arah, dan lainnya. Layanan ini merupakan layanan berbayar yang biayanya akan disesuaikan dengan penggunaan dan jenis API atau SDK yang digunakan. Aplikasi pada proyek ini juga menggunakan

beberapa SDK dan API dari Google Maps Platform, seperti Maps SDK for Android dan Places SDK for Android. (Amal W, 2015)

3.6. Git

Git adalah salah satu *version control system* yang digunakan dalam proses pengembangan perangkat lunak untuk mencatat perubahan dan versi pada file pengembangan. Selain itu, Git juga dapat menjadi alat kolaborasi karena perubahan dicatat dalam basis data yang tidak hanya terletak pada satu tempat. Basis data Git terdistribusi pada setiap sistem kolaborator pengembangan sehingga memudahkan kolaborasi dan manajemen proyek. (Chacon & Straub, 2014)

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Kebutuhan Sistem

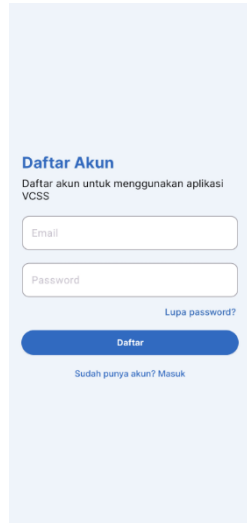
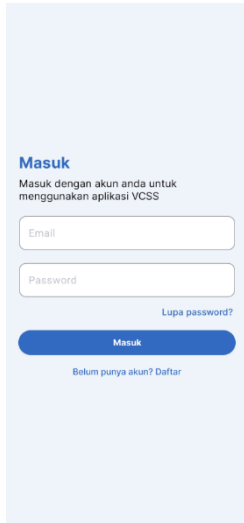
Sistem yang akan dikembangkan akan berusaha menyelesaikan masalah mobilitas di Kota Surabaya dengan menyediakan informasi yang dapat membantu penggunaanya mencapai tujuan. Informasi transportasi umum, situasi lalu lintas, dan navigasi menjadi fungsi utama dari sistem. Akan tetapi, sistem juga akan memberikan informasi terkait keamanan selama dalam perjalanan. Sehingga, kebutuhan fungsional sistem saat ini adalah sebagai berikut:

Tabel 4.1 Kebutuhan Fungsional Sistem

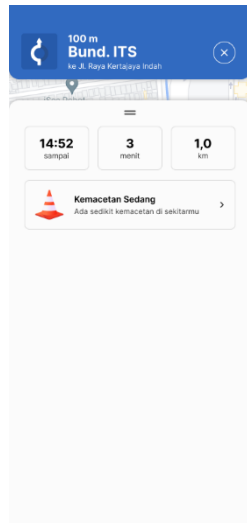
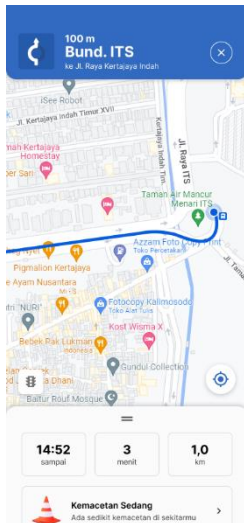
Kode	Deskripsi Kebutuhan
F-001	Mendapatkan informasi fasilitas umum
F-002	Mengganti foto akun
F-003	Mengganti nama
F-004	Mengganti email
F-005	Melakukan logout akun

4.2. Rancangan Antarmuka Sistem

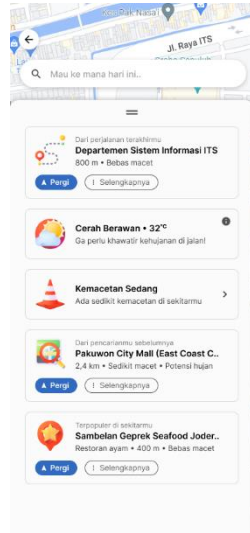
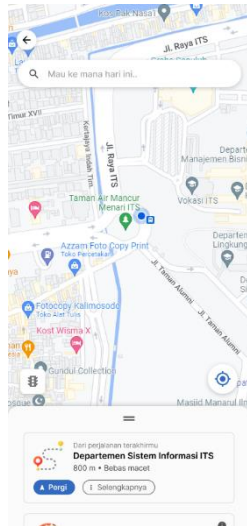
Pengembangan sistem telah berlangsung beberapa saat dan proses perancangan antarmuka telah dilaksanakan. Sehingga dalam kerja praktik ini, penulis hanya perlu melakukan implementasi dari rancangan antarmuka yang telah ada. Beberapa rancangan antarmuka sistem adalah sebagai berikut:



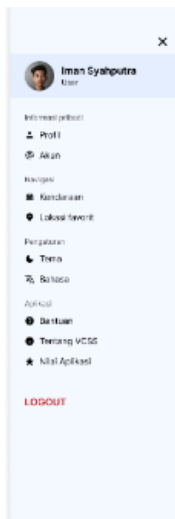
Gambar 4.1 Rancangan Antarmuka Halaman Autentikasi



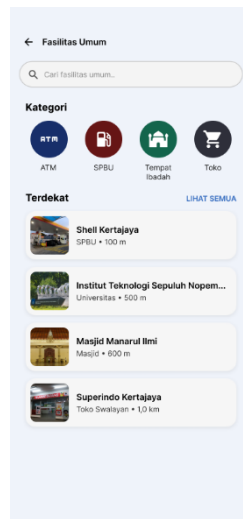
Gambar 4.2 Rancangan Antarmuka Halaman Dalam Navigasi



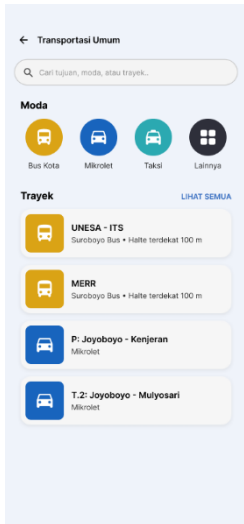
Gambar 4.3 Rancangan Halaman Navigasi



Gambar 4.4 Rancangan Antarmuka Overlay Menu



Gambar 4.5 Rancangan Antarmuka Halaman Fasilitas Umum



Gambar 4.6 Rancangan Antarmuka Halaman Transportasi Umum



Gambar 4.7 Rancangan Antarmuka Halaman Informasi Lalu Lintas

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI SISTEM

Implementasi dari sistem yang akan dikembangkan dibagi menjadi dua. Implementasi pertama adalah implementasi fungsi-fungsi sesuai analisis kebutuhan sistem. Sementara itu, implementasi kedua adalah implementasi antarmuka sesuai dengan rancangan antarmuka yang ada.

5.1. Implementasi Fungsi

5.1.1. Mendapatkan Informasi Fasilitas Umum

Pada halaman fasilitas umum akan ditampilkan beberapa informasi mengenai tempat yang berada di sekitar pengguna. Fungsi ini berusaha untuk menampilkan informasi fasilitas umum tersebut. Untuk itu, aplikasi perlu mengambil lokasi terkini pengguna terlebih dahulu.

Aplikasi akan memeriksa izin akses lokasi terlebih dahulu, yaitu `ACCESS_FINE_LOCATION` atau `ACCESS_COARSE_LOCATION`. Jika salah satu sudah diizinkan, maka akan diambil lokasi terakhir menggunakan Fused Location Provider dengan memanggil `fusedLocationClient.lastLocation`. Jika lokasi yang diterima ada atau tidak null, koordinat bujur dan lintang dari lokasi akan disimpan pada view model. Proses pengambilan lokasi ditunjukkan oleh Kode Sumber 5.1.

```
private fun getMyLastLocation() {
    if (requireContext().checkPermission(
        Manifest.permission.ACCESS_FINE_LOCATION) ||
        requireContext().checkPermission(
            Manifest.permission.ACCESS_COARSE_LOCATION)
    ) {
        fusedLocationClient.lastLocation.addOnSuccessListener {
            location: Location? ->
            if (location != null) {
                viewModel.setLocation(
                    LatLng(
```

```

        location.latitude,
        location.longitude)
    )
} else {
    viewModel.setLocation(null)
    Toast.makeText(
        requireContext(),
        "Location not found",
        Toast.LENGTH_SHORT).
        show()
    }
}
} else {
    requestPermissionLauncher.launch(
        arrayOf(
            Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.ACCESS_COARSE_LOCATION
        )
    )
}
}
}
}

```

Kode Sumber 5.1 Mengambil Lokasi Perangkat Terkini

Jika izin lokasi belum diberikan, aplikasi akan meminta izin untuk mengakses lokasi menggunakan *launcher*. Jika salah satu izin lokasi telah diberikan, aplikasi akan langsung mengambil lokasi dengan memanggil `getMyLastLocation()` kembali. Jika tidak, aplikasi akan memunculkan *alert dialog* menggunakan `MaterialAlertDialogBuilder()`. Saat pengguna memilih “Yes” pada *dialog*, pengguna akan diarahkan ke aplikasi Settings pada Android untuk mengizinkan izin terkait. Setelah itu, `getMyLastLocation()` akan dipanggil. Proses ini ditunjukkan dalam Kode Sumber 5.2.

```

private val waitSettingIntentLauncher =
    registerForActivityResult(
        ActivityResultContracts.StartActivityForResult()
    ) {
        getMyLastLocation()
    }
}

```

```

private val requestPermissionLauncher =
registerForActivityResult(
    ActivityResultContracts.RequestMultiplePermissions()
) { permissions ->
    when {
        permissions.getOrDefault(
            Manifest.permission.ACCESS_FINE_LOCATION,
            false) -> {
            getMyLastLocation()
        }
        permissions.getOrDefault(
            Manifest.permission.ACCESS_COARSE_LOCATION,
            false) -> {
            getMyLastLocation()
        }
        else -> {
            MaterialAlertDialogBuilder(requireContext())
                .setTitle("Warning")
                .setMessage("You need to allow location permission " +
                    "to use this app. " +
                    "Do you want to allow?")
                .setPositiveButton("Yes") { dialog, _ ->
                    dialog.dismiss()

                    val uri = Uri.fromParts(
                        "package",
                        requireActivity().packageName,
                        null
                    )
                    val intent = Intent(
                        Settings.ACTION_APPLICATION_DETAILS_SETTINGS
                    ).apply { data = uri }
                    waitSettingIntentLauncher.launch(intent)
                }
                .setNegativeButton("No") { dialog, _ ->
                    dialog.dismiss()
                }
                .show()
        }
    }
}
}
}

```

Kode Sumber 5.2 Meminta Izin Lokasi Perangkat

Lokasi yang didapatkan disimpan pada *shared view model* sehingga bisa digunakan di mana saja dalam aplikasi. Setelah lokasi didapatkan, aplikasi menggunakan lokasi tersebut dan Places SDK for Android dari Google Maps Platform untuk mendapatkan data tempat-tempat yang berada di sekitar pengguna. Aplikasi akan meminta data lokasi-lokasi terdekat pada Places API dengan menggunakan *request* `NearbySearchRequest()`. Pada *request* dikirim pula data lokasi dan radius. Data yang didapat kemudian diproses.

Proses pertama adalah menyaring data tempat berdasarkan tipenya. Data tempat pada Google Maps memiliki tiga tipe kategori. Tipe pertama adalah tipe-tipe tempat spesifik seperti `gas_station`, `department_store`, dan `beauty_salon`. Tipe kedua adalah tipe-tipe tempat yang lebih umum seperti `food`, `administrative_area_level_1`, dan `point_of_interest`. Tipe ketiga adalah tipe-tipe daerah dan bukan menunjukkan suatu tempat spesifik seperti `country`, `establishment`, dan `address`.

`PlaceSearchResponse` yang merupakan respons dari *request* tadi hanya mengembalikan tempat dengan tipe pertama dan kedua. Tetapi, aplikasi memerlukan tipe spesifik yang dapat ditampilkan yang menunjukkan tempat spesifik dan bukan tempat yang terlalu umum atau nama daerah. Untuk itu dilakukan penyaringan agar hanya didapatkan tempat dengan tipe pertama.

Selanjutnya, aplikasi juga membutuhkan foto dari tempat tersebut. Data foto yang didapat dari `PlaceSearchResponse` hanyalah nilai referensi. Untuk mendapatkan foto, aplikasi perlu melakukan *request* menggunakan `PhotoRequest()`. Hal ini dilakukan untuk setiap tempat yang didapat.

Aplikasi juga memerlukan data jarak dari tempat ke lokasi pengguna yang tidak tersedia di `PlaceSearchResponse`. Aplikasi harus menghitung jarak menggunakan API lain, yaitu `Distance Matrix API`. Untuk setiap data tempat, dilakukan *request* ke API tersebut untuk mendapatkan data jarak.

Setelah semua proses dilakukan, data-data tempat dapat dikirim ke *adapter* pada `RecyclerView`. *Adapter* tersebut akan

menampilkan data-data tersebut pada halaman. Proses-proses ini dapat ditunjukkan oleh Kode Sumber 5.3.

```
private fun getNearbyPlaces(location: LatLng) {
    with(viewModel.geoApiContext.value) {
        if (this != null) {
            val req = NearbySearchRequest(this)
                .location(location)
                .radius(500)
                .setCallback(
                    object : PendingResult.Callback<PlacesSearchResponse> {
                        override fun onResult(result: PlacesSearchResponse) {
                            val response = result.results

                            val filteredResponse =
                                response.filter { res ->
                                    res.types.any { it in PlaceTypes.TYPES_1 }
                                }

                            val results = filteredResponse.map { response ->
                                var photo: ImageResult? = null
                                if (response.photos != null) {
                                    photo = PhotoRequest(this@with)
                                        .photoReference(
                                            response.photos
                                                ?.firstOrNull()
                                                ?.photoReference
                                        )
                                        .maxHeight(100)
                                        .maxWidth(100)
                                        .await()
                                }

                                val distance = DistanceMatrixApiRequest(this@with)
                                    .origins(location)
                                    .destinations(response.geometry.location)
                                    .units(Unit.METRIC)
                                    .await()

                                Triple(response, photo, distance)
                            }

                            requireActivity().runOnUiThread {
                                showNearbyPlaces(results)
                            }
                        }
                    }
                )
        }
    }
}
```

```

        override fun onFailure(e: Throwable) {
            requireActivity().runOnUiThread {
                showNearbyPlacesError()
            }
        }
    })
}
}
}
}
}
}
}

```

Kode Sumber 5.3 Mendapatkan dan Memproses Data Tempat-Tempat Terdekat

5.1.2. Implementasi Mengganti Foto Akun

Dalam aplikasi, pengguna akan dapat mengganti foto akun mereka. Saat mendaftar, pengguna akan diberikan *placeholder* sebagai foto akun mereka dan dapat diubah pada pengaturan akun. Foto pengganti dapat diambil dari file yang ada pada penyimpanan perangkat pengguna. Untuk itu, akan diperiksa terlebih dahulu apakah aplikasi sudah memiliki izin `READ_EXTERNAL_STORAGE` dan `WRITE_EXTERNAL_STORAGE` atau belum.

Jika aplikasi belum memiliki izin, aplikasi akan meminta izin dengan *launcher* yang serupa saat meminta izin lokasi. Pengguna kemudian dapat memberikan izin terkait melalui *launcher* tadi. Jika izin sudah diberikan, aplikasi akan memulai *launcherImage* untuk memilih foto dari perangkat seperti yang ditunjukkan Kode Sumber 5.4.

```

btnChangePhoto.setOnClickListener {
    showChangePhotoLoading(true)

    if (
        requireContext().checkPermission(
            Manifest.permission.READ_EXTERNAL_STORAGE) &&
        requireContext().checkPermission(
            Manifest.permission.WRITE_EXTERNAL_STORAGE)
    ) {
        launcherImage.launch("image/*")
    } else {

```

```

requestPermissionLauncher.launch(
    arrayOf(
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    )
)
}
}

```

Kode Sumber 5.4 Memeriksa Izin Akses Penyimpanan Perangkat

Pada `launcherImage` diluncurkan, pengguna dapat memilih gambar yang ingin mereka gunakan sebagai pengganti foto. Konten yang didapat dari launcher akan berbentuk URI. Sebelum mengganti foto, foto baru perlu dimodifikasi supaya berbentuk persegi dengan rasio 1:1 karena rasio foto akun adalah 1:1.

Pemotongan dilakukan dengan mengubah URI ke Bitmap terlebih dahulu. Kemudian akan dibuat Bitmap baru menggunakan dimensi terkecil Bitmap sebelumnya sebagai dimensi lebar dan tinggi Bitmap baru. Setelah didapat Bitmap yang terpotong, Bitmap akan diubah kembali menjadi URI.

Penggantian foto dilakukan dengan memanfaatkan Firebase Authentication. Hal ini dilakukan karena data akun dibuat dengan Firebase sehingga perubahan informasi akun hanya dapat dilakukan melalui Firebase juga. Perubahan dilakukan dengan memanggil `auth.currentUser?.updateProfile()` dan `requestUserProfileChangeRequest`.

Foto yang baru di-*upload* dengan memanggil `setPhotoUri()` dengan URI foto baru sebagai argumen dalam `request`. Kemudian dari respons yang diterima, akan ditampilkan apakah penggantian foto berhasil atau tidak. Fungsi ini dapat dilihat pada Kode Sumber 5.5.

```

private lateinit var auth: FirebaseAuth
...
private val launcherImage = registerForActivityResult(
    ActivityResultContracts.GetContent()
) { uri ->
    if (uri != null) {

```

```

try {
    var bitmap = requireContext().uriToBitmap(uri)
    val dimen = bitmap.width.coerceAtMost(bitmap.height)
    bitmap = Bitmap.createBitmap(
        bitmap,
        (bitmap.width - dimen) / 2,
        (bitmap.height - dimen) / 2,
        dimen,
        dimen
    )
    val croppedUri = requireContext().bitmapToUri(bitmap)

    auth.currentUser?.updateProfile(
        UserProfileChangeRequest.Builder()
            .setPhotoUri(croppedUri)
            .build()
    )?.addOnCompleteListener { task ->

        if (task.isSuccessful) {
            Toast.makeText(
                requireContext(),
                "Photo changed successfully",
                Toast.LENGTH_SHORT
            ).show()
            showDetail()
        } else {
            Toast.makeText(
                requireContext(),
                "Failed to change photo",
                Toast.LENGTH_SHORT
            ).show()
        }
        showChangePhotoLoading(false)
    }

} catch (e: Exception) {
    Toast.makeText(
        requireContext(),
        "Failed to change photo",
        Toast.LENGTH_SHORT
    ).show()
}

```

```

        showChangePhotoLoading(false)
    }

    } else {
        showChangePhotoLoading(false)
    }
}
}

```

Kode Sumber 5.5 Meng-upload Foto Akun Baru

5.1.3. Implementasi Mengganti Nama Akun

Pengguna juga dapat mengganti nama pada akun mereka pada pengaturan akun. Penggantian nama juga menggunakan Firebase Authentication dan `UserProfileChangeRequest`. Pada `request builder` akan dipanggil `setDisplayname()` dengan argumen nama baru pada `view edtName`. Proses ini dapat dilihat pada Kode Sumber 5.6.

```

private lateinit var auth: FirebaseAuth
...
private fun tryChangeName() {
    binding.apply {
        auth.currentUser?.updateProfile(

            UserProfileChangeRequest.Builder()
                .setName(edtName.text.toString())
                .build()

        )?.addOnCompleteListener { task ->
            showLoading(false)

            if (task.isSuccessful) {
                Toast.makeText(
                    requireContext(),
                    "Name changed successfully",
                    Toast.LENGTH_SHORT
                ).show()

                findNavController().navigateUp()
            } else {

```

```

        Toast.makeText(
            requireContext(),
            "Failed to change name",
            Toast.LENGTH_SHORT
        ).show()
    }
}
}
}
}

```

Kode Sumber 5.6 Mengganti Nama Akun

5.1.4. Mengganti Email Akun

Pengguna dapat mengganti email akun pada pengaturan akun. Tetapi, email merupakan data krusial pada akun sehingga Firebase mengharuskan pengguna harus login dalam waktu dekat jika ingin mengganti email. Oleh karena itu, sebelum mengganti akun, aplikasi akan meminta kata sandi pengguna untuk melakukan autentikasi ulang. Setelah itu, baru aplikasi akan beralih ke layar penggantian email seperti yang ditunjukkan Kode Sumber 5.7.

```

private fun tryAuthenticate() {
    binding.apply {
        val isPasswordValid =
            ilPassword.validate(getString(R.string.password)) {
                if (it.isValidPassword()) null
                else getString(R.string.password_invalid)
            }

        if (!isPasswordValid) {
            showLoading(false)
            return
        }

        val credential =
            EmailAuthProvider.getCredential(
                auth.currentUser?.email!!,
                edtPassword.text.toString()
            )
    }
}

```

```

auth.currentUser
  ?.reauthenticate(credential)
  ?.addOnCompleteListener {

showLoading(false)
if (it.isSuccessful) {
  findNavController().navigate(
    R.id.action_changeEmailAuthenticationFragment_to_changeEmailFragment
  )
} else {
  Toast.makeText(
    requireContext(),
    "Authentication failed",
    Toast.LENGTH_SHORT
  ).show()
}
}
}
}
}

```

Kode Sumber 5.7 Autentikasi Ulang Kredensial

Pada *fragment* penggantian email juga akan digunakan Firebase Authentication untuk mengubahnya. Perubahan dilakukan dengan memanggil `updateEmail()` dengan argumen email baru. Jika berhasil, aplikasi akan kembali ke halaman pengaturan akun. Jika tidak dan *exception* yang diberikan adalah `FirebaseAuthRecentLoginRequiredException` yang berarti kredensial pengguna sudah usang, aplikasi akan kembali ke halaman autentikasi pengguna. Proses ini dapat dilihat pada Kode Sumber 5.8.

```

private fun tryChangeEmail() {
  binding.apply {
    val isEmailValid =
      ilEmail.validate(getString(R.string.email)) {
        if (it.isValidEmail()) null
        else getString(R.string.email_invalid)
      }
  }
}

```



```

if (!isEmailValid) {
    showLoading(false)
    return
}

auth.currentUser
?.updateEmail(edtEmail.text.toString())
?.addOnCompleteListener { task ->
    showLoading(false)

    if (task.isSuccessful) {
        Toast.makeText(
            requireContext(),
            "Email changed successfully",
            Toast.LENGTH_SHORT
        ).show()
        findNavController().navigateUp()
    } else {
        if (task.exception
            is FirebaseAuthRecentLoginRequiredException) {
            findNavController().navigate(
                R.id.action_changeEmailFragment_to_changeEmailAu
                thenticationFragment
            )
        } else {
            Toast.makeText(
                requireContext(),
                "Email change failed",
                Toast.LENGTH_SHORT
            ).show()
        }
    }
}
}
}

```

Kode Sumber 5.8 Mengganti Email Akun

5.1.5. Melakukan Logout Akun

Pengguna dapat keluar dari akun mereka dalam aplikasi. Proses ini menggunakan Firebase Authentication karena proses login juga menggunakan layanan yang sama. Sebelum melakukan logout, akan muncul *alert dialog* untuk meminta konfirmasi dari pengguna. Jika pengguna mengkonfirmasi aksi logout, dipanggil `auth.logout()` untuk logout dari akun. Aplikasi lalu akan kembali ke halaman autentikasi seperti yang ditunjukkan Kode Sumber 5.9.

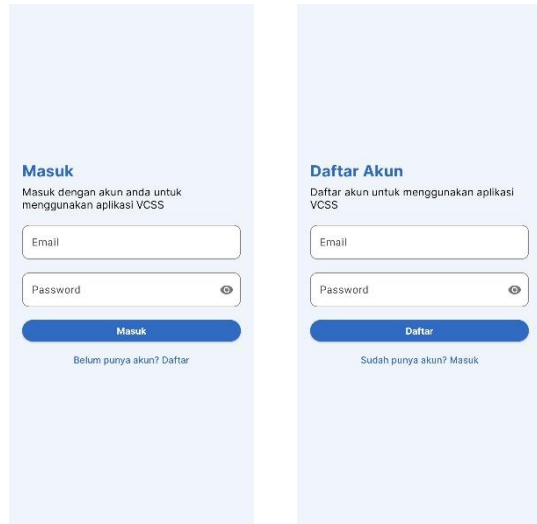
```
private fun tryLogout() {
    MaterialAlertDialogBuilder(requireContext()).apply {
        setTitle("Logout")
        setMessage("Are you sure you want to logout?")
        setPositiveButton("Yes") { dialog, _ ->
            dialog.dismiss()
            auth.signOut()
            findNavController().navigate(
                R.id.action_main_to_on_boarding
            )
        }
        setNegativeButton("No") { dialog, _ ->
            dialog.dismiss()
        }
        show()
    }
}
```

Kode Sumber 5.9 Melakukan Logout dari Akun

5.2. Implementasi Antarmuka

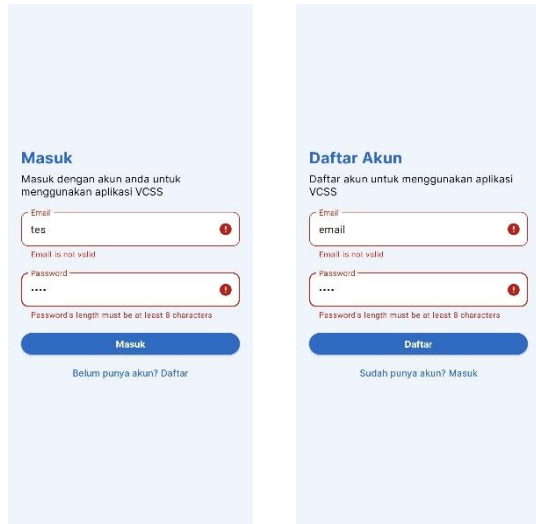
5.2.1. Implementasi Halaman Autentikasi

Sebelum pengguna menggunakan aplikasi, pengguna perlu login akun melalui halaman autentikasi. Gambar 5.1 menunjukkan implementasi dua halaman autentikasi, yaitu halaman masuk dan halaman daftar. Pada halaman terdapat dua view `EditText` untuk memasukkan informasi email dan kata sandi.



Gambar 5.1 Halaman Autentikasi

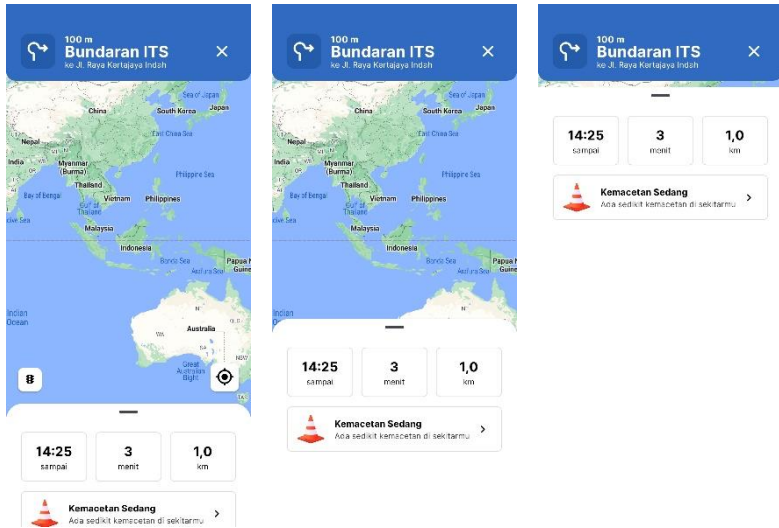
Formulir pada halaman ini juga akan melakukan validasi input sebelum mengirim informasi ke Firebase Authentication. Validasi dilakukan dengan mengecek pola yang sesuai untuk email dan kata sandi. Gambar 5.2 menunjukkan peringatan input tidak valid yang muncul saat informasi yang dimasukkan tidak valid.



Gambar 5.2 Halaman Autentikasi Menunjukkan Peringatan Input Tidak Valid

5.2.2. Implementasi Halaman Dalam Navigasi

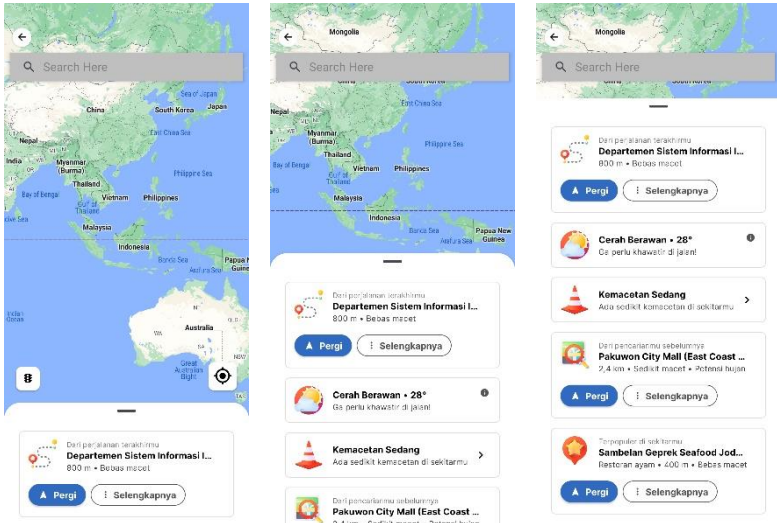
Gambar 5.3 menunjukkan halaman dalam navigasi. Di dalamnya terdapat beberapa *view* yang menunjukkan informasi navigasi. Terdapat peta yang dibuat menggunakan Google Maps *fragment* pada *FrameLayout*. Pada halaman juga terdapat *bottom sheet* yang dibuat dengan *CoordinatorLayout*. Pada Gambar 5.3 dapat dilihat *bottom sheet* yang dapat berinteraksi dengan input pengguna.



Gambar 5.3 Halaman Dalam Navigasi

5.2.3. Implementasi Halaman Navigasi

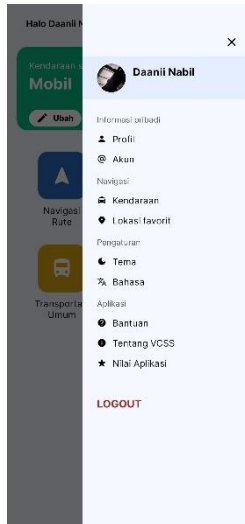
Pada aplikasi juga terdapat halaman navigasi yang akan menampilkan peta dan informasi terkait navigasi. Gambar 5.4 menunjukkan halaman navigasi. Peta pada halaman ini menggunakan Google Maps *fragment* pada *FrameLayout*. Selain itu, pada halaman ini juga terdapat *bottom sheet* yang dibuat menggunakan *CoordinatorLayout*.



Gambar 5.4 Halaman Navigasi

5.2.4. Implementasi Overlay Menu

Gambar 5.5 menunjukkan *overlay* menu pada halaman utama. Tata letak menu ini dibuat pada file tata letak yang berbeda dengan file tata letak untuk halaman utama. Untuk itu, *overlay* ini memanfaatkan *DrawerLayout* yang terdapat pada tata letak halaman utama dalam implementasinya.



Gambar 5.5 Overlay Menu pada Halaman Utama

5.2.5. Implementasi Halaman Fasilitas Umum

Gambar 5.6 menunjukkan halaman fasilitas umum. Pada halaman ini digunakan RecyclerView untuk menampilkan tempat-tempat yang ada di sekitar pengguna. Terdapat juga EditText yang berfungsi sebagai tempat pencarian nama tempat.



Gambar 5.6 Halaman Fasilitas Umum

5.2.6. Implementasi Halaman Transportasi Umum

Gambar 5.7 menunjukkan halaman transportasi umum. Terdapat beberapa view seperti TextInputLayout yang digunakan sebagai kolom pencarian dan beberapa Button yang dapat dipilih untuk mendapatkan informasi pada moda transportasi tertentu. Pada halaman ini juga digunakan RecyclerView untuk menampilkan daftar trayek.



Gambar 5.7 Halaman Transportasi Umum

5.2.7. Implementasi Halaman Informasi Lalu Lintas

Gambar 5.8 menunjukkan halaman informasi lalu lintas. Pada halaman ini terdapat beberapa *view* yang menunjukkan informasi lalu lintas dan informasi umum lainnya. Terdapat pula `TextInputLayout` yang digunakan sebagai kolom pencarian.



Gambar 5.8 Halaman Informasi Lalu Lintas

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

6.1. Tujuan Pengujian

Setelah implementasi selesai dilaksanakan, akan dilakukan pengujian terhadap implementasi tersebut. Pengujian aplikasi VCSS dilakukan guna menguji kesesuaian dan kesesuaian fungsionalitas dan antarmuka yang baru saja dikembangkan. Hasil dari pengujian tersebut kemudian dapat digunakan sebagai pertimbangan dalam implementasi berikutnya.

6.2. Kriteria Pengujian

Hasil dari pengujian didapatkan dengan memperhatikan apakah fungsionalitas baru tersebut berjalan sesuai dengan keinginan. Sehingga, hasil pengujian yang diharapkan adalah sebagai berikut:

1. Pengguna dapat mendapatkan informasi tempat-tempat sekitar
2. Pengguna dapat mengganti foto akun
3. Pengguna dapat mengganti nama akun
4. Pengguna dapat mengganti email akun
5. Pengguna dapat melakukan logout akun

6.3. Skenario Pengujian

Pengujian dilakukan dengan mencoba semua fungsionalitas yang baru saja ditambahkan dengan langkah-langkah pengujian sebagai berikut:

1. Login ke aplikasi VCSS.
2. Mengizinkan akses lokasi perangkat pada aplikasi.
3. Membuka menu fasilitas umum untuk beralih ke halaman fasilitas umum.
4. Memperhatikan apakah informasi tempat-tempat terdekat pada bagian “Terdekat” muncul dan sesuai seperti pada subbab 5.2.5.

5. Membuka menu pada halaman utama dan masuk pada pengaturan akun.
6. Melakukan penggantian foto dengan memilih tombol “Ganti Foto” dan mengizinkan akses penyimpanan perangkat.
7. Melakukan penggantian nama akun.
8. Melakukan penggantian email akun dengan melakukan autentikasi ulang menggunakan kata sandi terlebih dahulu.
9. Membuka menu pada halaman utama dan memilih tombol “Logout”

6.4. Evaluasi Pengujian

Evaluasi dilakukan terhadap hasil pengujian yang telah dilakukan. Hasil pengujian didapat dengan mengamati perilaku aplikasi sesuai dengan skenario pengujian. Penjabaran hasil evaluasi dijabarkan dalam Tabel 6.1.

Tabel 6.1 Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Pengguna dapat mendapatkan informasi tempat-tempat sekitar	Berhasil
Pengguna dapat mengganti foto akun	Berhasil
Pengguna dapat mengganti nama akun	Berhasil
Pengguna dapat mengganti email akun	Berhasil
Pengguna dapat melakukan logout akun	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan implementasi aplikasi VCSS pada kegiatan kerja praktik di PT. ini adalah sebagai berikut:

1. Implementasi fungsionalitas yang baru saja dikerjakan dapat memenuhi kebutuhan yang menjadi dasar pengembangan aplikasi VCSS.
2. Penggunaan Firebase dan Google Maps Platform dalam pengembangan aplikasi Android sangat membantu kecepatan pengembangan karena mudah diintegrasikan.

7.2. Saran

Saran untuk pengembangan selanjutnya untuk aplikasi VCSS adalah sebagai berikut:

1. Memperhatikan dan mengikuti lebih lanjut *best practices* terkait struktur file dalam pengembangan aplikasi Android.
2. Membangun *design guideline* untuk setiap komponen dan *layout* antarmuka pada tahap perencanaan agar lebih mudah diimplementasikan dan mengurangi perbedaan implementasi desain antar pengembang.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Amal W, R. (2015). *Learning Android Google Maps*. Packt Publishing.
- Android Studio Overview*. (n.d.). Retrieved May 4, 2023, from Android Developers: developer.android.com/studio/intro
- Application Fundamentals*. (n.d.). Retrieved May 4, 2023, from Android Developers: <https://developer.android.com/guide/components/fundamentals>
- Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
- Google Maps Platform Team. (2018, May 3). *Introducing Google Maps Platform*. Retrieved May 4, 2023, from Google Cloud: <https://cloud.google.com/blog/products/maps-platform/introducing-google-maps-platform>
- Grin, T. (n.d.). *Kotlin Media Kit*. Retrieved May 4, 2023, from Kotlin: <https://kotlinlang.org/assets/kotlin-media-kit.pdf>
- Horton, J. (2019). *Android Programming with Kotlin for Beginners*. Packt Publishing.
- Juliarto, R. (2020, November 25). *Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya*. Retrieved May 4, 2023, from Dicoding: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>
- Kumar S, A. (2018). *Mastering Firebase for Android Development*. Packt Publishing.
- Muhardian, A. (2017, February 8). *Apa itu Git dan Kenapa Penting bagi Programmer?* Retrieved April 5, 2023, from Petani Kode: <https://www.petanikode.com/git-untuk-pemula/>

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Daanii Nabil Ghinannafsi Kusnanta
Tempat, Tanggal Lahir : Situbondo, 25 April 2001
Jenis Kelamin : Laki-laki
Telepon : +6285155288581
Email : danynabilgk@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2019
Semester : 8 (Delapan)