

***MOBILE APPLICATION PSC 119 LAMPUNG TIMUR MENGGUNAKAN
FRAMEWORK FLUTTER ANDROID***

(Skripsi)

Oleh

NADIA MIRANTI



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

***MOBILE APPLICATION PSC 119 LAMPUNG TIMUR MENGGUNAKAN
FRAMEWORK FLUTTER ANDROID***

**Oleh
NADIA MIRANTI**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

MOBILE APPLICATION PSC 119 LAMPUNG TIMUR MENGGUNAKAN FRAMEWORK FLUTTER ANDROID

Oleh

NADIA MIRANTI

Kondisi darurat membutuhkan penanganan medis segera untuk menyelamatkan nyawa dan mencegah dampak kecelakaan yang serius. Upaya mengatasi masalah ini memerlukan solusi yang tepat dan efisien. Dalam kerangka meningkatkan layanan gawat darurat, Menteri Kesehatan menginisiasi pendirian *Public Safety Center (PSC)*, sebuah pusat koordinasi pelayanan gawat darurat untuk memastikan tanggapan yang tepat waktu dan akurat bagi masyarakat. Meskipun telah disediakan mobil ambulans, belum ada sarana yang mempermudah masyarakat mengakses layanan tersebut. Dalam konteks ini, untuk memperbaiki akses layanan darurat, dikembangkan sebuah aplikasi *front-end* menggunakan *Framework Flutter*. Aplikasi ini bertujuan untuk memberikan bantuan dalam kondisi darurat dengan fitur-fitur penting seperti laporan kejadian, tombol SOS untuk mengirim sinyal gawat darurat, riwayat laporan, dan informasi mengenai fasilitas kesehatan terdekat di Lampung Timur. Pengembangan aplikasi ini menggunakan metode Kanban dan dapat dipastikan setiap fitur telah diimplementasikan dengan baik dan sesuai dengan spesifikasi yang diharapkan. Melalui serangkaian pengujian *blackbox* dan penggunaan teknik *decision table* dengan total 14 skenario, aplikasi ini memberikan hasil uji sesuai dengan hasil yang diharapkan. Keberhasilan aplikasi ini menunjukkan efektivitasnya dalam menyediakan layanan darurat yang responsif dan bermanfaat bagi masyarakat Lampung Timur. Dalam konteks ini, pengembangan teknologi melalui aplikasi ini menjadi salah satu langkah penting dalam meningkatkan respons terhadap kondisi darurat dengan lebih efisien dan efektif.

Kata kunci — *Public Safety Center (PSC), Mobile Application, Front-End, Android, Flutter, Blackbox Testing*

ABSTRACT

MOBILE APPLICATION PSC 119 LAMPUNG TIMUR USING ANDROID FLUTTER FRAMEWORK

By

NADIA MIRANTI

Emergency situations require immediate medical attention to save lives and prevent serious accident repercussions. Dealing with this issue necessitates an appropriate and efficient solution. In the scope of enhancing emergency services, the Minister of Health has initiated the establishment of a Public Safety Center (PSC), a coordination center for emergency services to ensure timely and accurate responses for the public. Despite the provision of ambulance vehicles, there hasn't been a means to facilitate public access to these services. In this context, to improve emergency service accessibility, a front-end application was developed using the Flutter Framework. This application aims to provide assistance during emergencies with crucial features such as incident reporting, an SOS button for emergency signaling, report history, and information regarding nearby healthcare facilities in East Lampung. The application's development used the Kanban method, ensuring each feature was implemented properly and in accordance with the expected specifications. Through a series of black box testing and the use of decision table techniques involving a total of 14 scenarios, the application yielded test results consistent with the expected outcomes. The success of this application demonstrates its effectiveness in providing responsive emergency services beneficial to the East Lampung community. In this context, the technological development through this application stands as a crucial step in improving responses to emergencies more efficiently and effectively.

Keyword — Public Safety Center (PSC), Mobile Application, Front-End, Android, Flutter, Blackbox Testing

Judul Skripsi

**MOBILE APPLICATION PSC 119 LAMPUNG
TIMUR MENGGUNAKAN FRAMEWORK
FLUTTER ANDROID**

Nama mahasiswa

Nadia Miranti

Nomor Pokok Mahasiswa

1915061049

Program Studi

Teknik Informatika

Fakultas

Teknik



MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

Ir. Meizano Ardhi Muhammad, S.T., M.T., IPM.

Ir. Trisya Septiana, S.T., M.T., IPM.

NIP. 198105282012121001

NIP. 199009212019032025

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

Herlinawati, S.T., M.T.

Yessi Mulyani, S.T., M.T.

NIP. 197103141999032001

NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

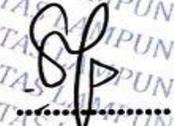
Ketua

Ir. Meizano Ardhi Muhammad, S.T., M.T., IPM.



Sekretaris

Ir. Trisya Septiana, S.T., M.T., IPM.



Penguji

Ir. Gigih Forda Nama, S.T., M.T.I., IPM.



2. Dekan Fakultas Teknik

Dr. Ing. Ir. Henny Fitriawan, S.T., M.Sc.

NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi : 19 Januari 2024

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “*Mobile Application PSC 119 Lampung Timur Menggunakan Framework Flutter Android*” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 19 Januari 2024

Pembuat pernyataan,



Nadia Miranti

NPM. 1915061049

RIWAYAT HIDUP



Penulis dilahirkan di Natar, pada tanggal 25 November 2000. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Amir Hamzah dan Ibu Yulianti.

Penulis menyelesaikan pendidikannya di SD N 4 Nxatar pada tahun 2013, SMP Al-Kautsar Bandar Lampung pada tahun 2016, dan SMA Al-Kautsar Bandar Lampung pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN.

Selama menjadi mahasiswa, penulis aktif mengikuti organisasi dan beberapa aktivitas sebagai berikut:

1. Anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Sosial dan Kewirausahaan, Divisi Kewirausahaan periode 2019/2020 sampai dengan periode 2020/2021.
2. Pada organisasi luar kampus, menjadi anggota dalam komunitas Gerakan Digital Ekosistem Nusantara (GRADIEN) yang merupakan wadah penulis dalam mengembangkan ilmu, bakat, serta kemampuan dalam bidang teknologi informasi dan komunikasi.
3. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas Pengembang Aplikasi Android dan Multi-Platform di Dicoding Indonesia pada tahun 2021.

4. Mengikuti program Magang Merdeka dari Kementerian Pendidikan dan Budaya sebagai *Mobile Developer* pada sebuah *software house* bernama Suitmedia Digital Agency pada tahun 2022.
5. Melaksanakan Kuliah Kerja Nyata pada bulan Januari sampai dengan Februari 2022 di Desa Trans Tanjung, Kecamatan Katibung, Kabupaten Lampung Selatan, Provinsi Lampung.
6. Melaksanakan Magang sebagai *Mobile Developer* pada CV. Gradien Digital Indonesia selama 1 tahun dari tanggal 12 Juli 2022 sampai dengan tanggal 12 Juli 2023.
7. Menjadi instruktur dalam Pelatihan Pengembangan Aplikasi Multi-Platform Menggunakan Framework Flutter yang diselenggarakan oleh Laboratorium Teknik Komputer Jurusan Teknik Elektro Universitas Lampung.

Prestasi yang pernah dicapai penulis antara lain:

1. Menjadi lulusan terbaik dalam program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya pada Kelas Pengembang Aplikasi Android dan *Multi-Platform Cycle 1* di Dicoding Indonesia.
2. Mendapatkan predikat proyek akhir terbaik dalam program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya pada Kelas Pengembang Aplikasi Android dan *Multi-Platform Cycle 1* di Dicoding Indonesia.
3. Menjadi pembicara pada *Expert Session SIB Dicoding X Kampus Merdeka Cycle 4* yang berlevel tingkat nasional.
4. Mendapat juara 1 pada perlombaan cipta puisi tingkat nasional yang diselenggarakan oleh non belmawa perorangan.

MOTTO

“Dan Dia Allah akan memberi manusia rezeki dari arah yang tak disangka-sangka. Dan barangsiapa yang bertawakal kepada Allah, niscaya Allah akan cukupkan (keperluan)nya. Sungguh, Allah melaksanakan urusan-Nya (menepati janji-Nya)”

(Q.S At-Talaq : 3)

“Dan janganlah engkau bersedih, sesungguhnya Allah bersama orang yang sabar”

(Q.S At-Taubah : 40)

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu”

(Ali bin Abi Thalib)

“When the world pushes you to your knees, you’re in the perfect position to pray.”

(Jalaluddin Rumi)

“Anyone who stops learning is old, whether at twenty or eighty. Anyone who keeps learning stays young. The greatest thing in life is to keep your mind young.”

(Henry Ford)

“Good things needs sabr”

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur kepada Allah SWT. atas segala Rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. shalawat serta salam teriring kepada Nabi Muhammad SAW sebagai suri tauladan *akhlakul karimah*.

KUPERSEMBAHKAN KARYA INI KEPADA:

“Ayah Amir Hamzah dan Bunda Yulianti atas segala kasih sayang yang telah membesarkan saya dari masih *zero* (nol) hingga menjadi *hero*.. Tanpa Ayah dan Bunda, Nadia tidak akan pernah bisa mencapai titik ini. Semoga Ayah dan Bunda selalu berada dibawah lindungan Allan dan dihadahi surga. Aamiin...”

“Terima kasih untuk kedua adikku Wulan Ayu Miranti dan Vanessa Syarla Miranti yang selalu menemani saya dalam kondisi apapun. Menjadi teman bermain, bercanda, berdiskusi, hingga bertengkar. Semoga kelak kalian akan menjadi sosok yang lebih sukses daripada uni.”

“Untuk diriku sendiri yang telah berhasil sampai pada tahap ini. Lihat Nadia, kita sudah sampai pada garis *finish!* Terima kasih atas segala pengorbanan dan jerih payah yang sudah kamu lakukan. Aku tahu ini semua tidak mudah bagimu. Kamu sudah melalui berbagai musim, air mata, drama, hingga jatuh bangun untuk sampai ke tahap ini. Ingat kalau perjuangan kita belum selesai. Ketahuilah, kamu hebat, *I’m so proud of you.*”

“Kepada sahabat-sahabat seperjuanganku, Nilam Cahya, Salma Irena Febriastia, Putri Anggia Cahyani, dan Sarah Mustika Dewi, terima kasih telah terlahir ke dunia dan menjadi bagian dari kehidupan perkuliahan saya. Tanpa kalian, kuliah akan terasa sangat berat dan monoton. Terima kasih atas segala dukungan dan bantuan sehingga skripsi ini dapat selesai dengan baik. Semoga persahabatan kita akan selalu terjalin selamanya.”

“Seluruh Keluarga Besar Teknik Elektro 2019, terima kasih telah hadir mewarnai hari-hariku di bangku kuliah. Semoga kita bisa menjadi orang-orang yang sukses di kemudian hari”

“Almamater tercinta, Universitas Lampung dan Jurusan Teknik Elektro”

SANWACANA

Puji syukur penulis panjatkan kepada kehadirat Allah SWT. yang telah memberikan rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “*Mobile Application PSC 119 Lampung Timur Menggunakan Framework Flutter Android*”. Selama masa penelitian penulis mendapatkan banyak bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Allah **وَ حَائِهٖ سُبُّ** atas segala rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar sampai selesai serta kepada Rasulullah Muhammad SAW yang telah menjadi suri tauladan selama penelitian berlangsung;
2. Ayah dan Bunda serta keluarga penulis yang selalu memberikan dukungan, arahan, dan motivasi hingga penelitian ini selesai.
3. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
4. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
5. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan Dosen Pembimbing Akademik yang telah membantu proses kelancaran pengerjaan penelitian dan perkuliahan;
6. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T., IPM selaku Pembimbing Utama yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukungan serta memudahkan penulis dalam menyelesaikan penelitian ini;
7. Ibu Ir. Trisyta Septiana, S.T., M.T., IPM selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;
8. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini;

9. CV. Gradien Digital Indonesia yang telah mewadahi penulis dalam melakukan penelitian;
10. Mbak Rika selaku *Admin* Program Studi Teknik Informatika yang telah banyak membantu penulis dalam urusan *administrasi* selama perkuliahan dan penelitian;
11. Nilam Cahya, Salma Irena Febriastia, Putri Anggia Cahyani, Sarah Mustika Dewi, dan segenap teman-teman TI 2019 yang telah menjadi rumah bagi penulis selama masa perkuliahan. Menemani, membantu dan memberikan pundak bagi penulis dikala susah dan sedih;
12. Teman-teman semasa SMA yang tergabung dalam sirkel “Kospley Mamak-Mamak” yang terlalu banyak untuk disebutkan satu-persatu, terima kasih atas segala motivasi dan dukungan yang telah diberikan;
13. Keluarga besar Teknik Elektro Angkatan 2019 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama;

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang Teknik Informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 19 Januari 2024
Penulis,

Nadia Miranti

DAFTAR ISI

	Halaman
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan Skripsi	3
II. TINJAUAN PUSTAKA	5
2.1 Gawat Darurat	5
2.2 PSC 119 Lampung Timur.....	5
2.3 Flutter	7
2.4 Android.....	7
2.5 Kanban Method	8
2.6 Trello	9
2.7 Application Programming Interface (API).....	10
2.8 JavaScript Object Notation (JSON)	10
2.9 Global Positioning System (GPS).....	11
2.10 Google Maps API.....	12
2.11 Blackbox Testing.....	13
2.12 Penelitian Terkait	16
III. METODE PENELITIAN	22
3.1 Waktu dan Tempat Penelitian	22
3.2 Jadwal Penelitian	22
3.3 Alat Dalam Penelitian	23
3.4 Metode Penelitian.....	24
3.5 Tahap Penelitian	25
3.5.1 Tahap <i>Requirements</i> (Pengumpulan Kebutuhan).....	25

3.5.2	Tahap Perancangan	28
3.5.3	Tahap Pengembangan Aplikasi.....	40
3.6	Tahap Testing	43
IV.	HASIL DAN PEMBAHASAN.....	44
4.1	Perancangan Antarmuka	44
4.1.1	<i>Flow</i> Pertama	45
4.1.2	<i>Flow</i> Kedua	48
4.1.3	<i>Flow</i> Ketiga	50
4.2	Pengembangan Aplikasi	52
4.2.1	<i>Flow</i> Pertama	52
4.2.2	<i>Flow</i> Kedua	60
4.2.3	<i>Flow</i> Ketiga	67
4.2.4	<i>Flow</i> Keempat	73
4.2.5	<i>Flow</i> Kelima	83
4.2.6	<i>Flow</i> Keenam	86
4.2.7	<i>Flow</i> Ketujuh	90
4.3	Evaluasi	95
4.4	Visualisasi Waktu Pengembangan	95
4.5	Tahap Pengujian	96
4.6	Analisis Keterbatasan	101
4.7	Capaian Penelitian	102
V.	KESIMPULAN DAN SARAN	107
5.1	Kesimpulan.....	107
5.2	Saran.....	108
	DAFTAR PUSTAKA	105

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Diagram Pengembangan Aplikasi Android PSC 119 Lampung Timur	6
Gambar 2.2 Desain Papan Kanban.....	8
Gambar 2.3 Tampilan Trello.....	10
Gambar 2.4 JSON Example	11
Gambar 2.5 Google Map API	13
Gambar 3.1 Tahapan Kanban pada Penelitian	25
Gambar 3.2 Format User Story	26
Gambar 3.3 <i>Use case diagram</i> aplikasi PSC 119 Lampung Timur	29
Gambar 3.4 <i>Activity Diagram</i> Melihat Informasi Faskes	30
Gambar 3.5 <i>Activity Diagram</i> Mengirim Laporan SOS	31
Gambar 3.6 <i>Activity Diagram</i> Mengirim Laporan Kejadian	32
Gambar 3.7 <i>Activity Diagram</i> Melihat Riwayat Laporan.....	33
Gambar 3.8 <i>Architecture Diagram</i> Aplikasi PSC 119 Lampung Timur	33
Gambar 3.9 Format taskcard pada papan Kanban	34
Gambar 3.10 Daftar tingkat prioritas pada label.....	34
Gambar 3.11 Breakdown user story pertama.....	36
Gambar 3.12 Breakdown user story kedua	36
Gambar 3.13 Breakdown user story ketiga.....	37
Gambar 3.14 Breakdown user story keempat	37
Gambar 3.15 Backlog aplikasi PSC 119 Lampung Timur.....	38
Gambar 3.16 Papan Kanban yang digunakan pada penelitian.....	38
Gambar 3.17 Kanban Board PSC 119 Lampung Timur Flow WIP.....	40

Gambar 3.18 Kanban Board PSC 119 Lampung Timur Flow Testing	41
Gambar 3.19 Kanban Board PSC 119 Lampung Timur Flow kembali ke WIP ...	42
Gambar 3.20 Kanban Board PSC 119 Lampung Timur Flow Done	42
Gambar 4.1 Backlog Perancangan Antarmuka	44
Gambar 4.2 Rancangan Antarmuka Aplikasi Menggunakan Figma	45
Gambar 4.3 Flow pertama proses perancangan antarmuka	45
Gambar 4.4 Rancangan Halaman Fasilitas Kesehatan	46
Gambar 4.5 Rancangan Halaman Detail Faskes	47
Gambar 4.6 Flow kedua proses perancangan antarmuka	48
Gambar 4.7 Rancangan Halaman Laporan Emergency	49
Gambar 4.8 Rancangan Halaman Laporan SOS	50
Gambar 4.9 Flow ketiga proses perancangan antarmuka	50
Gambar 4.10 Rancangan Halaman Riwayat	51
Gambar 4.11 Flow pertama proses pengembangan aplikasi	52
Gambar 4.12 Pengecekan API PSC 119 Lampung Timur menggunakan Postman	53
Gambar 4.13 Maps SDK for Android dari Google Cloud Platform	55
Gambar 4.14 Maps API Key	55
Gambar 4.15 Konfigurasi API Key pada AndroidManifest.xml	56
Gambar 4.16 Tampilan Home PSC 119 Lampung Timur	57
Gambar 4.17 Tampilan Ketika User Belum Memberi Izin Lokasi	58
Gambar 4.18 Kode Inisiasi Variabel Address Position	58
Gambar 4.19 Kode Program getPosition()	58
Gambar 4.20 Hasil Tampilan Setelah getPosition()	59
Gambar 4.21 Backlog proses pengujian pada flow pertama	59
Gambar 4.22 Flow kedua proses pengembangan aplikasi	61
Gambar 4.23 Deret Ikon Pada Laman Home	61

Gambar 4.24 Tampilan Laman Fasilitas Kesehatan	62
Gambar 4.25 Kode Program Pengecekan Izin Lokasi	63
Gambar 4.26 Hasil Pencarian Faskes	64
Gambar 4.27 Kode Program searchPlace	64
Gambar 4.28 Backlog proses pengujian pada flow kedua	65
Gambar 4.29 Flow ketiga dari proses pengembangan aplikasi	67
Gambar 4.30 Tampilan Laman Detail Faskes	68
Gambar 4.31 Kode Program Untuk Mendapatkan Detail Faskes	68
Gambar 4.32 Tampilan Rute Menuju Faskes Dengan Google Maps	69
Gambar 4.33 Kode Program Untuk Membuka Google Maps	70
Gambar 4.34 Button Call Center	70
Gambar 4.35 Menghubungi Call Center Faskes	71
Gambar 4.36 Kode Program Call Center	71
Gambar 4.37 Backlog proses pengujian pada flow ketiga	72
Gambar 4.38 Flow keempat dari proses pengembangan aplikasi	74
Gambar 4.39 Ikon Laporan Kejadian	74
Gambar 4.40 Kamera Laporan Kejadian PSC 119 Lampung Timur	75
Gambar 4.41 Kode Program Ikon Laporan Kejadian dan Kamera	75
Gambar 4.42 Preview Gambar	76
Gambar 4.43 Halaman Laporan Kejadian	77
Gambar 4.44 Daftar Faskes Yang Bisa Dipilih	78
Gambar 4.45 Kode Program Daftar Faksas	79
Gambar 4.46 Tampilan Laporan Yang Telah Terisi	79
Gambar 4.47 Kode Program Submit Laporan Kejadian	80
Gambar 4.48 Laporan Terkirim dan Laporan Gagal	81
Gambar 4.49 Dialog Peringatan Isi Laporan Terlebih Dahulu	81
Gambar 4.50 Backlog proses pengujian pada flow keempat	82

Gambar 4.51 Flow kelima dari proses pengembangan aplikasi	84
Gambar 4.52 Ikon SOS Pada Bottom Navigation Bar	84
Gambar 4.53 Kode Program Mengirim Laporan SOS	85
Gambar 4.54 Backlog proses pengujian pada flow kelima	85
Gambar 4.55 Flow keenam dari proses pengembangan aplikasi	87
Gambar 4.56 Tampilan Detail SOS	87
Gambar 4.57 Kode Program Cancel SOS	88
Gambar 4.58 Backlog proses pengujian pada flow keenam	88
Gambar 4.59 Flow ketujuh dari proses pengembangan aplikasi	90
Gambar 4.60 Tampilan Laman Riwayat Laporan	91
Gambar 4.61 Kode Program Tampilan Riwayat	92
Gambar 4.62 Kode Program Daftar Riwayat	92
Gambar 4.63 Detail Laporan SOS	93
Gambar 4.64 Backlog proses pengujian pada flow ketujuh	94
Gambar 4.65 Tampilan halaman Google Play Console	102
Gambar 4.66 Tampilan halaman buat aplikasi	103
Gambar 4.67 Tampilan halaman dashboard aplikasi	103
Gambar 4.68 Tampilan halaman buat rilis baru	104
Gambar 4.69 Menambahkan appbundle	104
Gambar 4.70 Menambahkan listingan play store	105
Gambar 4.71 Aplikasi PSC 119 Lampung Timur pada Google Play Store	106

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Contoh <i>Decision Tables</i>	16
Tabel 3.1 Jadwal Penelitian.....	22
Tabel 3.2 Alat yang digunakan pada penelitian	23
Tabel 3.3 <i>User Story</i> dari sistem yang dikembangkan.....	26
Tabel 3.4 Kebutuhan Fungsional	27
Tabel 3.5 Kebutuhan Non-Fungsional	28
Tabel 4.1 Dokumentasi API PSC 119 Lampung Timur	53
Tabel 4.2 Hasil pengujian blackbox testing pada fasilitas kesehatan	60
Tabel 4.3 Hasil pengujian blackbox testing pada fasilitas kesehatan	66
Tabel 4.4 Hasil pengujian blackbox testing pada fasilitas kesehatan	66
Tabel 4.5 Hasil pengujian blackbox testing pada fasilitas kesehatan	73
Tabel 4.6 Hasil pengujian blackbox testing pada laporan kejadian	83
Tabel 4.7 Hasil pengujian blackbox testing pada laporan SOS	86
Tabel 4.8 Hasil pengujian blackbox testing pada laporan SOS	89
Tabel 4.9 Hasil pengujian blackbox testing pada riwayat laporan	95
Tabel 4.10 Visualisasi Waktu Pengembangan	96
Tabel 4.11 Visualisasi Waktu Pengujian	96
Tabel 4.12 Hasil pengujian blackbox testing pada fasilitas kesehatan	98
Tabel 4.13 Hasil pengujian blackbox testing pada laporan kejadian	99
Tabel 4.14 Hasil pengujian blackbox testing pada laporan SOS.....	100
Tabel 4.15 Hasil pengujian blackbox testing pada riwayat laporan	101

I. PENDAHULUAN

1.1 Latar Belakang

Kondisi gawat darurat merupakan keadaan klinis pasien yang membutuhkan tindakan medis segera untuk penyelamatan nyawa dan pencegahan kecelakaan. Penanganan yang salah dan keterlambatan penanggulangan gawat darurat dapat menyebabkan hal yang fatal bagi korban [1]. Dilansir dari berita harian Detik.com, seorang pasien kecelakaan gawat darurat meninggal dunia akibat tidak mendapatkan ambulans dan mengalami keterlambatan penanganan sehingga nyawa korban tidak terselamatkan [2]. Hal ini merupakan dampak dari penanganan pada korban atau pasien yang kurang optimal.

Untuk menangani permasalahan ini serta dalam rangka meningkatkan mutu pelayanan kegawatdaruratan dalam Sistem Penanggulangan Gawat Darurat Terpadu (SPGDT), Menteri Kesehatan mengatur pembentukan *Public Safety Center (PSC)* oleh pemerintah kabupaten/kota yang mana merupakan pusat pelayanan yang menjamin kebutuhan masyarakat dalam hal-hal yang berhubungan dengan kegawatdaruratan di kabupaten/kota. Tujuan dari pembentukan PSC adalah sebagai wadah koordinasi untuk memberikan pelayanan gawat darurat secara cepat, tepat dan cermat bagi masyarakat [1].

Berdasarkan data tahun 2020, Lampung Timur memiliki total 1.051.994 jiwa penduduk yang mana terdiri dari 536.966 laki-laki dan 515.028 perempuan [3]. Dengan total penduduk yang banyak tersebut, tentu saja tingkat persentase terjadinya kondisi gawat darurat juga tinggi. Demi mempermudah pertolongan darurat, pemerintah kabupaten telah menyediakan mobil ambulans untuk mendukung PSC, namun sampai saat ini belum ada sebuah aplikasi yang dapat mempermudah masyarakat untuk menghubunginya. Dibutuhkannya

pengembangan aplikasi pada *smartphone* yang mana saat ini dimiliki oleh 370,1 juta dari total populasi Indonesia dan didominasi oleh sistem operasi Android yang berjumlah 93.2% dari total pengguna *smartphone* di Indonesia [4]. Selain itu penggunaan internet di Indonesia juga sudah mencapai 73.7% dari total populasi sehingga berdasarkan jumlah tersebut, maka tentu saja sarana akses PSC dalam bentuk aplikasi dapat sangat ramah dan mudah dijangkau bagi masyarakat ketimbang menghubungi nomor darurat menggunakan pulsa, mengingat pada saat ini pengguna ponsel lebih mendahulukan kuota daripada pulsa.

Untuk itu, dilakukan pengembangan *front-end* aplikasi menggunakan *framework* Flutter yang mana merupakan pekerjaan yang dilimpahkan dari Dinas Kesehatan Lampung Timur. Aplikasi ini melayani masyarakat dengan memberikan fitur-fitur yang membantu pada kondisi gawat darurat. Pengembangan berupa antarmuka untuk *user* yang berisikan beberapa fitur untuk menunjang pelaksanaan PSC, diantaranya laporan kejadian, tombol SOS untuk mengirim laporan sinyal gawat darurat, list fasilitas kesehatan terdekat, hingga informasi fasilitas kesehatan yang ada di Lampung Timur. Pekerjaan ini bukanlah lanjutan dari Magang Merdeka Belajar Kampus Merdeka (MBKM), melainkan bagian dari keterlibatan sebagai anggota komunitas *digital* Lampung yang dikenal sebagai Gradien.

1.2 Perumusan Masalah

Berdasarkan latar belakang, kajian masalah yang mendasari penelitian ini adalah bagaimana melakukan pengembangan aplikasi PSC 119 Lampung Timur menggunakan metode Kanban dengan *framework* Flutter untuk Android.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah mengembangkan aplikasi yang dapat membantu penanganan kegawatdaruratan di kabupaten Lampung Timur melalui aplikasi PSC 119 Lampung Timur.

1.4 Manfaat Penelitian

Penelitian ini mempermudah masyarakat kabupaten Lampung Timur dalam melakukan pelaporan situasi kondisi kegawatdaruratan yang ditemukannya sehingga dapat segera mendapatkan bantuan.

1.5 Batasan Masalah

Dalam penelitian ini, pembatasan masalah meliputi hal-hal sebagai berikut:

1. Spesifikasi aplikasi berdasarkan dari kebutuhan PSC 119 Lampung Timur.
2. Pengimplementasian dari aplikasi yang dikembangkan dilakukan oleh pihak PSC 119 Lampung Timur.
3. Pengembangan Aplikasi PSC 119 Lampung Timur berbasis Android hanya berfokus pada sisi *front-end* dan tidak mencakup API.

1.6 Sistematika Penulisan Skripsi

Sistematika penulisan yang digunakan pada skripsi ini terdiri dari 5 (lima) bab sebagai berikut:

BAB I : PENDAHULUAN

Bab ini membahas latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan skripsi.

BAB II : TINJAUAN PUSTAKA

Bab ini membahas dasar-dasar teori dan penelitian terkait yang berfungsi sebagai landasan dalam memahami permasalahan mengenai *Pengembangan Aplikasi Public Safety Center 119 pada Kabupaten Lampung Timur* yaitu penjabaran mengenai gawat darurat, PSC 119 Lampung Timur, Flutter, Android, *Kanban Method*, Trello, *Application Programming Interface (API)*, *JavaScript Object Notation (JSON)*, *Global Positioning System*

(GPS), Google Maps API, *Blackbox Testing*, serta penelitian terkait.

BAB III : METODE PENELITIAN

Bab ini berisikan metode penelitian yang digunakan dalam pengembangan *Aplikasi PSC 119 Lampung Timur*. Metode yang digunakan yaitu menggunakan *Kanban Method* yang mana akan dilakukan pengerjaan dua buah taskcard dalam satu *flow* (iterasi) yang urutan tahapannya yaitu *requirement gathering*, penentuan *user story*, perancangan aplikasi, penentuan *work on wait (WOW)*, tahap *work in progress (WIP)*, *done*, dan pelaporan.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi tentang pembahasan serta hasil yang diperoleh dalam penelitian.

BAB V : KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dan saran sebagai masukan untuk penelitian lanjutan di masa mendatang.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Gawat Darurat

Gawat darurat merupakan kondisi klinis dimana pasien membutuhkan tindakan medis secara cepat dan tanggap guna mencegah cacat dan kondisi terburuk yaitu kematian. Pasien sendiri merupakan orang yang secara mendadak berada dalam kondisi gawat dan mengancam nyawa atau kecacatan pada anggota tubuhnya. Kondisi gawat darurat dapat terjadi karena beberapa faktor, yaitu kecelakaan, bencana alam, penyakit, kejahatan lingkungan, dan sebagainya. Kriteria dari gawat darurat harus memenuhi kondisi mengancam nyawa, gangguan pada jalan nafas dan sirkulasi, penurunan kesadaran, membahayakan diri dan orang lain, serta memerlukan pertolongan segera [5].

2.2 PSC 119 Lampung Timur

Public Safety Center (PSC) atau Pusat Pelayanan Keselamatan Terpadu merupakan sebuah kebijakan dari Kementerian Kesehatan yang bertujuan untuk menjadi pusat pelayanan kebutuhan masyarakat dalam kondisi kegawatdaruratan. Menteri Kesehatan mengatur pembentukan PSC oleh pemerintah kabupaten/kota agar akses dan mutu pelayanan kegawatdaruratan meningkat. Hal tersebut diharapkan dapat menurunkan angka kematian dan kecacatan dengan memberikan penanganan cepat pada pasien gawat darurat [1].

PSC 119 Lampung Timur adalah sebuah aplikasi Android yang dibangun menggunakan *framework* Flutter dengan tujuan untuk melaksanakan pembentukan

PSC dan membantu penanganan kegawatdaruratan di Lampung Timur. Aplikasi ini merupakan pekerjaan yang dilimpahkan oleh Dinas Kesehatan Lampung Timur kepada sebuah komunitas di Lampung bernama Gradien. Dalam prosesnya, Dinas Kesehatan berperan sebagai *stakeholder* dan Gradien bertugas untuk men-*deliver* project sesuai *request* dari *stakeholder*. Dari Gradien, dibentuk sebuah tim pengembangan yang terdiri dari *Project Manager (PM)*, *Mobile Developer*, dan *Back-End Developer*. PM berperan sebagai pemimpin pada proses perencanaan dan pelaksanaan proyek, *Mobile Developer* berperan dalam pengembangan *front-end* aplikasi, dan *Back-End Developer* berperan merancang, mengelola, dan memelihara *database* yang digunakan oleh aplikasi. Sebelumnya pada tahun 2022, rancang bangun aplikasi PSC 119 Lampung Timur telah dilakukan dan berhasil membangun aplikasi dengan memiliki fitur dasar seperti autentikasi, edit profil, ubah kata sandi, dan panduan. Namun pengembangan untuk fitur utama di-*pending* sampai pada tahun 2023 dan akan dibahas pada penelitian ini. Fitur-fitur tersebut meliputi fitur fasilitas kesehatan, laporan kejadian, laporan sos, dan riwayat laporan.

Selain pada sisi Android, PSC 119 Lampung Timur juga tersedia versi *web*. Versi *web* ini dapat digunakan oleh admin dan pihak fasilitas kesehatan untuk mengelola seperti menambahkan data faskes dan menerima notifikasi laporan kejadian. Namun, versi *web* tidak memiliki interaksi secara langsung dengan aplikasi Android, sehingga dalam penelitian ini tidak akan dijelaskan secara rinci mengenai *website* PSC 119 Lampung Timur.

Diagram pengembangan dapat dilihat pada gambar 2.1.



Gambar 2.1 Diagram Pengembangan Aplikasi Android PSC 119 Lampung Timur

2.3 Flutter

Flutter merupakan *open source framework* oleh Google yang dapat digunakan untuk membangun dan mengembangkan sebuah aplikasi yang indah, dikompilasi secara native, dan multi-platform dari satu basis kode. Framework ini dapat dideploy ke banyak perangkat yaitu mobile, web, desktop, dan perangkat tertanam.

Flutter memiliki kelebihan seperti:

- 1.) Menggunakan bahasa pemrograman Dart yang dimana dipahami oleh perangkat *host* sehingga performanya cepat dan efektif serta mendekati aslinya.
- 2.) Menggunakan pustaka grafis Skia sumber terbuka yang dimiliki oleh Google untuk *rendering* UI sehingga dapat memberikan visual konsisten pada platform manapun.
- 3.) Mengutamakan kemudahan pada developer [6].

2.4 Android

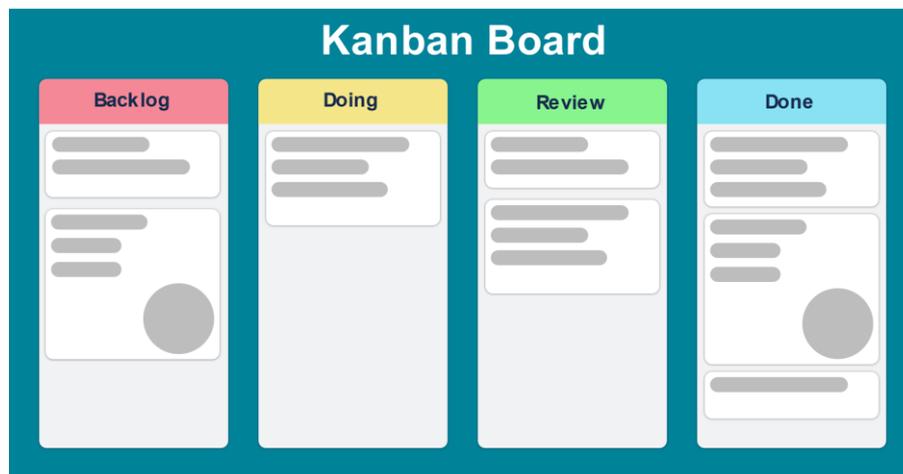
Android adalah sebuah sistem operasi yang dikembangkan oleh Google untuk *smartphone*, tablet, dan perangkat *mobile* lainnya. Sistem operasi ini adalah sistem operasi terpopuler pada platform *mobile* dimana telah memberdayakan ratusan juta perangkat seluler di lebih dari 190 negara di seluruh dunia. Android termasuk kit perangkat lunak pengembangan yang dapat digunakan untuk menulis kode dan merakit modul untuk membuat aplikasi bagi pengguna serta menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, Android mewakili ekosistem untuk aplikasi seluler.

Pada Android tersedia antarmuka berupa *touch screen* (UI) yang digunakan oleh pengguna untuk berinteraksi dengan aplikasi. Bentuk gerakan sentuh pada antarmuka berupa mengusap, mengetuk, dan mencubit untuk memanipulasi objek pada layar. Terdapat pula keyboard virtual yang dapat disesuaikan untuk input teks. Android dirancang untuk memberikan respons langsung terhadap input pengguna [7].

2.5 Kanban Method

Kanban berasal dari Bahasa Jepang yang artinya papan nama. Metode ini berupa *work flow* yang memvisualisasikan pekerjaan serta memaksimalkan efisiensi. Pada Kanban, terdapat kolom-kolom yang termasuk ke dalam flow, seperti *Work On Wait (WOW)*, *Work In Progress (WIP)*, *Review*, dan *Done*. Lalu tugas-tugas yang harus dikerjakan ditulis pada sebuah kartu dan diletakkan pada kolom WOW. Ketika mulai dikerjakan, maka kartu tugas dipindahkan ke kolom WIP. Setelah selesai, kartu tersebut dipindah ke alur selanjutnya yaitu *Review*, dan seterusnya.

Kanban dibuat untuk kelancaran dan keberlanjutan penyelesaian suatu project. Metode ini secara hati-hati dan efisien dapat mengontrol aliran dan kualitas pekerjaan untuk menyelesaikan masalah dengan segera. Ini juga dapat membatasi pekerjaan yang sedang berjalan sehingga tugas tidak menumpuk serta tim bisa menyesuaikan diri [8].



Gambar 2.2 Desain Papan Kanban

Berdasarkan Gambar 2.2 dapat diketahui desain dari papan Kanban dimana pada papan ini terdapat kolom-kolom sesuai dengan tahapan metode Kanban yang dilaksanakan. Pada contoh, terdapat kolom *Backlog*, *Doing*, *Review*, dan *Done*. *Backlog* disini berisi *card task* yang merupakan pecahan dari *user story*. *Task* yang dimasukkan pada *backlog* merupakan tugas-tugas *Work On Wait* yang nantinya digeser menuju kolom *Doing* ketika *task* tersebut dalam masa pengerjaan oleh

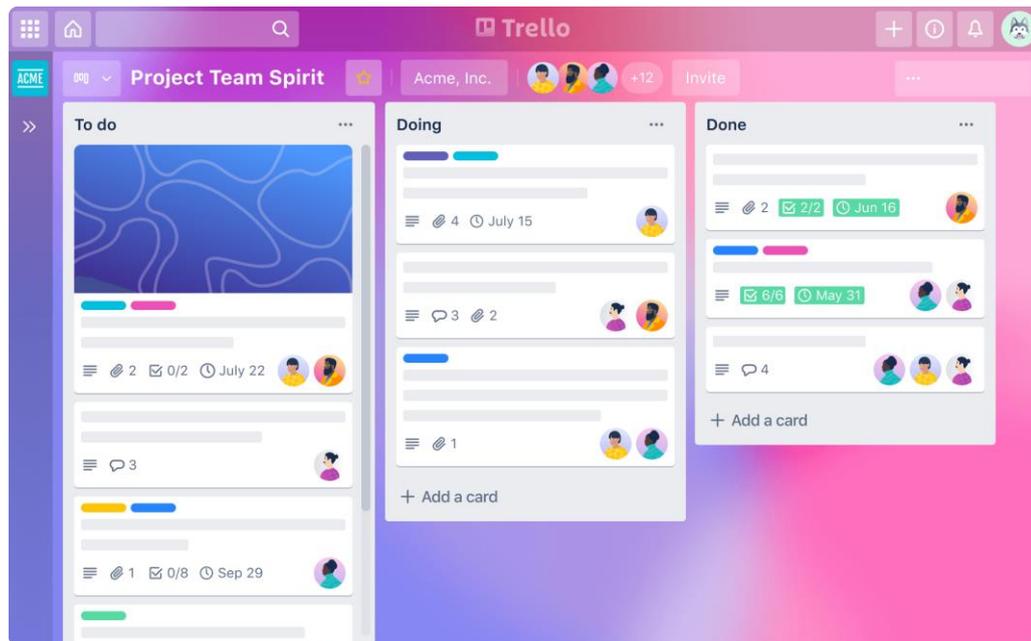
Developer. Ketika pengerjaan telah selesai, *card task* digeser menuju kolom *review* untuk memasuki tahap peninjauan. Setelah seluruh tugas selesai dan memberikan output yang sesuai dengan yang diinginkan, maka *card task* tersebut digeser menuju kolom terakhir yaitu kolom *Done*.

Pada metode Kanban, terdapat WIP Limit (*Work In Progress Limit*) yang merupakan batas *card* yang boleh masuk pada kolom WIP. Biasanya WIP Limit berkisar 2 atau lebih. Hal ini dilakukan untuk menghindari terlalu banyak task yang harus dikerjakan dan terbagi-baginya fokus pada *card-card* lainnya[9].

2.6 Trello

Trello merupakan aplikasi *task management* yang dapat digunakan untuk berkolaborasi dalam mengatur berbagai macam pekerjaan di satu tempat. Aplikasi ini berfungsi untuk memudahkan tim dalam mengembangkan proyek secara lebih efektif. Dapat dianalogikan Trello merupakan sebuah papan tulis putih yang dimana dapat ditempel berbagai macam *sticky notes* bertuliskan tugas-tugas baik yang belum, sedang, maupun telah selesai dikerjakan oleh tim. Dalam *notes* tersebut kita dapat melampirkan file-file pendukung serta kolom komentar. Fitur-fitur dalam Trello ialah :

- 1) *Boards* : berfungsi menjadi tampilan proyek yang didalamnya terdapat berbagai informasi.
- 2) *Cards* : fitur ini berguna untuk menjadi komponen untuk menuliskan tugas-tugas beserta ide yang dimiliki tim. Pada *cards*, dapat dicantumkan informasi-informasi lainnya untuk mempermudah komunikasi.
- 3) *Menu* : digunakan untuk menentukan atau mengubah konfigurasi khusus seperti manajemen anggota, manajemen *logs*, *power-ups*, hingga *filter cards*.
- 4) *Lists* : berfungsi sebagai wadah dari *cards* yang digunakan untuk menyusun kebutuhan proyek sesuai urutan pada kolom tertentu. Kolom-kolom yang terdapat pada Trello biasanya berupa *To Do* , *Doing*, dan *Done* [10].



Gambar 2.3 Tampilan Trello

2.7 Application Programming Interface (API)

API merupakan antarmuka *software-to-software* yang digunakan agar suatu aplikasi dapat berbicara satu sama lain serta melakukan pertukaran informasi antar perangkat melalui jaringan dan tanpa interaksi pengguna. API ini membantu memaparkan layanan bisnis atau data asset perusahaan kepada pengembang aplikasi. Pada prinsipnya, API dapat menghubungkan dua aplikasi, baik satu platform maupun *cross-platform* [11].

2.8 JavaScript Object Notation (JSON)

JSON atau *JavaScript Object Notation* adalah format file berbasis teks yang digunakan pada pertukaran data antara *server/API* dan klien. JSON sangat populer karena mudah dipahami, ringan, ringkas, menunjukkan data terstruktur berdasarkan syntax objek JavaScript, dan kompatibel dengan banyak Bahasa pemrograman. Format data JSON memungkinkan aplikasi berkomunikasi melalui jaringan, biasanya lewat RESTful API. Bentuk JSON yang sederhana terdiri dari Objek, Array, dan pasangan *key/value*.

```
{  
  "speaker" : {  
    "firstName": "Larson",  
    "lastName": "Richard",  
    "topics": [ "JSON", "REST", "SOA" ]  
  }  
}
```

Gambar 2.4 JSON *Example*

Bentuk format JSON yang valid dapat harus terdapat hal-hal berikut :

- 1) Objek diapit dengan tanda kurung kurawal { dan },
- 2) Array diapit dengan tanda kurung siku [dan] [12].

2.9 Global Positioning System (GPS)

GPS atau *Global Positioning System* merupakan sebuah sistem yang berfungsi untuk menentukan posisi dan letak di permukaan bumi berbasis satelit. GPS digunakan untuk menginformasikan penggunaannya mengenai keberadaan secara global. Data informasi yang diperoleh didapatkan dari satelit berupa sinyal radio dengan data digital. *Global Positioning System* bermula dari sebuah sistem navigasi ketentaraan yang dimiliki oleh Jabatan Pertahanan Amerika Serikat (DoD) pada tahun 1973. Awalnya, sistem ini dibuat hanya untuk tentara Amerika Serikat, namun kini telah terbuka untuk umum.

Saat ini, GPS telah beroperasi penuh serta memenuhi kriteria yang telah ditetapkan tahun 1960-an dimana sistem dapat memberikan penentuan posisi yang optimal, kecepatan tiga dimensi yang akurat, berkelanjutan, dan mendunia kepada pengguna. GPS dapat digunakan kapanpun dan dimanapun hingga 24 jam. Layanan ini dapat diakses secara gratis tanpa mengeluarkan biaya [13].

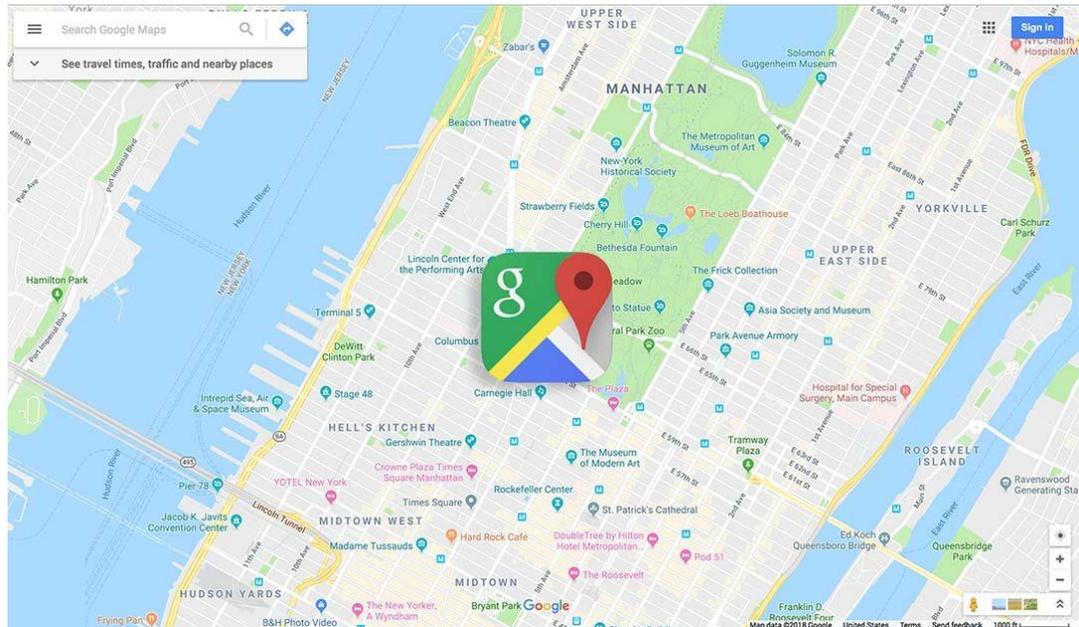
2.10 Google Maps API

Google Maps API merupakan layanan yang dapat digunakan oleh *software developer* untuk menampilkan peta, *geocoding*, tempat, dan konten lain dari Google pada halaman web atau aplikasi yang dikembangkan. Dengan menggunakan Google Maps API, seorang *developer* dapat juga mencantumkan peta dinamis, penanda yang menunjukkan suatu lokasi pada peta, bentuk, peristiwa di peta, *overlay* bumi, dan jendela info [14].

Pada Maps SDK untuk Android, tersedia beberapa *class* yang berfungsi untuk mengelola siklus proses. *Class* tersebut mendukung interaksi pengguna, seperti menetapkan status awal peta dan merespons input gestur pada *runtime*. *Class* dan antarmuka utama untuk menangani peta ini adalah *GoogleMap*, *SupportMapFragment*, *MapView*, dan *OnMapReadyCallback* [14].

Google Maps API menyediakan layanan dengan versi gratis dan berbayar. Namun pada versi gratis, akses yang tersedia terbatas pada beberapa fitur yang bisa dimanfaatkan oleh pengembang atau pemilik aplikasi. Berikut beberapa fitur yang tersedia dalam versi gratis Google Maps API:

1. Pemetaan yang Ditanamkan (*Embedded Maps*): Memungkinkan integrasi peta yang dapat disematkan di situs web atau aplikasi.
2. Penggunaan *Google Maps JavaScript API*: Penggunaan dasar API untuk menampilkan peta, menambahkan marker, dan beberapa fungsi dasar lainnya.
3. *Geocoding* dan *Reverse Geocoding*: Mengonversi alamat menjadi koordinat geografis atau sebaliknya.
4. Penggunaan Dasar *Places Library*: Menampilkan tempat-tempat terdekat, informasi tempat, dan pencarian tempat berdasarkan kriteria tertentu.
5. Penggunaan Dasar *Directions Service*: Mendapatkan rute antara dua atau lebih lokasi.



Gambar 2.5 Google Map API

2.11 Blackbox Testing

Blackbox Testing merupakan metode pengujian pada data berdasarkan persyaratan fungsional yang ditentukan tanpa mengetahui tentang detail implementasi dan struktur program akhir. Pengujian ini disebut juga pengujian berbasis data. *Blackbox Testing* mengacu pada pengujian fungsional dimana proses ditekankan pada pelaksanaan dan pemeriksaan masukan beserta keluarannya. Perangkat lunak yang diuji dianggap sebagai kotak hitam. Dalam proses pengujian, *input* dan *output* yang dilakukan dibandingkan dengan spesifikasi yang telah ditentukan untuk memvalidasi kebenaran.

Blackbox Testing berfokus memaksimalkan efektivitas pengujian dengan biaya yang minim. Pada fungsinya, pengujian ini dilakukan untuk menemukan masalah atau *bug* pada aplikasi yang dikembangkan. Tipe-tipe pada *Blackbox Testing* terdiri dari :

1) *Functional Testing*

Ini merupakan tipe pengujian yang berfokus pada fungsi fitur aplikasi secara spesifik. Pengujian fungsional ini lebih fokus pada aspek yang paling penting beserta integrasi dengan komponen utama.

2) *Non-Functional Testing*

Berfokus pada fitur tambahan dari perangkat lunak yang sifatnya non-fungsional. Tujuannya untuk menguji kemampuan aplikasi dalam menjalankan task tertentu.

Namun pengujian dengan metode ini memiliki beberapa kekurangan, yaitu :

- 1) Dapat terjadi kesalahan akibat kurangnya ketelitian.
- 2) Jika terjadi kesalahan, maka proses pengujian harus diulang kembali oleh daya ahli.
- 3) Penguji tanpa pengetahuan teknis memiliki kemungkinan melewatkan scenario tertentu sehingga dapat memunculkan *bug*

Beberapa teknik yang dapat dilakukan adalah :

b. *Requirements based testing*

Teknik ini dilakukan dengan menguji aplikasi berdasarkan persyaratan yang telah ditentukan dalam *Software Requirements Specification (SRS)*. Terdapat dua jenis persyaratan yaitu eksplisit dan implisit. Persyaratan eksplisit dinyatakan dan didokumentasikan dalam spesifikasi sedangkan persyaratan implisit atau tersirat adalah persyaratan yang tidak didokumentasikan tapi tetap dianggap sebagai syarat sebuah sistem.

c. *Positive and Negative Testing*

Pengujian positif akan membuktikan bahwa sebuah sistem telah melakukan apa yang seharusnya dilakukan sesuai dengan persyaratan, Ketika sebuah studi kasus memverifikasi bahwa proses uji yang dilakukan menghasilkan keluaran yang diharapkan, ini disebut pengujian positif. Bahkan apabila proses uji memberikan keluaran kesalahan ketika yang diharapkan adalah kesalahan, ini juga merupakan bagian dari pengujian positif.

Pengujian negatif dilakukan untuk memastikan bahwa perangkat lunak atau aplikasi tidak gagal ketika sebuah input tak terduga diberikan. Tujuan dari pengujian ini adalah untuk mencoba dan menghancurkan sistem agar dapat menemukan celah yang mungkin dapat bermasalah. Pengujian ini mencakup

skenario dimana produk tidak dirancang dan diberi kode. Dengan kata lain, nilai masukan mungkin belum dicantumkan dalam spesifikasi produk. Pengujian negatif adalah kondisi dimana keluaran menampilkan kesalahan ketika yang diharapkan adalah keberhasilan, dan keluaran menghasilkan keberhasilan ketika yang diharapkan adalah kesalahan. Sangat penting bagi penguji agar mengetahui kondisi negatif yang mungkin saja terjadi.

d. *Equivalence Partitioning*

Partisi Kesetaraan merupakan teknik uji perangkat lunak atau aplikasi yang melalui identifikasi sekumpulan kecil nilai input representatif yang menghasilkan sebanyak mungkin kondisi output berbeda. Pengujian ini dapat mengurangi total permutasi dan kombinasi input dan nilai output yang digunakan saat pengujian sehingga dapat meningkatkan cakupan dan mengurangi upaya dalam pengujian.

Partisi merupakan kumpulan dari input yang menghasilkan satu output yang diharapkan. Salah satu sampel dari partisi cukup untuk pengujian karena hasilnya akan sama pada nilai lainnya dan tidak akan menghasilkan kesalahan tambahan.

e. *Boundary Value Analysis*

Kondisi dan batasan merupakan sumber utama cacat produk. Yang dimaksud disini adalah situasi dimana berdasarkan nilai berbagai variable, sebuah tindakan tertentu harus diambil. *Boundary Value Analysis* adalah sebuah metode yang efektif dalam menangkap cacat yang terjadi pada batas. Pengujian ini merupakan proses uji dengan menentukan nilai batas atas dan batas bawah dari data yang diuji.

f. *Decision Tables*

Pada pengujian ini, kombinasi input dirangkum dalam sebuah tabel keputusan. Teknik *Decision Tables* digunakan untuk menguji perilaku sistem dengan beberapa kombinasi input yang berbeda. Metode ini disebut tabel *Cause-Effect*. Uji kasus dirancang untuk mencakup setiap kombinasi dan memastikan bahwa sistem berperilaku dengan benar untuk semua kemungkinan nilai masukan [15].

Tabel 2.1 Contoh *Decision Tables*

Condition	Case 1	Case 2	Case 3	Case 4
Kuantitas (Input)	True	True	False	False
Harga makanan (Input)	True	False	True	False
Output	Success	Error	Error	Error

2.12 Penelitian Terkait

Terdapat beberapa penelitian terkait yang dijadikan sebagai perbandingan serta rujukan mengenai metode serta hasil yang dicapai pada penelitian ini. Berikut merupakan ulasannya:

Penelitian yang dilakukan oleh Joei Ann Maghanoy dari FEU Institute of Technology Philippines membahas tentang pengembangan sebuah aplikasi yang dapat memfasilitasi masyarakat dalam melaporkan kondisi darurat seperti kecelakaan, kebakaran, kejahatan, dan insiden-insiden yang membutuhkan pertolongan cepat di Metro Manila [16]. Aplikasi yang dibuat berfokus untuk membantu masyarakat agar dapat dengan mudah menghubungi pihak-pihak yang berkaitan dengan kondisi darurat serta agar pemerintah daerah dapat cepat tanggap meluncurkan bantuannya ketika dalam kondisi genting dan sangat diperlukan. Alur kerja pada aplikasi *mobile* dari sisi *User* sebagai masyarakat Metro Manila dimulai dari *User* melakukan *login* pada aplikasi dan akan diarahkan pada halaman *citizen*. Setelah itu *User* dapat memilih tipe kejadian dan menginput detail terkait kejadian sehingga sistem dapat menghubungkan *user* pada Departmen yang sesuai dan yang paling dekat dari lokasi. Fitur lain yang dapat digunakan pada aplikasi adalah melihat map dan status apakah laporan yang telah diinput sudah dilihat atau belum.

Penelitian selanjutnya telah dilakukan oleh Ruth Simamora, Made Sudarma, dan I Made Arsa Suyadnya yang berjudul “Rancang Bangun Aplikasi *Ambulance Online* Berbasis Android” dari Universitas Udayana Denpasar [17]. Penelitian

berfokus untuk mempermudah masyarakat dalam memesan mobil ambulans yang berada di sekitarnya dengan cepat dan efisien. Aplikasi ambulans online ini dirancang berbasis Android dengan layanan Sistem Informasi Geografis (SIG) dan memiliki fitur pemesanan ambulans, peta yang terintegrasi dengan *Google Maps Api*, dan notifikasi dengan memanfaatkan *Firebase Cloud Messaging (FCM)*. Aplikasi yang dirancang diharapkan dapat membantu memberikan kemudahan dalam pelayanan tanggap darurat pada orang yang membutuhkan.

Penelitian yang dilakukan oleh Aditya Shukla, Bhavesh Solanki, dan Krunal Panchal dari *John College of Engineering and Management Mumbai, India*, membahas tentang pembuatan sistem aplikasi *Smart Ambulance* untuk menunjang pelaksanaan *Smart City* [18]. Pengembangan pada sektor penanganan kesehatan merupakan salah satu hal penting yang harus diperhatikan dalam membentuk *Smart City*. Karena kemacetan lalu lintas, seringkali ambulans terjebak dalam waktu yang lama hingga dapat membahayakan nyawa pasien. Selain itu, kecelakaan semakin meningkat dan tingkat hilangnya nyawa karena kecelakaan semakin tinggi. Apabila kecelakaan terjadi di wilayah yang tidak terjangkau seperti hutan dan sejenisnya, maka seseorang tidak dapat membagikan lokasi persisnya dan *driver* ambulans akan kesulitan menemukan lokasi. Faktor-faktor ini yang melatarbelakangi dilakukannya penelitian tentang *Smart Ambulance Service System*. Aplikasi yang dirancang merupakan aplikasi Android yang dapat memfasilitasi user dalam melaporkan kejadiannya serta mengirim gambar kepada *driver* ambulans. Gambar tersebut akan terkirim pada *driver* beserta GPS persis lokasi pengirim laporan. Terdapat pula beberapa fitur tambahan seperti *emergency of disaster*, *medical emergency call*, *user profile*, *map*, dan lain-lain yang dapat menunjang dalam pertolongan terhadap pasien yang membutuhkan.

Penelitian lainnya dilakukan oleh Nurul Faizah, Nurudin Santoso, dan Arief Andy Soebroto dari Universitas Brawijaya, Indonesia [19]. Penelitian ini berfokus pada pengembangan sistem aplikasi manajemen proyek menggunakan *Kanban framework*. Pada sebuah perusahaan *Software House* terdapat bermacam-macam proyek yang dibagi menjadi beberapa tim. Setiap proyek akan dipimpin oleh proyek manajer yang berbeda, namun proyek manajer tersebut dapat menjadi anggota tim

pada proyek lain. Dalam hal ini, akan ada banyak kebingungan dan rentan keliru pada proyek yang sedang dikerjakan. Penggunaan Kanban *framework* adalah untuk mencegah seorang proyek manajer kebingungan dalam memantau progress yang dipimpinya maupun yang dikerjakannya sebagai anggota. Dengan menggunakan sistem tersebut maka proyek manajer dapat memantau secara *real-time progress* dari anggota proyek yang telah selesai maupun yang harus dikerjakan.

Kemudian terdapat penelitian ini telah dilaksanakan oleh Muhammad Aliyya Ilmi, Fajar Pradana, dan Widhy Hayuhardhika Nugraha Putra dari Universitas Brawijaya [20]. Pada penelitian ini membahas mengenai manajemen sistem proyek perangkat lunak menggunakan metode Kanban di CV. Primavisi Globalindo. Hal ini berfungsi untuk mengurangi resiko kegagalan dalam pengerjaan proyek perangkat lunak di suatu perusahaan. Salah satu aspek penting adalah penjadwalan dimana meliputi penjadwalan sumber daya manusia dan tugas-tugas dalam pengerjaan proyek. Penelitian ini menerapkan unit testing terhadap 3 fitur utama sistem dan pengujian validasi pada 49 kebutuhan fungsional sistem yang dikembangkan. Pada penelitian ini metode yang digunakan adalah Kanban Board dimana terdapat *To do*, *Development* atau *Work in Progress*, *Test*, *Realese*, dan *Done*. Untuk *Work In Progress* sendiri telah dibatasi pada 5 task. Setiap kolom board tersusun sesuai dengan urutan pengerjaan yang telah ditentukan.

Selanjutnya, sebuah penelitian yang telah dilakukan oleh Bagus Ali Akbar dari Universitas Subang [21]. Penelitian ini membahas mengenai Perancangan Sistem Informasi Akademik Menggunakan Metode Kanban. Sistem dikembangkan untuk menunjang pelayanan akademik, kemahasiswaan, maupun pelayanan lainnya di Universitas Subang sehingga menghindari lambatnya pelayanan pada mahasiswa. Pada penelitian ini digunakan metode Kanban dan *Unified Modeling Language* untuk menggambarkan rancangannya. Identifikasi proses utama pada Kanban meliputi *To Do*, *Dev*, *Test*, *Realese*, dan *Done*. Untuk menentukan waktu penyelesaian (*Measure the Lead Time*) dibentuk sebuah tabel yang mendeskripsikan waktu penyelesaian tiap *task* sehingga dapat tergambar dengan jelas waktu penyelesaiannya.

Penelitian selanjutnya telah dilakukan oleh Sitti Aisa dari STMIK DIPANEGARA [22]. Penelitian berjudul Aplikasi Pencarian Bengkel Aktif dengan *Google Maps API* Berbasis *Web* yang bertujuan untuk mempermudah masyarakat mendapatkan informasi dan menjangkau lokasi bengkel di daerah kota Makassar. Proses pencarian lokasi dikembangkan menggunakan teknologi *Google Maps API* yang dapat menyediakan fasilitas untuk melihat lokasi bengkel tersebut. Penelitian ini menggunakan metode Waterfall dimana terdapat proses menentukan ruang lingkup aplikasi, menganalisis, membuat desain, menerapkan, menguji coba, serta menggunakan aplikasi. Hasil dari penelitian ini berhasil mengembangkan aplikasi yang dapat memberi kemudahan bagi pengguna untuk menemukan lokasi bengkel terdekat sehingga dapat memperbaiki kendaraannya lebih cepat dan efektif.

Penelitian yang telah dilakukan oleh Nurdin Nurdin, Sagaf S. Pettalongi, dan Mansur Mangasing dari Universitas Islam Negeri Palu, Indonesia, ini bertemakan Pengimplementasian Sistem Informasi Geografis berbasis *Google Maps API* untuk menentukan distribusi penerima beasiswa bidikmisi di daerah Sulawesi Tengah, Indonesia [23]. Penerima beasiswa ini adalah mahasiswa dari berbagai daerah dan latar belakang ekonomi di seluruh penjuru Indonesia. Namun sebaran mahasiswa belum pernah diteliti secara akurat. Penelitian ini mengembangkan Sistem Informasi Geografis (SIG) untuk memetakan persebaran mahasiswa penerima bidikmisi menggunakan *Google API* berdasarkan wilayah dan ekonomi geografis. Tujuannya untuk membantu perguruan tinggi untuk memantau penyaluran penerima beasiswa. Dari penelitian didapati hasil bahwa terdapat 350 siswa menerima bidikmisi dan berasal dari dua belas kabupaten di Sulawesi Tengah, Indonesia. Sistem yang dibangun berubah sebuah halaman website yang memiliki dua *role* yaitu *user* dan *admin*. Pada laman *user*, *user* dapat melakukan aktivitas seperti melihat sebaran penerima bidikmisi, melihat penerima bidikmisi, melihat data mahasiswa secara detail, melihat grafik jumlah penerima bidikmisi setiap tahunnya. Sedangkan pada laman *admin*, dapat melakukan beberapa hal seperti melakukan *login* dengan input *user* dan *password*, memasukan data mahasiswa/data *admin*, mencari data mahasiswa, dan melakukan *logout*. Pada sistem yang dibangun, penggunaan GIS dan *Google Maps API* terletak pada laman home yang mana menampilkan seluruh sebaran penerima beasiswa bidikmisi yang

ditandai lokasinya menggunakan marker. Untuk detail lengkap mengenai informasi penerima bidikmisi dapat dilakukan dengan men-*klik* marker yang ingin dilihat informasinya. Selain itu, *Maps* sebaran lokasi penerima bidikmisi juga dapat dilihat dengan mode satelit Google Maps.

Kemudian terdapat penelitian yang telah dilakukan oleh Farah Zakiyah Rahmanti, Oktavia Ayu Permata, Khodijah Amiroh, Philip Tobianto Daely, Anas Ittaquallah, dan Dimas Bagus Saputro dari Institut Teknologi Telkom Surabaya, Indonesia [24]. Tujuan dari Penelitian ini adalah mengembangkan Sistem Informasi Terintegrasi untuk Transportasi Umum di Surabaya dengan menggunakan *Global Positioning System* (GPS) dan Cloud Firestore. Dengan dilakukannya penelitian ini diharapkan dapat memudahkan penumpang mengakses serta memilih transportasi umum dengan memberikan rekomendasi rute terbaik, harga tiket yang murah, pemantauan posisi, dan visualisasi waktu tiba saat menggunakan transportasi umum di Surabaya. Sistem yang diberi nama Naik.in ini dibangun pada sistem aplikasi android dan terdiri pada dua role yaitu penumpang dan pengemudi. GPS pada sistem ini diimplementasikan untuk menentukan posisi (dengan menggunakan Google Map API) kemudian menyimpan data secara *real-time* menggunakan Firebase Cloud Firestore. Hal ini dapat memungkinkan penumpang untuk melacak kendaraan transportasi umum secara *real-time* melalui aplikasi dan berasal dari *positioning device* yang telah terpasang di setiap kendaraan.

Penelitian terakhir yang menjadi acuan merupakan penelitian yang telah dilakukan oleh Gita Indah Marthasari, Ani Tri Wahyuningsih, Muhammad Rizky Aviansyah, Muhammad Alfian Ramadhani, dan Zildan Rahmatullah dari Universitas Muhammadiyah Malang, Indonesia, yang mana mengangkat tema Pengujian Website Infotech Menggunakan Teknik Black-Box Decision Table [25]. Infotech adalah sebuah platform *website* berupa sistem informasi yang memberikan informasi mengenai praktikum, pelatihan, serta yang berhubungan dengan laboratorium informatika Universitas Muhammadiyah Malang. *Website* ini mendapatkan keluhan terkait *error* sistem, *server down*, kendala masuk, dan *bug* sistem yang membutuhkan perbaikan dan uji kualitas. Pada penelitian ini uji kualitas yang digunakan adalah Teknik *black-box testing* dengan metode *decision*

table. Fitur-fitur yang diuji yaitu merupakan fitur utama seperti *login* dan *submit file*. Dari pengujian yang telah dilakukan, didapati hasil bahwa dua fitur yang diuji yaitu fitur *login* dan fitur *submit file* telah memenuhi aturan atau syarat yang telah ditetapkan tanpa memunculkan *bug* atau *error*.

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada:

1. Waktu penelitian: April 2023 sampai dengan September 2023
2. Tempat penelitian: Program Studi Teknik Informatika Universitas Lampung

3.2 Jadwal Penelitian

Jadwal penelitian adalah sebagai berikut:

Tabel 3.1 Jadwal Penelitian

No	Aktivitas	Apr 2023	Mei 2023	Juni 2023	Juli 2023	Agus 2023	Sept 2023
1	Penentuan Kebutuhan	■					
2	Perancangan <i>User Story</i>	■					
3	Desain Aplikasi	■					
4	<i>Work On Wait (WOW)</i>		■	■			
5	<i>Work in Progress (WIP)</i>		■	■			
6	Pengujian			■	■		
7	Pelaporan				■	■	■

3.3 Alat Dalam Penelitian

Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut:

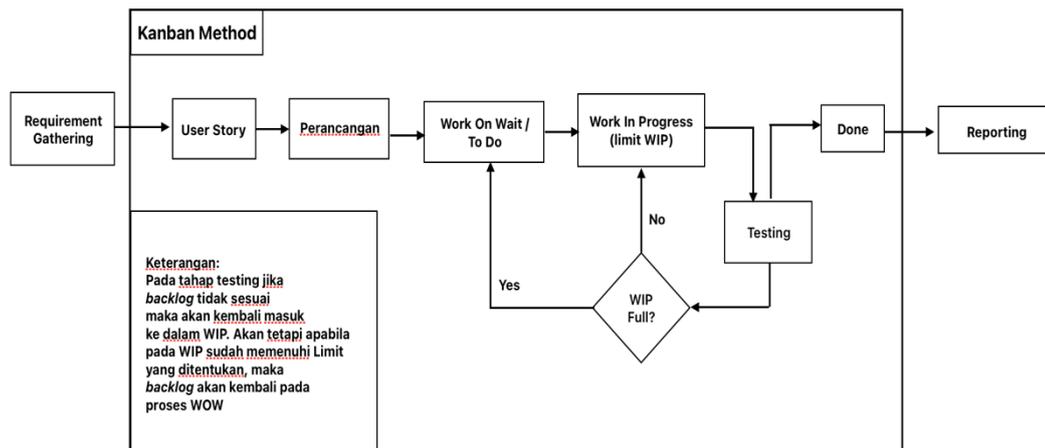
Tabel 3.2 Alat yang digunakan pada penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1.	Laptop	<i>Windows 11 64-bit Operating System, Intel Core i7-8550U CPU, RAM 16GB</i>	Perangkat keras yang digunakan untuk melakukan <i>coding</i> dan <i>compile</i> pada pemrograman.
2.	Flutter SDK	Versi 3.3.10	<i>Tools</i> yang digunakan untuk mengembangkan aplikasi <i>cross-platform</i> .
3.	Android Studio	Versi 11.0.11+9	<i>Integrated Development Environment (IDE)</i> resmi untuk pengembangan aplikasi Android serta menyediakan <i>Emulator</i> Android sebagai alat <i>run and debugging</i> aplikasi.
4.	Visual Studio Code	Versi 1.79.0	<i>Integrated Development Environment (IDE)</i> yang dapat melakukan <i>debugging</i> , <i>refactoring</i> , dan <i>compiling</i> untuk pengembangan aplikasi.
5.	Figma	<i>Website</i>	<i>Tools</i> aplikasi desain untuk membuat prototipe aplikasi.
6.	<i>Device Smartphone Android</i>	Android 12	Perangkat keras yang digunakan untuk menginstall, <i>testing</i> , dan <i>debugging</i> aplikasi.
7.	Postman	Versi 10.16	<i>Developing tools</i> yang digunakan untuk menguji API.
8.	Trello	<i>Website</i>	<i>Tools</i> manajemen tugas untuk pengimplementasian <i>Kanban Method</i> .

No	Nama Alat	Spesifikasi	Deskripsi
9.	Google Map API	<i>Maps SDK</i>	Layanan yang dapat menyertakan peta, <i>geocoding</i> , tempat, dan konten lainnya dari Google ke halaman aplikasi.
10.	API	API PSC 119 Lampung Timur	Digunakan untuk mengambil serta mengirim data untuk kebutuhan aplikasi PSC119 Lampung Timur

3.4 Metode Penelitian

Metode penelitian yang dilakukan mengikuti *Kanban Software Development Life Cycle Method* yang mana urutan tahapannya yaitu *requirement gathering*, penentuan *user story*, perancangan aplikasi, penentuan *work on wait (WOW)*, tahap *work in progress (WIP)*, *done*, dan pelaporan. Pada tahap *requirement gathering* dilakukan pengumpulan informasi untuk pengembangan aplikasi. Informasi-informasi yang dikumpulkan seperti kebutuhan fungsional dan kebutuhan non-fungsional. Setelah itu dilanjutkan pada tahap penentuan *user story*. *User story* yang telah dibuat ini kemudian dipecah menjadi beberapa *backlog* yang akan menjadi kumpulan *task-task* pada WOW. Saat memasuki tahap pengembangan, *developer* akan memindahkan *task* yang dikerjakan dari WOW ke WIP. Ditentukan *Limit WIP* yaitu terbatas pada dua buah *task*, dimana apabila limit terpenuhi, maka *task* lainnya harus menunggu pada WOW. Pada proses WIP ini lah *task-task* yang sudah ditentukan sebelumnya akan satu-persatu dikerjakan kemudian memasuki tahap testing ketika telah selesai. Apabila pada saat *testing* ditemukan *error* atau hasil tidak sesuai dengan syarat kebutuhan fungsionalitas, maka *task* akan kembali ke tahap WOW dan WIP. Proses ini akan terus berputar sampai pada tahap pengerjaan seluruh *task* telah selesai dan memberikan hasil sesuai dengan yang diinginkan. Setelah pengembangan selesai, maka akan memulai tahap pelaporan. Penggunaan Kanban dilakukan dari proses *user story*, perancangan, WOW, WIP, *testing*, dan *Done*.



Gambar 3.1 Tahapan Kanban pada Penelitian

3.5 Tahap Penelitian

Tahap penelitian dimulai dengan menentukan *requirements* aplikasi, perancangan, pengembangan aplikasi, testing, dan pelaporan. Pengembangan dari sistem ini menggunakan metode Kanban yang menggunakan visualisasi seperti kartu, kolom, ataupun papan untuk mengelola alur kerja secara efektif. Akan terdapat pula tahap penentuan *backlog* yang akan dipecah menjadi beberapa *task Work On Wait* pada papan Kanban. Tahapan yang dilakukan akan dijelaskan pada poin berikut:

3.5.1 Tahap *Requirements* (Pengumpulan Kebutuhan)

Pada tahap pengumpulan kebutuhan, dilakukan penentuan target pengguna, *user story*, serta analisis kebutuhan fungsional dan non-fungsional. Berdasarkan kebutuhan ini akan tergambar apa-apa saja yang perlu dikembangkan dalam aplikasi dan fitur apa saja yang perlu dibangun. Berikut merupakan tahapan yang dilakukan:

3.5.1.1 Target Pengguna

Pada tahap ini dilakukan penentuan target pengguna dari aplikasi PSC 119 Lampung Timur. Target pengguna aplikasi PSC 119 Lampung Timur merupakan masyarakat Lampung Timur yang mana spesifikasi kebutuhannya disampaikan oleh *Project Manager* dari komunitas Gradien. Proses pengujian aplikasi diwakili oleh *Project Manager* dari komunitas Gradien sebagai *user*.

3.5.1.2 Tahap Penentuan *User Story*

Pada tahap penentuan *user story* ini memecah seluruh kebutuhan fungsional menjadi beberapa *task* sebagai *backlog* yang akan menjadi acuan pada pengembangan aplikasi. Format penggunaan *user story* yang akan digunakan dapat dilihat pada Gambar 3.2. Bagian pengguna akan diisi oleh target dari pengguna aplikasi, bagian kegiatan akan diisi sesuai dengan *task* yang telah ditentukan sehingga menghasilkan hasil yang diharapkan sesuai dengan kebutuhan fungsionalitas aplikasi.

Sebagai [pengguna], saya ingin melakukan [kegiatan] agar [hasil yang diharapkan]

Gambar 3.2 Format *User Story*

Berdasarkan format diatas, ditentukan *user story* yang digunakan pada penelitian. Hal tersebut terangkum pada tabel 3.3.

Tabel 3.3 *User Story* dari sistem yang dikembangkan

No	<i>User Story</i>
1.	Sebagai <i>user</i> , saya ingin mengakses informasi mengenai fasilitas kesehatan terdekat dan seluruh fasilitas kesehatan yang ada di Lampung Timur agar saya dapat mengetahui informasi mengenai kamar kosong, stok darah, <i>call center</i> , hingga rute menuju fasilitas kesehatan.

2. Sebagai *user*, saya ingin melakukan pelaporan kejadian darurat yang saya temukan atau saya alami agar segera mendapatkan bantuan.
 3. Sebagai *user*, saya ingin mengirim laporan SOS dengan mudah ketika menemukan kondisi darurat agar bisa mendapatkan pertolongan dengan cepat.
 4. Sebagai *user*, saya ingin melihat riwayat pelaporan kejadian darurat yang pernah saya laporkan.
-

3.5.1.3 Tahap Analisis Kebutuhan

Pada tahap ini dilakukan analisis mengenai kebutuhan fungsional dan kebutuhan non-fungsional dari aplikasi PSC 119 Lampung Timur. Analisis ini kemudian menjadi acuan pengerjaan fitur-fitur aplikasi yang akan dikembangkan. Berikut merupakan hasil analisis kebutuhan fungsional aplikasi PSC 119 Lampung Timur:

a. Kebutuhan Fungsional

Kebutuhan ini mencakup kebutuhan utama yang terdapat pada fitur-fitur aplikasi sehingga memenuhi *user story* dan syarat aplikasi yang telah ditentukan. Berikut merupakan kebutuhan fungsional aplikasi yang dibangun dapat dilihat pada tabel 3.4.

Tabel 3.4 Kebutuhan Fungsional

No	Fitur	Kebutuhan Fungsional
1.	Fasilitas Kesehatan	Aplikasi menyediakan informasi fasilitas kesehatan terdekat yang berada di Lampung Timur
2.	Laporan Kejadian	Aplikasi menyediakan fitur pengiriman laporan kejadian gawat darurat

- | | | |
|----|-----------------|---|
| 3. | Laporan SOS | Aplikasi menyediakan fitur mengirim laporan SOS untuk kondisi darurat |
| 4. | Riwayat Laporan | Aplikasi menyediakan fitur riwayat laporan kejadian |

b. Kebutuhan Non-Fungsional

Kebutuhan ini merupakan kebutuhan pendukung/pelengkap kebutuhan fungsional. Pada kebutuhan ini berisi fitur-fitur yang mendukung berjalannya fitur utama. Berikut merupakan kebutuhan non-fungsional aplikasi yang dibangun dapat dilihat pada tabel 3.5.

Tabel 3.5 Kebutuhan Non-Fungsional

No	Deskripsi
1.	Interaksi dengan aplikasi menggunakan layar sentuh
2.	Waktu <i>loading</i> antar <i>screen</i> tidak lebih dari 3 detik
3.	Sistem selalu aktif selama 24 jam
4.	Sistem menyediakan fitur <i>retry</i> untuk memuat ulang halaman ketika terjadi masalah
5.	Sistem menyediakan <i>dialog error</i> ketika terjadi kesalahan

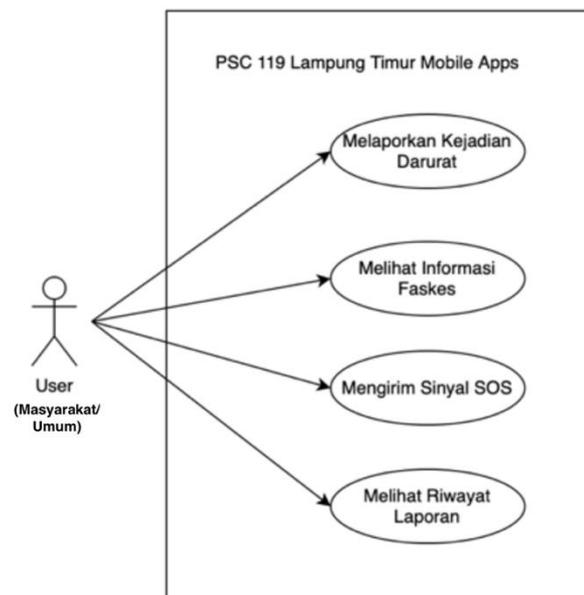
3.5.2 Tahap Perancangan

3.5.2.1 Tahap Perancangan Sistem

Tahap ini merupakan tahap perancangan sistem yang mana diantaranya membuat diagram *Unified Modelling Language* (UML) seperti *use case diagram* dan *activity diagram* dari pengembangan aplikasi PSC 119 Lampung Timur. *Use case diagram* dirancang berdasarkan *user story* yang telah dibuat sebelumnya. Kemudian dari diagram-diagram ini akan menjadi acuan dalam perancangan antarmuka aplikasi.

a. Use Case Diagram

Pada *use case diagram* sistem yang akan dibangun, terdapat 1 orang aktor yaitu *user* dan 4 buah aktivitas. Aktivitas yang dapat dilakukan seperti melaporkan kejadian darurat, melihat informasi faskes, mengirim laporan SOS, dan melihat riwayat laporan. Hasil dari penentuan *use case diagram* aplikasi PSC 119 Lampung Timur dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Use case diagram* aplikasi PSC 119 Lampung Timur

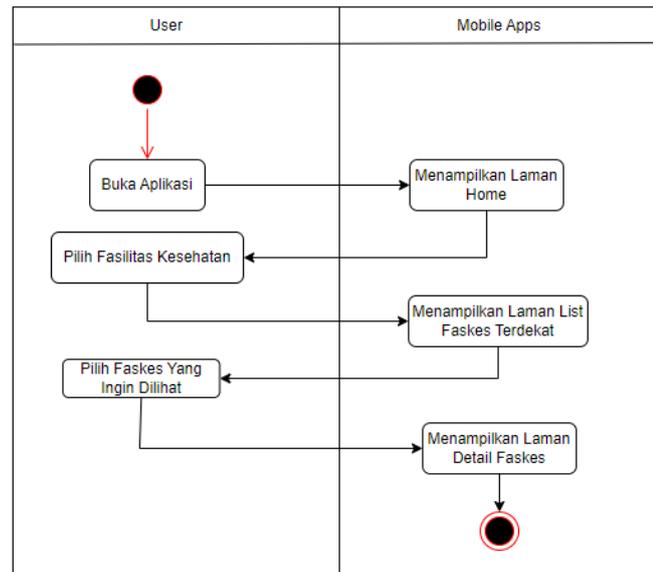
b. Activity Diagram

Activity Diagram merupakan diagram yang berguna untuk menggambarkan aktivitas aktor pada setiap fiturnya dalam menjalankan sistem berdasarkan fungsionalitas. Berikut merupakan rancangan *activity diagram*:

1. *Activity Diagram* Melihat Informasi Faskes

Pada proses melihat informasi faskes, aktivitas dimulai dari *user* membuka aplikasi dan sistem menampilkan laman *home*. Kemudian *user* akan memilih fitur Fasilitas Kesehatan. Sistem akan menampilkan halaman list dari fasilitas kesehatan terdekat yang ada di Lampung Timur. *User* akan memilih salah satu faskes yang ingin dilihat informasi detailnya. Dengan begitu, sistem

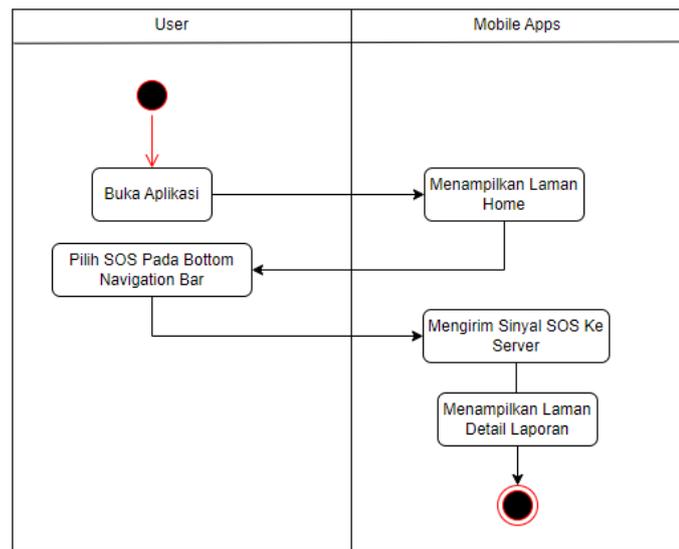
menampilkan halaman detail faskes yang telah dipilih, dan aktivitas selesai. Diagram dari aktivitas ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Activity Diagram* Melihat Informasi Faskes

2. *Activity Diagram* Mengirim Laporan SOS

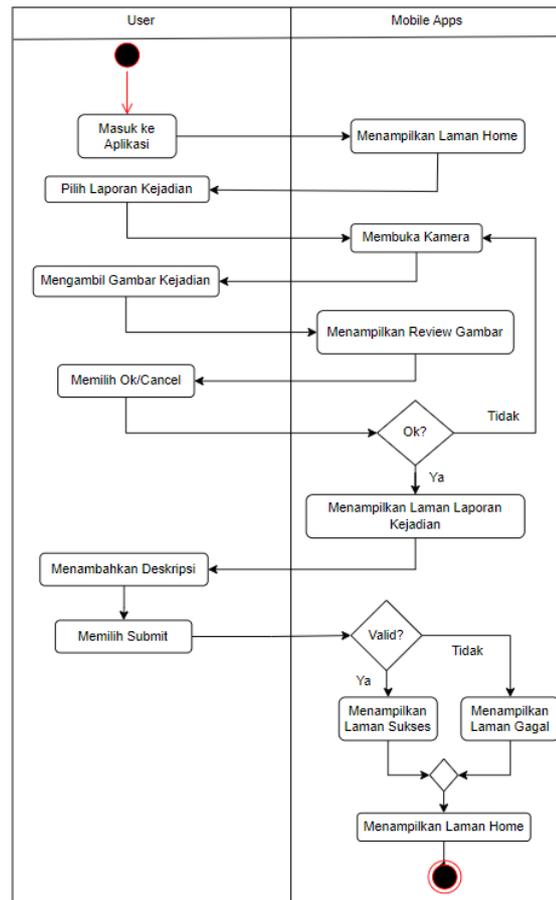
Pada proses melihat informasi faskes, aktivitas dimulai dari *user* membuka aplikasi sehingga tampil sebuah halaman *home* oleh sistem. Kemudian *user* akan menekan *icon* SOS pada *bottom navigation bar*. Sistem akan secara langsung mengirim laporan SOS ke server. Setelah selesai, sistem akan menampilkan laman detail laporan. Diagram dari aktivitas ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Mengirim Laporan SOS

3. Activity Diagram Mengirim Laporan Kejadian

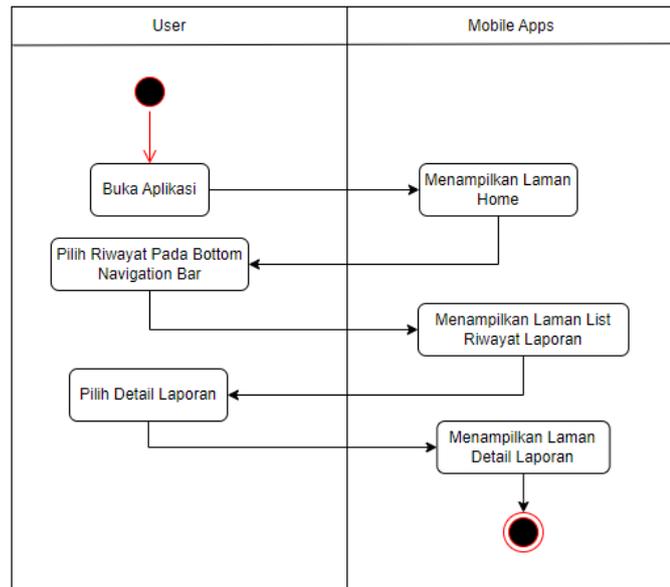
Pada proses mengirim laporan kejadian, aktivitas dimulai dari *user* masuk ke aplikasi, kemudian sistem akan menampilkan laman *home*. Setelah itu *user* memilih fitur Laporan Kejadian. Dari fitur ini, sistem akan langsung membuka kamera pada *device user*. Setelah *user* mengambil gambar, sistem akan menampilkan laman *preview* gambar agar *user* dapat mempertimbangkan gambar yang telah diambil. *User* dapat memilih *ok* atau *cancel*. Jika *ok*, maka akan ditampilkan laman laporan kejadian. Di laman ini, *user* dapat menambahkan deskripsi serta memilih faskes tujuan. Setelah *user* menekan *submit*, sistem akan memvalidasi apakah laporan yang dikirim *user* diterima server atau tidak. Jika ya, maka akan tampil ke laman sukses. Jika tidak, maka akan tampil laman gagal. Setelah itu, akan kembali ke halaman awal yaitu *home*. Diagram dari aktivitas ini dapat dilihat pada Gambar 3.6.



Gambar 3.6 Activity Diagram Mengirim Laporan Kejadian

4. Activity Diagram Melihat Riwayat Laporan

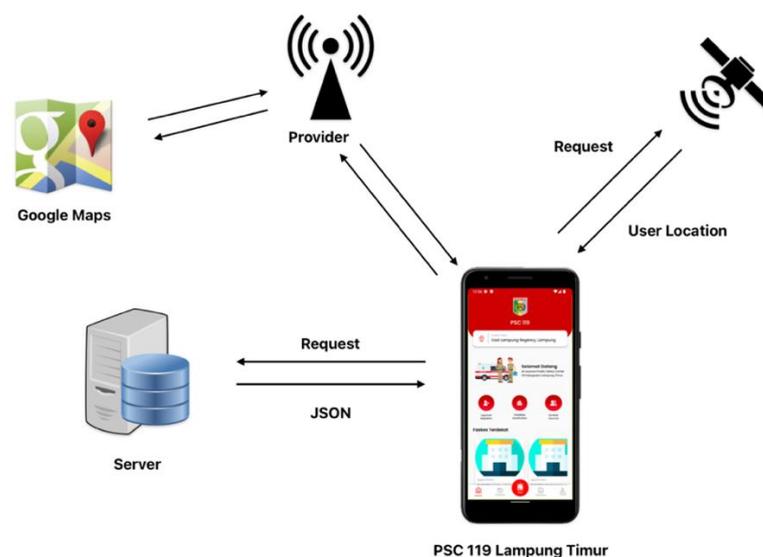
Pada proses melihat riwayat laporan, aktivitas dimulai dari ketika *user* membuka aplikasi, maka sistem akan menampilkan halaman *home*. Selanjutnya *user* akan memilih Riwayat pada *bottom navigation bar*. Dengan begitu, sistem akan menampilkan halaman list riwayat laporan kejadian yang pernah dikirim. *User* dapat memilih salah satu laporan untuk dilihat detailnya. Setelah itu, sistem akan menampilkan detail riwayat laporan tersebut. Diagram dari aktivitas ini dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity Diagram Melihat Riwayat Laporan

c. Architecture Diagram

Architecture Diagram merupakan representasi visual dari komponen sistem yang ingin dibangun. Hal ini dibuat dengan tujuan untuk menggambarkan perilaku aplikasi yang digunakan, berfokus pada bagaimana sistem berinteraksi satu sama lain dengan pengguna. Berikut *architecture diagram* dari sistem aplikasi PSC 119 Lampung Timur dapat dilihat pada Gambar 3.8.



Gambar 3.8 Architecture Diagram Aplikasi PSC 119 Lampung Timur

d. *Taskcard*

Taskcard merupakan kartu tugas yang tersusun pada sebuah kolom di papan kanban. Tugas-tugas tersebut didapat dari turunan *user story* yang telah ditentukan sebelumnya dan menghasilkan *backlog*. Format penulisan dari sebuah *taskcard* terdiri dari label, deskripsi tugas, dan *deadline* pengerjaan. Kartu-kartu tugas inilah yang akan dituliskan *backlog* untuk membantu proses pengembangan aplikasi. Adapun format penulisan *taskcard* yang digunakan pada penelitian ini dapat dilihat pada gambar 3.9.



Gambar 3.9 Format *taskcard* pada papan Kanban

Rincian dari penjelasan format *taskcard* adalah sebagai berikut:

1. Label

Label merupakan tanda atau status dari tugas yang dikerjakan. Status yang dimaksud dapat menunjukkan tingkat prioritas dari pengerjaan aplikasi. Berikut daftar dari tingkat prioritas yang digunakan pada penelitian akan dijelaskan melalui Gambar 3.10.



Gambar 3.10 Daftar tingkat prioritas pada label

Gambar 3.10 merupakan daftar tingkat prioritas pada label yang dicantumkan di *taskcard*. Deskripsi dari 5 buah label tersebut adalah sebagai berikut:

- a. *Slow* digambarkan dengan label berwarna hijau menandakan bahwa tugas tersebut memiliki batas waktu yang lebih dari 5 hari.
- b. *Medium* digambarkan dengan label berwarna oranye menandakan bahwa tugas tersebut memiliki batas waktu 2 sampai dengan 5 hari.
- c. *Urgent* digambarkan dengan label berwarna merah menandakan bahwa tugas tersebut memiliki batas waktu paling lambat 24 jam.
- d. *Verify* digambarkan dengan label berwarna ungu menandakan bahwa tugas tersebut perlu dilakukan verifikasi apakah sudah berfungsi dengan baik atau belum.
- e. *Done* digambarkan dengan label berwarna biru menandakan bahwa tugas tersebut telah selesai dikerjakan dan berfungsi dengan baik.

2. Deskripsi

Deskripsi merupakan keterangan atau informasi mengenai tugas yang dikerjakan. Hal ini berfungsi agar pengembang aplikasi dapat mengetahui tugas-tugasnya berdasarkan dari kalimat yang tertera pada deskripsi.

3. Deadline

Deadline adalah batas waktu pengerjaan dari sebuah tugas pada *taskcard*. *Deadline* akan dituliskan dalam format tanggal yang mana merupakan batas terakhir penyelesaian dari sebuah tugas. Setiap *taskcard* diberikan batas waktu/*deadline* agar pengerjaan lebih terstruktur dan sesuai target.

e. Backlog

Backlog berasal dari pemecahan *user story* yang telah ditentukan sebelumnya. *Backlog* berupa *task-task* yang akan diletakkan pada *work on wait* di *Kanban board*. Dengan *Backlog* ini, *developer* mendapat gambaran mengenai *task* yang harus dikerjakan dalam proses pengembangan aplikasi.

Dari *user story* pertama yang telah ditentukan, didapati 7 *backlog* yang dapat dilihat pada gambar 3.11.

User Story ...

Sebagai user, saya ingin mengakses informasi mengenai fasilitas kesehatan terdekat dan seluruh fasilitas kesehatan yang ada di Lampung Timur agar saya dapat mengetahui informasi mengenai kamar kosong, stok darah, call center, hingga rute menuju fasilitas kesehatan.

Backlog (Work On Wait)

<p>Medium</p> <p>Membuat Desain Halaman Fasilitas Kesehatan</p> <p>🕒 Apr 19</p>	<p>Urgent</p> <p>Buat Fungsi Faskes Terdekat</p> <p>🕒 May 15</p>
<p>Medium</p> <p>Membuat Desain Halaman Detail Fasilitas Kesehatan</p> <p>🕒 Apr 21</p>	<p>Medium</p> <p>Slicing UI Detail Fasilitas Kesehatan</p> <p>🕒 May 19</p>
<p>Medium</p> <p>Buat Fungsi GET User Location</p> <p>🕒 May 9</p>	<p>Urgent</p> <p>Buat Fungsi GET Detail Faskes</p> <p>🕒 May 20</p>
<p>Medium</p> <p>Slicing UI Fasilitas Kesehatan</p> <p>🕒 May 14</p>	

Gambar 3.11 *Breakdown user story* pertama

Kemudian dari hasil *breakdown user story* kedua yang mana merupakan fitur laporan kejadian, didapati 3 buah backlog yang dapat dilihat pada gambar 3.12.

User Story ...

Sebagai user, saya ingin melakukan pelaporan kejadian darurat yang saya temukan atau saya alami agar segera mendapatkan bantuan.

Backlog (Work On Wait)

<p>Medium</p> <p>Membuat Desain Halaman Laporan Kejadian</p> <p>🕒 Apr 25</p>	<p>Medium</p> <p>Buat Fungsi POST Laporan Kejadian</p> <p>🕒 May 29</p>
<p>Medium</p> <p>Slicing UI Laporan Kejadian</p> <p>🕒 May 25</p>	

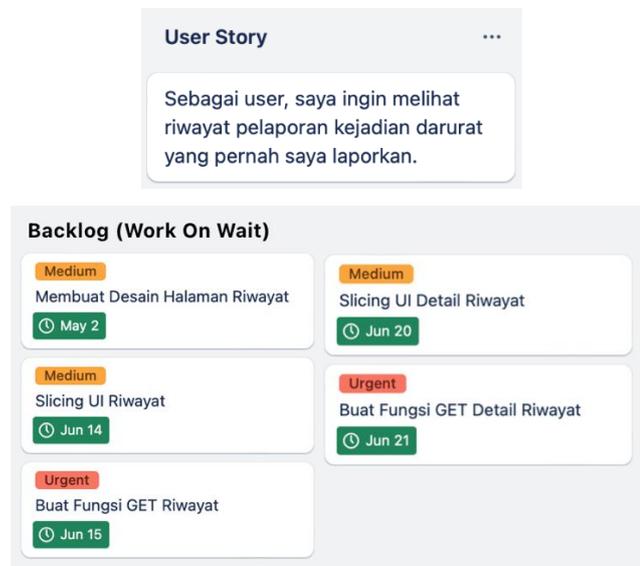
Gambar 3.12 *Breakdown user story* kedua

Pada *user story* ketiga yang mana merupakan fitur laporan SOS, didapati 4 buah backlog yang dapat dilihat pada gambar 3.13.



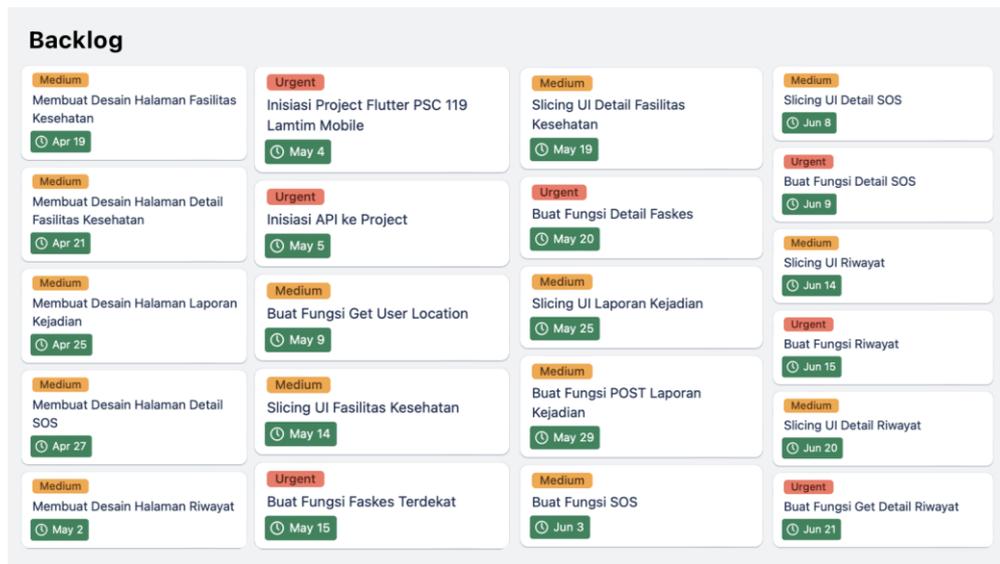
Gambar 3.13 *Breakdown user story* ketiga

Selanjutnya pada *breakdown user story* yang keempat telah menghasilkan 5 *backlog* yang dapat dilihat pada gambar 3.14.



Gambar 3.14 *Breakdown user story* keempat

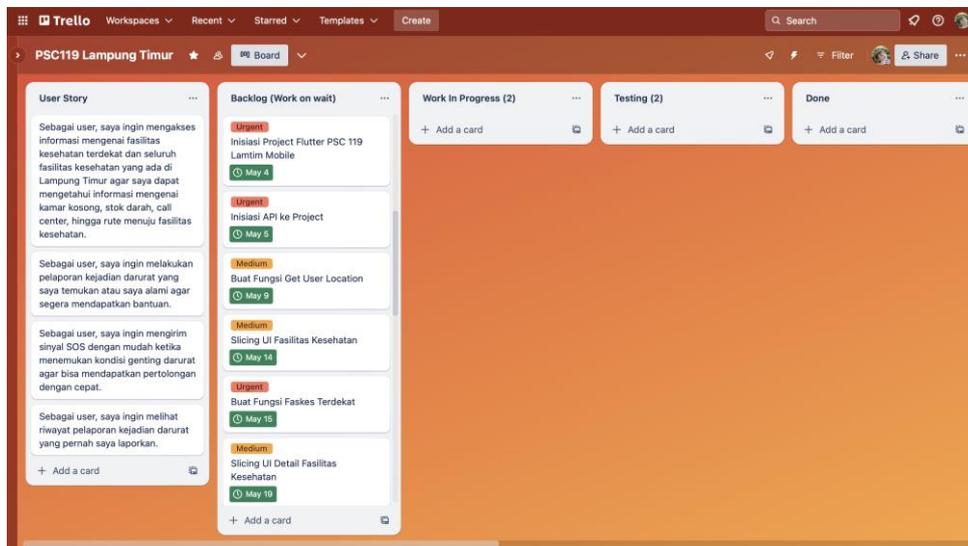
Sehingga keseluruhan *Backlog* dari pengembangan aplikasi PSC 119 Lampung Timur dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Backlog* aplikasi PSC 119 Lampung Timur

f. *Kanban Board*

Dari *backlog* yang telah ditentukan sebelumnya, dapat dirancang *taskcard* pada papan Kanban berdasarkan pada kolom-kolom tertentu sesuai dengan proses tahapan pengembangan aplikasi. Papan Kanban yang digunakan pada penelitian ini dapat dilihat pada gambar 3.16.



Gambar 3.16 Papan Kanban yang digunakan pada penelitian

Dari dapat gambar terlihat bahwa papan Kanban memiliki beberapa kolom yang merupakan gambaran *progress* pengembangan, seperti:

1. *User Story*

Kolom ini terdiri dari *task card* yang berisi *user story* PSC 119 Lampung Timur yang telah ditentukan.

2. *Backlog (Work On Wait)*

Kolom *backlog* berisi tugas-tugas yang perlu dikerjakan pada pengembangan aplikasi PSC 119 Lampung Timur.

3. *Work In Progress*

Kolom *Work In Progress* menampung *taskcard* yang sedang dalam masa pengerjaan atau pengembangan aplikasi. Dalam hal ini terdapat tanda (2) yang menandakan bahwa *Limit WIP* berjumlah 2 buah *taskcard*.

4. *Testing*

Kolom *testing* berisi tugas-tugas atau *taskcard* yang telah selesai dikerjakan dan sedang memasuki tahap pengujian apakah fungsi telah bekerja dengan baik.

5. *Done*

Kolom *done* merupakan kolom yang menampung *taskcard* yang telah selesai dikerjakan, lulus *testing*, dan dinyatakan bekerja dengan baik.

g. *Limit WIP*

Limit WIP merupakan *limit* atau batas jumlah *taskcard* yang boleh ada di kolom *work in progress* pada papan Kanban. *Limit* ini berfungsi untuk membatasi tugas yang dikerjakan pada proses pengembangan agar dapat terfokus pada pekerjaan yang ada di periode tertentu. Pada penelitian, *limit WIP* ditentukan pada 2 buah *taskcard*. Pembatasan ini ditentukan dengan mempertimbangkan dari kemampuan pengembang aplikasi dalam menyelesaikan suatu tugas pada satu alur kerja.

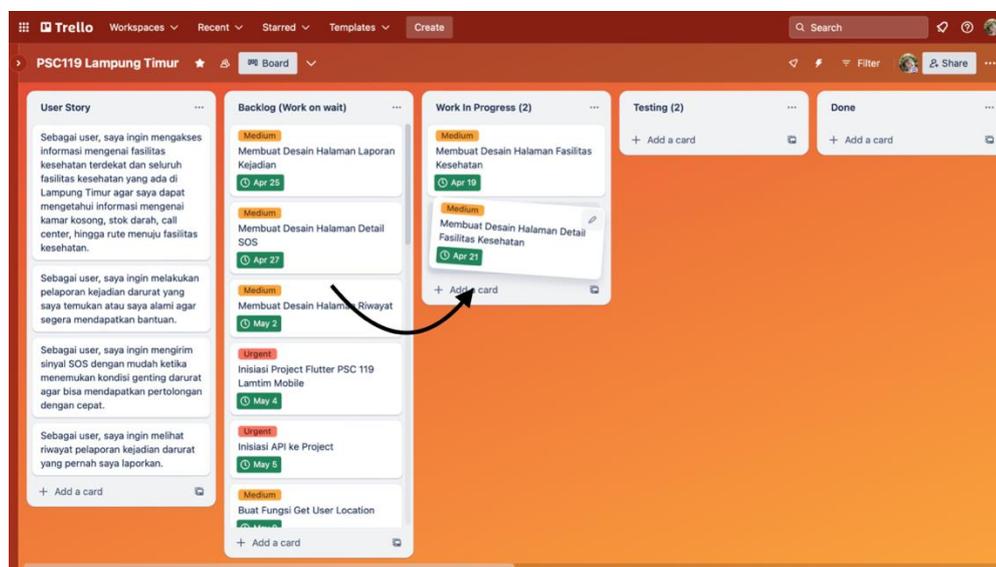
3.5.2.2 Tahap Perancangan Antarmuka

Pada tahap ini dilakukan perancangan antarmuka aplikasi yang dikembangkan. Melalui rancangan antarmuka, aplikasi dapat tergambar dengan

baik fitur dan fungsionalitasnya. Perancangan dibuat berdasarkan pada kebutuhan dan syarat yang telah ditentukan. Pengerjaan dari rancangan antarmuka sendiri dilakukan menggunakan *website Figma* yang merupakan sebuah *tools* aplikasi desain untuk membuat prototipe sistem. Aplikasi dirancang dengan memiliki 11 fitur utama. Fitur-fitur tersebut meliputi login, register, laporan kejadian, faskes terdekat, detail faskes, kirim laporan sos, kontak darurat, riwayat laporan, panduan, edit profil, change password, dan logout.

3.5.3 Tahap Pengembangan Aplikasi

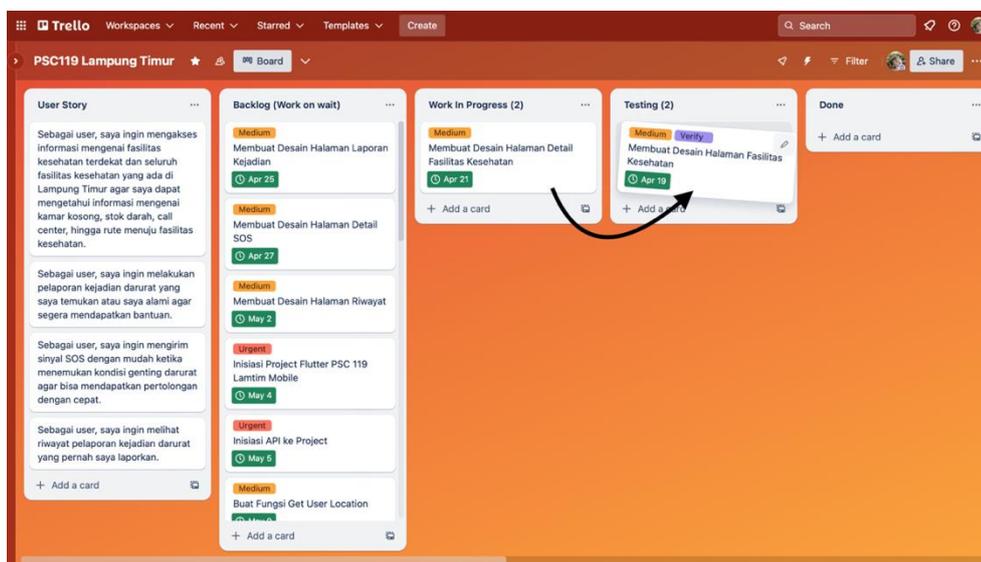
Metode penelitian yang digunakan adalah dengan *Kanban Board* yang memiliki beberapa *taskcard* berisi backlog tugas yang harus dikerjakan. Visualisasi alur kerja adalah dengan cara menggeser *taskcard* ke kolom-kolom pada proses yang sedang dijalankan. Apabila kolom WIP kosong, dapat menarik 2 buah *taskcard* dari kolom WOW dan memulai proses pengerjaan tugas sesuai dengan deskripsi kartu. Apabila pada kolom WIP terdapat 1 buah *taskcard*, maka dapat mengambil 1 buah *taskcard* dari kolom WOW sehingga *Limit WIP* terpenuhi dan proses pengerjaan dapat berlangsung efektif. Berikut gambaran dari visualisasi alur kerja dapat dilihat pada Gambar 3.17.



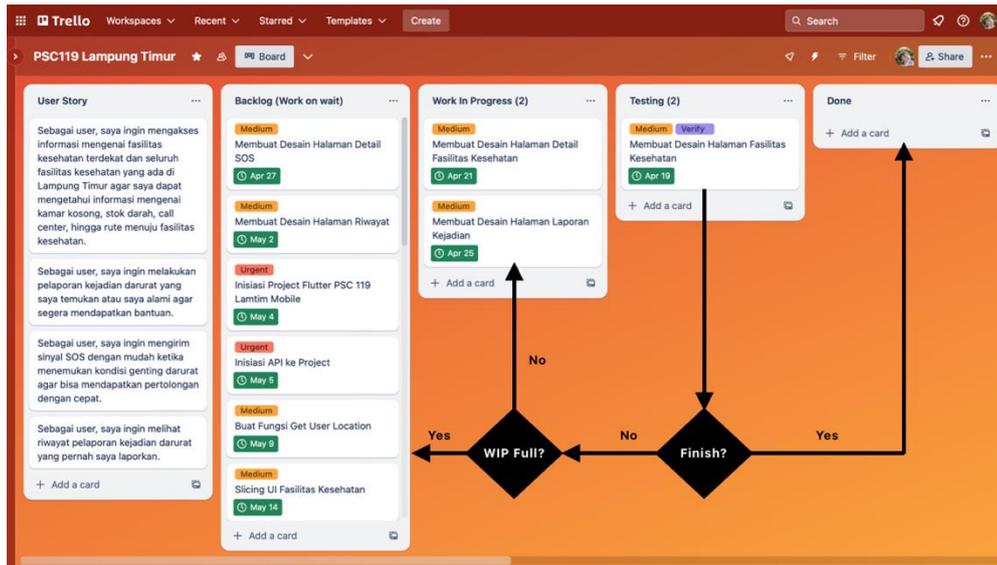
Gambar 3.17 *Kanban Board* PSC 119 Lampung Timur *Flow WIP*

Dari gambar yang tertera, terdapat satu tahap dimana pemindahan 1 *taskcard* dari kolom WOW menuju ke WIP. Dalam hal ini, artinya *taskcard* tersebut sedang berada dalam proses pengembangan. Kolom WIP memiliki *limit* berjumlah 2 buah kartu yang artinya apabila *limit* pada kolom telah terpenuhi, maka tidak dapat menambah *taskcard* lagi ke kolom WIP. Sehingga sisa *backlog* harus menunggu pada kolom WOW sampai proses pengerjaan *task* di kolom WIP telah selesai dan *limit* belum terpenuhi.

Selanjutnya, terdapat tahap pemindahan *taskcard* dari kolom WIP menuju ke kolom *testing*. Dalam hal ini mengartikan bahwa berarti *task* telah selesai dalam pengembangan dan memasuki tahap pengujian. *Taskcard* yang memasuki tahap pengujian akan memiliki label *verify* berwarna ungu sebagai penanda. Setelah *taskcard* tersebut dipindahkan, kolom WIP bersisa 1 buah *taskcard* yang berarti dapat ditambahkan 1 *taskcard* lagi dari kolom WOW untuk memasuki tahap pengembangan.

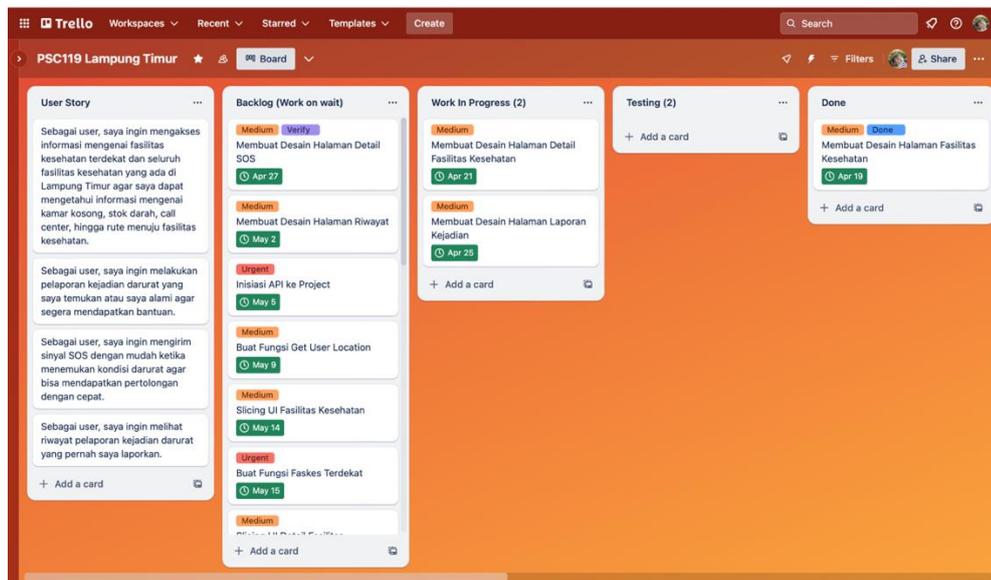


Gambar 3.18 Kanban Board PSC 119 Lampung Timur Flow Testing



Gambar 3.19 *Kanban Board* PSC 119 Lampung Timur *Flow* kembali ke WIP

Pada Gambar 3.19 tertera tampilan dari *Kanban board* pada tahap ketika pengujian telah selesai. Hasil dari pengujian bisa berupa selesai maupun perlu perbaikan. Apabila tugas tersebut telah dinyatakan selesai dan berfungsi dengan baik, maka *taskcard* dapat dipindahkan menuju ke kolom *done*. Namun jika hasil menyatakan perlu diadakan perbaikan, maka *taskcard* perlu dipindahkan lagi pada proses sebelumnya. Dalam hal ini jika limit WIP telah tercapai, *taskcard* harus menunggu lagi di kolom WOW.



Gambar 3.20 *Kanban Board* PSC 119 Lampung Timur *Flow Done*

3.6 Tahap Testing

Tahap *testing* bertujuan untuk menguji fitur yang telah dikerjakan apakah hasil yang diberikan telah sesuai dengan hasil yang diharapkan. Pada tahap ini pengujian bermaksud untuk menemukan kesalahan-kesalahan baik pada fungsionalitas maupun non-fungsionalitas. Teknik yang dilakukan untuk menguji sistem adalah *blackbox testing* dengan metode *decision table*.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Mengembangkan sebuah aplikasi Android PSC 119 Lampung Timur yang menyediakan pelayanan kepada masyarakat terkait pemenuhan *Public Safety Center* untuk daerah Lampung Timur dengan parameter keberhasilan adalah seluruh fitur telah berhasil dikembangkan yaitu fitur fasilitas kesehatan, laporan kejadian, laporan SOS, dan riwayat laporan.
2. Pengembangan dengan metode Kanban telah berjalan sesuai dengan tahapan dimana pada tiap *flow* pengerjaan terdiri dari dua buah *taskcard* yang prosesnya melalui *Work On Wait*, *Work In Progress*, *Testing*, dan *Done*. Aplikasi PSC 119 Lampung Timur telah sesuai dengan spesifikasi yang diinginkan dengan menggunakan 4 *user story* yang kemudian dipecah menjadi 19 *backlog* yang dituliskan pada sebuah *taskcard* sebagai acuan dari proses pengembangan aplikasi.
3. Pengujian dilakukan menggunakan *blackbox testing* dan teknik *decision table* dengan total 14 skenario pengujian. Dari proses uji, didapati bahwa hasil uji sesuai dengan hasil yang diharapkan. Aplikasi telah berhasil berjalan dengan baik dan memenuhi kebutuhan dari PSC 119 Lampung Timur.

5.2 Saran

Saran yang dapat diberikan untuk melanjutkan penelitian berdasarkan dari hasil penelitian adalah sebagai berikut:

1. Mengembangkan aplikasi lebih lanjut dengan memberikan fitur panduan pertolongan pertama pada kondisi gawat darurat.
2. Menambahkan fitur notifikasi untuk memberikan perkembangan ter-*update* dari riwayat laporan yang pernah dikirim oleh *user*.
3. Mengembangkan aplikasi pada versi iOS sehingga dapat menjangkau *user* tidak hanya pada kalangan pengguna Android namun juga pada pengguna iPhone dan iPad.
4. Melakukan *update* baik *library* maupun *dependency* yang dipakai pada aplikasi sehingga sistem selalu terbaharukan dan minim *error*.

DAFTAR PUSTAKA

- [1] Kemenkes RI, “Peraturan Menteri Kesehatan Republik Indonesia Nomor 19 Tahun 2016 Tentang Sistem Penanggulangan Gawat Darurat Terpadu,” 2016.
- [2] Anonim, “Geger Warga Klaten Meninggal Usai Tak Dipinjami Ambulans,” May 01, 2022. <https://www.detik.com/jateng/berita/d-6059369/geger-warga-klaten-meninggal-usai-tak-dipinjami-ambulans> (accessed Mar. 30, 2023).
- [3] BPS Lampung Timur, “Jumlah Penduduk Lampung Timur Berdasar Kecamatan (Jiwa), 2018-2020,” 2020. <https://lampungtimurkab.bps.go.id/indicator/12/33/1/jumlah-penduduk-lampung-timur-berdasar-kecamatan.html> (accessed May 26, 2023).
- [4] We Are Social & Hootsuite, “Digital Data Indonesia 2022,” 2022. <https://datareportal.com/reports/digital-2022-indonesia> (accessed May 26, 2023).
- [5] Kemenkes RI, “Peraturan Menteri Kesehatan Republik Indonesia Nomor 47 Tahun 2018 Tentang Pelayanan Kegawatdaruratan,” 2018.
- [6] R. Payne, “*Beginning App Development with Flutter (Create Cross-Platform Mobile Apps)*,” Apress, 2019.
- [7] D. Griffiths and D. Griffiths, “*Head First Android Development*,” O’Reily Media, Inc., 2021.
- [8] E. Brechner, “*Praise for Agile Project Management with Kanban*,” Microsoft Press, 2015.
- [9] D. J. Anderson and A. Charmichael, “*Essential Kanban Condensed*,” Lean-Kanban University, 2016.
- [10] Atlassian, “*Getting Started With Trello*,” 2023. <https://trello.com/guide> (accessed Jun. 20, 2023).

- [11] B. De, “*An Architect’s Guide to Developing and Managing APIs for Your Organization*,” Apress, 2017. doi: 10.1007/978-1-4842-1305-6.
- [12] T. Marrs, “*JSON at Work : Practical Data Integration for the Web*,” O’Reilly Media, Inc., 2017.
- [13] E. Kaplan and C. Hegarty, “*Understanding GPS Principles and Applications Second Edition*,” Artech House on Demand, 2006.
- [14] Google, “*Google Maps Platform Documentation*,” 2023. <https://developers.google.com/maps/documentation> (accessed Jun 21, 2023).
- [15] S. Desikan and G. Ramesh, “*Software Testing : Principles and Practice*,” Pearson Education India, 2007.
- [16] J. A. Maghanoy, “*iReportMo: An Emergency Report Android Mobile Application for Metro Manila*,” 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT), Dec 2019. doi: 10.1109/ICAIT.2019.8935906
- [17] R. Simamora, I. Made, and A. Suyadnya, “Rancang Bangun Aplikasi Ambulance Online Berbasis Android,” *SINTECH (Science and Information Technology) Journal*, vol. 3, no. 2, pp. 118-129, Oct 2020. doi: <https://doi.org/10.31598/sintechjournal.v3i2.633>
- [18] A. Shukla, B. Solanki, and K. Panchal, “*Smart Ambulance Service System*,” *IOSR Journal of Computer Engineering (IOSR-JCE)* vol. 22, no. 2, p. PP, Mar 2020, doi: 10.9790/0661-2202020106.
- [19] N. Faizah, N. Santoso, and A. A. Soebroto, “Pengembangan Sistem Aplikasi Manajemen Proyek menggunakan Kanban Framework,” *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 3 no. 10, pp. 9747–9754. Jan 2020.
- [20] M. A. Ilmi, F. Pradana, and W. Hayuhardhika Nugraha Putra, “*Software Project Management Systems Using Kanban Method in the CV. Primavisi Globalindo*,” *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 4, no. 2, pp. 215–231, Aug. 2020, doi: 10.29407/intensif.v4i2.14320.
- [21] B. A. Akbar, “Perancangan Sistem Informasi Akademik Menggunakan Metode Kanban,” *Information Management For Educators And*

- Professionals : Journal of Information Management* vol. 5, no. 2, pp. 33–42, 2021. doi: <https://doi.org/10.51211/imbi.v5i2.1546>
- [22] S. Aisa, “Aplikasi Pencarian Bengkel Aktif Dengan Google Maps Api Berbasis Web,” *Doubleclick: Journal of Computer and Information Technology* vol. 4 no. 2, pp. 61-69, February 2021. doi: <http://doi.org/10.25273/doubleclick.v4i2.8001>.
- [23] N. Nurdin, S. S. Pettalongi, and M. Mangasing, “Implementation of Geographic Information System Base On Google Maps API to Determine Bidikmisi Scholarship Recipient Distribution in Central Sulawesi Indonesia,” *Journal of Humanities and Social Sciences Studies* vol. 3 no. 12, pp. 38-53, 2021, doi: 10.32996/jhsss.
- [24] F. Zakiyah Rahmanti, O. Ayu Permata, K. Amiroh, P. Tobianto Daely, A. Ittaqullah, and D. Bagus Saputro, “An Improvement Using Global Positioning System (GPS) and Cloud Firestore for Integration of Information System in Surabaya Public Transportation,” *EDUTECH : Journal of Education And Technology*, vol. 5 no. 4, pp. 894-909, 2022. doi: <https://doi.org/10.29062/edu.v5i4.294>
- [25] G. Indah Marthasari *et al.*, “Pengujian Website Infotech Menggunakan Teknik Black-Box Decision Table,” *Jurnal Informatika Universitas Pamulan*, vol. 7, no. 1, pp. 115–119, doi: 10.32493/informatika.v7i1.17315.

