

**PENGEMBANGAN APLIKASI BERBASIS MOBILE UNTUK
PELAPORAN ANCAMAN KEJAHATAN DI PERUMAHAN
PANORAMA CIBODAS**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Syarief Noor Permadi
NIM: 165150200111034



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PERSETUJUAN

**PENGEMBANGAN APLIKASI BERBASIS MOBILE UNTUK PELAPORAN ANCAMAN
KEJAHATAN DI PERUMAHAN PANORAMA CIBODAS**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

**Disusun Oleh :
Syarief Noor Permadi
NIM: 165150200111034**

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Issa Arwani, S.Kom., M.Sc.
NIP: 19830922 201212 1 003

Muhammad Aminul Akbar, S.Kom., M.T.
NIK: 201607 891013 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Desember 2019

Syarief Noor Permadi

NIM: 165150200111034

PRAKATA

Puji kehadirat Allah SWT yang telah melimpahkan berkah dan rahmat-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Pengembangan Aplikasi Berbasis Mobile Untuk Pelaporan Ancaman Kejahatan di Perumahan Panorama Cibodas”.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Bapak Issa Arwani, S.Kom., M.Sc. dan bapak Muhammad Aminul Akbar, S.Kom., M.T. selaku pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika,
4. Bapak Adhitya Bhawiyuga, S.Kom., M.Sc. selaku dosen Penasihat Akademik yang selalu memberikan nasihat kepada penulis selama menempuh masa studi,
5. Bapak Supoyo dan Ibu Siti Mahmudah selaku orang tua penulis, serta seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
6. Bapak Sanduki selaku ketua RW 12 di Perumahan Panorama Cibodas beserta seluruh pengurus RW 12 dan seluruh warga Panorama Cibodas yang telah membantu proses penelitian skripsi,
7. Teman-teman ROAD to IF JUARA yang selalu memberikan dukungan dan semangat untuk menyelesaikan skripsi ini,
8. Insan Nurzaman Bangga Adi Pratama dan Zain Fikri Hanastyono yang telah menjadi teman diskusi dan sama-sama berjuang dalam mengerjakan skripsi,
9. Lailathul Fitri sebagai penyemangat penulis dalam menyelesaikan skripsi serta membantu pengujian aplikasi,
10. Seluruh pihak yang tidak dapat ditulis satu per satu yang telah memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyusunan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini, masih banyak kekurangan, sehingga penulis mengharapkan saran dan kritik yang membangun.

Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi seluruh pihak yang menggunakannya.

Malang, 18 Desember 2019

Penulis

syariefnoorp@gmail.com

ABSTRAK

Syarief Noor Permadi, Pengembangan Aplikasi Berbasis *Mobile* Untuk Pelaporan Ancaman Kejahatan di Perumahan Panorama Cibodas

Pembimbing: Issa Arwani, S.Kom., M.Sc. dan Muhammad Aminul Akbar, S.Kom., M.T.

Perumahan Panorama Cibodas merupakan griya hunian tempat tinggal penulis yang bertempat di Jalan Prabu Kian Santang, Kelurahan Sangiang Jaya, Kecamatan Periuk, Kota Tangerang. Berdasarkan pengamatan langsung yang dilakukan oleh penulis, pada Perumahan Panorama Cibodas memiliki 3 petugas keamanan yang melakukan tugas jaga secara bergantian dalam *3 shift*, sehingga pada 1 *shift* hanya terdapat 1 petugas keamanan yang berjaga. Jumlah petugas keamanan yang berjumlah 1 orang tentunya tidak akan dapat melakukan pengawasan secara maksimal. Jika terjadi tindak kejahatan yang lokasinya jauh dari pos keamanan, maka tidak dapat langsung ditangani oleh petugas keamanan. Sehingga untuk menjaga keamanan di lingkungan Perumahan Panorama Cibodas memerlukan bantuan dari warga Perumahan Panorama Cibodas. Pada penelitian ini penulis melakukan pengembangan aplikasi SKIP (Sistem Keamanan di Perumahan) untuk pelaporan ancaman kejahatan di lingkungan Panorama Cibodas dengan metode pengembangan *Prototyping* untuk membantu warga menjaga keamanan di lingkungan Perumahan Panorama Cibodas dari lingkungan rumah masing-masing. Metode pengembangan *Prototyping* dipilih karena penulis sebelumnya mengusulkan kebutuhan fungsional aplikasi dalam bentuk *prototype* lalu ditunjukkan kepada warga untuk mendapatkan *feedback* sampai mendapatkan kebutuhan fungsional akhir. Pengujian yang dilakukan pada aplikasi SKIP (Sistem Keamanan di Perumahan) adalah pengujian validasi (*blackbox*), pengujian *usability* dan pengujian *compatibility*. Pada pengujian validasi menghasilkan tingkat keberhasilan 100%. Pengujian *usability* menghasilkan *success rate* penggerjaan task sebesar 100% dan memiliki nilai akhir *System Usability Scale* (SUS) sebesar 80 yang termasuk ke dalam *grade B* dan *adjectiva rating* “good”. Pada pengujian *compatibility* aplikasi SKIP dapat berjalan dengan baik pada pengujian perangkat secara manual dan pengujian secara otomatis dengan Firebase Test Lab pada versi sistem operasi android minimum KitKat 4.4 dengan tingkatan API 19 dan maksimum Pie 9.0 dengan tingkatan API 28.

Kata kunci: pelaporan, Android, Firebase, *prototyping*, aplikasi berbasis *mobile*, *model view presenter* (MVP).

ABSTRACT

Syarief Noor Permadi, Mobile Based Application Development for Reporting Crime Threats in Panorama Cibodas Residence

Supervisors: Issa Arwani, S.Kom., M.Sc. and Muhammad Aminul Akbar, S.Kom., M.T.

Panorama Cibodas Residence is the author's residence located at Jalan Prabu Kian Santang, Kelurahan Sangiang Jaya, Kecamatan Periuk, Kota Tangerang. Based on direct observations by the author, Panorama Cibodas Residence has 3 security officers who carry out shift tasks in 3 shift, so that in 1 shift there is only 1 security officer on guard. The only 1 person security certainly will not be able to carry out whole residential area. If a crime occurs far from a security post, then it can't be dealt directly by security officers. So to maintain security at Panorama Cibodas Residence environment need help from the residents. In this research, the authors developed SKIP (Sistem Keamanan di Perumahan) application to report threats of crime in the Panorama Cibodas environment by developing Prototyping method to help residents maintain security in the Panorama Cibodas Residence. The prototype development method was chosen because the author's previously proposed the functional requirements of the application in the form of a prototype and then were shown to residents to get feedback until they got the final functional needs. Tests conducted on the SKIP (Sistem Keamanan di Perumahan) application is validation testing (blackbox), usability testing and compatibility testing. In validation testing results is 100% success rate. In usability testing results is 100% success rate for task execution by the respondent and has a final score on the System Usability Scale (SUS) 80 which falls into the grade B and adjectiva rating "good" category. In compatibility testing, SKIP application can run well on manual testing device and in automatically testing with Firebase Test Lab on the Android operating system version of minimum KitKat 4.4 with level API 19 and maximum Pie 9.0 with level API 28.

Keywords: reporting, Android, Firebase, prototyping, mobile application, model view presenter (MVP).

DAFTAR ISI

PERSETUJUANii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR PERSAMAAN.....	xvii
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Android	6
2.3 Firebase.....	7
2.3.1 Realtime Database	7
2.3.2 Cloud Messaging	7
2.4 JSON	7
2.5 GPS (<i>Global Positioning System</i>).....	7
2.6 Google Map.....	8
2.7 UML (<i>Unified Modelling Language</i>).....	8
2.8 <i>Model View Presenter (MVP)</i>	8
2.9 Prototyping	8
2.10 Teknik Pengujian	10

2.10.1 Pengujian Validasi	10
2.10.2 Pengujian <i>Usability</i>	10
2.10.3 Pengujian <i>Compatibility</i>	11
BAB 3 METODOLOGI	13
3.1 Studi Literatur	14
3.2 Analisis Kebutuhan	15
3.3 <i>Prototyping</i>	15
3.4 Evaluasi <i>Prototype</i>	15
3.5 Perancangan	15
3.6 Implementasi	16
3.7 Pengujian dan Analisa Hasil	16
3.7.1 Pengujian Validasi	16
3.7.2 Pengujian <i>Usability</i>	16
3.7.3 Pengujian <i>Compatibility</i>	17
3.8 Pengambilan Kesimpulan dan Saran	18
BAB 4 ANALISIS KEBUTUHAN	19
4.1 Gambaran Umum Aplikasi	19
4.2 Analisis Kebutuhan	19
4.2.1 Kebutuhan Fungsional Awal.....	19
4.2.2 Hasil Wawancara Uji Coba <i>Prototype</i>	20
4.3 Identifikasi Aktor.....	21
4.4 Kebutuhan Fungsional	22
4.5 Kebutuhan Non Fungsional.....	23
4.6 <i>Use Case Diagram</i>	24
4.7 <i>Use Case Scenario</i>	24
4.8 Analisis Data.....	32
4.9 Tampilan Antarmuka <i>Final Prototype</i>	32
BAB 5 PERANCANGAN.....	36
5.1 Perancangan Arsitektur Sistem.....	36
5.2 Perancangan <i>Sequence Diagram</i>	37
5.2.1 <i>Sequence Diagram</i> Buat Laporan.....	38
5.2.2 <i>Sequence Diagram</i> Lihat Laporan	39

5.2.3 <i>Sequence Diagram</i> Panggilan Darurat	40
5.2.4 <i>Sequence Diagram</i> Mengirim Pesan Darurat.....	41
5.3 Perancangan <i>Class Diagram</i>	42
5.4 <i>Physical Data Model</i>	44
5.5 Perancangan Antarmuka	44
5.5.1 Tampilan <i>Login</i>	45
5.5.2 Tampilan <i>Dashboard</i>	46
5.5.3 Tampilan Masukan Judul Laporan	47
5.5.4 Tampilan Perancangan Hasil Pengecekan Judul Laporan	48
5.5.5 Tampilan Perancangan Submit Laporan	49
5.5.6 Tampilan Perancangan Daftar Laporan	50
5.5.7 Tampilan Perancangan Detail Laporan	51
5.5.8 Tampilan Perancangan Panggilan Darurat.....	52
5.5.9 Tampilan Perancangan Panduan Aplikasi	53
5.5.10 Tampilan Perancangan Detail Panduan Aplikasi.....	54
5.5.11 Tampilan Perancangan Dialog Konfirmasi	55
5.5.12 Tampilan Perancangan Dialog Waktu Tunggu	56
5.5.13 Tampilan Perancangan Pesan Darurat.....	57
5.5.14 Tampilan Perancangan Tentang Aplikasi	58
5.6 Perancangan <i>Screenflow Diagram</i>	58
5.6.1 <i>Screenflow Diagram Login</i>	59
5.6.2 <i>Screenflow Diagram</i> Buat Laporan	59
5.6.3 <i>Screenflow Diagram</i> Lihat Laporan	60
5.6.4 <i>Screenflow Diagram</i> Panggilan Darurat	61
5.6.5 <i>Screenflow Diagram</i> Panduan Aplikasi	62
5.6.6 Screenflow Diagram Mengirim dan Menerima Pesan Darurat	62
BAB 6 IMPLEMENTASI	64
6.1 Spesifikasi Sistem	64
6.1.1 Spesifikasi Perangkat Keras.....	64
6.1.2 Spesifikasi Perangkat Lunak	64
6.2 Batasan-batasan Implementasi	65

6.3 Implementasi Data.....	65
6.4 Implementasi Algoritme	69
6.4.1 Algoritme membuat laporan.....	69
6.4.2 Algoritme Melihat Laporan	73
6.4.3 Algoritme Panggilan Darurat.....	74
6.4.4 Algoritme Mengirim Pesan Darurat.....	75
BAB 7 PENGUJIAN DAN ANALISIS.....	78
7.1 Pengujian Validasi (<i>Blackbox</i>)	78
7.1.1 Pengujian Validasi <i>Login</i>	78
7.1.2 Pengujian Validasi Buat Laporan.....	79
7.1.3 Pengujian Validasi Lihat Laporan	81
7.1.4 Pengujian Validasi Panggilan Darurat	82
7.1.5 Pengujian Validasi Panduan Aplikasi.....	83
7.1.6 Pengujian Validasi Mengirim Pesan Darurat.....	84
7.1.7 Pengujian Validasi Menerima Pesan Darurat	85
7.1.8 Pengujian Validasi Tentang Aplikasi.....	86
7.1.9 Analisis Hasil Pengujian Validasi	86
7.2 Pengujian <i>Usability</i>	87
7.2.1 Analisis Hasil Pengujian <i>Usability</i>	88
7.3 Pengujian <i>Compatibility</i>	91
7.3.1 Analisis Pengujian <i>Compatibility</i>	92
BAB 8 PENUTUP	93
8.1 Kesimpulan.....	93
8.2 Saran	94
Daftar referensi	95
LAMPIRAN A HASIL WAWANCARA UJI <i>PROTOTYPE</i>	97
LAMPIRAN B HASIL PENGUJIAN <i>USABILITY</i>	102

DAFTAR TABEL

Tabel 2.1 Interpretasi nilai akhir SUS <i>Score</i>	11
Tabel 3.1 Keterangan skor skala likert	16
Tabel 4.1 Kebutuhan fungsional awal.....	20
Tabel 4.2 Daftar responden uji <i>prototype</i>	20
Tabel 4.3 Kesimpulan hasil wawancara uji <i>prototype</i>	21
Tabel 4.4 Identifikasi aktor.....	21
Tabel 4.5 Kebutuhan fungsional	22
Tabel 4.6 Kebutuhan non fungsional	23
Tabel 4.7 Skenario <i>Login</i>	25
Tabel 4.8 Skenario Buat Laporan	25
Tabel 4.9 Skenario Lihat Laporan.....	27
Tabel 4.10 Skenario Panggilan Darurat.....	28
Tabel 4.11 Skenario Panduan Aplikasi	29
Tabel 4.12 Skenario Mengirim Pesan Darurat	29
Tabel 4.13 Skenario Menerima Pesan Darurat	31
Tabel 4.14 Skenario Tentang Aplikasi	31
Tabel 5.1 Informasi elemen pada tampilan <i>login</i>	45
Tabel 5.2 Informasi elemen pada tampilan <i>dashboard</i>	46
Tabel 5.3 Informasi elemen pada tampilan masukan judul laporan	47
Tabel 5.4 Informasi elemen pada tampilan hasil pengecekan judul laporan	48
Tabel 5.5 Informasi elemen pada tampilan submit laporan.....	49
Tabel 5.6 Informasi elemen pada tampilan daftar laporan	50
Tabel 5.7 Informasi elemen pada tampilan detail laporan.....	51
Tabel 5.8 Informasi elemen pada tampilan panggilan darurat.....	52
Tabel 5.9 Informasi elemen pada tampilan panduan aplikasi	53
Tabel 5.10 Informasi elemen pada tampilan detail panduan aplikasi.....	54
Tabel 5.11 Informasi elemen pada tampilan dialog konfirmasi	55
Tabel 5.12 Informasi elemen pada tampilan dialog waktu tunggu	56
Tabel 5.13 Informasi elemen pada tampilan pesan darurat.....	57
Tabel 5.14 Informasi elemen pada tampilan tentang aplikasi.....	58

Tabel 5.15 Keterangan <i>screenflow login</i>	59
Tabel 5.16 Keterangan <i>screenflow buat laporan</i>	59
Tabel 5.17 Keterangan <i>screenflow lihat laporan</i>	60
Tabel 5.18 Keterangan <i>screenflow panggilan darurat</i>	61
Tabel 5.19 Keterangan <i>screenflow panduan aplikasi</i>	62
Tabel 5.20 Keterangan <i>screenflow mengirim dan menerima pesan darurat</i>	62
Tabel 6.1 Spesifikasi perangkat keras	64
Tabel 6.2 Spesifikasi perangkat lunak	65
Tabel 6.3 <i>Source code method cekJudul()</i>	70
Tabel 6.4 Penjelasan <i>source code method addLaporan()</i>	70
Tabel 6.5 <i>Source code method addLaporan()</i>	71
Tabel 6.6 Penjelasan <i>source code method addLaporan()</i>	72
Tabel 6.7 <i>Source code method fetchLaporan()</i>	73
Tabel 6.8 Penjelasan <i>source code method fetchLaporan()</i>	74
Tabel 6.9 <i>Source code method fetchNomorDarurat()</i>	74
Tabel 6.10 Penjelasan <i>source code method fetchNomorDarurat()</i>	75
Tabel 6.11 <i>Source code method pushPesanDarurat()</i>	75
Tabel 6.12 Penjelasan <i>source code method pushPesanDarurat()</i>	77
Tabel 7.1 Pengujian validasi <i>login</i>	78
Tabel 7.2 Pengujian validasi buat laporan	79
Tabel 7.3 Pengujian validasi lihat laporan.....	81
Tabel 7.4 Pengujian validasi panggilan darurat	83
Tabel 7.5 Pengujian validasi panduan aplikasi.....	83
Tabel 7.6 Pengujian validasi mengirim pesan darurat.....	84
Tabel 7.7 Pengujian validasi menerima pesan darurat.....	85
Tabel 7.8 Pengujian validasi tentang aplikasi	86
Tabel 7.9 <i>Task Pengujian Usability</i>	87
Tabel 7.10 Daftar pernyataan kuesioner SUS	88
Tabel 7.11 Hasil penggerjaan <i>task</i> oleh responden.....	88
Tabel 7.12 Hasil pengisian kuesioner SUS.....	89
Tabel 7.13 Perhitungan akumulasi skor SUS.....	90
Tabel 7.14 Hasil pengujian <i>compatibility</i> secara manual.....	91

Tabel 7.15 Hasil pengujian compatibility menggunakan Firebase Test Lab 92

DAFTAR GAMBAR

Gambar 2.1 Lifecycle Android	6
Gambar 2.2 <i>Prototyping Paradigm</i>	9
Gambar 3.1 Diagram alir metodologi	13
Gambar 4.1 <i>Use case diagram</i>	24
Gambar 4.2 Tampilan antarmuka <i>final prototype</i> (bagian 1)	33
Gambar 4.3 Tampilan antarmuka <i>final prototype</i> (bagian 2)	33
Gambar 4.4 Tampilan antarmuka <i>final prototype</i> (bagian 3)	34
Gambar 4.5 Tampilan antarmuka <i>final prototype</i> (bagian 4)	34
Gambar 4.6 Tampilan antarmuka <i>final prototype</i> (bagian 5)	35
Gambar 4.7 Tampilan antarmuka <i>final prototype</i> (bagian 6)	35
Gambar 5.1 Skema Arsitektur MVP	36
Gambar 5.2 Arsitektur sistem	37
Gambar 5.3 <i>Sequence diagram</i> buat laporan	38
Gambar 5.4 <i>Sequence diagram</i> lihat laporan	39
Gambar 5.5 <i>Sequence diagram</i> panggilan darurat	40
Gambar 5.6 <i>Sequence diagram</i> mengirim pesan darurat.....	41
Gambar 5.7 <i>Class diagram</i>	43
Gambar 5.8 <i>Physical data model</i>	44
Gambar 5.9 Tampilan perancangan <i>login</i>	45
Gambar 5.10 Tampilan perancangan <i>dashboard</i>	46
Gambar 5.11 Tampilan perancangan masukan judul laporan.....	47
Gambar 5.12 Tampilan perancangan hasil pengecekan judul laporan	48
Gambar 5.13 Tampilan perancangan submit laporan	49
Gambar 5.14 Tampilan perancangan daftar laporan.....	50
Gambar 5.15 Tampilan perancangan detail laporan	51
Gambar 5.16 Tampilan perancangan panggilan darurat	52
Gambar 5.17 Tampilan perancangan panduan aplikasi.....	53
Gambar 5.18 Tampilan perancangan detail panduan aplikasi	54
Gambar 5.19 Tampilan perancangan dialog konfirmasi	55

Gambar 5.20 Tampilan perancangan dialog waktu tunggu.....	56
Gambar 5.21 Tampilan perancangan pesan darurat	57
Gambar 5.22 Tampilan perancangan tentang aplikasi	58
Gambar 5.23 <i>Screenflow diagram login</i>	59
Gambar 5.24 <i>Screenflow diagram buat laporan</i>	60
Gambar 5.25 <i>Screenflow diagram lihat laporan</i>	61
Gambar 5.26 <i>Screenflow diagram panggilan darurat</i>	61
Gambar 5.27 <i>Screenflow diagram panduan aplikasi</i>	62
Gambar 5.28 <i>Screenflow diagram mengirim dan menerima pesan darurat</i>	63
Gambar 6.1 Implementasi data token	66
Gambar 6.2 Implementasi data kendaraan	66
Gambar 6.3 Implementasi data laporan	67
Gambar 6.4 Implementasi data nomor darurat	68
Gambar 6.5 Implementasi data pengguna	68
Gambar 6.6 Implementasi data pesan darurat.....	69

DAFTAR PERSAMAAN

Persamaan 2.1 Rumus perhitungan SUS Score.....	11
Persamaan 7.1 Rumus perhitungan Success Rate.....	90

DAFTAR LAMPIRAN

Lampiran 1 Uji <i>prototype</i> dengan Ketua RW 12.....	97
Lampiran 2 Uji <i>prototype</i> dengan Ketua RT 01/RW 12	98
Lampiran 3 Uji <i>prototype</i> dengan Ketua RT 02/RW 12	99
Lampiran 4 Uji <i>prototype</i> dengan Koordinator Keamanan.....	100
Lampiran 5 Uji <i>prototype</i> dengan Koordinator Keamanan.....	101
Lampiran 6 Pengujian <i>usability</i> dengan Ketua RW 12 (lembar 1)	102
Lampiran 7 Pengujian <i>usability</i> dengan Ketua RW 12 (lembar 2)	103
Lampiran 8 Pengujian <i>usability</i> dengan Ketua RT 01/RW 12 (lembar 1).....	104
Lampiran 9 Pengujian <i>usability</i> dengan Ketua RT 01/RW 12 (lembar 2).....	105
Lampiran 10 Pengujian <i>usability</i> dengan Ketua RT 02/RW 12 (lembar 1).....	106
Lampiran 11 Pengujian <i>usability</i> dengan Ketua RT 02/RW 12 (lembar 2).....	107
Lampiran 12 Pengujian <i>usability</i> dengan Ibu Dwi Astuti (lembar 1).....	108
Lampiran 13 Pengujian <i>usability</i> dengan Ibu Dwi Astuti (lembar 2).....	109
Lampiran 14 Pengujian <i>usability</i> dengan Pemuda Karang Taruna (lembar 1)...	110
Lampiran 15 Pengujian <i>usability</i> dengan Pemuda Karang Taruna (lembar 2)... <td>111</td>	111
Lampiran 16 Foto pengujian <i>usability</i> dengan Ketua RW 12 dan Ibu Dwi Astuti	112
Lampiran 17 Foto pengujian <i>usability</i> dengan Ketua RT 01/RW 12	113
Lampiran 18 Foto pengujian <i>usability</i> dengan Ketua RT 02/RW 12	114
Lampiran 19 Foto pengujian <i>usability</i> dengan Pemuda Karang Taruna	115

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Menurut undang-undang Nomor 1 tahun 2011 tentang perumahan dan kawasan permukiman, perumahan adalah kumpulan rumah sebagai bagian dari permukiman, baik perkotaan maupun perdesaan, yang dilengkapi dengan prasarana, sarana, dan utilitas umum sebagai hasil upaya pemenuhan rumah yang layak huni. Kondisi lingkungan perumahan dengan penduduk yang memiliki tingkat mobilitas tinggi dan jam kerja yang panjang membuat lingkungan perumahan menjadi sepi dan hanya sebagian kecil anggota keluarga yang berada di rumah. Kondisi lingkungan perumahan yang sepi tentunya akan memicu terjadinya tindak kejahatan. Keamanan perumahan dari tindak kejahatan menjadi faktor utama seseorang untuk memilih lokasi tempat tinggal (Utaberta, 2014).

Perumahan Panorama Cibodas merupakan griya hunian tempat tinggal penulis yang bertempat di Jalan Prabu Kian Santang, Kelurahan Sangiang Jaya, Kecamatan Periuk, Kota Tangerang. Berdasarkan pengamatan langsung yang dilakukan oleh penulis, pada Perumahan Panorama Cibodas memiliki 3 petugas keamanan yang melakukan tugas jaga secara bergantian dalam 3 *shift*, sehingga pada 1 *shift* hanya terdapat 1 petugas keamanan yang berjaga. Jumlah petugas keamanan yang berjumlah 1 orang tentunya tidak akan dapat melakukan pengawasan secara maksimal. Jika terjadi tindak kejahatan yang lokasinya jauh dari pos keamanan, maka tidak dapat langsung ditangani oleh petugas keamanan. Selain itu, terdapat banyak kendaraan yang masuk dan keluar dari Perumahan Panorama Cibodas. Petugas keamanan tidak menghafal seluruh kendaraan warga sehingga petugas keamanan tidak dapat membedakan antara kendaraan warga dan kendaraan asing dari luar. Keberadaan warga asing di dalam lingkungan perumahan tanpa pengawasan dari petugas keamanan dan warga sekitar tentunya akan menjadi ancaman bagi keamanan di lingkungan perumahan.

Selain petugas keamanan, keamanan di lingkungan perumahan juga dapat diwujudkan oleh para warga perumahan. Warga perumahan yang peduli terhadap lingkungan akan selalu mengawasi rumahnya dan rumah-rumah disekitarnya. Jika seluruh warga perumahan memiliki kesadaran sosial untuk saling menjaga satu sama lain maka jumlah petugas keamanan yang sedikit tidak akan menghambat terwujudnya keamanan di lingkungan perumahan. Sebelumnya telah terdapat penelitian mengenai sistem keamanan perumahan oleh (Budianingsih & Riyanto, 2018) dengan konsep melaporkan keadaan darurat kepada petugas keamanan melalui SMS. Konsep tersebut sudah tepat, namun kurang cocok untuk diimplementasikan pada lingkungan perumahan yang hanya memiliki 1 petugas keamanan. Selain itu, konsep tersebut juga kurang memberikan dampak pada hubungan sosial antar warga perumahan.

Berdasarkan permasalahan yang telah dipaparkan, penulis akan berusaha mengembangkan aplikasi untuk pelaporan tindak kejahatan di Perumahan Panorama Cibodas. Aplikasi untuk pelaporan tindak kejahatan diharapkan dapat

membantu menciptakan keadaan aman di lingkungan perumahan meskipun hanya dengan 1 orang petugas keamanan. Aplikasi juga diharapkan dapat meningkatkan hubungan sosial antar warga perumahan. Dengan memanfaatkan sensor GPS (*Global Positioning System*) pada *smartphone* dapat menjadi validasi data pelaporan ancaman tindak kejahatan. Sensor GPS (*Global Positioning System*) akan mengirimkan lokasi saat melakukan pelaporan sehingga dapat divalidasi pengguna yang melaporkan di lokasi Perumahan Panorama Cibodas. Penulis akan menggunakan metode *prototyping* dalam mengembangkan aplikasi.

1.2 Rumusan Masalah

Setelah diuraikannya latar belakang pada penelitian ini, maka rumusan masalah dapat ditentukan dan rumusan masalah dari penelitian ini sebagai berikut:

1. Bagaimana hasil analisis kebutuhan dalam pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas dengan menggunakan metode *prototyping* ?
2. Bagaimana perancangan dan implementasi aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas ?
3. Bagaimana hasil pengujian validasi, *usability* dan *compatibility* pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas ?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mendapatkan kebutuhan fungsional dan non fungsional aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas berdasarkan pengembangan dengan metode *prototyping*.
2. Mengetahui hasil perancangan dan implementasi aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.
3. Mengetahui hasil pengujian validasi, pengujian *usability* dan pengujian *compatibility* pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.

1.4 Manfaat

Manfaat dari penelitian ini sebagai berikut:

1. Manfaat bagi penulis

Penulis dapat membuat aplikasi perangkat bergerak sesuai dengan materi yang telah dipelajari selama perkuliahan.

2. Manfaat bagi peneliti selanjutnya

Dapat menjadi acuan untuk pengembangan penelitian yang serupa.

3. Manfaat bagi pengguna

Membantu menjaga keamanan pada lingkungan Perumahan Panorama Cibodas, meningkatkan hubungan sosial antar warga Perumahan Panorama Cibodas dan diharap dapat menangani keadaan darurat lebih cepat.

1.5 Batasan Masalah

Dari latar belakang yang telah diuraikan pada penelitian ini, maka terdapat beberapa batasan dalam penelitian ini, diantaranya:

1. Aplikasi hanya akan diimplementasikan pada perumahan Panorama Cibodas.
2. Aplikasi hanya dapat digunakan oleh petugas keamanan dan warga Perumahan Panorama Cibodas.

1.6 Sistematika Pembahasan

BAB I : PENDAHULUAN

Berisikan latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat dari penelitian, batasan penelitian, dan sistematika pembahasan skripsi ini.

BAB II : LANDASAN KEPUSTAKAAN

Berisikan tentang kajian-kajian kepustakaan terkait yang digunakan sebagai referensi dalam melakukan penelitian.

BAB III : METODOLOGI

Memuat alur kerja penelitian dalam proses penyelesaian permasalahan penelitian.

BAB IV : ANALISIS KEBUTUHAN

Menjelaskan mengenai proses analisis kebutuhan pada aplikasi untuk pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.

BAB V : PERANCANGAN

Menjelaskan tentang perancangan dari aplikasi untuk pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.

BAB VI : IMPLEMENTASI

Menjelaskan tentang proses yang dilakukan pada implementasi aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.

BAB VII : PENGUJIAN

Menjelaskan tentang proses pengujian yang dilakukan pada aplikasi untuk pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.

BAB VIII : PENUTUP

Berisi kesimpulan dan saran dari hasil pengembangan aplikasi untuk pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisikan landasan kepustakaan yang terdiri dari kajian pustaka serta teori-teori dasar yang berkaitan dengan penelitian. Kajian pustaka berisi penelitian yang pernah dilakukan sebelumnya dan berkaitan dengan rancangan dan implementasi aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.

2.1 Kajian Pustaka

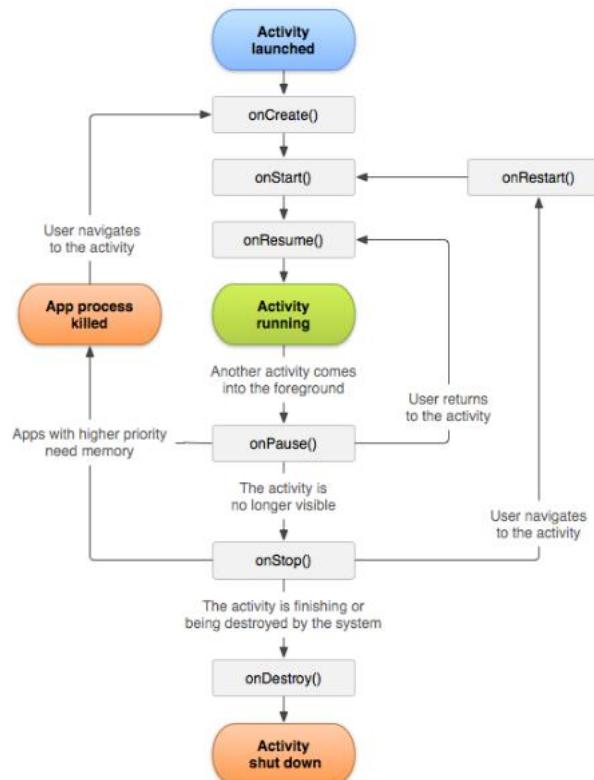
Penulis menemukan beberapa penelitian sebelumnya yang relevan terkait penelitian ini sehingga dapat membantu dalam penelitian yang dilakukan oleh penulis, beberapa penelitian tersebut diantaranya :

1. Penelitian yang berjudul “*Prototype Sistem Keamanan Cerdas Pada Komplek Perumahan*” (Budianingsih & Riyanto, 2018). Pada penelitian ini, pengembang menerapkan sebuah konsep sistem keamanan pada komplek perumahan. Penelitian ini fokus pada keamanan berbasis *Short Message Service (SMS)*. Jika terdapat suatu kejadian darurat, pemilik rumah akan menekan tombol darurat pada aplikasi kemudian akan mengirim *SMS* kepada petugas yang berisi data lokasi pemilik rumah dan kondisi keadaan darurat yang terjadi. Penulis akan menjadikan penelitian tersebut sebagai referensi dalam mengembangkan konsep sistem keamanan pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.
2. Penelitian dalam bentuk jurnal yang berjudul “*Implementasi Global Positioning System (GPS) dan Location Based Service (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek*” (Alfeno & Devi, 2017). Pada penelitian ini menerapkan *Global Positioning System (GPS)* dan layanan berbasis lokasi pada sistem informasi kereta api di wilayah Jabodetabek. Penulis akan menjadikan penelitian tersebut sebagai referensi untuk menggunakan sensor GPS pada aplikasi pelaporan ancaman tindak kejahatan. Sensor GPS akan memunculkan lokasi dari pelapor yang menjadi validasi lokasi pelaporan ancaman tindak kejahatan.
3. Penelitian yang berjudul “*Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android*” (Payara & Tanone, 2018). Pada penelitian tersebut menggunakan layanan *realtime database* dari firebase sebagai media penyimpanan data. Pada penelitian tersebut juga terdapat langkah-langkah konfigurasi project aplikasi dengan firebase dan pada bagian kesimpulan telah disimpulkan bahwa penggunaan firebase sebagai media penyimpanan memiliki keunggulan transfer data yang lebih cepat dan aman. Sehingga penelitian tersebut dijadikan sebagai referensi pengembangan aplikasi untuk pelaporan

ancaman tindak kejahatan yang menggunakan salah satu fitur firebase yaitu *realtime database* sebagai penyimpanan data pada aplikasi.

2.2 Android

Android merupakan sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi (Safaat, 2012). Awalnya, android dibuat oleh perusahaan kecil bernama Android Inc di Sillicon Valley. Kemudian Google Inc mengambil alih sistem operasi android pada tahun 2005 dan menjadikannya sebagai sistem operasi yang bersifat *open source*. Sistem operasi android yang bersifat *open source* membuat siapa pun boleh memanfaatkannya dengan gratis, termasuk dalam hal kode sumber yang digunakan untuk menyusun sistem operasi tersebut (Kadir, 2013). Android menyediakan *SDK* yang merupakan kumpulan software yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan), *emulator* (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan. Android memiliki *activity* selama aplikasi berjalan, pada android *activity* terdapat *lifecycle* seperti pada gambar 2.1.



Gambar 2.1 Lifecycle Android
(Sumber : developer.android.com)

Pada gambar 2.1 merupakan *diagram lifecycle* pada android yang menunjukkan *state activity* yang ada pada android seperti `onStart()`, `onResume()`, `onPause()`, `onDestroy()` dan yang lainnya. Pengembang dapat memanfaatkan *diagram lifecycle* untuk menyesuaikan alur pada aplikasi yang dikembangkan dengan alur *activity* pada android.

2.3 Firebase

Firebase adalah sebuah layanan yang dibuat oleh Google Inc untuk membantu *developer* mengembangkan sebuah aplikasi. Firebase memiliki beberapa fitur yaitu *realtime database*, *authentication*, *cloud messaging*, *cloud storage*, *analytics*, *remote config* dan *test lab*. Firebase dapat digunakan pada berbagai macam platform yaitu Android, IOS, dan web. Data pada firebase disimpan dalam bentuk JSON (*JavaScript Object Notation*) dan dilakukan sinkronisasi data secara berkala agar klien tidak perlu melakukan panggilan untuk mengubah data (Wadkar & Patil, 2018). Pada pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas menggunakan layanan Firebase Realtime Database dan Firebase Cloud Messaging.

2.3.1 Realtime Database

Firebase Realtime Database merupakan layanan dari firebase yang digunakan sebagai penyimpanan data dalam bentuk JSON dan disinkronkan secara *realtime* kepada seluruh klien yang terhubung. Pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas akan menggunakan layanan *realtime database* sebagai penyimpanan seluruh data yang diakses oleh pengguna secara online.

2.3.2 Cloud Messaging

Firebase Cloud Messaging merupakan layanan dari firebase yang digunakan untuk pengiriman dan penerimaan pesan dan notifikasi pada perangkat *mobile* dan web. Pada Firebase Cloud Messaging memiliki keunggulan untuk melakukan penargetan pesan secara spesifik, konten notifikasi yang dapat disesuaikan dan sudah banyak digunakan oleh para *developer* dalam pengembangan sistem.

2.4 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang sangat ringan, lebih mudah dibaca oleh manusia dan mudah untuk diterjemahkan pada komputer. JSON berupa format teks yang tidak bergantung dengan bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum. JSON dapat disebut sebagai bahasa pertukaran data karena memiliki sifat-sifat tersebut. JSON terdiri dari dua struktur yaitu kumpulan pasangan nama atau nilai yang sebagai objek dan daftar nilai yang terurut.

2.5 GPS (*Global Positioning System*)

Global Positioning System atau biasa disingkat dengan *GPS*, merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem navigasi pada *GPS* menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal akan diterima oleh *receiver* di permukaan bumi dan harus mengunci minimal 3 satelit untuk mendapatkan posisi 2D dalam *latitude* dan *longitude*, dan membutuhkan minimal 4 satelit untuk mendapatkan posisi 3D dalam *latitude*, *longitude*, dan *altitude* (Alfeno & Devi, 2017). Pada perangkat

smartphone masa kini telah dilengkapi dengan sensor *GPS* untuk memberikan layanan berbasis lokasi kepada penggunanya.

2.6 Google Map

Google Map adalah layanan yang dikembangkan oleh Google berbasis peta dan lokasi yang dapat diintegrasikan pada aplikasi. Google Map menyediakan *API* yang dapat digunakan pengembang untuk menambahkan layanan berbasis lokasi pada aplikasi. Layanan Google Map dapat digunakan secara gratis pada akun *Google Cloud Platform* untuk mendapatkan *API key*. Selanjutnya *API key* akan dipanggil pada projek aplikasi agar dapat menggunakan layanan dari Google Map *API*. Pada aplikasi ini, penulis akan menggunakan layanan dari Google Map untuk menampilkan lokasi ketika melakukan pelaporan ancaman tindak kejahatan.

2.7 UML (*Unified Modelling Language*)

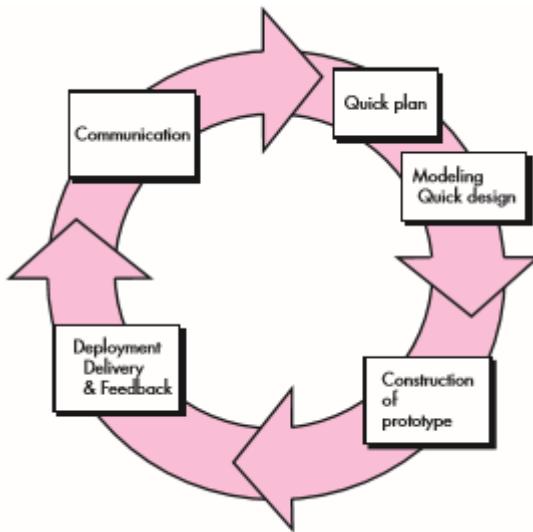
UML (*Unified Modelling Language*) adalah sebuah bahasa standar untuk melakukan pemodelan perangkat lunak. UML dapat digunakan untuk membangun, menggambarkan dan dokumentasi artifak pada proses pengembangan sistem perangkat lunak (Booch, et al., 2005). Konsep dari UML adalah setiap data yang akan diolah menjadi informasi akan dimodelkan dalam *Use Case Diagram* dan setiap kegiatan dalam masing-masing proses digambarkan dalam *Sequence Diagram*, *Activity Diagram*, *State Diagram*, *Class Diagram* dan *Deployment Diagram*. Pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas akan menggunakan *Use Case Diagram*, *Use Case Scenario*, *Sequence Diagram* dan *Class Diagram*.

2.8 Model View Presenter (MVP)

Menurut (Christoforus, et al., 2012) terdapat banyak teknik pemrograman yang dapat digunakan pada pengembangan aplikasi, salah satunya adalah *Model View Presenter* (MVP). Implementasi dari teknik MVP memiliki tujuan untuk memisahkan tanggung jawab pada bagian *Data Representation*, *User Interface* (UI) dan *Business Logic* aplikasi menjadi bentuk *model*, *view* dan *presenter*. *Model* dikenal sebagai *business object* yang memiliki tugas utama untuk menjaga data dan *method* pada aplikasi secara konsisten dapat diakses. *View* adalah bagian yang bertanggung jawab sebagai representasi antarmuka dengan pengguna. *Presenter* adalah bagian yang bertanggung jawab untuk memanipulasi model sebagai respon dari bagian *view* yang berasal dari pengguna.

2.9 Prototyping

Pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas menggunakan metode pengembangan *prototyping*. Pengembangan menggunakan metode tersebut karena kebutuhan awal yang belum pasti. *Prototyping* dilakukan secara iteratif yang memiliki beberapa tahapan yaitu *Communication*, *Quick Plan*, *Quick Design*, *Construction of Prototype*, dan *Deployment Delivery Feedback*.



Gambar 2.2 Prototyping Paradigm

(Sumber : Pressman, 2010)

Iterasi pada *prototyping* direncanakan dengan cepat menggunakan pemodelan *quick design*. *Quick design* berfokus pada representasi dari fitur-fitur pada perangkat lunak agar dapat terlihat oleh pengguna. *Quick design* mengarah pada pembangunan *prototype* untuk ditunjukkan kepada pengguna sebagai bahan evaluasi (Pressman, 2010). Penjelasan dari setiap tahapan pada *Prototyping Paradigm* adalah sebagai berikut :

1. *Communication / Komunikasi*

Melakukan pertemuan dengan calon pengguna untuk melakukan komunikasi agar kebutuhan-kebutuhan pada aplikasi dapat ditentukan.

2. *Quick Plan / Perencanaan Cepat*

Hasil kebutuhan yang telah diperoleh akan dilakukan perancangan untuk membuat *prototype*. Perencanaan akan dilakukan secara cepat agar *prototype* segera dapat dibuat.

3. *Modelling Quick Design / Pemodelan Desain Cepat*

Pemodelan secara terstruktur dalam bentuk *DFD(Data Flow Diagram)*, *ERD(Entity Relationship Diagram)* dan *Flowchart* untuk menggambarkan analisis dan desain sistem.

4. *Construction of Prototype / Pembuatan Prototype*

Setelah dilakukan perencanaan dan desain, maka akan dibangun *prototype* dari sistem perangkat lunak. *Prototype* diharapkan dapat menjadi gambaran mengenai sistem yang akan ditunjukkan kepada calon pengguna.

5. *Deployment Delivery & Feedback / Penyerahan dan Umpaman Balik*

Setelah pembuatan *prototype* selesai, maka *prototype* akan diserahkan kepada calon pengguna. Calon pengguna akan mencoba *prototype* dan

memberikan *feedback* mengenai *prototype* untuk dilanjutkan ke tahap implementasi. Jika masih terdapat kekurangan maka perlu dilakukan perbaikan dan kembali melakukan iterasi.

2.10 Teknik Pengujian

Teknik pengujian yang akan dilakukan pada pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan yaitu pengujian `validasi (*Blackbox Testing*), pengujian *usability* dan pengujian *compatibility*.

2.10.1 Pengujian Validasi

Pengujian validasi dapat didefinisikan sebagai pengujian untuk melakukan pengecekan apakah sebuah sistem telah berjalan dengan benar menggunakan serangkaian kasus uji yang dilakukan pengguna terhadap sistem (Sommerville, 2011). Pengujian validasi termasuk dalam kategori *Blackbox Testing*.

2.10.2 Pengujian *Usability*

Pengujian *usability* yaitu pengujian yang dilakukan untuk menguji kebutuhan non fungsional pada sistem dengan cara menguji sistem secara langsung kepada pengguna kemudian merekam respon dari para pengguna yang diukur dengan skala likert (Pratama, et al., 2019). Menurut (Nielsen, 2001) ukuran paling dasar *usability* dari sebuah sistem dapat diukur dengan matriks kualitas sebagai berikut:

- *Success rate* : tingkat keberhasilan pengguna melakukan tugas pada proses pengujian.
- *The time a task requires* : waktu yang dibutuhkan untuk melakukan tugas pada proses pengujian.
- *The error rate* : tingkat kesalahan pengguna saat melakukan pengujian.
- *User subjective satisfaction* : kepuasan pengguna saat menggunakan sistem secara subjektif.

Pada aplikasi ini akan melakukan pengujian usability dengan menggunakan matriks kualitas *success rate* dan *user subjective satisfaction*. *Success rate* digunakan untuk mengukur apakah pengguna benar-benar dapat menggunakan aplikasi tanpa mengalami masalah dan *user subjective satisfaction* untuk mengukur bagaimana tingkat kepuasan pengguna terhadap aplikasi yang telah dikembangkan. Pada pengukuran *user subjective satisfaction* menggunakan kuesioner SUS yang diukur dengan skala likert. Menurut (Budiaji, 2013) dalam (Likert, 1932) skala likert merupakan pengukuran dengan menggunakan beberapa butir pernyataan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan yaitu sangat setuju, setuju, tidak memutuskan, tidak setuju dan sangat tidak setuju. Kemudian untuk perhitungan bobot pada setiap pernyataan akan menggunakan *SUS Score*. Menurut (Aprilia, et al., 2015) dalam (Brooke, 1996) Nilai dari *SUS Score* memiliki rentang dari 0 hingga

100. Perhitungan *SUS Score* menurut John Brooke yaitu jawaban dari setiap pernyataan memiliki bobot skor kontribusi, pernyataan dikelompokkan menjadi pernyataan dengan nomor ganjil (pernyataan positif) dihitung dengan skor kontribusi dikurangi dengan 1 dan pernyataan dengan nomor genap (pernyataan negatif) dihitung dengan 5 dikurangi dengan skor kontribusi, lalu hasilnya akan diakumulasi dan dikalikan dengan 2,5. Gambaran dari rumus perhitungan *SUS Score* ditunjukkan pada persamaan 2.1.

$$\text{SUS Score} = ((P_1 - 1) + (5 - P_2) + (P_3 - 1) + (5 - P_4) + (P_5 - 1) + (5 - P_6) + (P_7 - 1) + (5 - P_8) + (P_9 - 1) + (5 - P_{10})) * 2.5 \quad (2.1)$$

Hasil dari perhitungan *SUS Score* dari setiap responden akan diakumulasi dan dihitung nilai rataannya yang dijadikan sebagai nilai akhir *SUS Score*. Menurut (Alathas, 2018) interpretasi nilai akhir *SUS Score* dikelompokkan seperti pada tabel 2.1.

Tabel 2.1 Interpretasi nilai akhir *SUS Score*

<i>SUS Score</i>	<i>Grade</i>	<i>Adjective Rating</i>
> 80.3	A	<i>Excellent</i>
68 – 80.3	B	<i>Good</i>
68	C	<i>Okay</i>
51-68	D	<i>Poor</i>
< 51	F	<i>Awful</i>

2.10.3 Pengujian *Compatibility*

Menurut (Pressman, 2010) setiap perangkat, sistem operasi dan kecepatan koneksi jaringan dapat memiliki pengaruh pada operasi yang dilakukan oleh sistem, setiap konfigurasi perangkat menghasilkan perbedaan dalam kecepatan pemrosesan sisi klien, resolusi tampilan dan kecepatan koneksi. Pengujian *compatibility* digunakan untuk memeriksa apakah sistem yang dikembangkan dapat dijalankan pada perangkat yang menjadi target untuk pengembangan dan perangkat yang diluar dari pengembangan sistem. Pengujian *compatibility* dapat dikatakan berhasil jika perangkat yang menjadi target pengembangan berhasil melakukan kasus uji yang telah ditentukan dan perangkat yang diluar dari pengembangan sistem tidak dapat untuk menjalankan kasus uji yang telah ditentukan.

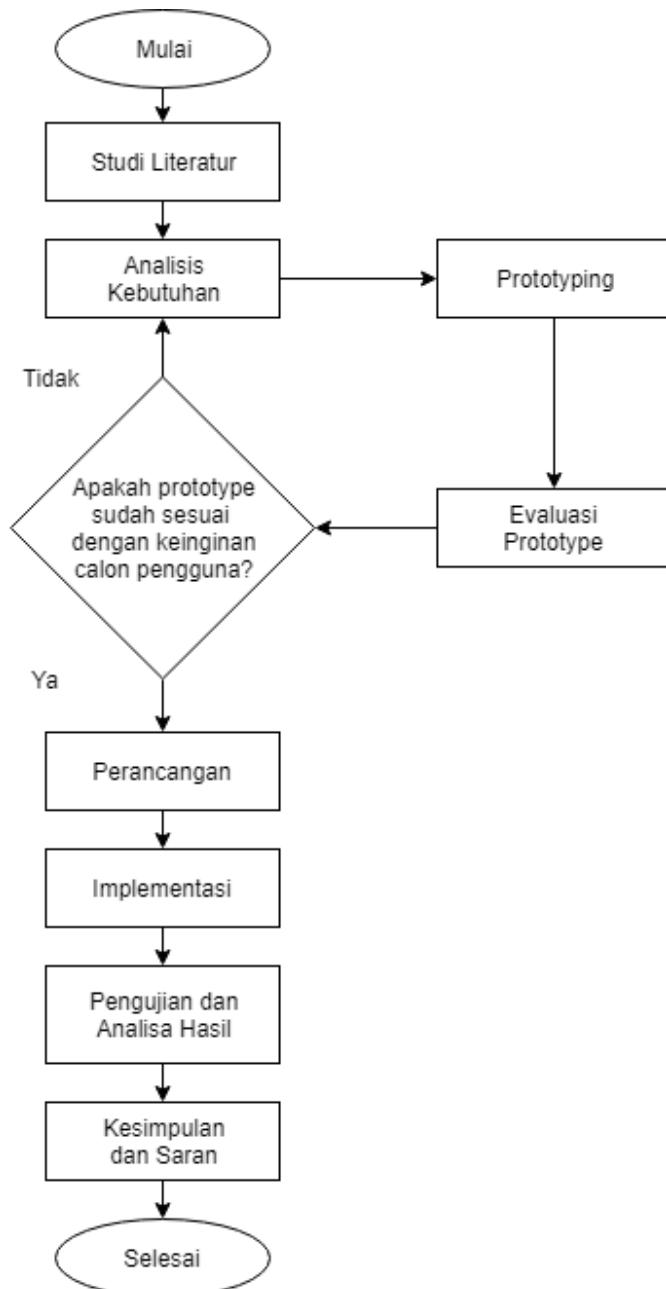
2.10.3.1 Firebase Test Lab

Firebase Test Lab merupakan infrastruktur pengujian yang disediakan oleh firebase untuk melakukan pengujian pada aplikasi berbasis *cloud*. Firebase Test Lab dapat melakukan pengujian pada *platform* android dan iOS secara bersamaan dalam satu operasi. Pengujian dapat melakukan konfigurasi perangkat untuk pengujian, melihat hasil pengujian termasuk hasil *log*, video dan *screenshot* pada

firebase console. Test Lab menggunakan perangkat uji sesungguhnya yang berjalan di pusat data Google untuk melakukan pengujian aplikasi. Firebase Test Lab diperlukan untuk menguji aplikasi ini supaya aplikasi dapat di uji secara otomatis untuk dijalankan pada berbagai versi sistem operasi android tanpa harus menghadirkan perangkat uji secara langsung.

BAB 3 METODOLOGI

Penelitian ini merupakan penelitian dengan tipe implementatif pengembangan, peneliti akan mengembangkan aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas. Subjek pada penelitian ini adalah warga, petugas keamanan dan ketua RW di Perumahan Panorama Cibodas. Peneliti melakukan pengembangan dengan metode *Software Development Life Cycle : Prototyping* dengan alur penelitian yang digambarkan pada gambar 3.1.



Gambar 3.1 Diagram alir metodologi

3.1 Studi Literatur

Pada studi literatur, peneliti mencari teori-teori dan penelitian yang dapat dijadikan referensi dalam melakukan penelitian. Terdapat 3 penelitian yang mendukung penelitian ini, yaitu :

1. Penelitian yang berjudul “Prototype Sistem Keamanan Cerdas Pada Komplek Perumahan” (Budianingsih & Riyanto, 2018). Pada penelitian ini, pengembang menerapkan sebuah konsep sistem keamanan pada komplek perumahan. Penelitian ini fokus pada keamanan berbasis *Short Message Service (SMS)*. Jika terdapat suatu kejadian darurat, pemilik rumah akan menekan tombol darurat pada aplikasi kemudian akan mengirim *SMS* kepada petugas yang berisi data lokasi pemilik rumah dan kondisi keadaan darurat yang terjadi. Penulis akan menjadikan penelitian tersebut sebagai referensi dalam mengembangkan konsep sistem keamanan pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas.
2. Penelitian dalam bentuk jurnal yang berjudul “Implementasi *Global Positioning System (GPS)* dan *Location Based Service (LSB)* pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek” (Alfeno & Devi, 2017). Pada penelitian ini menerapkan *Global Positioning System (GPS)* dan layanan berbasis lokasi pada sistem informasi kereta api di wilayah Jabodetabek. Penulis akan menjadikan penelitian tersebut sebagai referensi untuk menggunakan sensor GPS pada aplikasi pelaporan ancaman tindak kejahatan. Sensor GPS akan menjadi validasi lokasi pelaporan ancaman tindak kejahatan.
3. Penelitian yang berjudul “Penerapan *Firebase Realtime Database* Pada *Prototype* Aplikasi Pemesanan Makanan Berbasis Android” (Payara & Tanone, 2018). Pada penelitian tersebut menggunakan layanan *realtime database* dari firebase sebagai media penyimpanan data. Pada penelitian tersebut juga terdapat langkah-langkah konfigurasi *project* aplikasi dengan firebase dan pada bagian kesimpulan telah disimpulkan bahwa penggunaan firebase sebagai media penyimpanan memiliki keunggulan transfer data yang lebih cepat dan aman. Sehingga penelitian tersebut dijadikan sebagai referensi pengembangan aplikasi untuk pelaporan ancaman tindak kejahatan yang menggunakan salah satu fitur firebase yaitu *realtime database* sebagai penyimpanan data pada aplikasi.

Pada studi literatur juga menjelaskan dasar teori yang digunakan pada penelitian ini, teori-teori yang mendukung penelitian diantaranya :

1. Android
2. Firebase
3. JSON
4. GPS (Global Positioning System)

5. Google Map
6. UML (Unified Modelling Language)
7. Model View Presenter (MVP)
8. Prototyping
9. Pengujian Validasi
10. Pengujian Usability

3.2 Analisis Kebutuhan

Pada analisis kebutuhan akan mendeskripsikan segala kebutuhan dalam sistem secara umum dan lengkap. Tahap pertama dari analisis kebutuhan adalah pengembang akan mengusulkan kebutuhan pada aplikasi untuk pelaporan ancaman tindak kejahatan dan mengimplementasikan kebutuhan pada aplikasi dalam bentuk *prototype*. Setelah itu *prototype* akan ditunjukkan kepada para calon pengguna dan pengembang akan melakukan wawancara untuk mendapatkan *feedback* dari uji coba *prototype*. Tahap kedua dari analisis kebutuhan adalah hasil *feedback* dari uji coba *prototype* akan digunakan sebagai bahan evaluasi untuk menghasilkan kebutuhan fungsional dan non fungsional pada aplikasi untuk pelaporan ancaman tindak kejahatan di Perumahan Panorama Cibodas. Analisis kebutuhan akan menghasilkan identifikasi aktor pada aplikasi, *use case diagram* dan *use case scenario*. *Use case* didapatkan dari kebutuhan fungsional pada aplikasi yang telah melalui iterasi dengan metode *prototyping* hingga tahapan *final prototyping*.

3.3 Prototyping

Setelah pengembang mengusulkan kebutuhan-kebutuhan pada aplikasi, maka akan dilakukan pembuatan *prototype*. *Prototype* berupa rancangan antarmuka yang menggambarkan sistem perangkat lunak dan dapat berinteraksi dengan pengguna. *Prototype* akan diuji oleh calon pengguna untuk mendapatkan *feedback* sebagai bahan evaluasi.

3.4 Evaluasi Prototype

Setelah calon pengguna mencoba menjalankan *prototype*, calon pengguna akan memberikan *feedback* mengenai *protoype* yang sedang digunakan. *Feedback* dari calon pengguna akan dijadikan sebagai bahan evaluasi. Jika calon pengguna memberikan *feedback* berupa kekurangan maka harus dilakukan perbaikan dan kembali melakukan iterasi untuk membuat *prototype* lagi. Sedangkan jika calon pengguna telah puas dengan *protoype* yang digunakan, maka akan beralih ke tahap yang selanjutnya yaitu tahapan perancangan sistem perangkat lunak.

3.5 Perancangan

Pada tahap ini akan dilakukan perancangan untuk memodelkan aplikasi yang akan menyelesaikan permasalahan dari hasil analisis kebutuhan. Pada tahap

perancangan akan menggunakan metode OOD (*Object Oriented Design*) menggunakan tools UML (*Unified Modelling Language*) yang berupa *Sequence Diagram* dan *Class Diagram*. Selanjutnya pada perancangan antarmuka akan ditunjukkan tampilan antarmuka dari hasil *prototyping* dalam bentuk *screenflow diagram* dan penjelasan setiap elemen tampilan antarmuka.

3.6 Implementasi

Pada tahap implementasi merupakan tahapan untuk eksekusi dari hasil perancangan yang telah dilakukan sebelumnya. Implementasi akan dilakukan dengan metode OOP (*Object Oriented Programming*) dengan menggunakan bahasa pemrograman Kotlin dan untuk tampilan antarmuka menggunakan XML. Hasil dari *Class Diagram* akan digunakan sebagai acuan calon *class* yang akan diimplementasikan, sementara hasil dari *Sequence Diagram* akan digunakan sebagai acuan alur dari sistem perangkat lunak.

3.7 Pengujian dan Analisa Hasil

Pengujian akan dilakukan pada aplikasi yang telah melalui tahapan implementasi. Aplikasi akan diujikan kepada calon pengguna dengan harapan dapat beroperasi sesuai dengan keinginan calon pengguna. Selain itu juga akan dilakukan analisa hasil untuk mengukur apakah aplikasi telah berjalan sesuai dengan harapan.

3.7.1 Pengujian Validasi

Pengujian validasi akan dilakukan pada seluruh fungsional aplikasi dengan cara membuat kasus uji yang berisi prosedur pengujian, target yang ingin dicapai dan hasil dari pengujian.

3.7.2 Pengujian Usability

Pengujian *usability* pada aplikasi dibagi menjadi dua tahapan, yaitu responden melakukan *task* yang telah dibuat oleh pengembang dan pengisian kuesioner *System Usability Scale* (SUS). Pengujian akan berfokus pada tingkat keberhasilan pengguna dalam menjalankan *task* yang telah diberikan dan bagaimana kepuasan pengguna setelah menggunakan aplikasi. Pengujian *usability* akan dilakukan kepada 5 orang responden untuk menggunakan aplikasi dengan mengerjakan *task* yang diberikan kepada responden dan sebelumnya pengembang akan memberikan sedikit *clue* terkait *task* yang harus dilakukan. Setelah itu pengguna akan diminta untuk mengisi kuesioner *System Usability Scale* (SUS) yang berisi pernyataan untuk menjawab bagaimana hasil pengalaman pengguna setelah menggunakan aplikasi. Pertanyaan berjumlah 10 pernyataan dan dijawab dengan rentang angka skala likert yang ditunjukkan pada tabel 3.1.

Tabel 3.1 Keterangan skor skala likert

Skor	Keterangan
1	Sangat Tidak Setuju

Tabel 3.1 Keterangan skor skala likert (lanjutan)

2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Pernyataan akan dibagi menjadi pernyataan ganjil dan pernyataan genap. pernyataan ganjil merupakan pernyataan positif karena memiliki jawaban bagus jika responden menjawab dengan skor tertinggi. Sementara pernyataan genap merupakan pernyataan negatif karena memiliki jawaban bagus jika responden menjawab dengan skor terendah. Hasil dari kuesioner SUS akan dihitung skor nya pada setiap responden sesuai dengan teori yang ada pada bab sebelumnya. Setelah itu skor SUS pada setiap responden akan diakumulasi untuk menentukan nilai rataan yang akan menjadi nilai akhir skor SUS. Nilai akhir skor SUS akan menentukan apakah sistem yang diuji memiliki usabilitas yang baik atau buruk, sesuai dengan interpretasi skor yang dibuat oleh (Alathas, 2018).

Sementara itu, tingkat keberhasilan pengguna dalam mengerjakan task akan diukur seperti yang dilakukan oleh (Nielsen, 2001) yaitu dengan membuat kategori hasil *task* menjadi berhasil (B), hampir berhasil (HB) dan gagal (G). Perhitungan dari hasil tingkat keberhasilan pengguna dalam mengerjakan *task* yaitu dengan menghitung setiap *task* yang berhasil (B) dengan poin 1, hampir berhasil (HB) dengan poin 0,5 dan gagal (G) dengan poin 0. Kemudian seluruh poin diakumulasikan dan dibagi dengan jumlah *task* lalu dikali dengan 100%, maka akan didapatkan nilai dari tingkat keberhasilan/*success rate*.

3.7.3 Pengujian *Compatibility*

Pengujian *compatibility* pada aplikasi akan dilakukan pada lingkungan perangkat keras, perangkat lunak dan versi sistem operasi android yang berbeda-beda. Aplikasi diharapkan dapat berjalan pada perangkat keras, perangkat lunak dan versi sistem operasi android yang berbeda-beda sesuai dengan target yang telah ditentukan. Pengujian *compatibility* dilakukan dengan dua cara yaitu secara manual dan otomatis dengan bantuan Firebase Test Lab dengan metode Robo Test. Pada pengujian secara manual aplikasi akan diuji pada 3 perangkat dengan spesifikasi yang berbeda-beda. Pengujian *compatibility* dengan Firebase Test Lab dilakukan pada 5 perangkat uji dengan spesifikasi yang berbeda-beda untuk memenuhi persyaratan versi sistem operasi android yang telah ditargetkan. Hasil pengujian *compatibility* dikatakan berhasil apabila aplikasi dapat dijalankan pada seluruh perangkat pengujian yang memenuhi target versi sistem operasi android dan aplikasi tidak dapat dijalankan pada seluruh perangkat pengujian diluar dari versi sistem android yang telah ditargetkan.

3.8 Pengambilan Kesimpulan dan Saran

Pada tahap ini akan dilakukan penarikan kesimpulan berdasarkan seluruh tahapan penelitian yang telah dilakukan. Kesimpulan dibuat berdasarkan hasil dari pengujian dan analisa hasil yang telah dilakukan pada tahapan sebelumnya. Saran akan dibuat untuk menunjang penelitian berikutnya agar dapat memanfaatkan kelebihan dari penelitian ini dan tidak mengulangi kesalahan-kesalahan yang sama pada penelitian ini.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini membahas tentang analisis kebutuhan pada aplikasi berbasis *mobile* untuk pelaporan ancaman kejahatan di Perumahan Panorama Cibodas. Pada bab ini berisi analisis kebutuhan yang terdiri dari identifikasi aktor, kebutuhan fungsional non fungsional, *use case diagram* dan *use case scenario*.

4.1 Gambaran Umum Aplikasi

Pada penelitian ini mengembangkan sebuah aplikasi untuk pelaporan ancaman tindak kejahatan di lingkungan Perumahan Panorama Cibodas. Aplikasi untuk pelaporan ancaman tindak kejahatan di lingkungan Perumahan Panorama Cibodas akan diberi nama SKIP (**S**istem **K**eamanan **d**i **P**erumahan). Aplikasi SKIP memiliki konsep seperti siskamling yang dilakukan oleh seluruh warga dengan cara mengawasi lingkungan di sekitarnya, jika seluruh warga melakukan pengawasan pada lingkungan masing-masing maka seluruh lingkungan perumahan akan selalu aman meskipun dengan jumlah anggota petugas keamanan yang jumlahnya sedikit karena setiap warga dapat melaporkan sesuatu yang dapat menjadi potensi tindak kejahatan dan dapat dilihat oleh warga yang lainnya. Berdasarkan informasi laporan yang telah dibuat oleh warga, warga yang lain diharapkan menindaklanjuti laporan dengan cara memantau secara langsung atau bahkan bisa mendatangi pihak pembuat laporan untuk konfirmasi secara langsung. Laporan akan berisi informasi judul laporan, nama pembuat laporan, waktu pembuatan laporan, keterangan laporan, dan titik lokasi pelapor sebagai pertimbangan validasi laporan. Pada aplikasi ini terdapat menu tambahan untuk melakukan panggilan terhadap daftar nomor tanggap darurat seperti nomor telepon kepolisian, petugas keamanan, rumah sakit dan yang lainnya. Pada aplikasi ini juga terdapat tombol “*panic*” jika pengguna sedang membutuhkan bantuan secara cepat yang akan mengirimkan notifikasi darurat ke seluruh pengguna yang lainnya.

4.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mendapatkan kebutuhan pada aplikasi SKIP sesuai dengan yang dibutuhkan oleh warga Perumahan Panorama Cibodas. Tahapan analisis kebutuhan diawali dengan pengembang yang mengajukan kebutuhan fungsional awal pada aplikasi SKIP. Kemudian pengembang akan membuat *prototype* berdasarkan kebutuhan fungsional awal, lalu *prototype* akan diujikan kepada warga Perumahan Panorama Cibodas untuk mendapatkan saran dan tanggapan. Pengujian *prototype* dilakukan dengan cara melakukan wawancara dengan beberapa perwakilan warga Perumahan Panorama Cibodas. Saran dan tanggapan yang diberikan akan menjadi pertimbangan untuk penentuan kebutuhan fungsional akhir pada aplikasi.

4.2.1 Kebutuhan Fungsional Awal

Kebutuhan fungsional awal merupakan kebutuhan fungsional yang diajukan oleh pengembang pada aplikasi SKIP. Berikut adalah kebutuhan fungsional awal yang ditunjukkan pada tabel 4.1.

Tabel 4.1 Kebutuhan fungsional awal

No	Kode	Nama Kebutuhan
1	F-SKIP-1	<i>Login</i>
2	F-SKIP-2	Buat Laporan
3	F-SKIP-3	Lihat Laporan
4	F-SKIP-4	Panggilan Darurat
5	F-SKIP-5	Panduan Aplikasi
6	F-SKIP-6	Mengirim Pesan Darurat
7	F-SKIP-7	Menerima Pesan Darurat
8	F-SKIP-8	Tentang Aplikasi

4.2.2 Hasil Wawancara Uji Coba *Prototype*

Pada tahap sebelumnya telah ditentukan kebutuhan fungsional awal pada aplikasi serta melakukan pembuatan *prototype* berdasarkan daftar kebutuhan tersebut dan melakukan pengujian kepada calon pengguna. Selanjutnya pada tahap ini akan membahas mengenai hasil pengujian *prototype* yang dilakukan kepada 8 (delapan) orang perwakilan warga Perumahan Panorama Cibodas dengan informasi identitas yang ditunjukkan pada tabel 4.2. Bukti hasil uji coba *prototype* ditunjukkan pada *form* yang diisi oleh para responden pada bagian lampiran.

Tabel 4.2 Daftar responden uji *prototype*

No	Nama	Profesi	Umur	Jabatan di Perumahan Panorama Cibodas
1	Sanduki	Pegawai Swasta	56 thn	Ketua RW 12
2	Karsidi	Karyawan Swasta	43 thn	Ketua RT 01/RW 12
3	Asep M Syahrul	Pegawai Swasta	52 thn	Ketua RT 02/RW 12
4	Suhatril	Wiraswasta	52 thn	Ketua RT 03/RW 12
5	Maryulis	-	55 thn	Sekretaris RT 02/RW 12
6	Isa Ansori	Karyawan	45 thn	Sekretaris RT 03/RW 12
7	Sentot Alibasa	TNI (Purn)	57 thn	Koordinator keamanan di lingkungan RW 12
8	Sutarno	TNI	58 thn	Koordinator keamanan di lingkungan RW 12

Berdasarkan hasil wawancara yang telah dilakukan ketika uji coba *prototype* maka dapat dibuat beberapa poin kesimpulan yang ditunjukkan pada tabel 4.3. Kesimpulan dibuat berdasarkan hasil wawancara serta pengisian saran dan tanggapan pada form hasil uji coba *prototype*.

Tabel 4.3 Kesimpulan hasil wawancara uji *prototype*

No	Kesimpulan Hasil Wawancara Uji Coba <i>Prototype</i>
1.	Seluruh responden setuju dengan implementasi aplikasi untuk pelaporan ancaman kejahatan di lingkungan Perumahan Panorama Cibodas.
2.	Seluruh responden setuju dengan seluruh kebutuhan fungsional yang diajukan pada <i>prototype</i> untuk aplikasi pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.
3.	Seluruh responden telah setuju dengan tampilan antarmuka pada <i>prototype</i> untuk diimplementasikan pada aplikasi pelaporan ancaman kejahatan di Perumahan Panorama Cibodas.
4.	Sebagian responden menyarankan untuk membuat petunjuk pelaksanaan dan teknik secara lengkap karena sebagian warga Perumahan Panorama Cibodas awam terhadap teknologi.

Hasil kesimpulan pada tabel 4.3 menunjukkan bahwa seluruh responden telah menyetujui *prototype* yang dibuat oleh pengembang untuk diimplementasikan secara langsung menjadi aplikasi pelaporan ancaman kejahatan di Perumahan Panorama Cibodas. Sehingga dalam tahapan metodologi penelitian dengan metode *prototyping* pengembang tidak perlu melakukan iterasi untuk melakukan uji coba *prototype*.

4.3 Identifikasi Aktor

Pada tahapan identifikasi aktor akan menunjukkan aktor yang akan berinteraksi secara langsung dengan aplikasi SKIP. Hasil identifikasi aktor ditunjukkan pada tabel 4.4.

Tabel 4.4 Identifikasi aktor

Nama Aktor	Deskripsi
Pengguna	Merupakan warga Perumahan Panorama Cibodas yang telah memiliki akun terdaftar untuk melakukan login dan dapat membuat laporan, melihat laporan, melakukan panggilan darurat, melihat panduan aplikasi serta menekan tombol “panic” untuk meminta pertolongan.

4.4 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang mendeskripsikan apa saja fungsi yang bisa dilakukan pada aplikasi. Kebutuhan fungsional harus dipenuhi oleh sistem agar aplikasi dapat berjalan dengan baik dan sesuai dengan yang diinginkan oleh pengguna. Hasil kebutuhan fungsional didapatkan setelah melalui proses analisis kebutuhan. Pada analisis kebutuhan disimpulkan bahwa prototype yang diujikan telah sesuai dengan keinginan pengguna sehingga hasil akhir kebutuhan fungsional akan sama dengan kebutuhan fungsional awal yang diajukan oleh pengembang. Berikut adalah hasil kebutuhan fungsional akhir yang ditunjukkan pada tabel 4.5.

Tabel 4.5 Kebutuhan fungsional

Kode Fungsional	Nama Kebutuhan	Deskripsi
F-SKIP-1	Login	Sistem harus mampu memberikan layanan bagi guest untuk <i>login</i> dengan menginputkan <i>username</i> dan <i>password</i> .
F-SKIP-2	Buat Laporan	Sistem harus mampu memberikan layanan bagi pengguna untuk membuat laporan.
F-SKIP-3	Lihat Laporan	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat daftar laporan yang sebelumnya telah dibuat.
F-SKIP-4	Panggilan Darurat	Sistem harus mampu memberikan layanan bagi pengguna untuk melakukan panggilan darurat secara langsung terhadap daftar nomor panggilan darurat pada aplikasi.
F-SKIP-5	Panduan Aplikasi	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat panduan pada setiap menu pada aplikasi.

Tabel 4.5 Kebutuhan fungsional (lanjutan)

F-SKIP-6	Mengirim Pesan Darurat	Sistem harus mampu memberikan layanan bagi pengguna untuk mengirimkan pesan darurat kepada pengguna yang lainnya.
F-SKIP-7	Menerima Pesan Darurat	Sistem harus mampu memberikan layanan bagi pengguna untuk menerima pesan darurat dari pengguna lain yang mengirimkan pesan darurat.
F-SKIP-8	Tentang Aplikasi	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat informasi mengenai aplikasi secara umum.

4.5 Kebutuhan Non Fungsional

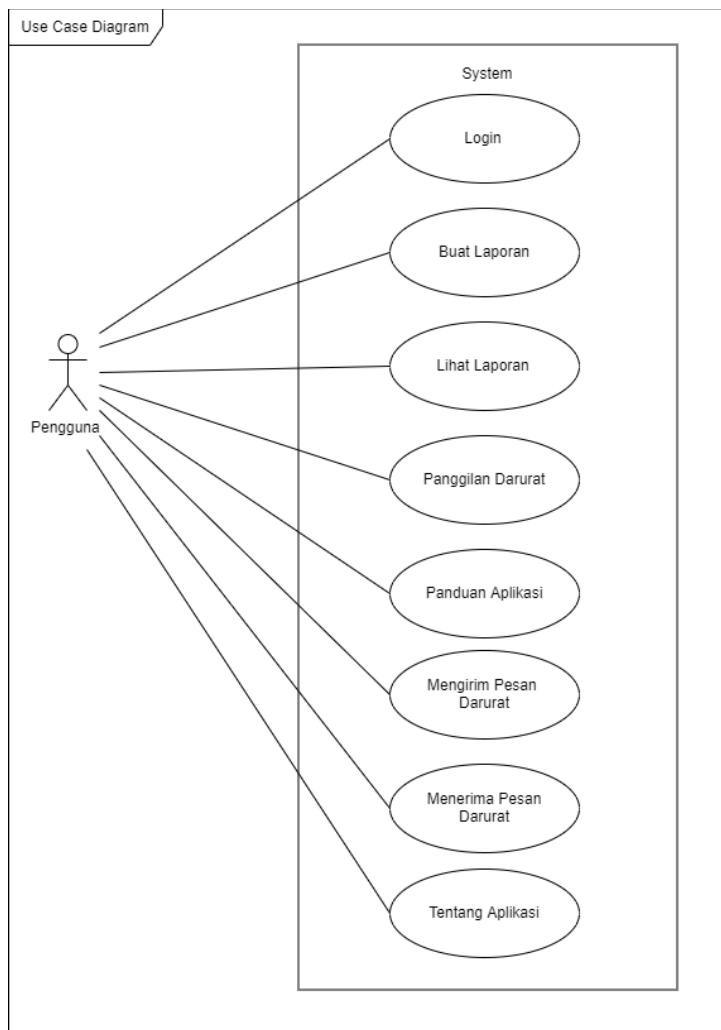
Kebutuhan non fungsional merupakan kebutuhan yang membahas tentang perilaku pada aplikasi. Kebutuhan non fungsional tidak harus dipenuhi oleh sistem namun dibutuhkan oleh pengguna yang bertujuan untuk meningkatkan kualitas aplikasi. Berikut adalah kebutuhan fungsional pada aplikasi yang ditunjukkan pada tabel 4.6.

Tabel 4.6 Kebutuhan non fungsional

Kode Fungsional	Nama	Deskripsi
NF-SKIP-1	<i>Usability</i>	Aplikasi dapat digunakan oleh pengguna dengan mudah dan memiliki nilai kegunaan.
NF-SKIP-2	<i>Compatibility</i>	Aplikasi dapat digunakan pada perangkat <i>mobile</i> dengan sistem operasi android minimal versi 4.4 KitKat dan maksimal versi 9.0 Pie.

4.6 Use Case Diagram

Use case diagram merupakan suatu diagram yang menggambarkan apa saja yang dapat dilakukan oleh aktor terhadap sistem. *Use case diagram* ditunjukkan pada gambar 4.1 terdapat satu aktor yaitu pengguna yang merupakan warga Perumahan Panorama Cibodas. Selain aktor, pada *use case diagram* juga terdapat *use case* yang merupakan seluruh elemen dari kebutuhan fungsional pada tabel 4.5.



Gambar 4.1 *Use case diagram*

4.7 Use Case Scenario

Setiap *use case* yang ada di *use case diagram* pada gambar 4.1 akan memiliki penjelasan detail mengenai proses yang terjadi pada setiap *use case*. Pada tabel 4.7 hingga tabel 4.13 akan menjelaskan secara detail skenario apa yang dilakukan oleh aktor dan sistem.

Tabel 4.7 Skenario Login

Item	Deskripsi
Kode	F-SKIP-1
Nama <i>Use case</i>	<i>Login</i>
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk <i>login</i> dengan menginputkan <i>username</i> dan <i>password</i>
Pra-kondisi	Aktor telah berada pada tampilan untuk <i>login</i>
Tindakan	<ol style="list-style-type: none">1. Aktor menginputkan <i>username</i> dan <i>password</i>2. Aktor menekan tombol “LOGIN”3. Sistem menampilkan <i>progress bar</i>4. Sistem mengirimkan inputan <i>username</i> dan <i>password</i> untuk verifikasi dengan <i>database</i> pengguna5. Sistem menghilangkan <i>progress bar</i>6. Sistem menampilkan <i>toast</i> “Login Berhasil!”7. Sistem menampilkan <i>activity dashboard</i>
Alternatif	<ol style="list-style-type: none">1. Sistem akan menampilkan <i>toast</i> “Verifikasi gagal, Periksa Kembali Username dan Password!” ketika <i>username</i> atau <i>password</i> tidak sesuai dengan yang ada pada <i>database</i> pengguna2. Sistem akan menampilkan <i>toast</i> “Proses Login Gagal, Pastikan Anda Terhubung Dengan Internet” ketika kondisi <i>smartphone</i> sedang tidak terhubung dengan internet3. Sistem akan menampilkan <i>toast</i> “Proses Login Gagal, Data Tidak Dapat Diakses” saat sistem mengalami masalah mengakses <i>database</i>
Post-kondisi	Aktor berhasil menuju <i>activity dashboard</i>

Tabel 4.8 Skenario Buat Laporan

Item	Deskripsi
Kode	F-SKIP-2
Nama <i>Use case</i>	Buat Laporan
Aktor	Pengguna

Tabel 4.8 Skenario Buat Laporan (lanjutan)

Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk membuat laporan
Pra-kondisi	Aktor telah berada pada tampilan untuk membuat laporan
Tindakan	<ol style="list-style-type: none"> 1. Aktor memasukkan data judul laporan 2. Aktor menekan tombol “CEK” 3. Sistem melakukan verifikasi data judul laporan dengan <i>database</i> nomor kendaraan warga Perumahan Panorama Cibodas 4. Sistem menampilkan hasil verifikasi data judul laporan 5. Aktor menekan tombol “BUAT LAPORAN” 6. Sistem menampilkan <i>progress bar</i> dan teks “Sedang Dalam Proses...” 7. Sistem melakukan pengambilan data lokasi aktor saat ini 8. Sistem menghilangkan <i>progress bar</i> dan teks “Sedang Dalam Proses...” 9. Sistem menampilkan <i>activity</i> untuk submit laporan yang berisi judul laporan, keterangan laporan, dan lokasi pelapor 10. Aktor menekan tombol “SUBMIT” 11. Sistem menampilkan <i>toast</i> “Upload Laporan Berhasil!” 12. Sistem menampilkan <i>activity dashboard</i>
Alternatif	<ol style="list-style-type: none"> 1. Sistem akan menampilkan <i>toast</i> “Proses Pengecekan Gagal, Pastikan Anda Terhubung Dengan Internet” saat sistem gagal melakukan verifikasi data judul laporan dengan <i>database</i> nomor kendaraan warga Perumahan Panorama Cibodas karena masalah koneksi internet 2. Sistem akan menampilkan <i>toast</i> “Proses Pengecekan Gagal, Data Tidak Dapat Diakses” saat sistem gagal melakukan verifikasi data judul laporan dengan <i>database</i> nomor kendaraan warga Perumahan Panorama Cibodas karena sistem tidak dapat mengakses <i>database</i>

Tabel 4.8 Skenario Buat Laporan (lanjutan)

	<ol style="list-style-type: none">3. Sistem akan menampilkan <i>toast</i> “Gagal Mendapatkan Data Lokasi, Silahkan Coba Lagi” saat sistem gagal melakukan pengambilan data lokasi aktor4. Sistem akan menampilkan <i>toast</i> “Silahkan Aktifkan Sensor GPS/Location Service Terlebih Dahulu” jika pengguna belum mengaktifkan sensor GPS/Location Service5. Sistem akan menampilkan <i>toast</i> “Upload Laporan Gagal, Pastikan Anda Terhubung Dengan Internet” saat sistem gagal melakukan <i>push</i> data laporan ke <i>database</i> karena masalah koneksi internet6. Sistem akan menampilkan <i>toast</i> “Upload Laporan Gagal, Data Tidak Dapat Diakses” saat sistem gagal melakukan <i>push</i> data laporan ke <i>database</i> karena tidak dapat mengakses <i>database</i>
Post-kondisi	Sistem berhasil menyimpan data laporan dan aktor dialihkan ke <i>activity dashboard</i>

Tabel 4.9 Skenario Lihat Laporan

Item	Deskripsi
Kode	F-SKIP-3
Nama <i>Use case</i>	Lihat Laporan
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat daftar laporan yang sebelumnya telah dibuat
Pra-kondisi	Aktor telah berada pada tampilan <i>dashboard</i> (halaman utama)
Tindakan	<ol style="list-style-type: none">1. Aktor menekan tombol menu “Lihat Laporan”2. Sistem menampilkan daftar data laporan3. Aktor dapat menekan salah satu laporan untuk melihat rincian dari laporan tersebut4. Sistem menampilkan detail laporan yang dipilih oleh aktor5. Aktor menekan tombol “LIHAT LOKASI PELAPOR”6. Sistem menampilkan lokasi pelapor dalam tampilan <i>google maps</i>

Tabel 4.9 Skenario Lihat Laporan (lanjutan)

Alternatif	<ol style="list-style-type: none">1. Sistem akan menampilkan <i>toast</i> “Gagal Mendapatkan Data Laporan, Pastikan Anda Terhubung Dengan Internet” saat sistem gagal mendapatkan daftar data laporan karena masalah koneksi internet2. Sistem akan menampilkan <i>toast</i> “Gagal Mendapatkan Data Laporan, Data Tidak Dapat Diakses” saat sistem gagal mendapatkan daftar data laporan karena sistem tidak dapat mengakses <i>database</i>
Post-kondisi	Aktor melihat detail informasi laporan

Tabel 4.10 Skenario Panggilan Darurat

Item	Deskripsi
Kode	F-SKIP-4
Nama <i>Use case</i>	Panggilan Darurat
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk melakukan panggilan darurat secara langsung terhadap daftar nomor panggilan darurat pada aplikasi
Pra-kondisi	Aktor telah berada pada tampilan <i>dashboard</i> (halaman utama)
Tindakan	<ol style="list-style-type: none">1. Aktor menekan tombol menu “Panggilan Darurat”2. Sistem melakukan pengambilan data nomor darurat dari <i>database</i> nomor darurat3. Sistem menampilkan daftar data nomor darurat4. Aktor dapat memilih salah satu nomor darurat yang tersedia dan menekan logo telepon5. Sistem akan mengarahkan ke tampilan untuk melakukan panggilan kepada nomor darurat yang dipilih
Alternatif	<ol style="list-style-type: none">1. Sistem akan menampilkan <i>toast</i> “Gagal Mendapatkan Data Nomor Darurat, Pastikan Anda Terhubung Dengan Internet” saat sistem gagal mendapatkan daftar data nomor darurat karena masalah koneksi internet
Post-kondisi	Aktor melihat daftar nomor darurat dan dapat melakukan panggilan kepada nomor darurat yang dipilih

Tabel 4.11 Skenario Panduan Aplikasi

Item	Deskripsi
Kode	F-SKIP-5
Nama <i>Use case</i>	Panduan Aplikasi
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat panduan pada setiap menu pada aplikasi.
Pra-kondisi	Aktor telah berada pada tampilan daftar panduan menu aplikasi
Tindakan	<ol style="list-style-type: none">1. Aktor menekan tombol menu “Panduan Aplikasi”2. Sistem menampilkan daftar panduan menu aplikasi3. Aktor dapat menekan salah satu menu pada aplikasi yang tersedia untuk melihat rincian panduan dari menu aplikasi tersebut4. Sistem menampilkan tampilan rincian panduan dari menu aplikasi yang dipilih
Alternatif	-
Post-kondisi	Aktor dapat melihat panduan menu aplikasi yang dipilih

Tabel 4.12 Skenario Mengirim Pesan Darurat

Item	Deskripsi
Kode	F-SKIP-6
Nama <i>Use case</i>	Mengirim Pesan Darurat
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk mengirimkan pesan darurat kepada pengguna yang lainnya
Pra-kondisi	Aktor telah berada pada tampilan <i>dashboard</i> (halaman utama)

Tabel 4.12 Skenario Mengirim Pesan Darurat (lanjutan)

Tindakan	<ol style="list-style-type: none"> 1. Aktor menekan tombol “PANIC” yang berada pada pojok kanan atas tampilan <i>dashboard</i> 2. Sistem menampilkan dialog dengan teks “Anda Menekan Tombol Panic! Apakah Benar Anda Sedang Dalam Keadaan Darurat ?” 3. Aktor menekan tombol “Ya” 4. Sistem menampilkan <i>progress bar</i> 5. Sistem melakukan pengambilan data lokasi aktor saat ini 6. Sistem melakukan penyimpanan data pesan darurat ke <i>database</i> 7. Sistem mengirimkan pesan darurat kepada pengguna yang lainnya 8. Sistem menampilkan dialog informasi berhasil mengirimkan pesan darurat dan menampilkan waktu tunggu selama 20 detik 9. Sistem menghilangkan waktu tunggu dan menampilkan tombol “OK” 10. Aktor menekan tombol “OK” 11. Sistem menghilangkan dialog dan kembali ke tampilan <i>dashboard</i>
Alternatif	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>toast</i> “Gagal Mendapatkan Data Lokasi,Silahkan Coba Lagi” jika sistem gagal melakukan pengambilan data lokasi aktor 2. Sistem menampilkan <i>toast</i> “Pembuatan Pesan Darurat Gagal, Pastikan Anda Terhubung Dengan Internet” jika sistem gagal menyimpan data pesan darurat ke <i>database</i> karena masalah koneksi internet 3. Sistem menampilkan <i>toast</i> “Pembuatan Pesan Darurat Gagal, Database Tidak Dapat Diakses” jika sistem gagal menyimpan data pesan darurat ke <i>database</i> karena sistem tidak dapat mengakses <i>database</i>
Post-kondisi	Sistem berhasil mengirimkan pesan darurat kepada pengguna yang lainnya

Tabel 4.13 Skenario Menerima Pesan Darurat

Item	Deskripsi
Kode	F-SKIP-7
Nama <i>Use case</i>	Menerima Pesan Darurat
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk menerima pesan darurat dari pengguna lain yang mengirimkan pesan darurat
Pra-kondisi	Aplikasi berada pada <i>task background device</i> dari aktor
Tindakan	<ol style="list-style-type: none">1. Sistem menampilkan tampilan pesan darurat yang berisi informasi data nama, alamat di perumahan, dan lokasi saat mengirimkan pesan darurat disertai dengan <i>device</i> yang bergetar selama 10 detik sebelum aktor menekan tombol “OK”2. Aktor menekan tombol “OK”3. Sistem menampilkan tampilan yang sebelumnya sedang diakses oleh aktor
Alternatif	<ol style="list-style-type: none">1. Sistem dapat gagal menerima pesan darurat jika <i>device</i> aktor tidak terhubung dengan internet, namun pesan darurat akan masuk setelah koneksi internet telah tersambung
Post-kondisi	Sistem berhasil menampilkan pesan darurat yang dikirimkan oleh pengguna lainnya

Tabel 4.14 Skenario Tentang Aplikasi

Item	Deskripsi
Kode	F-SKIP-8
Nama <i>Use case</i>	Tentang Aplikasi
Aktor	Pengguna
Deskripsi	Sistem harus mampu memberikan layanan bagi pengguna untuk melihat informasi aplikasi secara umum
Pra-kondisi	Aktor telah berada pada tampilan <i>dashboard</i> (halaman utama)
Tindakan	<ol style="list-style-type: none">1. Aktor menekan tombol menu “Tentang Aplikasi”2. Sistem menampilkan <i>activity</i> Tentang Aplikasi

Tabel 4.14 Skenario Tentang Aplikasi (lanjutan)

Alternatif	-
Post-kondisi	Aktor dapat melihat informasi aplikasi secara umum

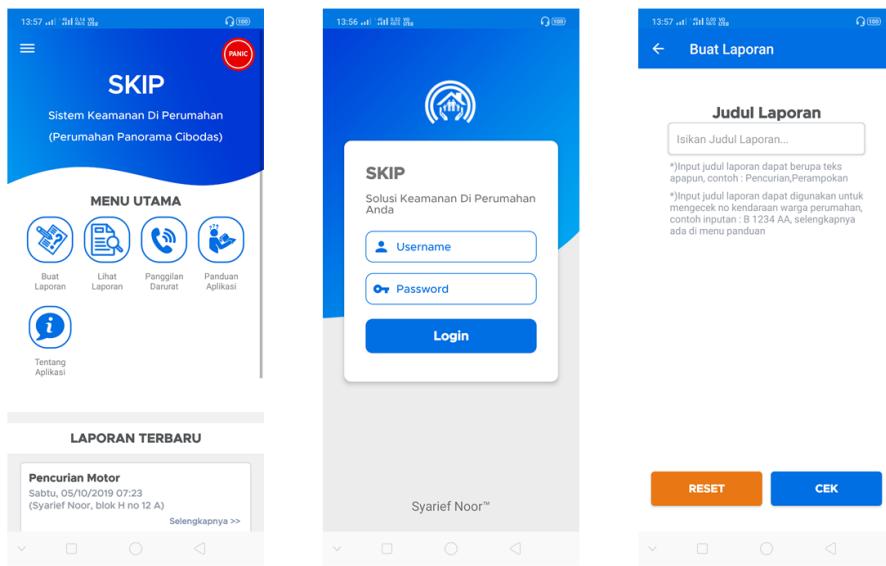
4.8 Analisis Data

Analisis data dilakukan untuk mengetahui data apa saja yang diperlukan pada aplikasi. Analisis data diperoleh dari hasil analisis kebutuhan fungsional dan kebutuhan non fungsional yang telah dilakukan sebelumnya. Hasil dari analisis data adalah sebagai berikut :

1. Data pengguna berupa id pengguna, *username*, *password*, *email*, nama lengkap, alamat, nomor *handphone*. Data *username* dan *password* digunakan untuk melakukan *login*.
2. Data laporan berupa id laporan, judul, nama pembuat laporan, alamat pembuat laporan, keterangan laporan, waktu pembuatan laporan dan lokasi pelapor. Data laporan digunakan untuk melakukan *posting* laporan dan melihat informasi laporan.
3. Data nomor darurat berupa id nomor darurat, nama instansi, tipe dan nomor handphone. Data nomor darurat digunakan oleh pengguna untuk melakukan panggilan secara langsung kepada nomor darurat yang dipilih.
4. Data pesan darurat berupa id pesan, token pengirim, nama pengirim, alamat pengirim, waktu pengiriman dan lokasi pengirim pesan darurat. Data pesan darurat digunakan untuk mengirim dan menerima pesan darurat.
5. Data kendaraan berupa id kendaraan, nomor plat kendaraan, jenis kendaraan, nama pemilik dan alamat pemilik. Data kendaraan digunakan untuk dilakukan pengecekan nomor plat kendaraan dengan judul laporan yang terdapat pada fungsionalitas buat laporan.
6. Data token berupa id token, nama pengguna, dan token. Data token digunakan untuk identitas pengiriman dan penerimaan notifikasi pemberitahuan laporan baru dan pesan darurat.

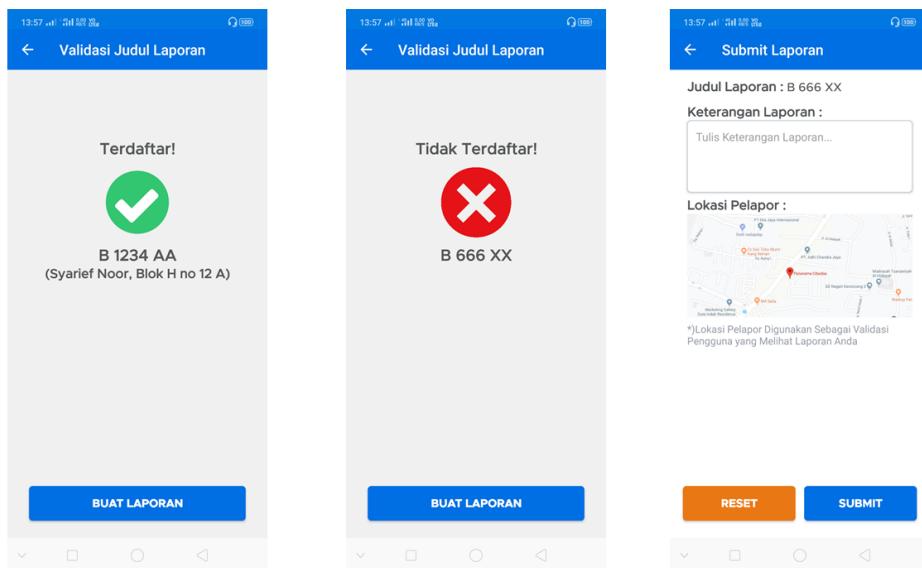
4.9 Tampilan Antarmuka *Final Prototype*

Pada bagian ini penulis akan mencantumkan tampilan antarmuka *final prototype* dari hasil *prototyping* yang telah dilakukan. Tampilan antarmuka *final prototype* ditunjukkan pada gambar 4.2 sampai dengan gambar 4.7.



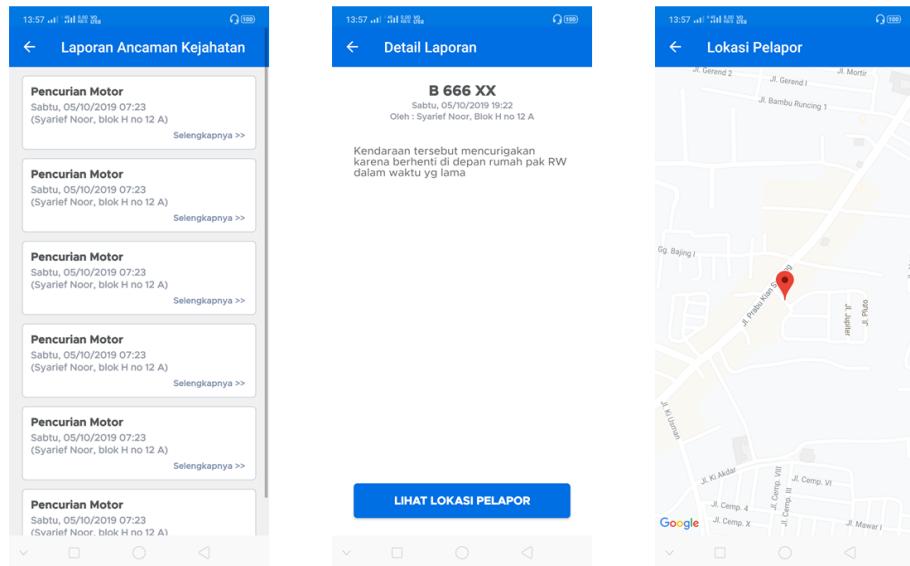
Gambar 4.2 Tampilan antarmuka *final prototype* (bagian 1)

Gambar 4.2 menunjukkan tampilan antarmuka *final prototype* pada bagian *dashboard*, *login* dan tampilan pengecekan judul laporan.



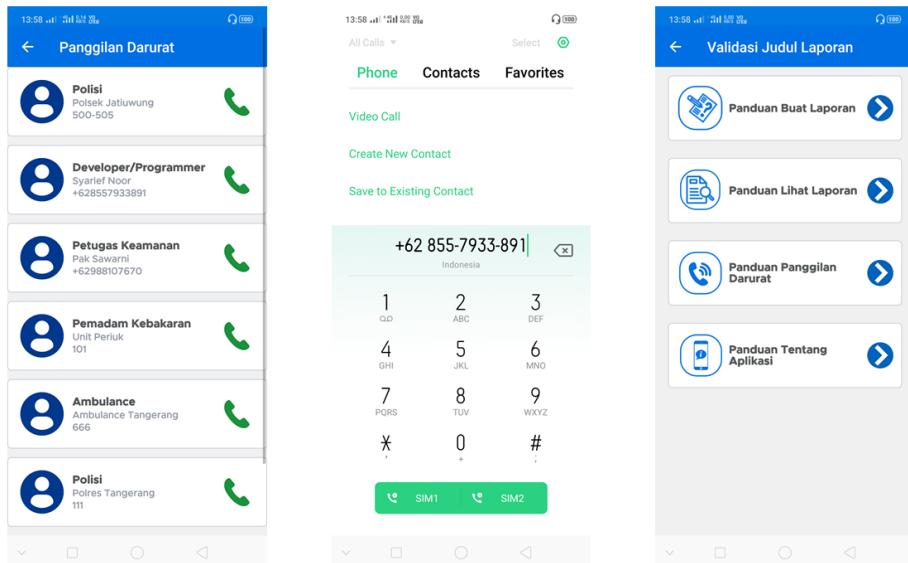
Gambar 4.3 Tampilan antarmuka *final prototype* (bagian 2)

Pada gambar 4.3 menunjukkan tampilan antarmuka *final prototype* pada bagian tampilan hasil pengecekan terdaftar, hasil pengecekan tidak terdaftar dan tampilan untuk submit laporan.



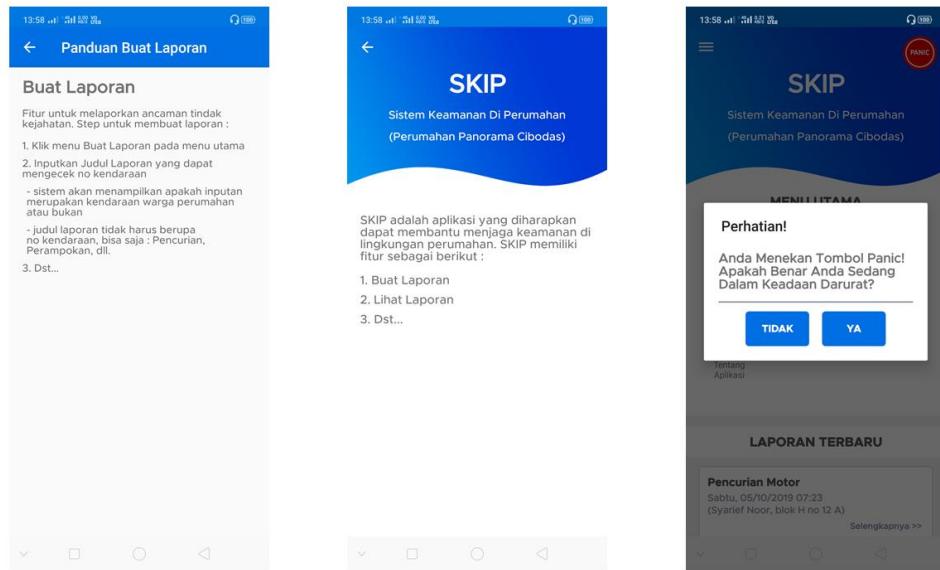
Gambar 4.4 Tampilan antarmuka *final prototype* (bagian 3)

Pada gambar 4.4 menunjukkan tampilan antarmuka *final prototype* pada bagian tampilan daftar laporan, detail laporan dan tampilan lokasi pelapor.



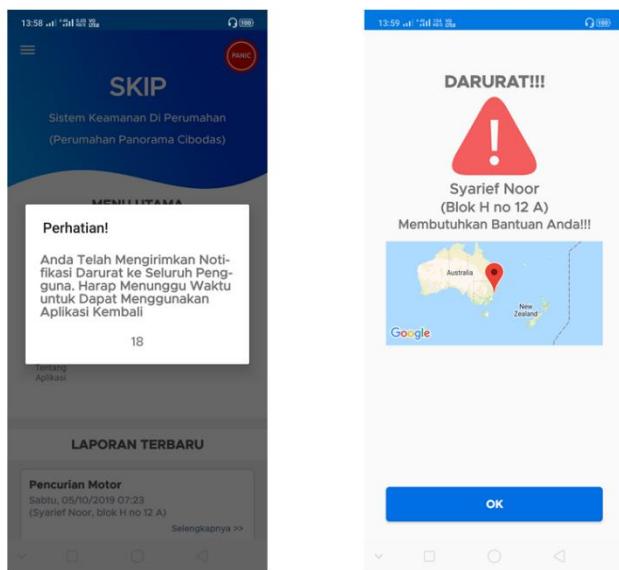
Gambar 4.5 Tampilan antarmuka *final prototype* (bagian 4)

Pada gambar 4.5 menunjukkan tampilan antarmuka *final prototype* pada bagian tampilan daftar nomor darurat, tampilan untuk melakukan panggilan, dan tampilan pilihan panduan menu pada aplikasi.



Gambar 4.6 Tampilan antarmuka *final prototype* (bagian 5)

Pada gambar 4.6 menunjukkan tampilan antarmuka *final prototype* pada bagian tampilan panduan menu buat laporan, tampilan tentang aplikasi, dan tampilan dialog konfirmasi untuk mengirim pesan darurat.



Gambar 4.7 Tampilan antarmuka *final prototype* (bagian 6)

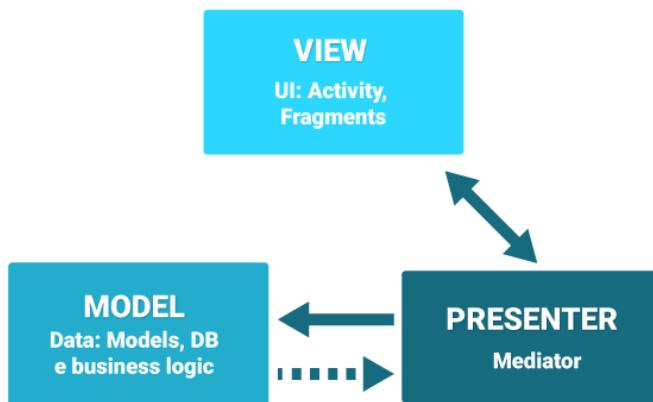
Pada gambar 4.7 menunjukkan tampilan antarmuka *final prototype* pada bagian tampilan dialog setelah mengirim pesan darurat dan tampilan pesan darurat yang masuk.

BAB 5 PERANCANGAN

Pada bab ini membahas tentang perancangan dari aplikasi SKIP. Perancangan akan terdiri dari perancangan arsitektur sistem, *physical data model*, *sequence diagram*, *class diagram* serta perancangan antarmuka yang terdiri dari *screenflow diagram* dan perancangan antarmuka itu sendiri.

5.1 Perancangan Arsitektur Sistem

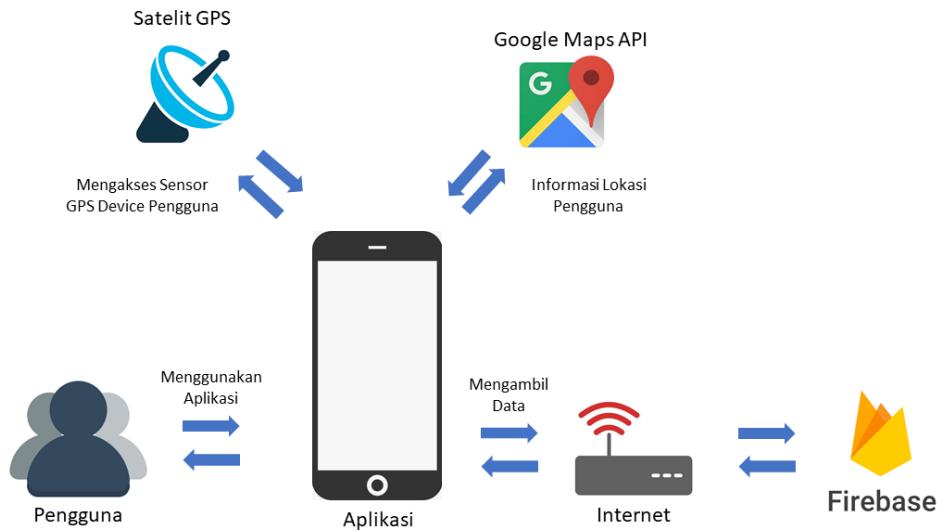
Aplikasi SKIP dibangun dengan menggunakan pendekatan arsitektur *Model View Presenter* (MVP). MVP merupakan pola perancangan aplikasi yang memisahkan fungsi untuk tampilan dan fungsional yang berjalan pada aplikasi. Pengembang memilih menggunakan MVP dengan harapan dapat membuat alur program yang mudah dipahami oleh pembaca atau pengembang lainnya. *Model* merupakan bagian yang berisi *logic* dan interaksi data dengan *database*. *View* merupakan bagian yang berisi tampilan untuk menampilkan data dan interaksi dengan pengguna. *Presenter* merupakan bagian yang menjadi penghubung komunikasi antara *View* dengan *Model*. Skema dari arsitektur MVP ditunjukkan pada gambar 5.1.



Gambar 5.1 Skema Arsitektur MVP

(Sumber : Sianturi, 2018)

Sementara itu skema aplikasi SKIP secara umum ditunjukkan pada gambar 5.2. Pada gambar tersebut menunjukkan alur dari aplikasi yang digunakan oleh pengguna, aplikasi berhubungan dengan sensor GPS dan Google Maps API untuk mendapatkan data lokasi, aplikasi perlu terhubung dengan internet untuk dapat mengakses data yang tersimpan pada Firebase.

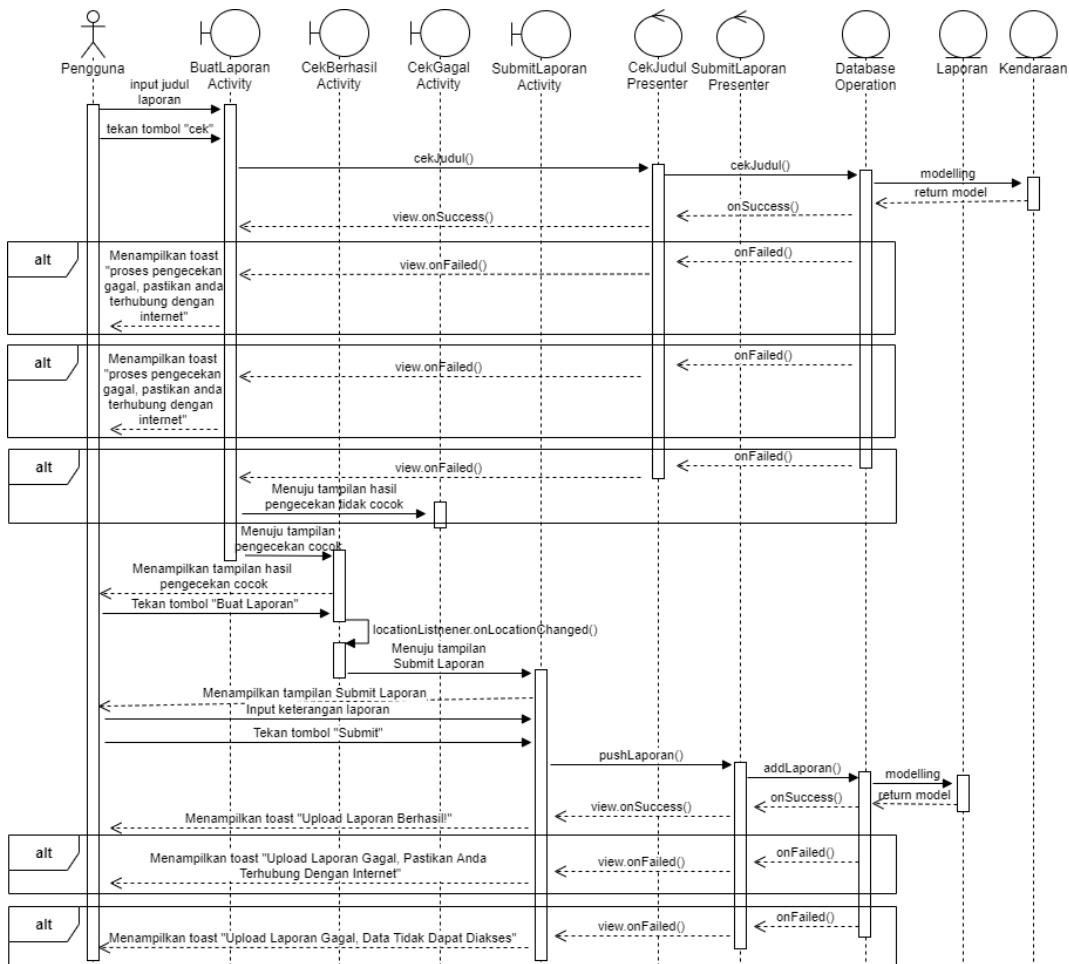


Gambar 5.2 Arsitektur sistem

5.2 Perancangan *Sequence Diagram*

Perancangan *sequence diagram* berisi gambaran interaksi yang terjadi pada sistem secara berurutan untuk mencapai tujuan dari kebutuhan fungsional. Pada perancangan *sequence diagram* tidak menggambarkan *sequence diagram* untuk seluruh kebutuhan fungsional, namun hanya fitur utama pada sistem. Berikut adalah hasil perancangan *sequence diagram* yang ditunjukkan pada sub bab selanjutnya.

5.2.1 Sequence Diagram Buat Laporan



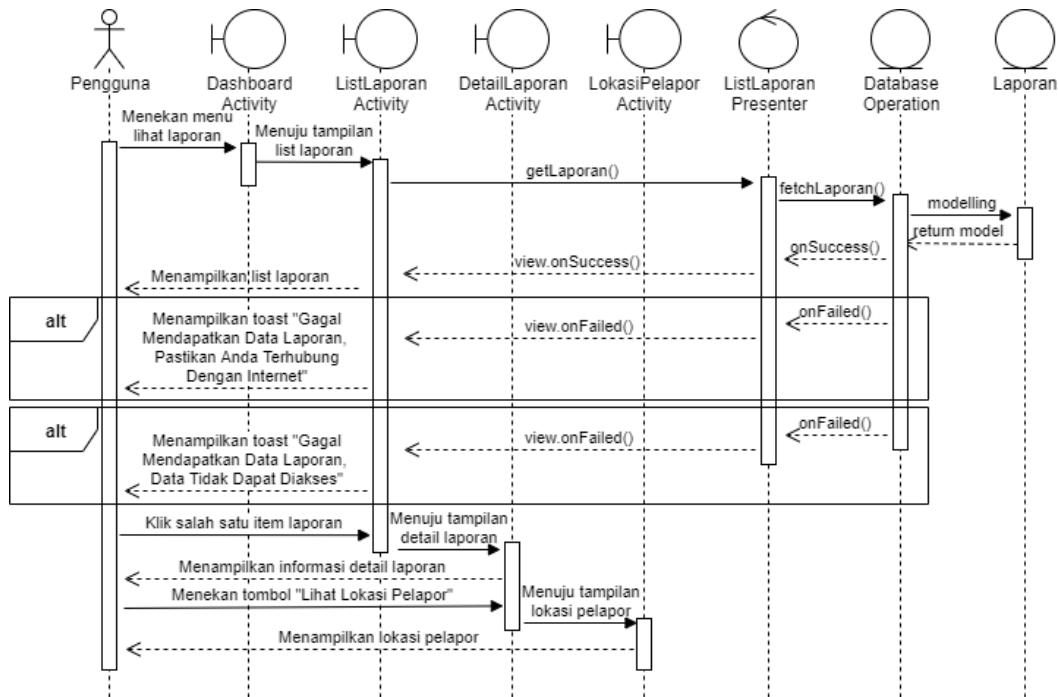
Gambar 5.3 Sequence diagram buat laporan

Pada gambar 5.3 menunjukkan *sequence diagram* dari fungsionalitas buat laporan. Diawali dengan pengguna memasukkan judul laporan dan menekan tombol “cek”, kemudian akan menuju tampilan Hasil Cek Judul pada *class* HasilCekJudulActivity. Setelah itu memanggil *method* cekJudul() pada *class* CekJudulPresenter dan dilanjutkan memanggil *method* cekJudul() pada *class* DatabaseOperation. Jika pengecekan berhasil maka akan memanggil *method* onSuccess() pada *class* CekJudulPresenter dan dilanjutkan memanggil *method* onSucess() melalui objek *view* pada *class* HasilCekJudulActivity. Lalu akan menampilkan hasil pengecekan judul laporan apakah terdaftar atau tidak terdaftar kepada pengguna. Jika pengecekan gagal maka akan memanggil *method* onFailed() pada *class* CekJudulPresenter dan dilanjutkan memanggil *method* onFailed() melalui objek *view* pada *class* BuatLaporanActivity. Lalu akan menampilkan *toast* sesuai dengan kondisi kegagalan yaitu masalah koneksi internet atau *database* yang tidak dapat diakses.

Alur selanjutnya yaitu pengguna akan menekan tombol “Buat Laporan” pada *class* CekBerhasilActivity. Kemudian akan melakukan pengambilan data lokasi pengguna saat ini melalui variabel *locationListener*. Ketika data lokasi telah

didapatkan akan menuju tampilan `SubmitLaporanActivity`. Selanjutnya pengguna memasukkan keterangan laporan dan menekan tombol “Submit”, lalu sistem memanggil *method* `pushLaporan()` pada *class* `SubmitLaporanPresenter` dan dilanjutkan memanggil *method* `addLaporan()` pada *class* `DatabaseOperation`. Selanjutnya dilakukan pemodelan data pada *class* `Laporan`. Jika penambahan data laporan berhasil maka akan memanggil *method* `onSuccess()` pada *class* `DatabaseOperation` dan dilanjutkan *method* `onSuccess()` melalui objek *view* pada *class* `SubmitLaporanActivity`. Lalu akan menampilkan *toast* pemberitahuan upload laporan berhasil. Jika penambahan data laporan gagal maka akan memanggil *method* `onFailed()` pada *class* `DatabaseOperation` dan dilanjutkan *method* `onFailed()` melalui objek *view* pada *class* `SubmitLaporanActivity`. Lalu akan menampilkan *toast* sesuai dengan kondisi kegagalan yaitu masalah koneksi internet atau *database* yang tidak dapat diakses.

5.2.2 Sequence Diagram Lihat Laporan



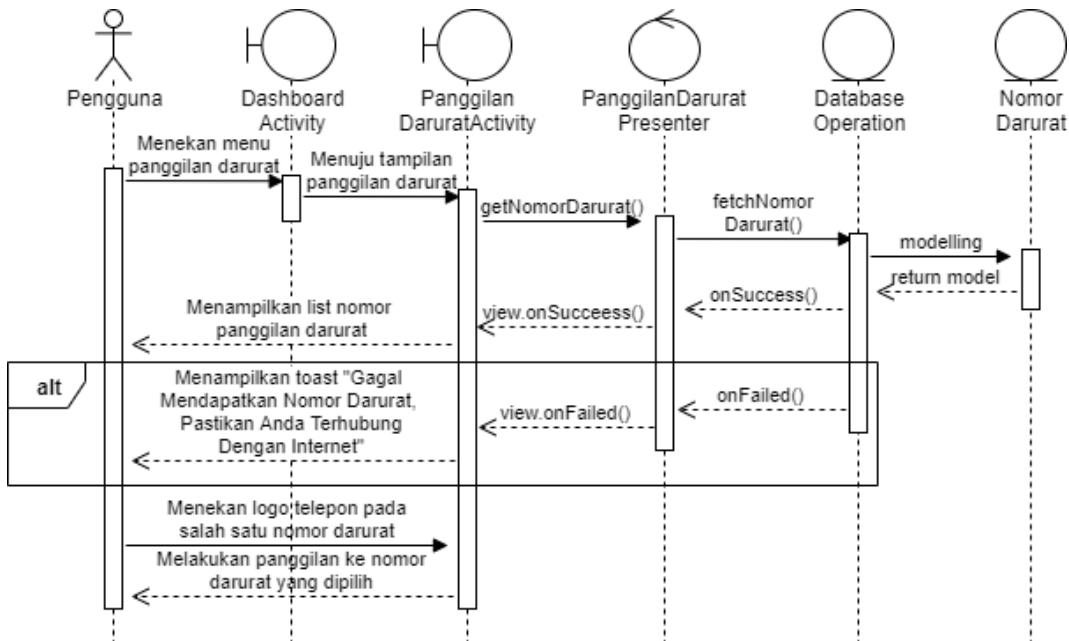
Gambar 5.4 Sequence diagram lihat laporan

Pada gambar 5.4 menunjukkan *sequence diagram* dari fungsionalitas lihat laporan. Diawali dengan pengguna menekan menu lihat laporan pada *DashboardActivity*, lalu sistem akan menuju ke tampilan *list laporan*. Kemudian sistem memanggil *method* `getLaporan()` pada *class* `ListLaporanPresenter` dan dilanjutkan memanggil *method* `fetchLaporan()` pada *class* `DatabaseOperation`. Jika pengecekan berhasil maka akan memanggil *method* `onSuccess()` pada *class* `ListLaporanPresenter` dan dilanjutkan memanggil *method* `onSuccess()` melalui objek *view* pada *class* `ListLaporanActivity`. Lalu akan menampilkan *list* data laporan kepada pengguna. Jika pengecekan gagal maka akan memanggil *method* `onFailed()` pada *class* `ListLaporanPresenter` dan dilanjutkan memanggil *method* `onFailed()` melalui objek *view* pada *class* `ListLaporanActivity`. Lalu akan

menampilkan *toast* pemberitahuan gagal menampilkan data laporan yang disebabkan masalah koneksi internet atau *database* yang tidak dapat diakses.

Alur selanjutnya yaitu pengguna akan menekan salah satu item laporan, lalu sistem akan menuju ke tampilan detail laporan dan menampilkan detail data pada laporan yang dipilih. Kemudian pengguna menekan tombol “Lihat Lokasi Pelapor” dan sistem akan menuju ke tampilan lokasi pelapor untuk menampilkan lokasi pelapor pada tampilan google maps di `LokasiPelaporActivity`.

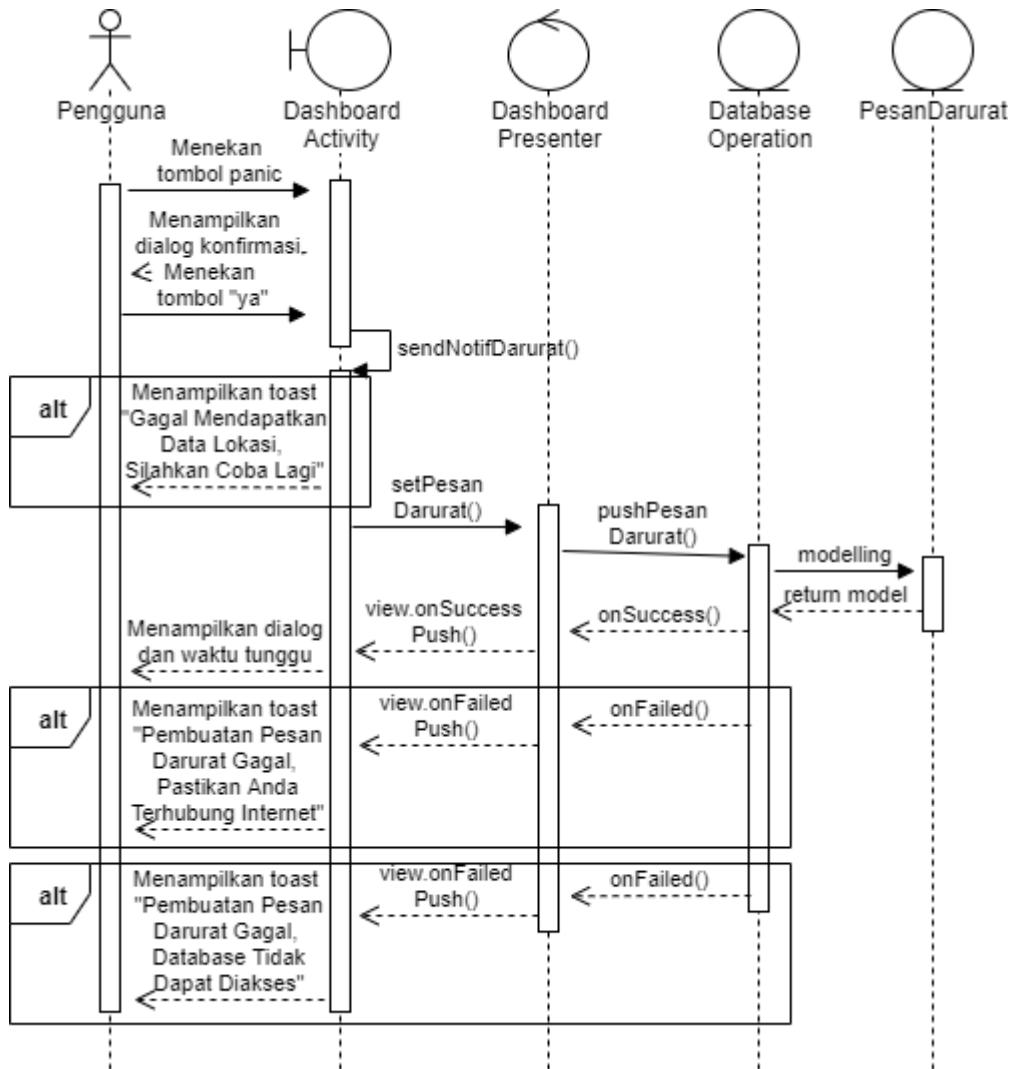
5.2.3 Sequence Diagram Panggilan Darurat



Gambar 5.5 Sequence diagram panggilan darurat

Pada gambar 5.5 menunjukkan *sequence diagram* dari fungsionalitas panggilan darurat. Diawali dengan pengguna menekan menu panggilan darurat, lalu sistem akan menuju tampilan panggilan darurat. Kemudian sistem memanggil *method* `getNomorDarurat()` pada *class* `PanggilanDaruratPresenter` dan dilanjutkan memanggil *method* `fetchNomorDarurat()` pada *class* `DatabaseOperation`. Jika pengambilan data berhasil maka akan dilakukan pemodelan pada *class* `NomorDarurat`, lalu memanggil *method* `onSuccess()` pada *class* `PanggilanDaruratPresenter` dan dilanjutkan memanggil *method* `onSuccess()` melalui objek *view* pada *class* `PanggilanDaruratActivity`. Lalu akan menampilkan *list* nomor panggilan darurat kepada pengguna. Jika pengambilan data gagal maka akan memanggil *method* `onFailed()` pada *class* `PanggilanDaruratPresenter` dan dilanjutkan memanggil *method* `onFailed()` melalui objek *view* pada *class* `PanggilanDaruratActivity`. Lalu akan menampilkan *toast* pemberitahuan gagal menampilkan *list* nomor panggilan darurat. Selanjutnya pengguna menekan logo telepon pada salah satu nomor darurat yang dipilih, lalu sistem akan melakukan panggilan kepada nomor telepon yang dituju.

5.2.4 Sequence Diagram Mengirim Pesan Darurat



Gambar 5.6 Sequence diagram mengirim pesan darurat

Pada gambar 5.6 menunjukkan *sequence diagram* dari fungsionalitas mengirim pesan darurat. Diawali dengan pengguna menekan tombol “panic”, lalu sistem akan menampilkan dialog konfirmasi telah menekan tombol “panic”. Kemudian pengguna menekan tombol “Ya”, lalu sistem memanggil *method* `sendNotifDarurat()` pada *class* `DashboardActivity`. Jika pengambilan data lokasi gagal maka sistem akan menampilkan *toast* pemberitahuan gagal melakukan pengambilan data lokasi kepada pengguna. Selanjutnya sistem memanggil *method* `setPesanDarurat()` pada *class* `DashboardPresenter` dan dilanjutkan memanggil *method* `pushPesanDarurat()` pada *class* `DatabaseOperation`. Jika pengiriman pesan berhasil maka sistem melakukan pemodelan pada *class* `PesanDarurat`, lalu memanggil *method* `onSuccess()` pada *class* `DashboardPresenter` dan dilanjutkan memanggil *method* `onSuccessPush()` melalui objek *view* pada *class* `DashboardActivity`. Lalu akan menampilkan dialog berhasil dan menampilkan waktu tunggu. Jika pengiriman pesan berhasil maka sistem memanggil *method* `onFailed()` pada *class* `DashboardPresenter` dan dilanjutkan memanggil *method*

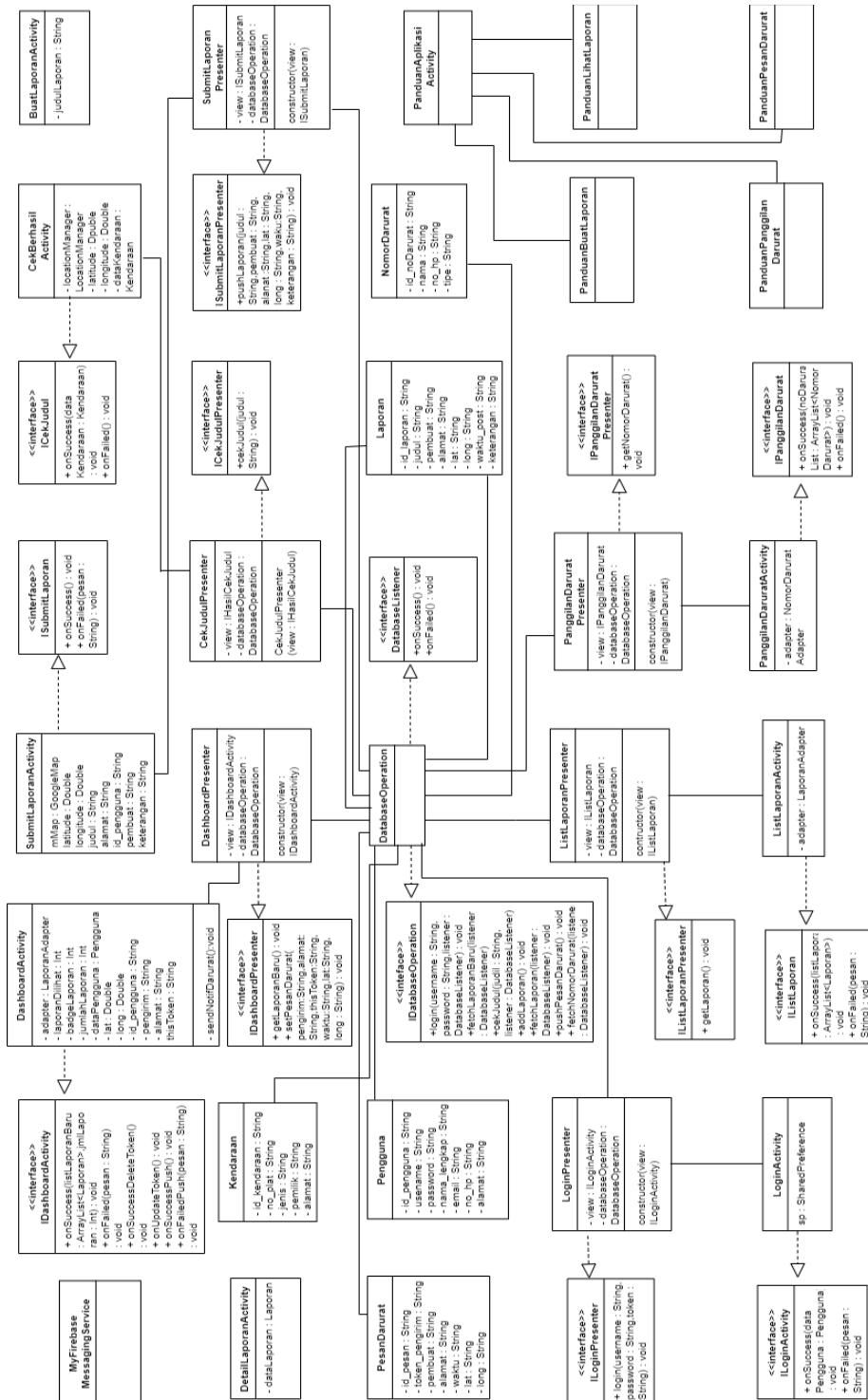
`onFailedPush()` melalui objek `view` pada `class DashboardActivity`. Lalu akan menampilkan `toast` pemberitahuan gagal mengirim pesan darurat kepada pengguna dengan penyebab masalah koneksi internet atau `database` tidak dapat diakses.

5.3 Perancangan Class Diagram

Pada gambar 5.7 merupakan gambaran *class diagram* dari aplikasi SKIP. Pada *class diagram* tersebut terdapat dua macam relasi yaitu asosiasi dan implementasi untuk *interface*. Aplikasi SKIP dikembangkan dengan arsitektur MVP (*Model View Presenter*) sehingga *class* yang dibuat akan dikelompokkan berdasarkan konsep MVP. Pada bagian model berisi *class-class* yang berhubungan dengan entitas dan *database*, pada *class diagram* diatas yaitu `class DatabaseOperation` sebagai pusat operasi yang dilakukan terhadap *database*, *interface* `DatabaseListener` sebagai kontrak untuk hasil proses operasi, *interface* `IDatabaseOperation` sebagai kontrak *method* untuk operasi pada `class DatabaseOperation`.

Pada bagian *presenter* berisi *class-class* yang bertujuan menghubungkan antara *view* dengan model. *Presenter* diperlukan agar pada bagian *view* dikhususkan untuk menangani bagian tampilan, model untuk proses olah data, sedangkan *presenter* sebagai penghubung antara keduanya. Pada *class diagram* diatas *presenter* terdiri dari *class* dan *interface* sebagai kontrak yang berisi *method* untuk menghubungkan *view* dengan model. Beberapa *presenter* pada *class diagram* diatas yaitu `LoginPresenter` dengan *interface* `ILoginPresenter`, `PanggilanDaruratPresenter` dengan *interface* `IPanggilanDaruratPresenter`, `SubmitLaporanPresenter` dengan *interface* `ISubmitLaporanPresenter`, dan yang lainnya.

Pada bagian *view* berisi *class-class* yang berkaitan dengan tampilan. Pada *class diagram* diatas bagian *view* terdiri dari *class* dan *interface* jika pada *class* tersebut terdapat proses olah data. *Interface* berisi kontrak yang berisi *method-method* untuk menerima respon setelah melakukan proses olah data. Beberapa *view* pada *class diagram* diatas yaitu `LoginActivity` dengan *interface* `ILoginActivity` karena untuk melakukan *login* membutuhkan verifikasi dengan *database*, `LihatLaporanActivity` dengan *interface* `ILihatLaporan` untuk proses pengambilan data laporan dari *database*, `TentangAplikasiActivity` dan yang lainnya.



Gambar 5.7 Class diagram

5.4 Physical Data Model

Physical data model adalah pemodelan rancangan basis data yang digunakan pada aplikasi SKIP. Pada *physical data model* yang ditunjukkan gambar 5.8, terdapat 6 tabel yaitu pengguna, laporan, nomor_darurat, kendaraan, data_token dan pesan_darurat. Pada setiap tabel terdapat kolom beserta tipe data. Di dalam tabel pengguna terdapat kolom id_pengguna yang merupakan *primary key*, username, password, email, nama_lengkap, alamat dan no_hp. Di dalam tabel laporan terdapat kolom id_laporan sebagai *primary key*, judul, keterangan, pembuat, alamat, lat, long dan waktu_post. Di dalam tabel pesan_darurat terdapat kolom id_pesan sebagai *primary key*, pembuat, alamat, token_pengirim, lat, long dan waktu. Pada tabel nomor_darurat terdapat kolom id_noDarurat sebagai *primary key*, nama, no_hp dan tipe. Pada tabel kendaraan terdapat kolom id_kendaraan sebagai *primary key*, jenis, no_plat, alamat dan pemilik. Pada tabel data_token terdapat kolom id sebagai *primary key*, pengguna, dan token.

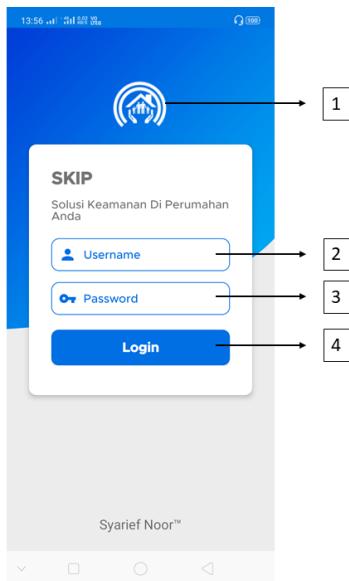


Gambar 5.8 *Physical data model*

5.5 Perancangan Antarmuka

Pada bagian perancangan antarmuka menjelaskan mengenai informasi detail dari setiap tampilan yang ada pada aplikasi SKIP. Perancangan antarmuka pada aplikasi SKIP menggunakan screenshot dari tampilan *prototype* yang telah diujikan kepada calon pengguna.

5.5.1 Tampilan *Login*



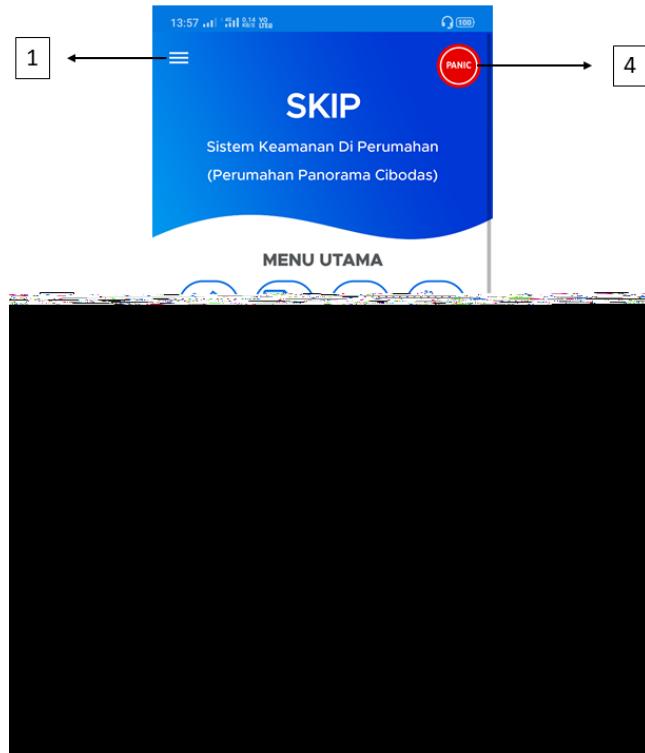
Gambar 5.9 Tampilan perancangan *login*

Penjelasan setiap elemen yang ada pada gambar 5.9 tampilan *login* dijelaskan pada tabel 5.1 berikut :

Tabel 5.1 Informasi elemen pada tampilan *login*

No	Penjelasan	Detail Fungsi
1	Logo aplikasi SKIP	Digunakan untuk memperjelas aplikasi yang sedang dibuka oleh pengguna
2	Masukan <i>username</i>	Digunakan untuk memasukkan masukan <i>username</i>
3	Masukan <i>password</i>	Digunakan untuk memasukkan masukan <i>password</i>
4	Tombol <i>Login</i>	Digunakan untuk melakukan proses <i>login</i>

5.5.2 Tampilan *Dashboard*



Gambar 5.10 Tampilan perancangan *dashboard*

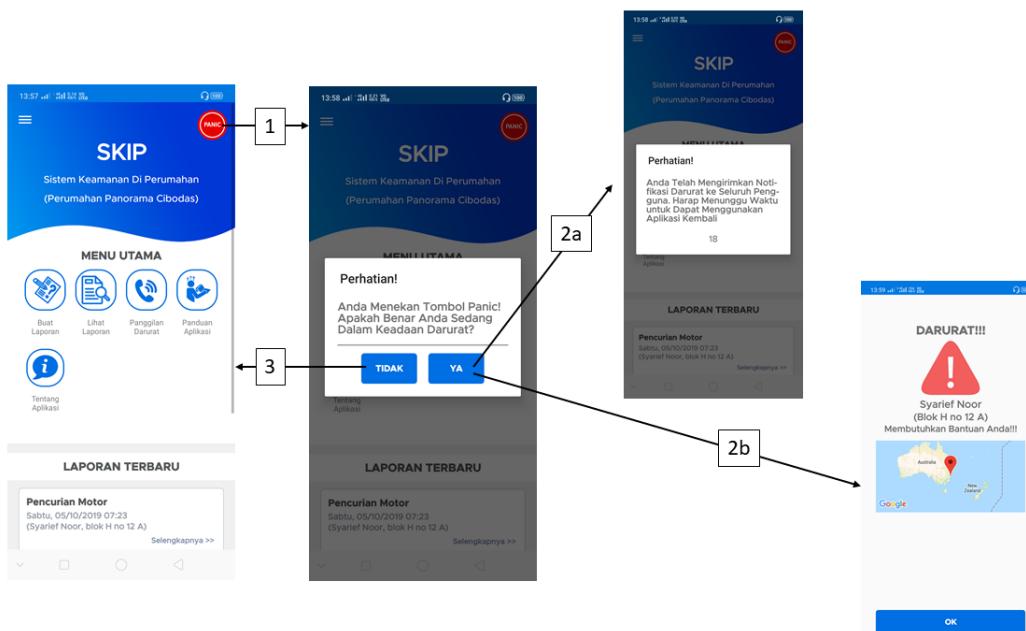
Penjelasan setiap elemen yang ada pada gambar 5.10 tampilan perancangan *dashboard* dijelaskan pada tabel 5.2 berikut :

Tabel 5.2 Informasi elemen pada tampilan *dashboard*

No	Penjelasan	Detail Fungsi
1	Tombol navigasi	Digunakan untuk menunjukkan tampilan menu navigasi
2	Tombol menu buat laporan	Digunakan untuk menuju tampilan menu buat laporan
3	Tombol menu tentang aplikasi	Digunakan untuk menuju tampilan menu tentang aplikasi
4	Tombol "Panic"	Digunakan untuk mengirimkan pesan darurat kepada pengguna lainnya
5	Tombol menu panduan aplikasi	Digunakan untuk menuju tampilan menu panduan aplikasi
6	Tombol menu panggilan darurat	Digunakan untuk menuju tampilan menu panggilan darurat

Tabel 5.20 Keterangan screenflow mengirim dan menerima pesan darurat (lanjutan)

3	Setelah pengguna menekan tombol “Tidak” pada dialog, sistem akan mengarahkan kembali ke tampilan dashboard
---	--



Gambar 5.28 Screenflow diagram mengirim dan menerima pesan darurat

BAB 6 IMPLEMENTASI

Pada bab ini membahas mengenai implementasi dari aplikasi SKIP. Beberapa hal yang terkait dengan implementasi aplikasi SKIP yaitu spesifikasi perangkat keras dan perangkat lunak, batasan-batasan implementasi, implementasi data, dan algoritme.

6.1 Spesifikasi Sistem

Pada bagian spesifikasi sistem menjelaskan mengenai detail informasi dari perangkat keras dan perangkat lunak yang digunakan dalam pengembangan aplikasi SKIP.

6.1.1 Spesifikasi Perangkat Keras

Pengembangan aplikasi SKIP dilakukan dengan menggunakan perangkat komputer dan *smartphone* untuk pengujian dengan spesifikasi seperti pada tabel 6.1.

Tabel 6.1 Spesifikasi perangkat keras

Komponen	Spesifikasi
ASUS X550Z	1. <i>Processor</i> AMD A10-7400P 2. <i>Memory</i> 8GB DDR3L 3. <i>Graphics</i> AMD Radeon® R5 M230 + Radeon® R7 M265 DX Dual Graphics with 2GB DDR3 VRAM
Xiaomi Redmi 3 Pro	1. <i>Processor</i> Snapdragon 615 1.5 GHz 2. RAM 3GB 3. <i>Memory Internal</i> 32GB
Oppo K3	1. <i>Processor</i> Snapdragon 710 2.2 GHz 2. RAM 6GB 3. <i>Memory Internal</i> 64GB

6.1.2 Spesifikasi Perangkat Lunak

Pengembangan aplikasi SKIP menggunakan perangkat keras yang didukung oleh perangkat lunak dengan spesifikasi seperti pada tabel 6.2 berikut.

Tabel 6.2 Spesifikasi perangkat lunak

Komponen	Spesifikasi
ASUS X550Z	1. OS, Windows 10 Pro 2. IDE, Android Studio 3. Browser, Opera
Xiaomi Redmi 3 Pro	1. Android Lollipop 5.1.1
Oppo K3	1. Android Pie 9

6.2 Batasan-batasan Implementasi

Aplikasi SKIP memiliki beberapa batasan implementasi dalam pengembangannya. Batasan-batasan tersebut antara lain sebagai berikut :

1. Aplikasi SKIP hanya dapat dijalankan pada smartphone dengan sistem operasi android minimal versi 4.4 (KitKat) dan maksimum versi 9.0 (Pie).
2. Aplikasi SKIP membutuhkan koneksi internet untuk interaksi dengan *database*.
3. Aplikasi SKIP memerlukan sensor GPS yang aktif untuk menggunakan layanan berbasis lokasi dari Google Map API.

6.3 Implementasi Data

Implementasi data berisi bentuk data hasil dari implementasi yang telah dilakukan pada aplikasi SKIP dengan menggunakan firebase *realtime database*. Implementasi data dibuat sesuai dengan pemodelan yang telah dilakukan pada bab sebelumnya yaitu pada bagian *Physical Data Model*. Berikut adalah hasil implementasi data pada aplikasi SKIP.



Gambar 6.1 Implementasi data token

Pada gambar 6.1 berisi struktur data token pada aplikasi SKIP. Data token digunakan sebagai identitas untuk pengiriman dan penerimaan notifikasi dengan menggunakan Firebase Cloud Messaging. Pada data token terdapat kolom id sebagai identitas di *database*, kolom pengguna yang berisi informasi nama pengguna, kolom token yang berisi token dari *smartphone* yang digunakan oleh pengguna.



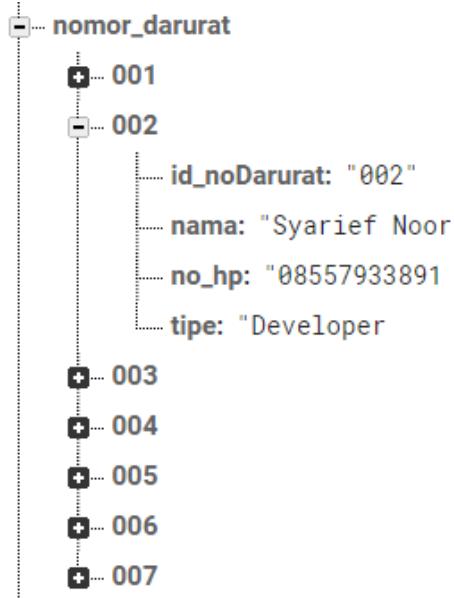
Gambar 6.2 Implementasi data kendaraan

Pada gambar 6.2 berisi struktur data kendaraan pada aplikasi SKIP. Data kendaraan digunakan untuk melakukan pengecekan nomor plat kendaraan pada saat melakukan *input* judul laporan. Pada data kendaraan terdapat kolom alamat, kolom id_kendaraan sebagai identitas tiap data, kolom jenis kendaraan, kolom no_plat dan kolom nama pemilik.



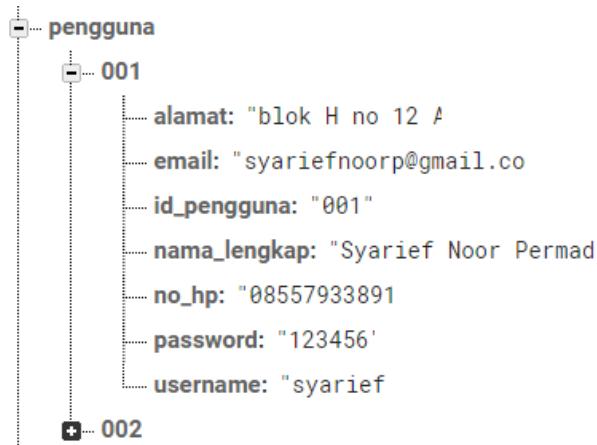
Gambar 6.3 Implementasi data laporan

Pada gambar 6.3 berisi struktur data laporan pada aplikasi SKIP. Data laporan berasal dari pengguna yang melaporkan suatu keadaan melalui aplikasi SKIP. Seluruh data laporan akan ditampilkan kepada pengguna. Pada data laporan terdapat kolom alamat, kolom id_laporan sebagai identitas tiap data, kolom judul laporan, kolom keterangan laporan, kolom lat yang berisi koordinat *latitude*, kolom long yang berisi koordinat *longitude*, kolom pembuat yang berisi nama pembuat laporan dan kolom waktu_post yang berisi hari, tanggal dan jam waktu pelaporan.



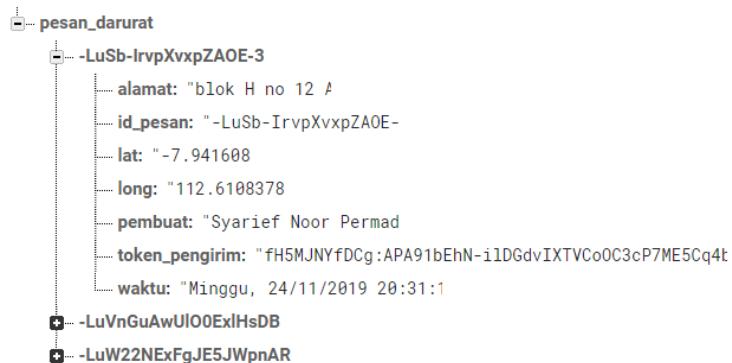
Gambar 6.4 Implementasi data nomor darurat

Pada gambar 6.4 berisi struktur data nomor darurat pada aplikasi SKIP. Data nomor darurat akan ditampilkan pada menu panggilan darurat agar pengguna dapat melakukan panggilan secara langsung kepada nomor darurat yang dituju. Pada data nomor darurat terdapat kolom id_noDarurat sebagai identitas dari tiap data, kolom nama, kolom no_hp dan kolom tipe yang berisi keterangan instansi.



Gambar 6.5 Implementasi data pengguna

Pada gambar 6.5 berisi struktur data pengguna pada aplikasi SKIP. Data pengguna digunakan untuk melakukan *login* pada aplikasi SKIP dan sebagai tambahan data identitas untuk data yang lainnya. Pada data pengguna terdapat kolom alamat, kolom email, kolom id_pengguna sebagai identitas tiap data, kolom nama_lengkap, kolom no_hp, kolom password dan kolom username. Data pengguna didaftarkan secara manual oleh calon pengguna kepada *developer*.



Gambar 6.6 Implementasi data pesan darurat

Pada gambar 6.6 berisi struktur data pesan darurat pada aplikasi SKIP. Data pesan darurat berasal dari pengguna yang melakukan pelaporan pesan darurat dengan menekan tombol *panic*. Pada data pesan darurat terdapat kolom alamat, kolom *id_pesan* sebagai identitas data, kolom *lat* yang berisi koordinat *latitude* saat melakukan pelaporan, kolom *long* yang berisi koordinat *longitude* saat melakukan pelaporan, kolom pembuat laporan, kolom *token_pengirim* yang berisi token sebagai identitas pengirim notifikasi dan kolom *waktu* pelaporan yang berisi hari, tanggal dan jam pelaporan.

6.4 Implementasi Algoritme

Pada bagian implementasi algoritme berisi pemaparan kode program pada fungsi-fungsi utama aplikasi SKIP dan disertai dengan penjelasan alur kerjanya. Fungsi-fungsi yang akan dijelaskan pada implementasi algoritme yaitu pada fitur membuat laporan, melihat laporan, melakukan panggilan darurat dan mengirim pesan darurat.

6.4.1 Algoritme membuat laporan

Implementasi kode program pada method `cekJudul()` yang berada pada *class DatabaseOperation*, pada kode program tersebut digunakan untuk melakukan pengecekan judul laporan apakah judul laporan berupa nomor plat kendaraan warga perumahan Panorama Cibodas atau bukan, jika judul laporan cocok dengan *database* kendaraan warga Perumahan Panorama Cibodas maka fungsi akan mengembalikan data kendaraan melalui *listener*. Kode program dari method `cekJudul()` dapat dilihat pada tabel 6.3 dan penjelasan algoritme pada tabel 6.4. Kemudian setelah melakukan pengecekan judul, pengguna dapat melanjutkan untuk membuat laporan yang ditangani oleh method `addLaporan()` yang berada pada *class DatabaseOperation*. Kode program dari method `addLaporan()` dapat dilihat pada tabel 6.5 dan penjelasan algoritme pada tabel 6.6.

Nama *Class* : *DatabaseOperation*

Nama *Method* : `cekJudul()`

Tabel 6.3 Source code method cekJudul()

No	Source Code
1	override fun cekJudul(judul: String, listener: DatabaseListener.CekJudulListener) {
2	//operasi di database
3	if (!isConnectInternet()){ //cek koneksi internet
4	listener.onFailed("offline")
5	}else{
6	ref_kendaraan.addListenerForSingleValueEvent(object : ValueEventListener{
7	override fun onCancelled(p0: DatabaseError) { listener.onFailed("error") }
8	override fun onDataChange(dataSnapshot : DataSnapshot) { kendaraanList.clear() for (kendaraanSnapshot in dataSnapshot.getChildren()) { var kendaraan = kendaraanSnapshot.getValue(Kendaraan::class.java) kendaraanList.add(kendaraan!!) } var statusJudul = false for (i in 0..kendaraanList.size-1){ if(judul.equals(kendaraanList[i]. no_plat,true)){ statusJudul = true dataKendaraan = kendaraanList[i] } } if (statusJudul){ listener.onSuccess(dataKendaraan) }else{ listener.onFailed("not found") } } }) }
36	}
37	}

Tabel 6.4 Penjelasan source code method addLaporan()

Baris	Penjelasan
4-6	Pengecekan koneksi internet, jika koneksi internet <i>offline</i> maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code> , jika koneksi internet <i>online</i> maka akan melakukan proses selanjutnya
7-8	Memanggil <code>listener</code> pada objek <code>ref_kendaraan</code> yang mengacu pada <code>database kendaraan</code> untuk melakukan akses data pada <code>database kendaraan</code>
9-11	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
15-20	Perulangan untuk mendapatkan seluruh data kendaraan yang dimasukkan ke dalam <code>list kendaraanList</code>

Tabel 6.4 Penjelasan *source code method addLaporan()* (lanjutan)

22-28	Perulangan pada kendaraanList untuk melakukan pengecekan apakah pada kendaraanList ada objek yang memiliki plat nomor sama dengan judul laporan. Jika terdapat yang sama, maka akan menginisialisasi variabel statusJudul dengan nilai true dan menyimpan objek kendaraan tersebut pada variabel dataKendaraan
29-33	Seleksi kondisi, jika variabel statusJudul bernilai true maka akan memanggil method onSuccess() melalui objek listener, jika variabel statusJudul bernilai false maka akan memanggil method onFailure() melalui objek listener

Nama Class : DatabaseOperation

Nama Method : addLaporan()

Tabel 6.5 Source code method addLaporan()

No	Source Code
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	<pre> override fun addLaporan(judul: String, pembuat: String, alamat: String, lat: String, long: String, waktu : String, keterangan : String, listener: DatabaseListener.BaseListener) { //operasi di database if (!isConnectInternet()){ listener.onFailed("offline") }else{ ref_laporan.addValueEventListener(object : ValueEventListener{ override fun onCancelled(p0: DatabaseError) { listener.onFailed("error") } override fun onDataChange(p0: DataSnapshot) { val uniqueId = ref_laporan.push().key!! //generate unique key val root = ref_laporan.child(uniqueId) root.setValue(Laporan(uniqueId, judul, pembuat,alamat,lat,long,waktu,keterangan)) ref_token.addListenerForSingleValueEvent (object : ValueEventListener{ override fun onCancelled(p0: DatabaseError?) { listener.onFailed("error") } override fun onDataChange(p0: DataSnapshot) { listToken.clear() for(tokenSnapshot in p0.chidren) { var token = tokenSnapshot .getValue(Token::class.java) listToken.add(token!!) } } }) } }) } } </pre>

Tabel 6.5 Source code method addLaporan() (lanjutan)

```

40                     if (listToken.size>1){
41                         val thisToken =
42                             Rak.grab<Token>("token")
43                             for (i in 0..listToken.size-1){
44                                 if (listToken[i].pengguna.
45                                     equals(thisToken.pengguna)){
46                                     continue
47                                 }
48                                 FirebaseNotification
49                                 Helper.initialize(serverKey)
50                                     .defaultJson(true, null)
51                                     .title(pembuat)
52                                     .message(judul)
53                                     .receiverFirebaseToken
54                                     (listToken[i].token)
55                                         .send()
56                                         }
57                                         listener.onSuccess()
58                                         }else{
59                                         listener.onSuccess()
60                                         }
61                                         }
62                                         })
63                                         }
64                                         }
65                                         }
66 }

```

Tabel 6.6 Penjelasan source code method addLaporan()

Baris	Penjelasan
12-14	Pengecekan koneksi internet, jika koneksi internet <i>offline</i> maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code> , jika koneksi internet <i>online</i> maka akan melakukan proses selanjutnya
15-16	Memanggil <code>listener</code> pada objek <code>ref_laporan</code> yang mengacu pada <code>database laporan</code> untuk melakukan akses data pada <code>database laporan</code>
17-19	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
21	Inisialisasi variabel <code>uniqueId</code> dengan <code>id</code> unik dari firebase melalui objek <code>ref_laporan</code>
23	Inisialisasi variabel <code>root</code> dengan objek <code>ref_laporan</code> yang merujuk pada <code>child id</code> unik yang telah dibuat sebelumnya pada variabel <code>uniqueId</code>
24-25	Menambahkan data ke dalam <code>database laporan</code> dengan memanggil method <code>setValue()</code> melalui variabel <code>root</code>
28-31	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>

Tabel 6.6 Penjelasan *source code method addLaporan()* (lanjutan)

35-39	Perulangan untuk mendapatkan seluruh data token yang dimasukkan ke dalam list <code>listToken</code>
40	Seleksi kondisi jika <code>size</code> dari <code>listToken</code> bernilai lebih dari 1
43-62	Perulangan untuk mengirimkan notifikasi dengan bantuan <i>library</i> <code>FirebaseNotificationHelper</code> ke seluruh objek token yang ada pada <code>listToken</code> dengan kondisi objek token yang memiliki nilai token sama dengan token pembuat laporan maka akan dilewati oleh perulangan

6.4.2 Algoritme Melihat Laporan

Pada fitur melihat laporan, implementasi kode program ada pada *method* `fetchLaporan()` yang berada pada *class* `DatabaseOperation`. Kode program pada *method* `fetchLaporan()` berisi alur untuk melakukan pengambilan seluruh data laporan dari *database* laporan untuk ditampilkan kepada pengguna. Kode program dari *method* `fetchLaporan()` dapat dilihat pada tabel 6.7 dan penjelasan algoritmenya pada tabel 6.8.

Nama *Class* : `DatabaseOperation`

Nama *Method* : `fetchLaporan()`

Tabel 6.7 *Source code method fetchLaporan()*

No	Source Code
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	override fun fetchLaporan(listener: DatabaseListener .GetLaporanListener) { //operasi di database if (!isConnectInternet()){ listener.onFailed("offline") }else{ ref_laporan.addValueEventListener(object : ValueEventListener { override fun onCancelled(p0: DatabaseError) { listener.onFailed("error") } override fun onDataChange(p0: DataSnapshot) { listLaporan.clear() for(laporanSnapshot in p0.getChildren()) { var laporan = laporanSnapshot.get Value(Laporan::class.java) listLaporan.add(laporan!!) } listener.onSuccess(listLaporan) } }) }

Tabel 6.8 Penjelasan source code method `fetchLaporan()`

Baris	Penjelasan
4-6	Pengecekan koneksi internet, jika koneksi internet <i>offline</i> maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code> , jika koneksi internet <i>online</i> maka akan melakukan proses selanjutnya
7-8	Memanggil <code>listener</code> pada objek <code>ref_laporan</code> yang mengacu pada <code>database laporan</code> untuk melakukan akses data pada <code>database laporan</code>
9-11	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
14-18	Perulangan untuk mengambil seluruh data laporan dari <code>database laporan</code> melalui objek <code>ref_laporan</code> yang diinisialisasikan ke dalam <code>list listLaporan</code>
19	Memanggil method <code>onSuccess()</code> melalui objek <code>listener</code>

6.4.3 Algoritme Panggilan Darurat

Pada fitur panggilan darurat, implementasi kode program ada pada `method fetchNomorDarurat()` yang berada pada `class DatabaseOperation`. Kode program pada `method fetchNomorDarurat()` berisi alur untuk melakukan pengambilan seluruh data nomor darurat dari `database nomor darurat` untuk ditampilkan kepada pengguna. Setelah nomor darurat berhasil ditampilkan maka pengguna dapat memilih nomor darurat untuk melakukan panggilan kepada nomor tersebut. Kode program dari `method fetchNomorDarurat()` dapat dilihat pada tabel 6.9 dan penjelasan algoritme pada tabel 6.10.

Nama Class : `DatabaseOperation`

Nama Method : `fetchNomorDarurat()`

Tabel 6.9 Source code method `fetchNomorDarurat()`

No	Source Code
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	override fun fetchNomorDarurat(listener: DatabaseListener.NomorDaruratListener) { //operasi di database if (!isConnectInternet()){ listener.onFailed() }else{ ref_nomor.addValueEventListener(object : ValueEventListener { override fun onCancelled(p0: DatabaseError) { listener.onFailed() } override fun onDataChange(p0: DataSnapshot) { noDaruratList.clear() for (nomorSnapshot in p0.getChildren()) { var noDarurat = nomorSnapshot.getValue(NomorDarurat::class.java) } } }) } }

Tabel 6.9 Source code method `fetchNomorDarurat()` (lanjutan)

```

17             noDaruratList.add(noDarurat!!)
18         }
19     }
20     }
21   })
22 }
23 }
```

Tabel 6.10 Penjelasan source code method `fetchNomorDarurat()`

Baris	Penjelasan
4-6	Pengecekan koneksi internet, jika koneksi internet <i>offline</i> maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code> , jika koneksi internet <i>online</i> maka akan melakukan proses selanjutnya
7-8	Memanggil <code>listener</code> pada objek <code>ref_nomor</code> yang mengacu pada <i>database</i> nomor darurat untuk melakukan akses data pada <i>database</i> nomor darurat
9-11	Kondisi jika sistem gagal mengakses <i>database</i> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
14-18	Perulangan untuk mengambil seluruh data laporan dari <i>database</i> nomor darurat melalui objek <code>ref_nomor</code> yang diinisialisasikan ke dalam <i>list</i> <code>noDaruratList</code>
19	Memanggil method <code>onSuccess()</code> melalui objek <code>listener</code>

6.4.4 Algoritme Mengirim Pesan Darurat

Pada fitur mengirim pesan darurat, implementasi kode program ada pada *method* `pushPesanDarurat()` yang berada pada *class* `DatabaseOperation`. Kode program pada *method* `pushPesanDarurat()` berisi alur untuk menyimpan data pesan darurat ke *database* pesan darurat dan mengirim notifikasi ke pengguna yang lainnya. Kode program dari *method* `pushPesanDarurat()` dapat dilihat pada tabel 6.11 dan penjelasan algoritme pada tabel 6.12.

Nama *Class* : `DatabaseOperation`

Nama *Method* : `pushPesanDarurat()`

Tabel 6.11 Source code method `pushPesanDarurat()`

No	Source Code
1	override fun pushPesanDarurat(
2	pengirim : String,
3	alamat : String,
4	thisToken : String,
5	waktu : String,
6	lat: String,
7	long: String,
8	listener: DatabaseListener.BaseListener

Tabel 6.11 Source code method pushPesananDarurat() (lanjutan)

```
9  ) {
10     //operasi di database
11     if (!isConnectInternet()){
12         listener.onFailed("offline")
13     }else{
14         ref_token.addListenerForSingleValueEvent(object :
15 ValueEventListener{
16             override fun onCancelled(p0: DatabaseError?) {
17                 listener.onFailed("error")
18             }
19             override fun onDataChange(p0: DataSnapshot) {
20                 listToken.clear()
21                 for (tokenSnapshot in p0.children) {
22                     var token = tokenSnapshot.getValue(
23 Token::class.java)
24                     listToken.add(token!!)
25                 }
26                 println("token size : "+listToken.size)
27                 if (listToken.size>0){
28                     ref_pesan.addListener
29 ForSingleValueEvent(object : ValueEventListener{
30             override fun onCancelled(p0:
31 DatabaseError?) {
32                 listener.onFailed("error")
33             }
34             override fun onDataChange(p0:
35 DataSnapshot?) {
36                 val uniqueId =
37                 ref_pesan.push().key!! //generate unique key
38                 val root = ref_pesan.child(
39 uniqueId)
40                 root.setValue(PesananDarurat
41 (uniqueId,thisToken,pengirim,alamat,waktu,lat,long))
42                 for (i in 0..listToken.size-1){
43                     if (listToken[i].pengguna.
44 equals(pengirim)){ //nanti diubah jadi cuma nama
45                         continue
46                     }
47                     FirebaseNotification
48 Helper.initializeApp(serverKey)
49                         .defaultJson(true, null)
50                         .title("pesan darurat")
51                         .message(uniqueId)
52                         .receiverFirebaseToken
53 (listToken[i].token)
54                         .send()
55                     }
56                     listener.onSuccess()
57                 }
58             }
59         }
60     }
61 }
62 }
63 }
```

Tabel 6.12 Penjelasan *source code method* pushPesanDarurat()

Baris	Penjelasan
11-13	Pengecekan koneksi internet, jika koneksi internet <i>offline</i> maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code> , jika koneksi internet <i>online</i> maka akan melakukan proses selanjutnya
14-15	Memanggil <code>listener</code> pada objek <code>ref_token</code> yang mengacu pada <code>database</code> data token untuk melakukan akses data pada <code>database</code>
16-18	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
21-25	Perulangan untuk mendapatkan seluruh data token yang dimasukkan ke dalam list <code>listToken</code>
27	Seleksi kondisi jika <code>size</code> dari <code>listToken</code> bernilai lebih dari 1
28-29	Memanggil <code>listener</code> pada objek <code>ref_pesan</code> yang mengacu pada <code>database</code> pesan darurat untuk melakukan akses data pada <code>database</code>
30-33	Kondisi jika sistem gagal mengakses <code>database</code> , maka akan memanggil method <code>onFailed()</code> melalui objek <code>listener</code>
36-37	Inisialisasi variabel <code>uniqueId</code> dengan <code>id</code> unik dari firebase melalui objek <code>ref_pesan</code>
38-39	Inisialisasi variabel <code>root</code> dengan objek <code>ref_pesan</code> yang merujuk pada <code>child id</code> unik yang telah dibuat sebelumnya pada variabel <code>uniqueId</code>
40-41	Menambahkan data ke dalam <code>database</code> pesan darurat dengan memanggil method <code>setValue()</code> melalui variabel <code>root</code>
42-58	Perulangan untuk mengirimkan notifikasi dengan bantuan <i>library</i> <code>FirebaseNotificationHelper</code> ke seluruh objek token yang ada pada <code>listToken</code> dengan kondisi objek token yang memiliki nilai token sama dengan token pengirim pesan darurat maka akan dilewati oleh perulangan

BAB 7 PENGUJIAN DAN ANALISIS

Pada bab ini membahas terkait pengujian yang dilakukan pada aplikasi SKIP. Setelah selesai melakukan tahapan implementasi, aplikasi SKIP akan diuji dengan berbagai macam pengujian, yaitu pengujian validasi (*blackbox*), pengujian *usability* dan pengujian *compatibility*. Pada setiap pengujian akan dibuat analisis berdasarkan hasil pengujian yang telah dilakukan.

7.1 Pengujian Validasi (*Blackbox*)

Pengujian validasi dilakukan pada aplikasi SKIP dengan tujuan untuk memastikan seluruh fungsionalitas pada aplikasi telah berjalan sesuai dengan pemodelan *use case scenario*. Pengujian validasi dilakukan pada setiap fungsionalitas yang menjalankan skenario pengujian untuk menentukan apakah fungsionalitas tersebut valid atau tidak valid. Jika fungsionalitas dapat berjalan sesuai dengan prosedur dan sesuai dengan hasil yang diharapkan, maka fungsionalitas tersebut memiliki status valid. Jika fungsionalitas tidak dapat berjalan sesuai dengan prosedur dan atau tidak sesuai dengan hasil yang diharapkan, maka fungsionalitas tersebut memiliki status tidak valid. Pengujian validasi diuji oleh Insan Nurzaman Adi Bangga Pratama, mahasiswa Teknik Informatika FILKOM UB angkatan tahun 2016. Pengujian dilakukan pada *device* Realme 2 dengan versi Android Pie 9.0.

7.1.1 Pengujian Validasi *Login*

Pada tabel 7.1 menunjukkan hasil pengujian validasi pada fungsionalitas *login*. Pengujian validasi *login* dilakukan sesuai dengan prosedur pada tabel 7.1 dan hasil yang diharapkan yaitu pengguna berhasil untuk *login* jika memasukkan *username* dan *password* yang benar kemudian sistem menampilkan tampilan *dashboard* disertai dengan *toast* notifikasi berhasil *login*. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas *login* seperti pada *use case scenario*.

Tabel 7.1 Pengujian validasi *login*

Kode Kebutuhan	F-SKIP-1
Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none">1. Pengguna menginputkan <i>username</i> dan <i>password</i>2. Pengguna menekan tombol “LOGIN”
Expected Result	<ol style="list-style-type: none">1. Proses <i>login</i> berhasil, sistem menampilkan tampilan <i>dashboard</i> dan menampilkan <i>toast</i> “Login Berhasil!” jika pengguna melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar disertai dengan kondisi <i>device</i> yang terhubung dengan internet

Tabel 7.1 Pengujian validasi *login* (lanjutan)

	<p>2. Proses <i>login</i> gagal, sistem menampilkan <i>toast</i> “Verifikasi Gagal, Periksa Kembali Username dan Password!” jika pengguna melakukan <i>login</i> dengan <i>username</i> atau <i>password</i> yang tidak sesuai dengan data pada <i>database</i></p> <p>3. Proses <i>login</i> gagal, sistem menampilkan <i>toast</i> “Proses Login Gagal, Pastikan Anda Terhubung Dengan Internet” jika pengguna melakukan <i>login</i> dengan kondisi <i>device</i> tidak terhubung dengan internet</p>
Result	<p>1. Proses <i>login</i> berhasil, sistem menampilkan tampilan <i>dashboard</i> dan menampilkan <i>toast</i> “Login Berhasil!” ketika pengguna melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar disertai dengan kondisi <i>device</i> yang terhubung dengan internet</p> <p>2. Proses <i>login</i> gagal, sistem menampilkan <i>toast</i> “Verifikasi gagal, Periksa Kembali Username dan Password!” ketika pengguna melakukan <i>login</i> dengan <i>username</i> atau <i>password</i> yang tidak sesuai</p> <p>3. Proses <i>login</i> gagal, sistem menampilkan <i>toast</i> “Proses Login Gagal, Pastikan Anda Terhubung Dengan Internet” ketika pengguna melakukan <i>login</i> dengan kondisi <i>device</i> tidak terhubung dengan internet</p>
Status	Valid

7.1.2 Pengujian Validasi Buat Laporan

Pada tabel 7.2 menunjukkan hasil pengujian validasi pada fungsionalitas buat laporan. Pengujian validasi buat laporan dilakukan sesuai dengan prosedur pada tabel 7.2 dan hasil yang diharapkan yaitu sistem berhasil menyimpan data laporan yang dibuat oleh pengguna dan sistem menampilkan *toast* notifikasi berhasil *upload* laporan. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas buat laporan seperti pada *use case scenario*.

Tabel 7.2 Pengujian validasi buat laporan

Kode Kebutuhan	F-SKIP-2
Nama Kasus Uji	Buat Laporan

Tabel 7.2 Pengujian validasi buat laporan (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Pengguna menekan tombol menu “Buat Laporan” di tampilan <i>dashboard</i> 2. Sistem menampilkan tampilan <i>input</i> judul laporan 3. Pengguna memasukkan data judul laporan 4. Pengguna menekan tombol “CEK” 5. Sistem menampilkan tampilan hasil verifikasi data judul laporan, jika <i>input</i> judul laporan berupa nomor plat kendaraan 6. Pengguna menekan tombol “BUAT LAPORAN” 7. Sistem menampilkan tampilan untuk <i>submit</i> laporan 8. Pengguna memasukkan data keterangan laporan dan memeriksa informasi lokasi pelaporan 9. Pengguna menekan tombol “SUBMIT”
Expected Result	<ol style="list-style-type: none"> 1. Proses <i>upload</i> laporan berhasil, sistem berhasil menyimpan data laporan di <i>database</i> dan menampilkan <i>toast</i> “Upload Laporan Berhasil!” ketika pengguna telah melakukan seluruh proses <i>upload</i> laporan dengan kondisi <i>device</i> yang selalu terhubung dengan internet 2. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Proses Pengecekan Gagal, Pastikan Anda Terhubung Dengan Internet” ketika pengguna melakukan pengecekan judul laporan dengan kondisi <i>device</i> yang tidak terhubung dengan internet 3. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Silahkan Aktifkan Sensor GPS/Location Service Terlebih Dahulu” ketika pengguna belum mengaktifkan sensor GPS/<i>Location Service</i> saat sistem melakukan proses pengambilan data lokasi pelapor 4. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Upload Laporan Gagal, Pastikan Anda Terhubung Dengan Internet” ketika pengguna melakukan <i>upload</i> laporan dengan kondisi <i>device</i> yang tidak terhubung dengan internet

Tabel 7.2 Pengujian validasi buat laporan (lanjutan)

Result	1. Proses <i>upload</i> laporan berhasil, sistem berhasil menyimpan data laporan di <i>database</i> dan menampilkan <i>toast</i> “Upload Laporan Berhasil!” ketika pengguna telah melakukan seluruh proses <i>upload</i> laporan dengan kondisi <i>device</i> yang selalu terhubung dengan internet 2. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Proses Pengecekan Gagal, Pastikan Anda Terhubung Dengan Internet” ketika pengguna melakukan pengecekan judul laporan dengan kondisi <i>device</i> yang tidak terhubung dengan internet 3. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Silahkan Aktifkan Sensor GPS/Location Service Terlebih Dahulu” ketika pengguna belum mengaktifkan sensor <i>GPS/Location Service</i> saat sistem melakukan proses pengambilan data lokasi pelapor 4. Proses <i>upload</i> laporan gagal, sistem menampilkan <i>toast</i> “Upload Laporan Gagal, Pastikan Anda Terhubung Dengan Internet” ketika pengguna melakukan <i>upload</i> laporan dengan kondisi <i>device</i> yang tidak terhubung dengan internet
Status	Valid

7.1.3 Pengujian Validasi Lihat Laporan

Pada tabel 7.3 menunjukkan hasil pengujian validasi pada fungsionalitas lihat laporan. Pengujian validasi lihat laporan dilakukan sesuai dengan prosedur pada tabel 7.3 dan hasil yang diharapkan yaitu sistem berhasil menampilkan seluruh data laporan yang dapat dilihat oleh pengguna pada aplikasi dan pengguna dapat melihat detail informasi data laporan beserta lokasi pelaporan. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas lihat laporan seperti pada *use case scenario*.

Tabel 7.3 Pengujian validasi lihat laporan

Kode Kebutuhan	F-SKIP-3
Nama Kasus Uji	Lihat Laporan
Prosedur	1. Pengguna menekan tombol menu “Lihat Laporan” di tampilan <i>dashboard</i>

Tabel 7.3 Pengujian validasi lihat laporan (lanjutan)

	<ol style="list-style-type: none"> 2. Sistem menampilkan tampilan yang berisi seluruh data laporan 3. Pengguna menekan tombol “Selengkapnya” pada laporan yang ingin dilihat detail informasinya 4. Sistem menampilkan tampilan detail informasi laporan 5. Pengguna menekan tombol “LIHAT LOKASI PELAPOR” <p>Sistem menampilkan lokasi pelapor dalam tampilan <i>google maps</i></p>
Expected Result	<ol style="list-style-type: none"> 1. Sistem berhasil menampilkan seluruh data laporan yang dapat dilihat oleh pengguna pada aplikasi dan pengguna dapat melihat detail informasi laporan beserta lokasi pelaporan 2. Sistem menampilkan <i>toast</i> “Gagal Mendapatkan Data Laporan, Pastikan Anda Terhubung Dengan Internet” jika pengguna membuka menu lihat laporan dengan kondisi <i>device</i> tidak terhubung dengan internet
Result	<ol style="list-style-type: none"> 1. Sistem berhasil menampilkan seluruh data laporan yang dapat dilihat oleh pengguna pada aplikasi dan pengguna dapat melihat detail informasi laporan beserta lokasi pelaporan 2. Sistem menampilkan <i>toast</i> “Gagal Mendapatkan Data Laporan, Pastikan Anda Terhubung Dengan Internet” ketika pengguna membuka menu lihat laporan dengan kondisi <i>device</i> tidak terhubung dengan internet
Status	Valid

7.1.4 Pengujian Validasi Panggilan Darurat

Pada tabel 7.4 menunjukkan hasil pengujian validasi pada fungsionalitas panggilan darurat. Pengujian validasi panggilan darurat dilakukan sesuai dengan prosedur pada tabel 7.4 dan hasil yang diharapkan yaitu sistem berhasil menampilkan seluruh data nomor darurat yang dapat dilihat oleh pengguna pada aplikasi dan pengguna dapat melakukan panggilan secara langsung kepada nomor yang dipilih. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas panggilan darurat seperti pada *use case scenario*.

Tabel 7.4 Pengujian validasi panggilan darurat

Kode Kebutuhan	F-SKIP-4
Nama Kasus Uji	Panggilan Darurat
Prosedur	<ol style="list-style-type: none"> 1. Pengguna menekan tombol menu “Panggilan Darurat” di tampilan <i>dashboard</i> 2. Sistem menampilkan tampilan yang berisi seluruh data nomor darurat 3. Pengguna memilih salah satu nomor darurat yang tersedia dan menekan logo telepon
Expected Result	<ol style="list-style-type: none"> 1. Sistem berhasil menampilkan seluruh data nomor darurat dan pengguna dapat melakukan panggilan secara langsung kepada nomor darurat yang dipilih 2. Sistem menampilkan <i>toast</i> “Gagal Mendapatkan Data Nomor Darurat, Pastikan Anda Terhubung Dengan Internet” jika pengguna menuju menu panggilan darurat dengan kondisi <i>device</i> tidak terhubung dengan internet
Result	<ol style="list-style-type: none"> 1. Sistem berhasil menampilkan seluruh data nomor darurat dan pengguna dapat melakukan panggilan secara langsung kepada nomor darurat yang dipilih 2. Sistem menampilkan <i>toast</i> “Gagal Mendapatkan Data Nomor Darurat, Pastikan Anda Terhubung Dengan Internet” ketika pengguna menuju menu panggilan darurat dengan kondisi <i>device</i> tidak terhubung dengan internet
Status	Valid

7.1.5 Pengujian Validasi Panduan Aplikasi

Pada tabel 7.5 menunjukkan hasil pengujian validasi pada fungsionalitas panduan aplikasi. Pengujian panduan aplikasi dilakukan sesuai dengan prosedur pada tabel 7.5 dan hasil yang diharapkan yaitu sistem berhasil menampilkan informasi panduan menu aplikasi yang dapat dilihat oleh pengguna pada aplikasi.

Tabel 7.5 Pengujian validasi panduan aplikasi

Kode Kebutuhan	F-SKIP-5
Nama Kasus Uji	Panduan Aplikasi
Prosedur	<ol style="list-style-type: none"> 1. Pengguna menekan tombol menu “Panduan Aplikasi” di tampilan <i>dashboard</i>

Tabel 7.5 Pengujian validasi panduan aplikasi (lanjutan)

	2. Sistem menampilkan tampilan daftar panduan menu pada aplikasi 3. Pengguna menekan salah satu pilihan menu untuk melihat informasi panduan
Expected Result	Sistem berhasil menampilkan informasi panduan menu aplikasi yang dapat dilihat oleh pengguna pada aplikasi
Result	Sistem berhasil menampilkan informasi panduan menu aplikasi yang dapat dilihat oleh pengguna pada aplikasi
Status	Valid

7.1.6 Pengujian Validasi Mengirim Pesan Darurat

Pada tabel 7.6 menunjukkan hasil pengujian validasi pada fungsionalitas mengirim pesan darurat. Pengujian validasi mengirim pesan darurat dilakukan sesuai dengan prosedur pada tabel 7.6 dan hasil yang diharapkan adalah sistem berhasil menyimpan data pesan darurat untuk didistribusikan kepada pengguna yang lainnya dan sistem menampilkan dialog informasi berhasil mengirimkan pesan darurat. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas mengirim pesan darurat seperti pada *use case scenario*.

Tabel 7.6 Pengujian validasi mengirim pesan darurat

Kode Kebutuhan	F-SKIP-6
Nama Kasus Uji	Mengirim Pesan Darurat
Prosedur	1. Pengguna menekan tombol “PANIC” yang berada di pojok kanan atas tampilan <i>dashboard</i> 2. Sistem menampilkan dialog konfirmasi 3. Pengguna menekan tombol “YA” pada dialog
Expected Result	1. Pengiriman pesan darurat berhasil, sistem menampilkan dialog informasi berhasil mengirimkan pesan darurat dan menampilkan waktu tunggu selama 20 detik jika pengguna melakukan pengiriman pesan darurat dengan kondisi <i>device</i> yang terhubung dengan internet 2. Pengiriman pesan darurat gagal, sistem menampilkan <i>toast</i> “Pembuatan Pesan Darurat Gagal, Pastikan Anda Terhubung Dengan Internet” jika pengguna melakukan pengiriman pesan darurat

Tabel 7.6 Pengujian validasi mengirim pesan darurat (lanjutan)

	dengan kondisi <i>device</i> tidak terhubung dengan internet
Result	1. Pengiriman pesan darurat berhasil, sistem menampilkan dialog informasi berhasil mengirimkan pesan darurat dan menampilkan waktu tunggu selama 20 detik jika pengguna melakukan pengiriman pesan darurat dengan kondisi <i>device</i> yang terhubung dengan internet 2. Pengiriman pesan darurat gagal, sistem menampilkan <i>toast</i> "Pembuatan Pesan Darurat Gagal, Pastikan Anda Terhubung Dengan Internet" jika pengguna melakukan pengiriman pesan darurat dengan kondisi <i>device</i> tidak terhubung dengan internet
Status	Valid

7.1.7 Pengujian Validasi Menerima Pesan Darurat

Pada tabel 7.7 menunjukkan hasil pengujian validasi pada fungsionalitas menerima pesan darurat. Pengujian validasi menerima pesan darurat dilakukan sesuai dengan prosedur pada tabel 7.7 dan hasil yang diharapkan yaitu sistem berhasil menerima pesan darurat dengan menampilkan tampilan pesan darurat yang dapat dilihat pengguna pada aplikasi. Selain itu juga akan diujikan kondisi alternatif pada fungsionalitas menerima pesan darurat seperti pada *use case scenario*.

Tabel 7.7 Pengujian validasi menerima pesan darurat

Kode Kebutuhan	F-SKIP-7
Nama Kasus Uji	Menerima Pesan Darurat
Prosedur	1. Pengguna sedang membuka aplikasi SKIP atau aplikasi SKIP sedang berada pada <i>task background device</i> pengguna 2. Sistem menerima notifikasi pesan darurat
Expected Result	1. Sistem menampilkan tampilan pesan darurat disertai dengan <i>device</i> yang bergetar selama 10 detik 2. Sistem gagal menampilkan pesan darurat jika <i>device</i> pengguna tidak terhubung dengan internet
Result	1. Sistem menampilkan tampilan pesan darurat disertai dengan <i>device</i> yang bergetar selama 10 detik

Tabel 7.7 Pengujian validasi menerima pesan darurat (lanjutan)

	2. Sistem gagal menampilkan pesan darurat jika <i>device</i> pengguna tidak terhubung dengan internet dan pesan darurat akan masuk setelah <i>device</i> tersambung kembali dengan internet
Status	Valid

7.1.8 Pengujian Validasi Tentang Aplikasi

Pada tabel 7.8 menunjukkan hasil pengujian validasi pada fungsionalitas tentang aplikasi. Pengujian validasi tentang aplikasi dilakukan sesuai dengan prosedur pada tabel 7.8 dan hasil yang diharapkan yaitu sistem berhasil menampilkan informasi tentang aplikasi yang dapat dilihat oleh pengguna pada aplikasi.

Tabel 7.8 Pengujian validasi tentang aplikasi

Kode Kebutuhan	F-SKIP-8
Nama Kasus Uji	Tentang Aplikasi
Prosedur	1. Pengguna menekan tombol menu “Tentang Aplikasi” di tampilan <i>dashboard</i>
Expected Result	Sistem berhasil menampilkan tampilan informasi tentang aplikasi yang dapat dilihat oleh pengguna pada aplikasi
Result	Sistem berhasil menampilkan tampilan informasi tentang aplikasi yang dapat dilihat oleh pengguna pada aplikasi
Status	Valid

7.1.9 Analisis Hasil Pengujian Validasi

Berdasarkan hasil pengujian validasi yang telah dilakukan oleh pengujinya sebelumnya, aplikasi SKIP dapat menjalankan seluruh fungsionalitas dengan baik dan tanpa mengalami kendala. Selain itu aplikasi SKIP juga dapat menjalankan seluruh skenario termasuk skenario alternatif sesuai dengan *expected result* yang telah direncanakan. Dapat disimpulkan bahwa aplikasi SKIP dapat berjalan sesuai dengan kebutuhan yaitu dengan tingkat keberhasilan sebesar 100% pada seluruh fungsionalitas. Aplikasi SKIP diharapkan dapat berjalan sesuai kebutuhan dengan tingkat keberhasilan 100% juga pada perangkat lainnya yang telah ditargetkan pada pengujian *compatibility*.

7.2 Pengujian *Usability*

Pengujian *usability* merupakan salah satu pengujian dari sisi non fungsional yang dilakukan pada aplikasi SKIP. Pengujian *usability* dibagi menjadi dua tahap yaitu pengujian *usability* dan pengisian kuesioner *System Usability Scale* (SUS). Pengujian *usability* dilakukan kepada 5 responden untuk menggunakan aplikasi selama kurang lebih 10 menit dengan mengerjakan *task* yang diberikan kepada responden dan pengujian akan memberikan sedikit *clue*. *Task* yang diberikan yaitu untuk menjalankan fungsionalitas yang ada pada aplikasi SKIP yang ditunjukkan pada tabel 7.9.

Tabel 7.9 Task Pengujian *Usability*

No	Task	Detail
1	<i>Login</i>	Pengguna akan diminta untuk melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi

Setelah mengerjakan task, pengguna akan diminta untuk mengisi kuesioner *System Usability Scale* (SUS) yang berisi 10 pernyataan. Pernyataan akan dibagi menjadi pernyataan ganjil dan pernyataan genap. Pernyataan ganjil merupakan pernyataan positif karena memiliki jawaban bagus jika responden menjawab

dengan skor tertinggi. Sementara pernyataan genap merupakan pernyataan negatif karena memiliki jawaban bagus jika responden menjawab dengan skor terendah. Pernyataan pada kuesioner SUS ditunjukkan pada tabel 7.10.

Tabel 7.10 Daftar pernyataan kuesioner SUS

No	Pertanyaan
1	Aplikasi ini berguna bagi saya
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)
3	Aplikasi ini mudah untuk digunakan
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan
6	Aplikasi ini tidak konsisten
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan
9	Saya akan menggunakan aplikasi ini untuk seterusnya
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas

7.2.1 Analisis Hasil Pengujian *Usability*

Setelah melakukan pengujian *usability* dan mendapatkan skor SUS dari kuesioner SUS yang telah diisi oleh responden, maka tahapan selanjutnya adalah melakukan analisis berdasarkan hasil pengujian *usability* kepada seluruh responden. Hasil pengujian *usability* terdiri dari tabel 7.11 yang menunjukkan *task* yang dilakukan oleh responden apakah berhasil (B), hampir berhasil (HB) atau gagal (B).

Tabel 7.11 Hasil penggerjaan *task* oleh responden

Task	Responden				
	R1	R2	R3	R4	R5
1	B	B	B	B	B
2	B	B	B	B	B
3	B	B	B	B	B
4	B	B	B	B	B
5	B	B	B	B	B
6	B	B	B	B	B

Tabel 7.11 Hasil penggerjaan *task* oleh responden (lanjutan)

7	B	B	B	B	B
8	B	B	B	B	B
9	B	B	B	B	B
Keterangan : B = Berhasil, HB = Hampir Berhasil, G = Gagal					

Keterangan Responden :

R1 : Bagas Sudarsono (Pemuda Karang Taruna)

R2 : Bapak Sanduki (Ketua RW 12)

R3 : Ibu Dwi Astuti (Perwakilan ibu-ibu di Perumahan Panorama Cibodas)

R4 : Bapak Karsidi (Ketua RT 01/RW 12)

R5 : Bapak Asep (Ketua RT 02/RW 12)

Perhitungan tingkat keberhasilan responden dalam mengerjakan *task* yang telah diberikan dihitung dengan persamaan 7.1 dengan detail perhitungan sebagai berikut.

$$\text{Success rate} = \frac{(\text{jumlah B} + (\text{jumlah HB} \times 0.5))}{\text{jumlah tugas} \times \text{jumlah responden}} \times 100\% \quad (7.1)$$

$$\text{Success rate} = \frac{(45 + (0 \times 0.5))}{9 \times 5} \times 100\%$$

$$\text{Success rate} = 100\%$$

Sementara itu, hasil dari pengisian kuesioner SUS pada kelima responden yang ditunjukkan pada tabel 7.12. Bukti dari pengisian kuesioner SUS yang asli terdapat pada bagian lampiran.

Tabel 7.12 Hasil pengisian kuesioner SUS

Nomor Pernyataan	Responden				
	R1	R2	R3	R4	R5
1	4	4	5	4	5
2	1	2	2	2	1
3	4	4	5	4	5
4	4	2	2	4	5
5	4	4	5	4	5
6	2	2	2	2	1
7	5	4	5	4	5

Tabel 7.12 Hasil pengisian kuesioner SUS (lanjutan)

8	1	2	2	2	1
9	4	4	4	4	5
10	1	2	2	2	1

Perhitungan skor SUS dilakukan pada setiap responden berdasarkan hasil kuesioner yang telah diisi oleh kelima responden. Perhitungan skor SUS dilakukan dengan teori yang telah dipaparkan pada bab sebelumnya, maka didapatkan hasil perhitungan yang ditunjukkan pada tabel 7.13.

Tabel 7.13 Perhitungan akumulasi skor SUS

Nomor Pernyataan	Responden				
	R1	R2	R3	R4	R5
1	3	3	4	3	4
2	4	3	3	3	4
3	3	3	4	3	4
4	1	3	3	1	0
5	3	3	4	3	4
6	3	3	3	3	4
7	4	3	4	3	4
8	4	3	3	3	4
9	3	3	3	3	4
10	4	3	3	3	4
Jumlah	32	30	34	28	36
Dikali 2,5 (x 2,5)	80	75	85	70	90
Rata-rata skor SUS/Nilai akhir skor SUS					80

Berdasarkan perhitungan akumulasi skor SUS pada tabel 7.13, rata-rata dari skor SUS memiliki nilai sebesar 80 yang berarti sistem memiliki kualifikasi *grade B* dengan *adjectiva rating* “good” dan responden memiliki tingkat keberhasilan melakukan *task* sebesar 100%. Dengan hasil tersebut maka aplikasi SKIP dapat dikatakan berhasil digunakan oleh para responden tanpa mengalami masalah dan aplikasi SKIP dianggap cukup memuaskan oleh para responden.

7.3 Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk memastikan aplikasi SKIP dapat berjalan dengan baik pada sistem operasi android dengan versi minimum KitKat 4.4 dan versi maksimum Pie 9.0. Pengujian dilakukan dengan dua cara yaitu melakukan instalasi aplikasi SKIP pada beberapa *smartphone* android yang akan dijadikan perangkat uji dengan versi yang berbeda-beda secara manual dan menggunakan bantuan Firebase Test Lab. Pada pengujian secara manual seluruh perangkat uji akan menjalankan *task* seperti pada pengujian *usability*. Sementara itu pengujian *compatibility* pada Firebase Test Lab dilakukan secara otomatis yaitu dengan metode Robo Test. Hasil pengujian dikatakan valid apabila seluruh perangkat uji berhasil menjalankan seluruh *task* tanpa menyebabkan aplikasi *crash* dan aplikasi tidak dapat berjalan pada perangkat uji dengan versi dibawah dan diatas target yang telah ditentukan. Berikut adalah hasil pengujian *compatibility* secara manual yang dapat dilihat pada tabel 7.14.

Tabel 7.14 Hasil pengujian *compatibility* secara manual

Perangkat	API Level 21 (5.1)	API Level 27 (8.1)	API Level 28 (9.0)
Use Case			
Login	✓	✓	✓
Mengecek plat nomor kendaraan	✓	✓	✓
Membuat laporan	✓	✓	✓
Melihat laporan	✓	✓	✓
Melakukan panggilan darurat	✓	✓	✓
Melihat panduan menu aplikasi	✓	✓	✓
Mengirim pesan darurat	✓	✓	✓
Menerima pesan darurat	✓	✓	✓
Melihat informasi tentang aplikasi	✓	✓	✓

Hasil pengujian *compatibility* dengan menggunakan Firebase Test Lab dengan metode Robo Test dapat dilihat pada tabel 7.15.

Tabel 7.15 Hasil pengujian compatibility menggunakan Firebase Test Lab

Perangkat \ Pengujian	API Level 18 (4.3)	API Level 19 (4.4)	API Level 23 (6.0)	API Level 25 (7.1)	API Level 29 (10.0)
<i>Firebase Test Lab (Run a Robo Test)</i>	✗	✓	✓	✓	✗

7.3.1 Analisis Pengujian *Compatibility*

Berdasarkan hasil pengujian *compatibility* yang ditunjukkan pada tabel 7.14 dan 7.15, aplikasi SKIP dapat berjalan sesuai dengan target versi android yang telah ditentukan. Maka aplikasi SKIP dapat dikatakan valid atau lulus uji *compatibility*.

BAB 8 PENUTUP

Pada bab ini terdiri dari kesimpulan dan saran yang dibuat berdasarkan penelitian yang telah dilakukan pada bab-bab sebelumnya. Pada bagian kesimpulan berisi hasil dari penelitian dan bagian saran berisi saran untuk penelitian selanjutnya.

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dimulai dari proses analisis kebutuhan, perancangan, implementasi dan pengujian, maka dapat dibuat kesimpulan sebagai berikut :

1. Analisis kebutuhan pada pengembangan aplikasi SKIP dengan metode *Prototyping* bermula dari pengembang yang mengajukan kebutuhan pada aplikasi kemudian dibuat dalam bentuk *prototype* untuk ditunjukkan kepada calon pengguna dan mendapatkan *feedback*. Pengembang menempuh sekali tahap iterasi karena calon pengguna telah menyetujui kebutuhan pada aplikasi yang telah diajukan dalam bentuk *prototype*. Hasil analisis kebutuhan pada aplikasi SKIP dibagi menjadi 2 yaitu kebutuhan fungsional yang berjumlah 8 dan kebutuhan non fungsional yang berjumlah 2.
2. Perancangan yang dilakukan pada aplikasi SKIP menghasilkan perancangan arsitektur sistem, *sequence diagram*, *class diagram*, *physical data model*, perancangan antarmuka dan *screenflow diagram*.
3. Aplikasi SKIP diimplementasikan pada sistem operasi android yang memanfaatkan layanan firebase untuk penyimpanan data dan *push* notifikasi dan layanan google map API untuk menggunakan layanan berbasis lokasi. Aplikasi SKIP diimplementasikan dengan bahasa pemrograman Kotlin dan menggunakan arsitektur *model view presenter* (MVP).
4. Terdapat 3 pengujian yang dilakukan pada aplikasi SKIP yaitu pengujian validasi (*blackbox*), pengujian *usability* dan pengujian *compatibility*. Pengujian validasi menghasilkan bahwa aplikasi SKIP memiliki tingkat keberhasilan 100% pada seluruh fungsionalitas. Pada pengujian *usability* menghasilkan *success rate* 100% dan nilai akhir skor *System Usability Scale* (SUS) sebesar 80 yang termasuk ke dalam *grade B* dan *adjectiva rating* “good”. Pada pengujian *compatibility* yang dilakukan secara manual dan otomatis dengan Firebase Test Lab menghasilkan bahwa aplikasi SKIP dapat berjalan sesuai dengan versi sistem operasi android yang telah ditargetkan yaitu minimum versi KitKat 4.4 dan maksimum versi Pie 9.0.

8.2 Saran

Saran yang diberikan pada aplikasi SKIP untuk proses pengembangan selanjutnya yaitu sebagai berikut :

1. Menambahkan aktor baru sebagai admin yang memiliki fitur untuk melakukan penghapusan data yang sudah usang di database agar mengurangi beban proses pengambilan data di aplikasi.
2. Menambahkan fitur yang berisi daftar pesan darurat agar pengguna dapat melihat informasi daftar pesan darurat yang telah dibuat.
3. Menambahkan fitur sinkronisasi data yang memanfaatkan penyimpanan data pada ponsel untuk menangani masalah jika ponsel sedang mengalami gangguan jaringan internet.
4. Melakukan migrasi kode program ke versi Android X agar aplikasi dapat berjalan pada versi sistem operasi android terbaru yaitu Android Q 10.0.
5. Membuat mekanisme notifikasi yang dapat masuk meskipun pengguna sedang tidak membuka aplikasi.

DAFTAR REFERENSI

- Alathas, H., 2018. Bagaimana Mengukur Kebergunaan Produk dengan System Usability Scale (SUS) Score. [Online] Tersedia di : <https://medium.com/kelasux/bagaimana-mengukur-kebergunaan-produk-dengan-system-usability-scale-sus-score-2d6843ca780a> [Diakses 8 Desember 2019]
- Alfeno, S. & Devi, R. E. C., 2017. Implementasi Global Positioning System (GPS) dan Location Based Service (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek. *Jurnal Sisfotek Global*, VII(2), pp. 27-33.
- Aprilia, I., Santoso, P. I. & Ferdiana, R., 2015. Pengujian Usability Website Menggunakan System Usability Scale Website Usability Testing using System Usability Scale. *IPTEK-KOM*, XVII(1), pp. 31-38.
- Booch, G., J. Rumbaugh, and I. Jacobsen, The Unified Modelling Language User Guide. 2d ed., Addison-Wesley, 2005.
- Budiaji, W., 2013. Skala Pengukuran dan Jumlah Respon Skala Likert. *Jurnal Ilmu Pertanian dan Perikanan*, II(2), pp. 127-133.
- Budianingsih & Riyanto, A., 2018. Prototype Sistem Keamanan Cerdas Pada Komplek Perumahan. *Jurnal Pendidikan Informatika dan Sains*, VII(1), pp. 146-154.
- Christoforus, A., Nugroho, A. & Setiyanti, A. A., 2012. Implementasi Model View Presenter dan Object Relational Mapping NHibernate pada Aplikasi eStop Card Berbasis Web. *Informatika*, VIII(2), pp. 189-207.
- Firebase. *Firebase Cloud Messaging*. [Online] Tersedia di : <https://firebase.google.com/docs/cloud-messaging> [Diakses 25 Oktober 2019]
- Firebase. *Firebase Test Lab*. [Online] Tersedia di : <https://firebase.google.com/docs/test-lab/?hl=id> [Diakses 1 Desember 2019]
- Firebase. *Firebase Realtime Database*. [Online] Tersedia di : <https://firebase.google.com/docs/database?hl=id> [Diakses 25 Oktober 2019]
- Google. *Understand The Activity Lifecycle*. [Online] Tersedia di : <https://developer.android.com/guide/components/activities/activity-lifecycle> [Diakses 25 Oktober 2019]
- Kadir, A., 2013. *From Zero to A Pro - Pemrograman Aplikasi Android*. Yogyakarta: Andi Yogyakarta.
- Nielsen, J., 2001. *Usability Metrics*. [Online] Tersedia di : <https://www.nngroup.com/articles/usability-metrics/> [Diakses 11 Desember 2019]

- Nielsen, J., 2001. *Success Rate : The Simplest Usability Metric*. [Online] Tersedia di : <https://www.nngroup.com/articles/usability-metrics/> [Diakses 11 Desember 2019]
- Payara, G. R. & Tanone, R., 2018. Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*, IV(3), pp. 397-406.
- Pratama, M. A., Kharisma, A. P. & Brata, A. H., 2019. Pengembangan Aplikasi Sportyway : Aplikasi Pencari Teman Untuk Berolahraga Bersama Berbasis Lokasi. III(3), pp. 2761-2767.
- Pressman, R., 2010. Software Engineering: A Practitioner's Approach. 7th Edition ed. New York: McGraw-Hill.
- Safaat, N., 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Sianturi, Eminarti. 2018. *Android MVP Series : Membangun Aplikasi Android dengan Arsitektur MVP*. [Online] Tersedia di : <https://medium.com/easyread/android-mvp-series-membangun-aplikasi-android-dengan-arsitektur-mvp-fbf1f77ecaec> [Diakses 1 Desember 2019]
- Sommerville, Ian. 2011. *Software Engineering 9th Edition*. London : Pearson.
- Utaberta, N., 2014. Interaksi Sosial dan Kriminalitas di Perumahan. XIV(1), pp. 189-192.
- Wadkar, M. C. & Patil, P. P., 2018. Traditional Infrastructure vs Firebase Infrastructure. *International Journal of Trend in Scientific Research and Development*, II(4), pp. 2050-2053.

LAMPIRAN A HASIL WAWANCARA UJI PROTOTYPE

A.1 Form Hasil Wawancara Uji Prototype

Form Hasil Wawancara Uji Coba Prototype

Nama	Sanduki
Alamat	Prm. Panorama Cibodas Blok. H No.11
Umur	56 Th.
Profesi/Instansi	Pegawai Swasta
Jabatan di Perumahan Panorama Cibodas	Ketua RW.
Saran atau tanggapan berdasarkan hasil uji coba prototipe	- cukup bagus dan In shaa Allah bermanfaat di lingkungan kami.
Tanda Tangan	Tangerang, 11 Oktober 2019   <p>RW. 012 PANORAMA CIBODAS KELURAHAN : SANJIANG JAYA KECAMATAN : PERIUK KOTA TANGERANG</p>

Lampiran 1 Uji prototype dengan Ketua RW 12

Form Hasil Wawancara Uji Coba Prototype

Nama	KARSIDI
Alamat	Panorama Cibodas Jl. Venus J/6
Umur	43 Thn
Profesi/Instansi	Karyawati Swasta
Jabatan di Perumahan Panorama Cibodas	Ketua RT. 01
Saran atau tanggapan berdasarkan hasil uji coba prototype	<p>Sayang rasanya bagus aplikasi ini. Bahanwanya banyak sedangkan perumahan sendiri memiliki IT seperti itu.</p>
Tanda Tangan	<p style="text-align: center;">Tangerang</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> RT.001 RW.12 KELURAHAN : SANGGANG JAYA KECAMATAN : PERIUK KOTA TANGERANG (KARSIDI) </div>

Lampiran 2 Uji prototype dengan Ketua RT 01/RW 12

Form Hasil Wawancara Uji Coba Prototype

Nama	Asep M Syahrul
Alamat	Panorama Cibodas Blok C No.14
Umur	52 Tahun
Profesi/Instansi	# Pegawai Swasta
Jabatan di Perumahan Panorama Cibodas	# Ketua RT
Saran atau tanggapan berdasarkan hasil uji coba prototype	Cukup bagus Ide ini kami dukung untuk meningkatkan sistem keamanan & komunikasi yang cepat dan tepat di lingkungan kami (RW 012 / RT 02 Panorama Cibodas)
Tanda Tangan	<p align="center">Tangerang, 13 Oktober 2019 .</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> RT.002 RW.12 KELURAHAN : SANGIANG JAYA KECAMATAN : PERIUK KOTA TANGERANG </div> <p align="center">(Asep.M.S.)</p>

Lampiran 3 Uji prototype dengan Ketua RT 02/RW 12

Form Hasil Wawancara Uji Coba Prototype

Nama	SENTOT ACIBAS
Alamat	PANORAMA CIBODAS Blok J. no.11 RT.01/12
Umur	57 TH
Profesi/Instansi	PENSIUNAN. TNI
Jabatan di Perumahan Panorama Cibodas	KOORDINATOR KEAMANAN LAPANGAN.
Saran atau tanggapan berdasarkan hasil uji coba prototype	<ul style="list-style-type: none"> - SANGAT MENYUKING DENGAN APLIKASI YANG BERPENGARUH LANTAR KEPENTINGAN KIM. - MUDAH MUDAH DENGAN HANYA APLIKASI TERSEBUT BISA MEMBUAT WARDA PANORAMA LEBIH NYAMAN DAN AMAN. - LEBIH MEMUDAHKAN PELAPORAN DENGAN KEJADIAN DENGAN CEPAT.
Tanda Tangan	<p>Tangerang, 13 - OCT - 2019</p>  (SENTOT .AB.)

Lampiran 4 Uji prototype dengan Koordinator Keamanan

Form Hasil Wawancara Uji Coba Prototype

Nama	JUTARNO'
Alamat	PERUM PANORAMA CIBODAS BLOK G. NO.9.
Umur	58 TH.
Profesi/Instansi	KORDINATOR KEAMANAN . TNI
Jabatan di Perumahan Panorama Cibodas	KORDINATOR KEAMANAN PERUMAHAN
Saran atau tanggapan berdasarkan hasil uji coba prototype	<ul style="list-style-type: none"> - SANGAT POSITIF DENG RENCANA APLIKASI TERSEBUT - - SANGAT MEMBANTU KEAMANAN DAN KENYAMANAN WARGA . - SANGAT MEMBANTU ANGGOTA KEAMANAN X6 SDH APA ..
Tanda Tangan	<p>Tangerang, 13 . NOV. 2019.</p>  <p>(JUTARNO)</p>

Lampiran 5 Uji prototype dengan Koordinator Keamanan

LAMPIRAN B HASIL PENGUJIAN USABILITY

B.1 Form Hasil Pengujian *Usability*

Pengujian *Usability* (Aplikasi SKIP)

Nama : Sanduki
Umur : 55 th
Alamat : panorama cibodas Blok H/11
Pekerjaan : Swasta

Tanda Tangan : 

1. Tugas untuk responden

No	Task	Detail	Hasil
1	Login	Pengguna akan diminta untuk melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar	✓
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan	✓
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan	✓
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan	✓
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih	✓
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi	✓
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat	✓
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat	✓
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi	✓



Scanned with
CamScanner

Lampiran 6 Pengujian *usability* dengan Ketua RW 12 (lembar 1)

2. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Aplikasi ini berguna bagi saya				✓	
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)		✓			
3	Aplikasi ini mudah untuk digunakan				✓	
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini		✓			
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan				✓	
6	Aplikasi ini tidak konsisten		✓			
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat				✓	
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan		✓			
9	Saya akan menggunakan aplikasi ini untuk seterusnya				✓	
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas		✓			



Scanned with
CamScanner

Lampiran 7 Pengujian *usability* dengan Ketua RW 12 (lembar 2)

Pengujian Usability (Aplikasi SKIP)

Nama : KARSIDI
 Umur : 42 Thn.
 Alamat : perumahan Cibodas T/C
 Pekerjaan : Karyawan Swast. /Keluarga RT. 01

Tanda Tangan :



1. Tugas untuk responden

No	Task	Detail	Hasil
1	Login	Pengguna akan diminta untuk melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar	✓
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan	✓
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan	✓
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan	✓
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih	✓
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi	✓
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat	✓
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat	✓
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi	✓



Scanned with
CamScanner

Lampiran 8 Pengujian *usability* dengan Ketua RT 01/RW 12 (lembar 1)

2. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Aplikasi ini berguna bagi saya				✓	
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)		✓			
3	Aplikasi ini mudah untuk digunakan				✓	
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini				✓	
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan				✓	
6	Aplikasi ini tidak konsisten		✓			
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat				✓	
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan		✓			
9	Saya akan menggunakan aplikasi ini untuk seterusnya				✓	
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas		✓			



Scanned with
CamScanner

Lampiran 9 Pengujian *usability* dengan Ketua RT 01/RW 12 (lembar 2)

Pengujian Usability (Aplikasi SKIP)

Nama : Acep Mu S. Tanda Tangan : 
 Umur : 52 th
 Alamat : Panorama Cibodas C-14
 Pekerjaan : Surasta / ket RT. 02

1. Tugas untuk responden

No	Task	Detail	Hasil
1	Login	Pengguna akan diminta untuk melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar	✓
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan	✓
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan	✓
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan	✓
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih	✓
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi	✓
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat	✓
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat	✓
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi	✓

Lampiran 10 Pengujian *usability* dengan Ketua RT 02/RW 12 (lembar 1)

2. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Aplikasi ini berguna bagi saya					✓
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)	✓				
3	Aplikasi ini mudah untuk digunakan					✓
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini					✓
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan					✓
6	Aplikasi ini tidak konsisten	✓				
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat					✓
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan	✓				
9	Saya akan menggunakan aplikasi ini untuk seterusnya					✓
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas	✓				

Lampiran 11 Pengujian *usability* dengan Ketua RT 02/RW 12 (lembar 2)

Pengujian Usability (Aplikasi SKIP)

Nama : Dwi Astuti
 Umur : 48 K
 Alamat : Cluster Panorama Cibodas Blok H
 No. 11.
 Pekerjaan : Ibu Rumah Tangga.

Tanda Tangan :



1. Tugas untuk responden

No	Task	Detail	Hasil
1	Login	Pengguna akan diminta untuk melakukan login dengan username dan password yang telah terdaftar	✓
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan	✓
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan	✓
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan	✓
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih	✓
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi	✓
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat	✓
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat	✓
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi	✓



Scanned with
CamScanner

Lampiran 12 Pengujian usability dengan Ibu Dwi Astuti (lembar 1)

2. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Aplikasi ini berguna bagi saya					✓
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)		✓			
3	Aplikasi ini mudah untuk digunakan					✓
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini		✓			
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan					✓
6	Aplikasi ini tidak konsisten		✓			
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat					✓
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan		✓			
9	Saya akan menggunakan aplikasi ini untuk seterusnya				✓	
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas		✓			



Scanned with
CamScanner

Lampiran 13 Pengujian *usability* dengan Ibu Dwi Astuti (lembar 2)

Pengujian *Usability* (Aplikasi SKIP)

Nama : BAGAS SUDARSONO
Umur : 23
Alamat : PANORAMA CIBODAS G/S
Pekerjaan : PEMUDA KARANG TARUNA

Tanda Tangan : 

1. Tugas untuk responden

No	Task	Detail	Hasil
1	Login	Pengguna akan diminta untuk melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang telah terdaftar	✓
2	Mengecek plat nomor kendaraan	Pengguna akan diminta untuk mengecek sebuah plat nomor kendaraan untuk memastikan apakah kendaraan tersebut milik warga Perumahan Panorama Cibodas atau bukan	✓
3	Membuat laporan	Pengguna akan diminta untuk membuat laporan	✓
4	Melihat laporan	Pengguna akan diminta untuk melihat sebuah informasi data laporan	✓
5	Melakukan panggilan darurat	Pengguna akan diminta untuk melakukan panggilan kepada nomor darurat yang dipilih	✓
6	Melihat panduan menu aplikasi	Pengguna akan diminta untuk melihat salah satu panduan menu aplikasi	✓
7	Mengirim pesan darurat	Pengguna akan diminta untuk mengirim pesan darurat	✓
8	Menerima pesan darurat	Pengguna akan dikondisikan untuk menerima pesan darurat	✓
9	Melihat informasi tentang aplikasi	Pengguna akan diminta untuk melihat informasi tentang aplikasi	✓



Scanned with
CamScanner

Lampiran 14 Pengujian *usability* dengan Pemuda Karang Taruna (lembar 1)

2. Kuesioner System Usability Scale (SUS)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Aplikasi ini berguna bagi saya				✓	
2	Aplikasi ini terlalu kompleks (banyak hal yang tidak perlu)	✓				
3	Aplikasi ini mudah untuk digunakan				✓	
4	Saya membutuhkan bantuan untuk menggunakan aplikasi ini				✓	
5	Aplikasi ini memiliki fitur seperti yang saya harapkan pada aplikasi di bidang keamanan perumahan				✓	
6	Aplikasi ini tidak konsisten		✓			
7	Saya merasa kebanyakan orang akan mudah menggunakan aplikasi ini dengan cepat					✓
8	Saya merasa aplikasi ini sulit/rumit untuk digunakan	✓				
9	Saya akan menggunakan aplikasi ini untuk seterusnya				✓	
10	Aplikasi ini tidak sesuai dengan warga Perumahan Panorama Cibodas	✓				✗



Scanned with
CamScanner

Lampiran 15 Pengujian *usability* dengan Pemuda Karang Taruna (lembar 2)

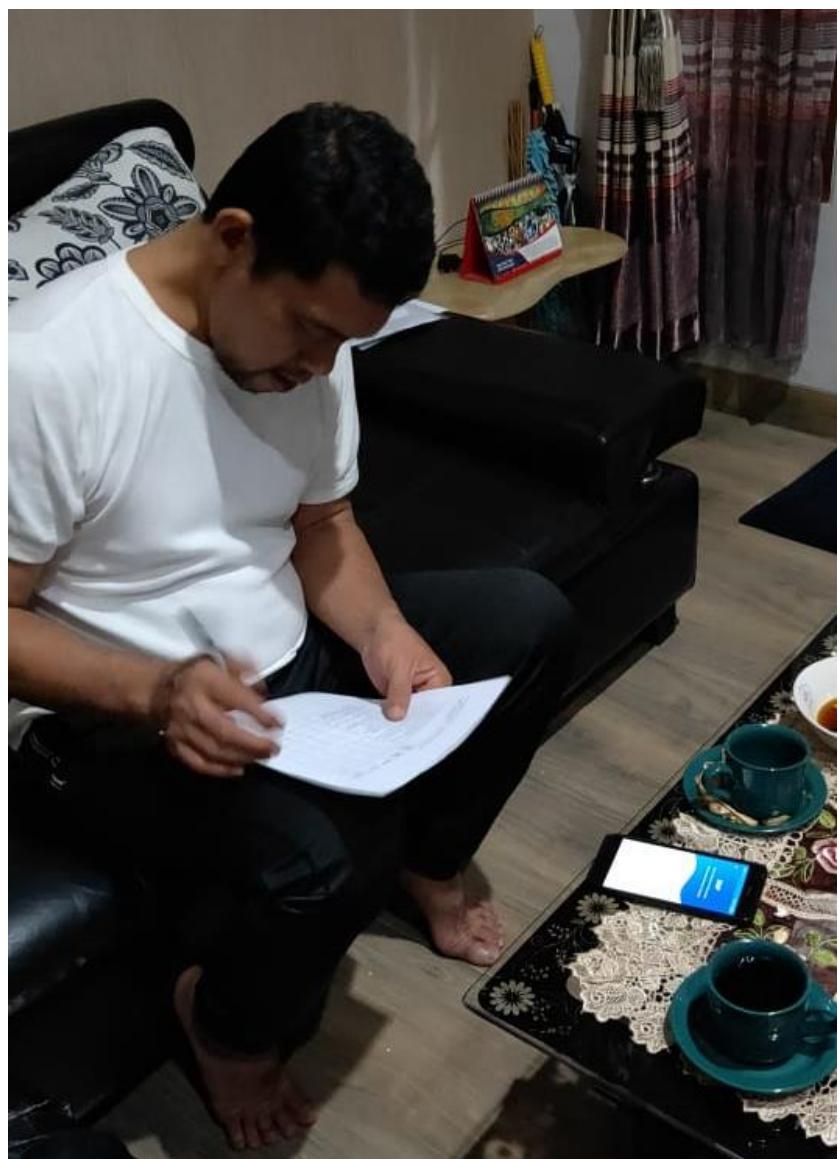
B.2 Foto Kegiatan Pengujian *Usability*



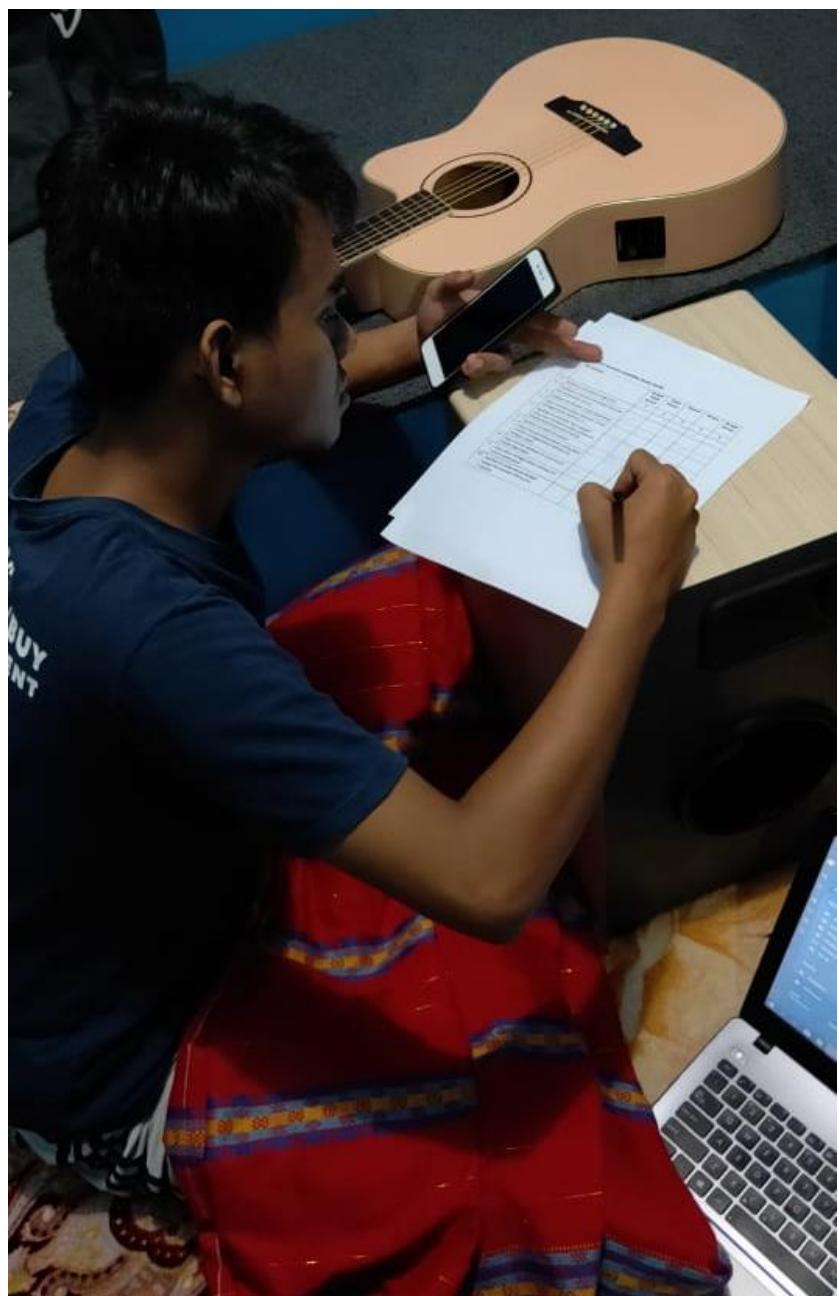
Lampiran 16 Foto pengujian *usability* dengan Ketua RW 12 dan Ibu Dwi Astuti



Lampiran 17 Foto pengujian *usability* dengan Ketua RT 01/RW 12



Lampiran 18 Foto pengujian *usability* dengan Ketua RT 02/RW 12



Lampiran 19 Foto pengujian *usability* dengan Pemuda Karang Taruna