

**ANALISIS DAN PERANCANGAN SISTEM PENGADUAN
MASYARAKAT KABUPATEN TANGERANG BERBASIS
*ANDROID PLATFORM***

TUGAS AKHIR

**diajukan untuk memenuhi syarat
mendapatkan gelar Sarjana Komputer**



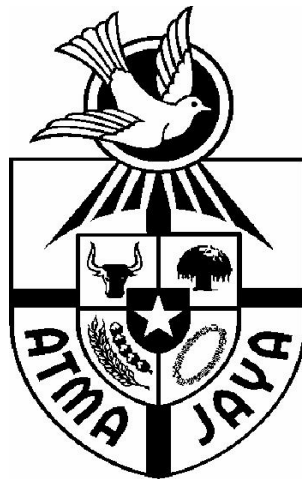
Disusun oleh:
Kevin Kurniawan
201804560004

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK INDONESIA ATMA JAYA
JAKARTA
2022**

***ANALYSIS AND DESIGN OF TANGERANG
REGION'S ANDROID BASED MOBILE
REPORTING SYSTEM***

FINAL ASSIGNMENT

***Proposed as a requirements for
Bachelor's degree in Information System***



***Created by:
Kevin Kurniawan
201804560004***

***INFORMATION SYSTEM DEPARTMENT
FACULTY OF ENGINEERING
ATMA JAYA CATHOLIC UNIVERSITY OF INDONESIA
JAKARTA
2022***



**FORMULIR
PERNYATAAN PEMBIMBING TUGAS AKHIR MAHASISWA**

PROGRAM STUDI SISTEM INFORMASI

NAMA : Kevin Kurniawan

NIM : 201804560004

JUDUL TUGAS AKHIR : Analisis dan Perancangan Sistem Pengaduan

Masyarakat Kabupaten Tangerang Berbasis

Android Platform

Tanggal Ujian T.A : 27 Januari 2022

Menyatakan bahwa **Tugas Akhir dan Makalah** telah dikoreksi sesuai dengan hasil sidang.

Jakarta, 31 Januari 2022

Mengetahui KaProdi

**(Julius Victor Manuel Bata,
S.Kom., M.T)**

Pembimbing Tugas Akhir

**(Ferdian Aditya Pratama,
S.Kom., M.Cs)**

LEMBAR PENGESAHAN

Yang bertanda tangan di bawah ini menyatakan bahwa:

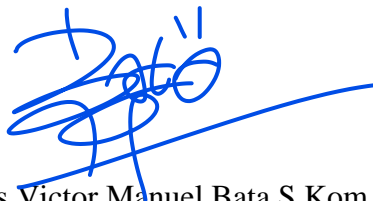
Nama : Kevin Kurniawan
NIM : 201804560004
: Analisis dan Perancangan Sistem Pengaduan
Judul Tugas Akhir Masyarakat Kabupaten Tangerang Berbasis
Android Platform
Tanggal Ujian : 27 Januari 2022

telah dinyatakan lulus ujian Tugas Akhir dan buku Tugas Akhir telah diperbaiki,
diperiksa dan disetujui oleh pembimbing (dan Co. Pembimbing) Tugas Akhir.

Jakarta, 31 Januari 2022

Mengetahui,

Ketua Program Studi Sistem Informasi



Julius Victor Manuel Bata S.Kom., M.T

Pembimbing Tugas Akhir



Ferdian Aditya Pratama,
S.Kom., M.Cs

LEMBAR PERNYATAAN

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini mahasiswa:

N a m a : Kevin Kurniawan
N I M : 201804560004
Judul Tugas Akhir : Analisis dan Perancangan Sistem Pengaduan
Masyarakat Kabupaten Tangerang Berbasis *Android*
Platform

menyatakan bahwa Tugas Akhir ini adalah hasil karya sendiri. Tugas Akhir ini bukan merupakan duplikasi sebagian atau seluruh dari karya orang lain yang sudah pernah dipublikasikan atau yang sudah pernah dipakai untuk mendapatkan gelar di salah satu Perguruan Tinggi. Bagian yang bersumber dari karya orang lain dicantumkan dengan cara referensi sebagaimana mestinya.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi berupa pembatalan Tugas Akhir apabila terbukti melakukan duplikasi dari Tugas Akhir yang sudah ada.

Tangerang, 14 Januari 2022



Kevin Kurniawan



**FORMULIR
PERNYATAAN TANDA SELESAI TUGAS AKHIR**

Yang bertanda tangan di bawah ini, pembimbing Tugas Akhir dari mahasiswa :

Nama : Kevin Kurniawan

NIM / St. ID : 201804560004 / 12018000397

Judul : Analisis dan Perancangan Sistem Pengaduan Masyarakat Kabupaten
Tangerang Berbasis *Android Platform*

Menyatakan bahwa mahasiswa tersebut di atas dapat mengikuti ujian tugas akhir
periode (bulan / tahun) : Januari 2022

Demikian pernyataan ini dibuat agar mahasiswa dapat mengikuti sidang tugas akhir.

Jakarta, 20 Januari 2022

Dosen pembimbing,

Ferdian Aditya Pratama, S.Kom., M.Cs

ABSTRAK

Perancangan Sistem Pengaduan Masyarakat Kabupaten Tangerang dirancang berdasar pada sistem pelaporan sebelumnya yang kurang efektif, pembuatan sistem pelaporan aduan dengan bentuk sosial media dan juga banyaknya kerusakan lingkungan serta kerusakan fasilitas umum yang sering terjadi di Kabupaten Tangerang. Penelitian ini dilakukan agar dapat menciptakan sebuah sistem yang dapat menjadi acuan dari pergantian sistem yang sudah ada, dengan melakukan beberapa tahapan metode yaitu analisis masalah, pengumpulan kebutuhan, perancangan sistem, implementasi, pengujian, dan pelaporan. Pengujian akan dilakukan dengan *usability testing* untuk menguji kelayakan sebuah sistem yang dibuat. Hasil dari penelitian yang dilakukan berupa sebuah aplikasi *mobile* dengan sistem operasi *Android* yang dapat menjadi acuan pergantian dari sistem yang sudah ada. Pengujian yang dilakukan terhadap penelitian ini mendapatkan hasil bahwa sistem yang dibuat layak untuk digunakan dengan nilai skor *SUS* 82,875 dan mendapatkan *grade scale* B (*Acceptable*).

Kata kunci: sistem, pengaduan, *android*, *firebase*, pemrograman

ABSTRACT

The design of the Tangerang Regency Community Complaint System designed based on the previous reporting system which was less effective, the creation of a complaint reporting system in the form of social media and also the amount of environmental damage and damage to public facilities that often occur in Tangerang Regency. This research was conducted in order to create a system that can be used as a reference for changing the existing system, by carrying out several stages of the method, namely problem analysis, requirements gathering, system design, implementation, testing, and reporting. Testing will be carried out with usability testing to test the feasibility of a system made. The results of the research conducted are in the form of an Android mobile operating system application that can be used as a reference for replacing the existing system. Tests carried out on this research obtained the results that the system made was feasible to use with an SUS score of 82,875 and got a grade scale B (Acceptable).

Keyword: system, report, android, firebase, programming

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **Analisis dan Perancangan Sistem Pengaduan Masyarakat Kabupaten Tangerang Berbasis *Android Platform***. Adapun tugas akhir ini diajukan sebagai salah satu syarat menempuh ujian sarjana komputer Program Studi Sistem Informasi di Universitas Katolik Indonesia Atma Jaya.

Pada penyusunan tugas akhir ini, tentu masih jauh dari kata sempurna dan masih banyak kekurangan. Namun penulis sudah berusaha semaksimal mungkin untuk menyusun tugas akhir ini dengan harapan dapat memiliki manfaat bagi banyak pihak. Oleh karena itu, penulis selalu menerima saran dan kritik yang diberikan agar dapat membangun tugas akhir ini menjadi lebih baik.

Tidak lupa penulis mengucapkan terima kasih kepada pihak yang terkait dalam penyusunan tugas akhir ini karena telah mendukung, membimbing dan memberikan semangat sehingga tugas akhir ini dapat diselesaikan. Berikut pihak-pihak yang terkait dalam penyusunan tugas akhir ini:

1. Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya kepada penulis.
2. Bapak Julius Victor Manuel Bata, S.Kom., M.T. selaku kepala Program Studi Sistem Informasi di Universitas Katolik Indonesia Atma Jaya.
3. Bapak Ferdian Adiya Pratama, S.Kom., M.Cs. selaku dosen pembimbing yang telah memberikan waktu, pikiran dan tenaganya serta pengetahuan, saran, arahan, dan bimbingan dalam penyusunan tugas akhir ini sampai selesai.

4. Kedua orang tua penulis yang telah menjadi *support system* terbaik dalam hidup selama ini.
5. Jawangi Unedo Dexius Sinaga karena telah membantu mendapatkan ide pembuatan tugas akhir.
6. Grup “Slurzs” yang telah memberikan dukungan dan semangat serta saran dan kritik mengenai aplikasi yang dibuat.
7. Teman-teman seperjuangan Sistem Informasi angkatan 2018 di Universitas Katolik Indonesia Atma Jaya terima kasih atas jalinan persahabatan dan kontribusi yang diberikan.
8. Serta pihak-pihak yang terkait yang tidak dapat disebutkan satu persatu yang telah memberikan bantuan secara langsung dan tidak langsung.

DAFTAR ISI

ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xvi
DAFTAR KODE PROGRAM	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah	4
1.4. Tujuan dan Manfaat	5
1.5. Sistematika Penulisan	5
BAB 2 TEORI PENDUKUNG.....	7
2.1. Kajian Pustaka	7
2.2. Landasan Teori	9
2.2.1. Sistem.....	9
2.2.2. Pengaduan	12

2.2.3. <i>Android</i>	13
2.2.4. <i>Firebase</i>	14
2.2.5. <i>Unified Modeling Language</i>	15
BAB 3 RANCANGAN PENELITIAN	21
3.1 Metodologi Penelitian.....	21
BAB 4 HASIL DAN PEMBAHASAN	36
4.1. Hasil & Implementasi	36
4.1.1. Halaman <i>login</i>	39
4.1.2. Halaman <i>register</i>	42
4.1.3. Halaman memasukkan <i>username</i>	43
4.1.4. Halaman utama masyarakat	45
4.1.5. Fungsi <i>posting</i> pengaduan.....	51
4.1.6. Fungsi notifikasi.....	62
4.1.7. Fungsi perubahan status, balasan sementara dan tindaklanjut	64
4.2. Hasil Analisis <i>Usability Testing</i>	69
BAB 5 SIMPULAN DAN SARAN.....	71
5.1. Simpulan	71
5.2. Saran	71
DAFTAR PUSTAKA	72
LAMPIRAN.....	76

DAFTAR TABEL

Tabel 2. 1.Kajian pustaka terdahulu.....	7
Tabel 2. 2.Simbol-simbol pada <i>use case diagram</i>	17
Tabel 2. 3.Simbol atau komponen pada <i>activity diagram</i>	20
Tabel 3. 1.Tabel pertanyaan kuesioner <i>usability testing</i>	33
Tabel 4. 1.Data responden dan jumlah skor <i>SUS</i>	69

DAFTAR GAMBAR

Gambar 1. 1.Grafik masalah sistem pelaporan sebelumnya	3
Gambar 3. 1.Alur penelitian.....	21
Gambar 3. 2. <i>Use case diagram</i> Sistem Pengaduan Masyarakat Kabupaten Tangerang	23
Gambar 3. 3. <i>Class diagram</i> Sistem Pengaduan Masyarakat Kabupaten Tangerang .	25
Gambar 3. 4. <i>Activity diagram</i> posting pengaduan	27
Gambar 3. 5. <i>Activity diagram</i> proses pemberian komentar & dukungan	29
Gambar 3. 6. <i>Activity diagram</i> proses penindaklanjutan pengaduan	30
Gambar 3. 7.Skor skala kelayakan sebuah sistem	35
Gambar 4. 1. <i>Menu SDK Manager</i>	36
Gambar 4. 2.Gambaran sistem secara keseluruhan.....	38
Gambar 4. 3.Halaman <i>login</i>	40
Gambar 4. 4.Halaman <i>register</i>	42
Gambar 4. 5.Halaman memasukkan <i>username</i>	44
Gambar 4. 6.Halaman utama masyarakat	46
Gambar 4. 7.Tombol <i>upvote</i> dan <i>downvote</i>	47
Gambar 4. 8.Tombol komentar	49
Gambar 4. 9.Halaman pemberian komentar	49
Gambar 4. 10.Tombol <i>posting</i> pengaduan	51
Gambar 4. 11.Halaman pemberian bukti aduan.....	52
Gambar 4. 12.Menampilkan bukti aduan yang dipilih oleh pengguna melalui galeri/kamera	55

Gambar 4. 13.	<i>Field</i> untuk memasukkan judul, catatan, kategori dan lokasi aduan...	56
Gambar 4. 14.	Hasil setelah tombol "Tambahkan lokasi" ditekan.....	59
Gambar 4. 15.	Tombol kirim	59
Gambar 4. 16.	Contoh data yang dimasukkan setelah menekan tombol "Kirim"	61
Gambar 4. 17.	Halaman notifikasi	62
Gambar 4. 18.	Detail dari aduan yang diajukan masyarakat.....	65
Gambar 4. 19.	Tombol berikan bukti tindakan	66
Gambar 4. 20.	Pemberian bukti tindakan dan deskripsi penindakan	67

DAFTAR LAMPIRAN

A- 1.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 1	76
A- 2.Pengujian sistem terhadap masyarakat Kabupten Tangerang 2.....	77
A- 3.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 3	78
A- 4.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 4.....	79
B- 1.Data <i>e-mail</i> dan <i>password</i> pengguna di <i>Firebase Authentication</i>	80
B- 2.Contoh data <i>posting</i> aduan pada <i>Firebase Realtime Database</i>	81
B- 3.Contoh data pemerintah dan masyarakat pada <i>Firebase Realtime Database</i>	82
C- 1.Koding fungsi navigasi komentar dan tombol <i>upvote</i> di <i>Android Studio</i>	83
C- 2.Koding fungsi memasukkan foto ke dalam database <i>Firebase Storage</i>	84
C- 3.Koding <i>xml</i> untuk membuat <i>design</i> sebuah halaman.....	85

DAFTAR KODE PROGRAM

Kode Program 4. 1.Fungsi <i>login</i>	40
Kode Program 4. 2.Fungsi <i>register</i>	43
Kode Program 4. 3.Fungsi memasukkan <i>username</i>	45
Kode Program 4. 4.Fungsi <i>Upvote</i>	47
Kode Program 4. 5.Fungsi <i>Downvote</i>	48
Kode Program 4. 6.Fungsi memberikan komentar	50
Kode Program 4. 7.Pembukaan galeri.....	53
Kode Program 4. 8.Pembukaan kamera	53
Kode Program 4. 9.Menampilkan bukti aduan ke imageView	54
Kode Program 4. 10.Mengambil informasi <i>latitude</i> dan <i>longitude</i> lokasi pengguna	56
Kode Program 4. 11.Pengiriman data aduan kedalam basis data	60
Kode Program 4. 12.Fungsi notifikasi	63
Kode Program 4. 13.Fungsi perubahan status.....	66
Kode Program 4. 14.Pemberian bukti tindakan	67

BAB 1

PENDAHULUAN

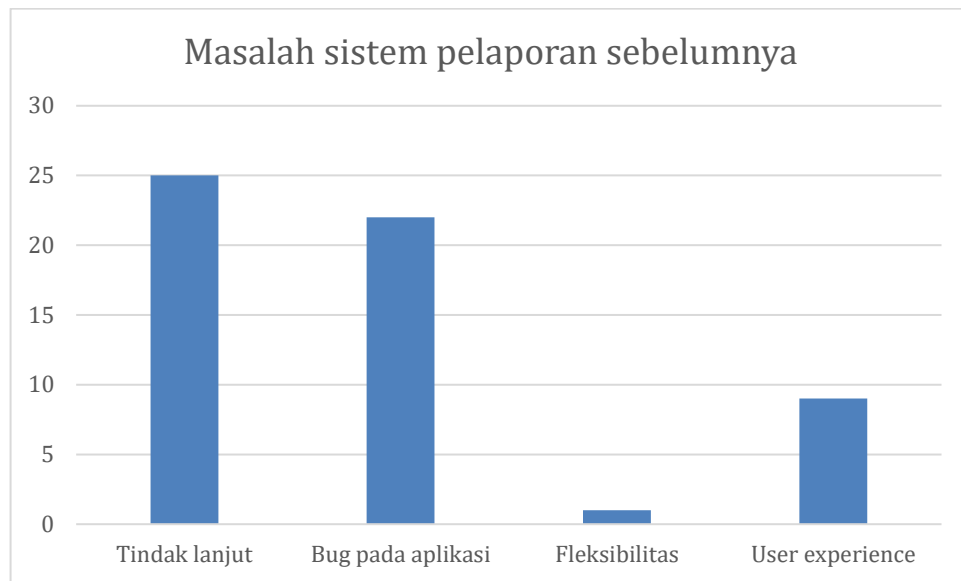
1.1. Latar Belakang

Perkembangan teknologi informasi sudah masuk ke dalam berbagai kalangan, baik anak-anak, remaja, dewasa maupun yang sudah usia lanjut masih termotivasi untuk terus mengikuti perkembangan zaman pada segi teknologi informasi, terutama pada bidang sosial media, *e-commerce* dan teknologi digital lainnya yang terhubung dengan internet (Sholihin et al., 2018). Dengan adanya perkembangan teknologi informasi yang sangat pesat, hal tersebut dapat dijadikan solusi kendala kerusakan lingkungan yang ada di Indonesia. Saat ini bumi mengalami tingkat kerusakan lingkungan tingkat kritis. Beberapa kerusakan lingkungan yang sering terjadi di Indonesia adalah jalanan yang bolong, sampah berserakan, got mampet, fasilitas kota yang rusak dan sebagainya. Tanpa disadari, hal ini sudah menjadi bagian dari kehidupan manusia (Juanda, 2018). Hal ini menunjukkan bahwa saat ini masalah lingkungan yang banyak terjadi di Indonesia perlu mendapat perhatian dari pemerintah provinsi (Deny Suryani, 2018) agar menciptakan lingkungan yang sehat dan nyaman.

Salah satu cara dalam mengatasi kerusakan lingkungan yang terjadi di Indonesia adalah dengan terhubungnya antara masyarakat dengan pemerintah tentang masalah yang terjadi di lingkungan sekitar. Dengan dibuatnya sebuah *platform* khusus untuk hal tersebut, dalam hal ini dibuatnya sistem pengaduan masyarakat yang bertujuan untuk menghubungkan pemerintah dengan masyarakat terkait masalah kerusakan lingkungan yang ada di Kabupaten Tangerang.

Selain terhubungnya antara masyarakat dengan pemerintah, masyarakat juga terhubung dengan masyarakat lain, karena dengan dibuatnya *platform* yang berbentuk sosial media, maka masyarakat dapat saling mendukung dalam pengajuan keluhan yang dilakukan, dengan fitur-fitur yang tersedia pada *platform*. Dibuatnya *platform* dengan basis *mobile application* dan berbentuk sosial media karena sekarang ini kalangan remaja secara intens mengakses sosial media dan menjelajahi internet pada *smartphone* mereka, bahkan sampai meninggalkan aktivitas lain (Alhakim, 2018). Hal tersebut juga memicu pada ketergantungan kalangan remaja terhadap *smartphone* dalam kehidupan sehari-hari. Fakta ini dapat menjadi alasan dasar pembuatan *platform* ini berbasis *mobile application*, karena pada umumnya lebih menarik bagi kalangan remaja (Utomo et al., 2020).

Sistem pelaporan yang sudah ada di Indonesia dapat diakses melalui *website* <https://www.lapor.go.id/>. Namun sistem pelaporan tersebut kurang efektif, karena berdasarkan aplikasi yang dibuat berbasis *Android* pada pusat pengunduhan aplikasi *playstore*, aplikasi "SP4N LAPOR!" mendapatkan ulasan buruk dari para pengguna. Masalah yang sering terjadi pada aplikasi tersebut antara lain terdapat banyak *bug* pada sistem *network* pada aplikasi, pemerintah jarang menindaklanjuti pengaduan yang diajukan, pelapor aduan bersifat tidak anonim, tidak adanya beberapa fitur penting yang dibutuhkan oleh masyarakat, dan lainnya.



Gambar 1. 1.Grafik masalah sistem pelaporan sebelumnya

Pada Gambar 1.1 dapat disimpulkan bahwa masalah yang kerap timbul pada sistem pelaporan sebelumnya cenderung ke arah tindak lanjut dari pemerintah, banyaknya *bug* pada aplikasi dan *user experience* yang kurang memuaskan. Oleh karena itu, dibuatlah *platform* khusus dengan fokus untuk menyelesaikan masalah yang terjadi di lingkungan sekitar dengan domisili khusus untuk menghubungkan masyarakat dengan pemerintah Kabupaten Tangerang. Dengan difokuskannya pengaduan melalui daerah tertentu, pemerintah dapat fokus membenahi masalah lingkungan yang terjadi di daerah tertentu, baik melalui pemerintah daerah maupun pemerintah pusat. Aplikasi juga dibuat menjadi lebih mudah digunakan oleh pengguna serta meminimalisir *bug* yang terjadi pada pelaporan sebelumnya, selain itu pada sistem pelaporan sebelumnya terdapat pelemparan yang terjadi antar pemerintah terhadap aduan yang diberikan oleh masyarakat jadi segi tindak lanjut menjadi terhambat karena pelemparan yang sering terjadi, hal ini juga menjadi salah satu dasar dilakukannya penelitian ini.

Platform yang saat ini sangat banyak digunakan oleh pengguna internet adalah *Android* (Ananda, 2018). *Android* merupakan sistem operasi pada perangkat keras *smartphone* yang berdasar pada versi modifikasi dari *linux* dan perangkat *open source* lainnya (Wahyu Setia Bintara, 2021). Oleh karena itu, sistem dibuat dengan menggunakan *platform Android Studio*. *Android Studio* adalah perangkat lunak (*software*) resmi yang dibuat oleh *Google* yang ditujukan untuk para pengembang *platform Android* (Loki Fadilah, 2019). Penggunaan *platform Android* ditujukan agar pelaporan aduan lebih mudah dan fleksibel, selain itu karena dasar dari aplikasi pengaduan ini berdasarkan bentuk sosial media maka dapat memudahkan masyarakat Kabupaten Tangerang untuk menggunakan *platform* ini sebagai sarana pemecahan solusi kerusakan lingkungan serta keluhan lain yang dialami oleh masyarakat. Tampilan sosial media dapat memudahkan para kalangan remaja karena saat ini remaja milenial sangat akrab dan cepat beradaptasi dengan teknologi, dengan *smartphone* sistem *Android* yang banyak memberikan fitur-fitur yang memudahkan para remaja milenial untuk mendapatkan informasi yang mereka inginkan (Ainiyah, 2018).

1.2. Rumusan Masalah

1. Bagaimana perancangan sistem pengaduan masyarakat untuk Kabupaten Tangerang?
2. Bagaimana proses pengujian sistem pengaduan masyarakat Kabupaten Tangerang?

1.3. Batasan Masalah

Batasan yang digunakan dalam pembuatan sistem ini yaitu:

1. Pengguna berdomisili di Kabupaten Tangerang.
2. Aplikasi dibuat menggunakan *platform Android Studio*.
3. Layanan basis data menggunakan *Firebase*.
4. Proses pengujian dilakukan dengan *usability testing*.
5. Pengguna terverifikasi sebagai aparat/pemerintah dapat mengubah status proses dan memberikan balasan pada pengaduan.

1.4. Tujuan dan Manfaat

Tujuan dari penelitian ini adalah membuat sistem pengaduan masyarakat yang dapat digunakan oleh masyarakat dan pemerintah Kabupaten Tangerang untuk memecahkan masalah kerusakan lingkungan.

Manfaat dari penelitian ini yaitu:

1. Pemerintah dan masyarakat Kabupaten Tangerang menjadi terhubung dalam segi penyelesaian masalah yang terjadi di masyarakat.
2. Pemerintah menjadi terfokus dalam menyelesaikan masalah untuk wilayah kerjanya sendiri.
3. Dapat menjadi acuan untuk pergantian sistem yang sudah ada.
4. Masyarakat menjadi lebih mudah untuk mengajukan pengaduan masalah.

1.5. Sistematika Penulisan

Sistematika penulisan dalam tugas akhir ini mengacu pada susunan sebagai berikut:

Bab 1 Pendahuluan membahas tentang masalah awal yang menjadi urgensi dari latar belakang penelitian, rumusan masalah, tujuan penelitian ini dilakukan, batasan masalah dan sistematika penulisan yang menjadi alur dari gambaran besar penulisan.

Bab 2 Landasan Teori membahas tentang penelitian terdahulu yang serupa dengan penelitian ini, teori-teori yang mendukung penelitian, sehingga penelitian dapat dibuat dan dilakukan secara terstruktur. Sumber dari teori-teori yang digunakan diperoleh melalui buku, jurnal terdahulu, sumber *online*, karya ilmiah yang memiliki keterkaitan dengan topik penelitian ini.

Bab 3 Metodologi Penelitian menjelaskan tentang gambaran alur keseluruhan penelitian dilakukan, yang terdiri dari menganalisis masalah yang terjadi, pengumpulan kebutuhan, perancangan system berupa *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*. Serta membahas tentang bagaimana pengujian sistem dilakukan.

Bab 4 Hasil dan Pembahasan menjelaskan tentang hasil dari pembuatan sistem pengaduan masyarakat Kabupaten Tangerang yang bernama Tecom berupa aplikasi *mobile* dan pengujian sistem yang dilakukan dengan *usability testing*. Penjelasan hasil pada bab ini berupa *screenshot interface* sistem Tecom dan potongan kode program fungsi utama yang ada pada sistem.

Bab 5 Simpulan dan Saran berisi rangkuman keseluruhan dari hasil penelitian yang menjadi jawaban atas tujuan penelitian ini dilakukan. Pada bagian saran berisi catatan untuk penelitian selanjutnya.

BAB 2 TEORI PENDUKUNG

2.1. Kajian Pustaka

Terdapat penelitian terdahulu yang sudah dilakukan sebelumnya, yang sudah dilakukan dan berkaitan dengan penelitian ini. Rangkuman penelitian terdahulu yang pernah dilakukan, ditunjukkan pada Tabel 2.1.

Tabel 2. 1.Kajian pustaka terdahulu

No	Peneliti	Tahun	Judul	Pembahasan
1	Faris D. Nurzam, Ika Nur Fajri, Donni Prabowo	2017	Rancang Bangun Aplikasi Media Laporan Aspirasi dengan <i>Firestore Cloud Messaging</i> Berbasis <i>Mobile</i>	Penelitian ini dilakukan dengan tujuan agar para instansi mendengar suara aspirasi masyarakat dengan memberikan pelayanan yang baik
2	Dekra Setia Darmawan, Dodi Ruhyadi, Kuswinanti	2019	Aplikasi Layanan Pengaduan Masyarakat Berbasis <i>Android</i>	Penelitian ini dilakukan dengan tujuan keluhan masyarakat dapat tersampaikan kepada para instansi, serta dilakukannya perbaikan terhadap keluhan tersebut
3	Andi Jumardi, Achmad Solichin	2016	Prototype Aplikasi Layanan Pengaduan Masyarakat Berbasis <i>Android</i> dan <i>Web Service</i>	Tujuan diberlakukannya penelitian ini yaitu untuk mengatasi permasalahan sampah yang terjadi di Indonesia agar laporan pengaduan tersampaikan kepada instansi/masyarakat setempat

Penelitian yang dilakukan oleh Faris D. Nurzam yaitu pembuatan aplikasi laporan aspirasi rakyat berbasis *mobile* menggunakan *firebase cloud messaging*. Penelitian ini berlatar belakang pada tidak tercapainya aspirasi masyarakat karena tidak adanya sarana yang memadai untuk hal tersebut, maka dibuatlah aplikasi ini agar

memudahkan aspirasi rakyat untuk sampai kepada para instansi. Menurut studi terdahulu dalam penelitian ini, masyarakat melakukan aspirasi melalui media kabar sebesar 53.8%, melalui radio sebesar 33.91% dan melalui fasilitas *SMS* sebesar 30.65%. Hal tersebut juga menjadi dasar dalam pembuatan aplikasi laporan aspirasi masyarakat yang dilakukan oleh Faris, dengan hasil dibuatnya aplikasi yang berfungsi penuh sesuai dengan tujuan, dan difokuskan pada fitur *push notification*, di mana setelah pengguna menuliskan sebuah aspirasi melalui *form*, maka instansi akan menerima notifikasi ada aspirasi masuk ke dalam ponsel pintar mereka. Menu dan fitur yang terdapat pada aplikasi penelitian ini ada *dashboard*, *form* untuk menulis aspirasi, *login*, melihat daftar laporan/aspirasi, melihat dan mengganti status laporan dan *push notification*.

Penelitian yang dilakukan oleh Dekra Setia Darmawan yaitu pembuatan aplikasi layanan pengaduan masyarakat berbasis *Android* yang berlatarbelakang pada instansi kelurahan menjadi peran terpenting dalam pelayanan masyarakat terutama dalam pelayanan pembuatan kartu tanda penduduk (KTP), kartu keluarga (KK) atau pelayanan masyarakat lainnya dengan tujuan agar tidak terjadinya kecurangan, pungutan liar atau kegiatan sejenis yang dapat membuat masyarakat tidak nyaman. Hasil dari penelitian ini adalah terbentuknya aplikasi pengaduan masyarakat yang berguna untuk menjadi alat pemantauan instansi kelurahan terhadap pelanggaran yang terjadi menggunakan *Android*.

Penelitian yang dilakukan oleh Andi Jumardi yaitu pembuatan aplikasi pengaduan masyarakat yang berbasis *Android* dan *website*. Platform *Android* sebagai sarana pengajuan aduan, tujuan dilakukannya penelitian ini adalah untuk mengatasi masalah sampah yang terjadi di Indonesia, sebagai *platform* atau sarana pengajuan

aduan tentang sampah yang mengganggu aktivitas masyarakat. Hasil dari penelitian yang dilakukan ini adalah terbentuknya aplikasi untuk melakukan pengaduan masyarakat tentang permasalahan lingkungan yang terjadi. *Menu* pada aplikasi ini yaitu *form* untuk memasukkan laporan, *list* laporan yang diadukan, dan peta laporan.

Dari ketiga penelitian terdahulu tersebut dengan penelitian yang dilakukan, memiliki kesamaan dalam beberapa aspek yaitu basis aplikasi *Android*, penggunaan layanan *Firebase*, *MySQL* sebagai sistem basis data. Perbedaan yang terdapat pada pembuatan sistem ini dengan penelitian terdahulu, ada sistem yang berbasis *website*, sedangkan penelitian ini memiliki basis *Android*. Penelitian ini dilakukan pengembangan dalam beberapa fitur yang dibutuhkan dalam sebuah sistem pengaduan masyarakat yang tidak ada pada penelitian terdahulu, seperti fitur lokasi di mana lokasi masalah yang diadukan, anonimitas dari pengguna diutamakan dan memiliki tampilan sosial media. Beberapa tujuan dari penelitian terdahulu juga memiliki kesamaan dengan penelitian ini, yaitu menghubungkan masyarakat dengan pemerintah agar menciptakan solusi yang terjadi pada lingkungan masyarakat, seperti masalah sampah di lingkungan sekitar, pengaduan tentang administrasi masyarakat dan lainnya.

2.2. Landasan Teori

2.2.1. Sistem

Sistem secara umum, dapat diartikan sebagai suatu kesatuan objek nyata atau abstrak yang terdiri dari komponen-komponen atau unsur yang saling berkaitan, saling mendukung, saling terhubung secara keseluruhan yang memiliki satu tujuan yang sama secara efektif dan efisien (M Prawiro, 2018). Sistem merupakan seperangkat elemen, subsistem, dan rakitan yang terintegrasi untuk mencapai tujuan yang

ditentukan. Hal ini termasuk produk perangkat keras, perangkat lunak, dan *firmware*, proses, orang, informasi, teknologi, kota, layanan, dan elemen pendukung lainnya (Dori et al., 2020).

Pembuatan sebuah sistem dilakukan memiliki dasar yaitu untuk mempermudah maupun memperbaiki sistem yang sudah ada agar lebih efektif dan lebih efisien, sistem ini dibuat agar wilayah Kabupaten Tangerang memiliki sebuah *platform* untuk memaparkan keluhan kesah masyarakat sekitar mengenai hal yang terjadi. Pemerintah juga dapat melihat berbagai macam hal yang terjadi di masyarakat secara detail melalui laporan yang diajukan lewat sistem notifikasi yang dibuat, sistem melaporkan pengaduan yang berbentuk sosial media yang menyangkut dengan foto, deskripsi dan kategori. Selain itu pemerintah juga dapat langsung melihat sistem lokasi yang dibuat agar memudahkan tindakan yang dilakukan terhadap aduan.

Berdasarkan sistem yang akan dibuat, sistem merupakan sebuah rangkaian yang terdiri atas beberapa komponen pendukung yang membuat sebuah hal menjadi satu kesatuan yang juga memiliki tujuan tertentu. Sistem juga memiliki karakteristik, bahwa sebuah rangkaian yang memiliki ciri yang dapat dikatakan hal itu merupakan sebuah sistem, menurut (Muhammad Muslihudin & Oktafianto, 2016) berikut merupakan karakteristik sebuah sistem:

1. Komponen sistem adalah sistem yang saling berhubungan, bagian-bagian yang beragam yang bersama-sama mencapai beberapa tujuan. Sistem terdiri dari tujuan, sasaran dan elemen yang dapat dikenali dan dilengkapi berdasarkan tujuan, bukan seperangkat elemen yang teratur.
2. Batasan sistem (*Boundary*) adalah area yang membatasi sistem dari suatu sistem ke sistem lain atau lingkungan eksternal.

3. Lingkungan di luar sistem (*Environment*) adalah lingkungan di luar batas sistem yang mempengaruhi operasi sistem. Lingkungan eksternal dapat merusak sistem. Lingkungan eksternal yang menguntungkan adalah bentuk energi dari sistem, dan lingkungan eksternal yang tidak menguntungkan perlu dibatasi dan dikendalikan. Jika tidak, kontinuitas sistem akan terganggu.
4. Sistem penghubung (*Interface*) adalah media koneksi antara satu subsistem dengan subsistem lainnya. Penghubung ini memungkinkan sumber daya mengalir dari satu sistem ke sistem lain dengan menghubungkan subsistem yang dapat berinteraksi dengan subsistem lain untuk membentuk satu kesatuan.
5. Sistem masukan (*Input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa *input* pemeliharaan dan *input* sinyal. Pekerjaan pemeliharaan adalah energi yang digunakan untuk menjalankan sistem. Sinyal *input*, di sisi lain adalah energi yang diproses untuk mendapatkan *output*.
6. Sistem keluaran (*Output*) adalah energi yang diproses dan dipecah menjadi keluaran yang berguna bagi subsistem lain.
7. Sistem sasaran adalah sistem yang memiliki maksud atau tujuan. Jika sistem tidak memiliki batas target, operasi sistem tidak akan berguna.

Berdasarkan kedua pemaparan tersebut tentang sistem, dapat disimpulkan bahwa sistem adalah suatu kesatuan yang saling berhubungan yang memiliki komponen yang terhubung, mendukung satu sama lain untuk mencapai tujuan tertentu dalam berbagai macam bentuk produk seperti perangkat lunak, perangkat keras, dan lain-lain.

2.2.2. Pengaduan

Menurut Kamus Besar Bahasa Indonesia, Pengaduan adalah penyambungan, proses, cara, perbuatan mengadu, ungkapan rasa tidak senang atau tidak puas akan hal-hal tidak begitu penting, tetapi perlu diperhatikan. Menurut Kitab Undang-Undang Hukum Acara Pidana (KUHP) Pasal 1 Butir 25, Pengaduan adalah pemberitahuan disertai permintaan oleh pihak yang berkepentingan kepada pejabat yang berwenang untuk menindak menurut hukum seorang yang telah melakukan tindak pidana aduan yang merugikannya.

Sarana pengaduan yang dimiliki oleh masyarakat Kabupaten Tangerang melalui internet yaitu melalui <https://www.lapor.go.id/> pada *platform* tersebut masyarakat dapat melakukan pengaduan, mengajukan aspirasi dan permintaan informasi kepada pemerintah, dengan memberikan beberapa informasi yang akan dikirimkan kepada pemerintah seperti pada pengaduan, masyarakat dapat memberikan judul pengaduan, deskripsi pengaduan, tanggal, lokasi kejadian, kategori serta lampiran berupa gambar. Selain itu pada bagian aspirasi masyarakat dapat langsung memberikan judul, deskripsi aspirasi, asal pelapor dan kategori sebagai informasi yang dikirimkan kepada pemerintah. Pada bagian klasifikasi permintaan laporan kolom yang diberikan untuk melapor sama dengan kolom pada pengajuan aspirasi.

Selain pada *website*, terdapat sistem pelaporan yang dapat digunakan juga oleh masyarakat Kabupaten Tangerang yaitu aplikasi LAPOR! pada tempat *install* aplikasi di *Android* dan *iOS* masyarakat dapat mengajukan aduan dengan mengirimkan deskripsi dan foto maupun deskripsi saja. Masyarakat lain juga dapat memberikan dukungan berupa *like*, namun ada beberapa faktor yang membatasi masyarakat dalam melakukan pengaduan seperti *loading* yang terjadi terus menerus (kesalahan *server*)

dan juga segi tindak lanjut yang ditangani oleh pemerintah sangat minim menurut ulasan aplikasi di *playstore*, ini didasarkan pada penggunaan aplikasi ini umum untuk satu negara. Hal ini juga menjadi dasar sebagai pembuatan sistem secara terpusat pada satu kabupaten/wilayah saja sehingga pemerintah dapat fokus dalam melakukan tindakan. Sebuah sistem pengaduan dibutuhkan oleh masyarakat agar pemerintah dapat melakukan perbaikan pada keluhan yang terjadi di lingkungan sekitar.

Berdasarkan pernyataan tersebut, pengaduan berarti suatu hal yang berkaitan dengan kepuasan masyarakat terhadap layanan yang diberikan oleh pemerintah, berdasarkan hal tersebut berarti sebuah aksi permintaan masyarakat kepada para pejabat yang berwenang untuk menindaklanjuti masalah yang terjadi.

2.2.3. Android

Android merupakan sebuah sistem operasi yang dirancang oleh *Google* yang berbasis *kernel linux* untuk kinerja perangkat *smartphone*. Jadi, *Android* digunakan lewat sentuhan pada layar *gadget* (Putra, 2019). *Android* adalah sistem operasi dengan sumber terbuka (*open source*) dengan lisensi *apache*, karena *Android* sistem operasi terbuka, maka lisensi perizinan tersebut memungkinkan para pengembang untuk memodifikasi dan mendistribusikan perangkat lunak secara bebas (Fikri Akhdi Maulana, 2014).

Pada penelitian ini, bahasa pemrograman yang digunakan yaitu Java. Java merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi *native Android*. Bahasa pemrograman *Java* dapat digunakan untuk berbagai macam *platform* seperti *desktop*, *Android*, dan *linux*. Dalam pembuatan aplikasi *Android* untuk penelitian ini, diperlukan beberapa *tools* yaitu:

1. JDK

JDK atau *Java Development Kit* adalah implementasi dari *platform Java SE*, *Java EE* atau *Java ME* yang dirilis dalam format biner oleh Oracle Corporation dan ditujukan untuk pengembang *Java* di Solaris, Linux, Mac OS X, atau Windows. Dikatakan JDK berisi JVM pribadi dan beberapa suber daya lain untuk melengkapi resep aplikasi *Java*. Ini adalah pengembangan perangkat lunak (SDK) yang paling banyak digunakan sejak *platform Java* diperkenalkan.

2. Android SDK

Android SDK adalah alat untuk programmer yang ingin mengembangkan aplikasi berbasis *Google Android*. Android SDK berisi satu set lengkap alat pengembangan. *Android SDK* terdiri dari debugger, *library*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. *Android* saat ini mendukung arsitektur x86 di Linux, Mac OS X dan Windows XP atau Vista. Persyaratan termasuk JDK, Apache Ant, dan Python, IDE yang didukung secara resmi adalah Eclipse 3.2 atau yang lebih baru dengan *plugin Android Development Tools* (ADT). Hal ini memungkinkan pengembang untuk mengubah file *Java* menggunakan editor teks.

3. AVD

Android Virtual Device (AVD) merupakan emulator yang memungkinkan para pengembang menjalankan aplikasi yang dibuat secara langsung.

2.2.4. *Firestore*

Menurut (Dicoding Intern, 2020) *Firestore* merupakan layanan dari *Google* guna memberikan kemudahan bagi para pengembang untuk mengembangkan aplikasinya,

Firebase saat ini menjadi solusi untuk para pengembang agar bisa mempercepat pekerjaan pembuatan aplikasi.

Pada penelitian ini, menggunakan beberapa jenis-jenis dan fitur yang terdapat pada *Firebase*. Berikut adalah beberapa fitur yang digunakan pada penelitian ini:

1. *Firebase Cloud Messaging*: *Firebase Cloud Messaging* merupakan layanan dari *Firebase* yang digunakan untuk mendapatkan fitur *push notification* pada perangkat *Android* dan *iOS* (Andika Wasisto, 2018).
2. *Firebase Authentication*: *Firebase Authentication* merupakan sebuah layanan dari *Firebase* yang digunakan untuk membuat sistem *login* dan *register*, terdapat juga untuk *login* dengan *facebook*, *twitter*, dan sebagainya. Dengan *firebase authentication* para pengembang tidak perlu khawatir untuk membuat metode konfirmasi data pengguna untuk aplikasi mereka (Avin Riyan, 2018).
3. *Firebase Real Time Database*: *Firebase Real Time Database* merupakan sebuah layanan dari *Firebase* yang digunakan sebagai media penyimpanan data dari aplikasi secara *realtime* (Sanad, 2019).

2.2.5. Unified Modeling Language

Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam desain dan manufaktur perangkat lunak berorientasi objek. *UML* adalah standar penulisan yang di dalamnya terdapat proses bisnis yang menulis kelas-kelas dalam bahasa tertentu (M Teguh Prihandoyo, 2018). Beberapa diagram *UML* yang digunakan untuk merancang sistem pada penelitian ini adalah sebagai berikut:

1. *Use Case Diagram*

Use Case Diagram merupakan diagram yang bekerja dengan menggambarkan interaksi khas antara pengguna (user) suatu sistem dengan sistem yang terpisah melalui riwayat penggunaan suatu sistem. *Use Case Diagram* terdiri dari aktor dan interaksi yang dibuat, aktor-aktor dapat berupa orang, perangkat keras, sistem lain atau mereka yang berinteraksi dengan sistem (T Bayu Kurniawan, 2020).

Menurut (Michael Kharisma Hutauruk, 2019) *Use Case Diagram* memiliki 3 komponen, yaitu:

a. Sistem

Meliputi keterbatasan sistem dalam hal aktor yang menggunakannya (di luar sistem) dan fitur yang perlu disediakan (di dalam sistem).

b. Aktor











Sesuatu di luar sistem yang digunakan sistem untuk melakukan sesuatu. Ini bisa berupa orang, sistem, atau perangkat yang berperan dalam keberhasilan pengoperasian sistem.

c. *Use Case*

Deskripsi fungsional dari sebuah sistem. Dengan demikian antara konsumen dan pengguna sistem akan mengerti atau mengerti fungsi dari sistem yang dibangun.

Use Case Diagram memiliki beberapa simbol untuk mendeskripsikan *use case* secara umum, berikut merupakan penjelasan simbol-simbol yang terdapat pada *Use Case Diagram* pada Tabel 2.2 menurut (Flin Setyadi, 2021).

Tabel 2. 2.Simbol-simbol pada *use case diagram*

Simbol	Deskripsi
 Aktor	Merupakan orang atau benda (misalnya, perangkat atau sistem lain) yang berinteraksi dengan sistem. Aktor hanya berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kendali atas <i>use case</i> .
 <i>Dependency</i>	Suatu hubungan yang menyatakan perubahan yang terjadi pada elemen independen mempengaruhi elemen yang bergantung pada elemen yang tidak independen.
 <i>Generalization</i>	Hubungan di mana objek anak berbagi perilaku dan struktur data objek di atas objek induk.
 <i>Include</i>	Hubungan antar <i>use case</i> di mana <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> tersebut untuk menjalankan fungsinya.
 <i>Extend</i>	Hubungan antar <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat independen tanpa <i>use case</i> tambahan.
 <i>Association</i>	Menghubungkan <i>link</i> antar elemen.
 <i>System boundary</i>	Untuk memperlihatkan batasan sistem dalam <i>use case</i> .
 <i>Use case</i>	Untuk menggambarkan bagaimana aktor menggunakan sistem.
 <i>Collaboration</i>	Interaksi aturan dan elemen lain yang bekerja sama memungkinkan perilaku yang lebih besar dari jumlah elemennya.
 <i>Note</i>	Elemen fisik yang ada saat aplikasi berjalan dan yang mewakili sumber daya komputasi.

2. *Class Diagram*

Class diagram merupakan sebuah *diagram* yang menunjukkan interaksi antar kelas dalam sistem. *Class diagram* dibuat berdasarkan diagram *use case* dan diagram sekuensial yang telah dibuat sebelumnya (Selatan et al., 2018).

Menurut (Alifia Seftin Oktriwina, 2021) terdapat beberapa komponen yang ada pada *class diagram*, komponen-komponen tersebut yaitu:

a. Komponen atas

Bagian ini berisi nama kelas yang selalu diperlukan dalam *classifier* atau objek.

b. Komponen tengah

Komponen ini berisi atribut kelas yang digunakan untuk menggambarkan kualitas kelas. Ini hanya diperlukan ketika menjelaskan contoh spesifik dari suatu kelas.

c. Komponen bawah

Bagian ini merupakan komponen diagram kelas yang berisi operasi kelas yang ditampilkan dalam format daftar. Dengan demikian, setiap operasi mengambil jalurnya sendiri.

d. Komponen tambahan

- 1) *Class*, mewakili suatu objek atau sekelompok objek yang memiliki struktur serupa.
- 2) *Signals*, simbol yang mewakili komunikasi asinkron satu arah antara objek aktif.


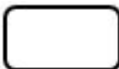




- 3) Tipe data dibentuk dengan mengeksekusi *classifier* yang menentukan nilai data. Tipe data ini dapat menghasilkan tipe dan jumlah primitif.
- 4) *Packages* untuk mengatur pengklasifikasi terkait pada diagram. Komponen ini dilambangkan dengan bentuk persegi panjang.
- 5) *Interface* merupakan satu set atribut yang mendefinisikan satu set perilaku terkait. Komponen ini mirip dengan kelas, tetapi harus memiliki setidaknya satu kelas untuk mengimplementasikannya.
- 6) *Enumerations* menggambarkan tipe data yang ditentukan pengguna. Komponen enumerasi berisi sekelompok pengidentifikasi yang mewakili nilai enumerasi.
- 7) Objek yaitu item yang dapat ditambahkan ke diagram kelas untuk mewakili contoh atau prototipe tertentu.
- 8) *Artifacts* merupakan elemen model yang mewakili entitas tertentu dalam sistem perangkat lunak. Misalnya, dokumen, *database*, *file* yang dapat dieksekusi, komponen perangkat lunak, dll.

3. *Activity Diagram*

Activity diagram menurut (Yuli Syafitri, 2016) suatu diagram yang menggambarkan aliran aktivitas yang berbeda dalam sistem yang akan dirancang, bagaimana setiap aliran dimulai, keputusan apa yang dapat dibuat dan bagaimana aktivitas berakhir.

Activity diagram memiliki beberapa simbol untuk penyusunannya, menurut (Ahmad Ansori, 2020) berikut beberapa simbol yang ada pada *activity diagram* ditunjukkan pada Tabel 2.3.

Tabel 2. 3.Simbol atau komponen pada *activity diagram*

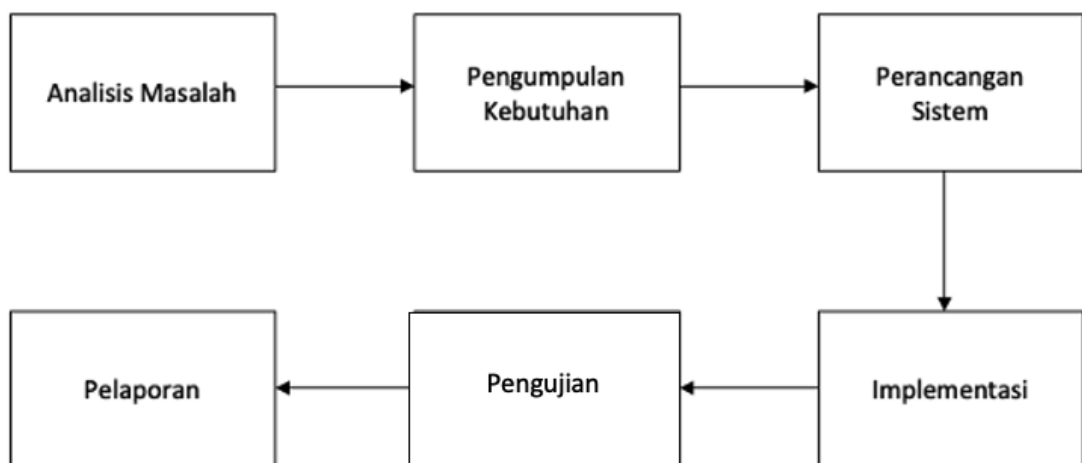
Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Percabangan di mana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / <i>Join</i>	Penggabungan di mana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

BAB 3

RANCANGAN PENELITIAN

3.1 Metodologi Penelitian

Pengertian metodologi penelitian menurut (Sugiyono, 2015) merupakan sebuah cara ilmiah untuk memperoleh, menemukan, mengembangkan, dan menunjukkan data yang berguna yang kemudian dapat digunakan untuk memahami, memecahkan dan mengantisipasi masalah tertentu.



Gambar 3. 1. Alur penelitian

Gambar 3.1 merupakan alur penelitian yang terdiri atas enam tahapan sebagai berikut:

1. Analisis masalah

Masalah utama yang dapat menjadi patokan pada dibuatnya penelitian ini adalah Kabupaten Tangerang tidak mempunyai sistem pengaduan yang secara khusus dipegang oleh pemerintah daerah Kabupaten Tangerang, sehingga terdapat masalah lingkungan yang kerap terjadi yang belum dilakukannya penanganan pada masalah

tersebut. Sistem pengaduan yang sudah ada tidak di fokuskan pada satu daerah untuk penanganan pemerintah, melainkan dikumpulkan menjadi satu sistem yang membuat pelemparan antar pemerintah atas pengaduan yang diajukan. Menurut suara rakyat yang menggunakan sistem tersebut, hal itu sering terjadi dan malah membuat masalah tidak terselesaikan bahkan dihapuskan dari keberadaan di basis data aplikasi, hal ini masuk kedalam aspek tindak lanjut pemerintah yang menjadi salah satu masalah pada sistem pengaduan yang sudah ada. Selain itu, masyarakat juga mengalami masalah *bug* pada aplikasi yang sering terjadi mulai dari lambatnya koneksi ke *server*, tombol yang tidak bisa ditekan, alur aplikasi yang tidak sesuai dan lain-lain. Fitur penting yang tidak tersedia pada sistem pelaporan sebelumnya, seperti lokasi di mana sistem pelaporan tersebut dilakukan yang langsung dapat diarahkan ke *Google Maps* untuk melakukan tindakan.

2. Pengumpulan kebutuhan

Pada tahapan ini, penelitian dilakukan dengan studi literatur yang berguna untuk mendukung teori-teori yang terkait dengan masalah yang dibahas. Pada penelitian ini, studi literatur dilakukan dengan membaca jurnal-jurnal, buku, artikel pada *website* serta penelitian yang sudah dilakukan sebelumnya, dilakukannya studi literatur guna mendapatkan referensi yang menjadi alasan kuat untuk diberlakukannya penelitian ini. Selain itu, dilakukannya penelitian ini juga berdasarkan hal yang terjadi di lingkungan sekitar Kabupaten Tangerang karena terdapat banyaknya masalah lingkungan yang terjadi, seperti jalanan bolong, sampah berserakan dan lainnya yang belum diberlakukannya tindak lanjut dari pemerintah atas masalah tersebut. Studi literatur

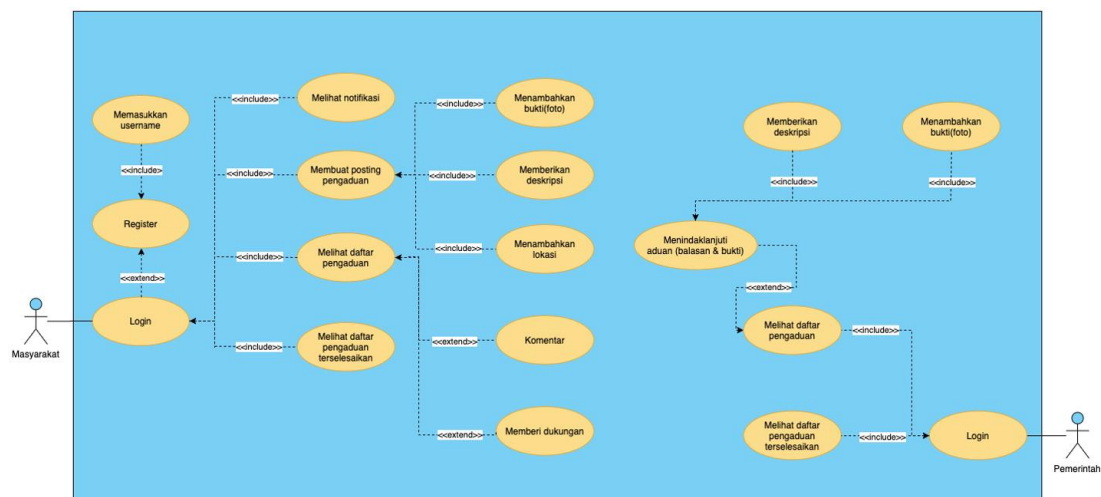
juga digunakan untuk penyusunan kuesioner sebagai pengumpulan data untuk pengembangan fitur pada aplikasi yang akan dibuat.

3. Perancangan sistem

Pada tahapan ini, perancangan sistem menggunakan UML (*Unified Modeling Language*). UML adalah bahasa pemodelan visual yang berguna untuk mendefinisikan, memvisualisasikan, membuat, dan mendokumentasikan sistem perangkat lunak. Selain itu, UML berguna untuk memahami desain, konfigurasi, pemeliharaan dan pengendalian informasi tentang sistem yang dibangun (Leslie Hendric Spits Warnars, 2017).

a. *Use Case Diagram*

Use Case Diagram pada sistem ini ditunjukkan pada Gambar 3.2. Dalam aplikasi sistem pengaduan masyarakat ini, terdapat dua aktor yaitu Masyarakat dan Pemerintah.



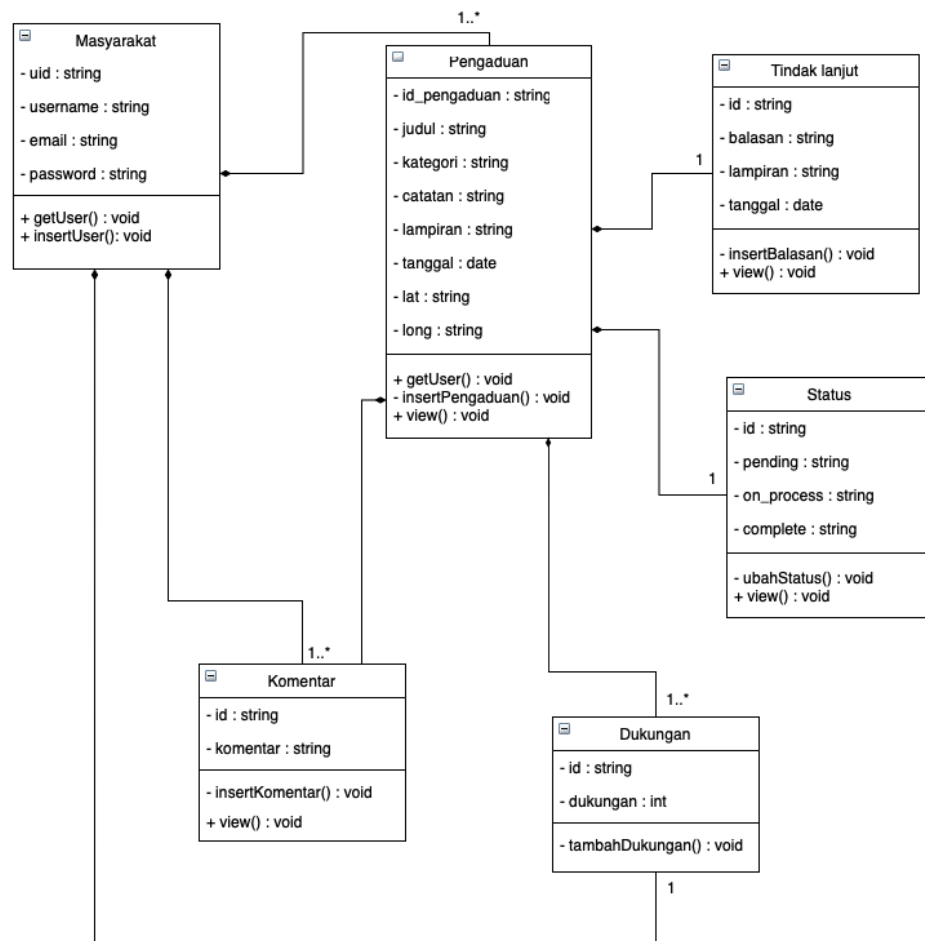
Gambar 3. 2. *Use case diagram* Sistem Pengaduan Masyarakat Kabupaten Tangerang

Para aktor tersebut memiliki hak dan kewajiban masing-masing dalam penggunaan fitur pada aplikasi. Berikut merupakan penjelasan tiap aktor:

- 1) *Register*: Masyarakat dapat melakukan registrasi untuk menggunakan aplikasi, data pada saat registrasi yaitu *e-mail*, *password*, dan *username*.
- 2) *Login*: Masyarakat dan Pemerintah dapat melakukan *login* kedalam aplikasi sesuai *role* masing-masing.
- 3) Memasukkan *username*: Untuk membuat *posting* dan menggunakan aplikasi, masyarakat harus memasukkan *username* sebagai tanda pengenal.
- 4) Membuat *posting* pengaduan: Masyarakat dapat melakukan pengaduan dengan fitur *posting*.
- 5) Memberikan deskripsi: Masyarakat harus menuliskan *detail* dari pengaduan yang diajukan.
- 6) Menambahkan bukti (foto): Masyarakat harus memberikan bukti pengaduan berupa foto.
- 7) Menambahkan lokasi: Masyarakat harus menambahkan lokasi di mana pengaduan diajukan.
- 8) Melihat daftar pengaduan: Masyarakat dan Pemerintah dapat melihat daftar pengaduan yang diajukan oleh masyarakat.
- 9) Melihat daftar pengaduan terselesaikan: Masyarakat dan pemerintah dapat melihat daftar pengaduan yang sudah diselesaikan oleh pemerintah.
- 10) Komentar: Masyarakat bisa memberikan komentar terhadap pengaduan yang dilakukan oleh warga lain.
- 11) Memberi dukungan: Antar masyarakat dapat mendukung pengaduan satu sama lain agar lebih cepat terlihat oleh pihak pemerintah dengan cara *like*.

- 12) Menindaklanjuti pengaduan: Pemerintah harus menindaklanjuti pengaduan yang diajukan oleh masyarakat dengan memberikan deskripsi dan mengirim bukti perbaikan yang dilakukan.
- 13) Melihat notifikasi: Masyarakat dapat melihat status proses pengaduan.

b. *Class Diagram*



Gambar 3. 3. *Class diagram* Sistem Pengaduan Masyarakat Kabupaten Tangerang

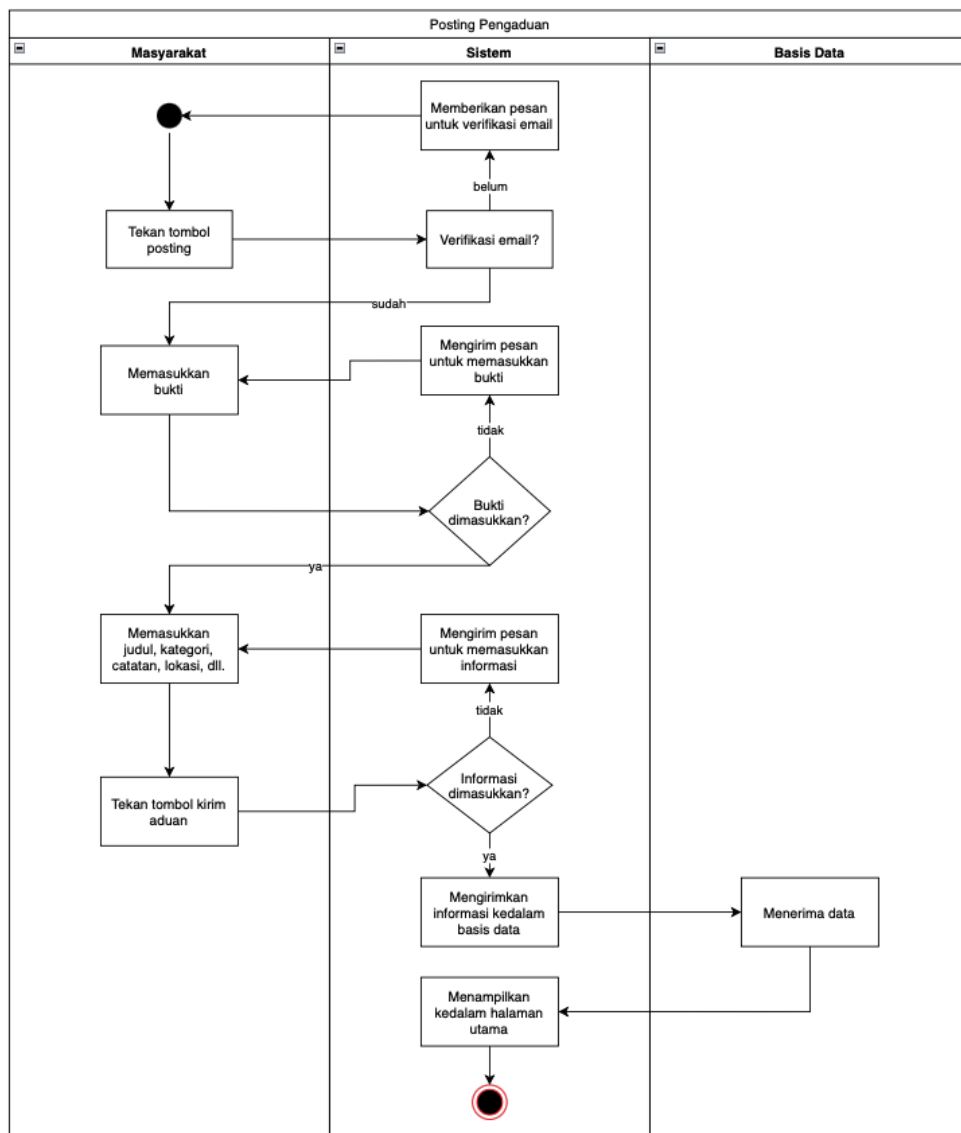
Pada Gambar 3.3 ditunjukkan *class diagram* dari sistem yang akan dibuat, terdapat enam kelas yang ada pada sistem ini yaitu:

- 1) Masyarakat: Memiliki beberapa properti seperti *uid*, *username*, *email* dan *password* saja karena untuk tujuan *anonymity*. Kelas ini juga memiliki beberapa metode yaitu *getUser()* untuk verifikasi login pengecekan kedalam basis data & untuk identifikasi pembuatan *posting*. *insertUser()* untuk masyarakat melakukan registrasi ke aplikasi.
- 2) Pengaduan: Memiliki beberapa properti seperti *id_pengaduan*, *judul*, *kategori*, *catatan*, *lampiran*, *tanggal*, *lat*, *long* untuk *lat* dan *long* untuk informasi lokasi pengaduan dilakukan. Kelas ini memiliki beberapa metode yaitu *getUser()* untuk menjadi identifikasi pembuat *posting*. *insertPengaduan()* berguna untuk memasukkan pengaduan kedalam basis data. *view()* untuk menampilkan daftar pengaduan dari basis data ke halaman utama dan halaman pengaduan yang terselesaikan.
- 3) Tindak lanjut: Memiliki beberapa properti seperti *id*, *balasan*, *lampiran* dan *tanggal balasan* berupa teks yang diberikan dari pemerintah kepada aduan yang diajukan. Kelas ini memiliki beberapa metode yaitu *insertBalasan()* untuk memasukkan balasan ke aduan. *view()* untuk menampilkan balasan kepada aduan yang diajukan.
- 4) Status: Memiliki beberapa properti seperti *id*, *pending*, *on_process* dan *complete* guna memberikan informasi jalannya aduan yang diajukan. Status memiliki beberapa metode yaitu *ubahStatus()* untuk pemerintah mengubah status aduan dan *view()* untuk masyarakat melihat status aduan.
- 5) Komentar: Memiliki beberapa properti yaitu *id* dan *komentar*. Kelas ini memiliki beberapa metode yaitu *insertKomentar()* untuk memasukkan komentar kepada *posting* dan *insertBalasan()* untuk memasukkan balasan

yang diberikan dari pemerintah. *view()* untuk melihat daftar komentar yang telah dimasukkan oleh para pengguna.

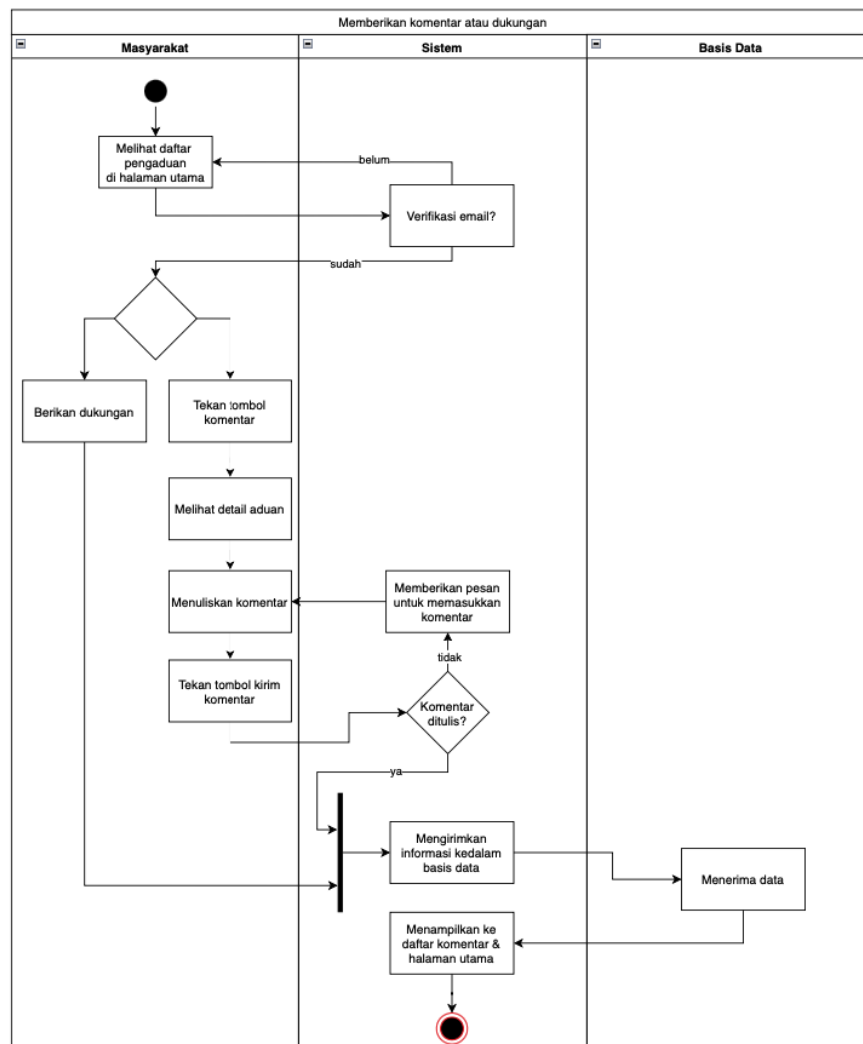
- 6) Dukungan: Memiliki beberapa properti yaitu id dan dukungan. Kelas ini memiliki 1 metode untuk menambahkan jumlah dukungan kedalam *posting*.

c. Activity Diagram



Gambar 3. 4.Activity diagram posting pengaduan

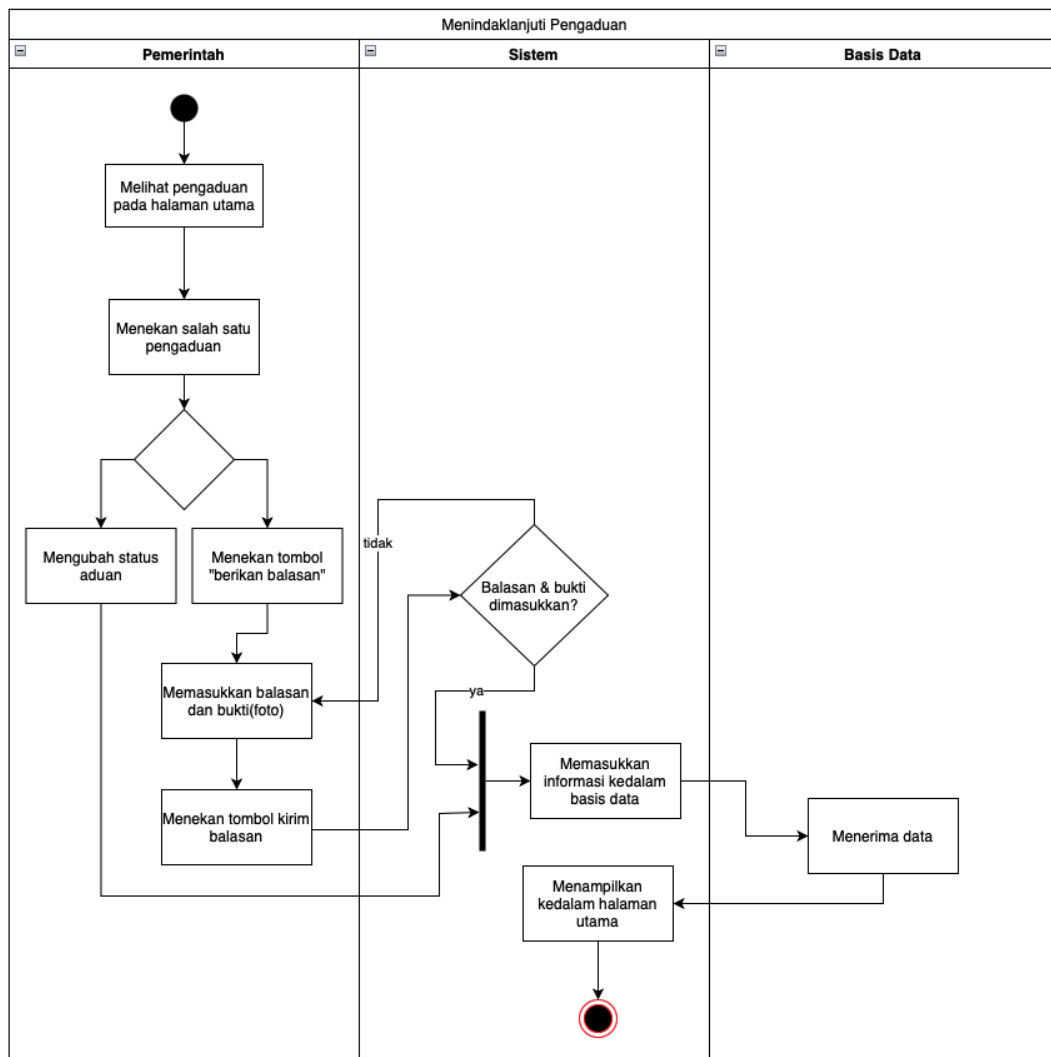
Pada Gambar 3.4 ditunjukkan *activity diagram* pada saat masyarakat melakukan sebuah *posting* pengaduan, pertama-tama pada halaman utama pengguna menekan tombol *posting* yang berada di tengah setelah itu sistem akan cek apakah pengguna sudah melakukan verifikasi *e-mail* atau belum, apabila belum maka sistem akan memberikan pesan untuk verifikasi *e-mail* apabila sudah akan langsung diarahkan kepada memasukkan bukti/foto, setelah memasukkan foto selanjutnya bisa menekan tombol *next* lalu sistem akan memeriksa apakah foto sudah dimasukkan, apabila belum maka pengguna diberikan pesan untuk memasukkan bukti/foto terlebih dahulu, setelah sudah dimasukkan selanjutnya memasukkan beberapa *field* seperti judul, kategori berupa *combobox*, catatan dan lokasi setelah itu bisa tekan tombol kirim, namun sistem akan memeriksa apakah *field* sudah diisi semua, apabila belum maka sistem akan mengirimkan pesan agar pengguna mengisikan *field* yang ada. Setelah itu pengguna akan tekan tombol *kirim* pada proses itulah sistem akan memasukkan informasi kedalam basis data, lalu memunculkannya di halaman utama.



Gambar 3. 5. *Activity diagram* proses pemberian komentar & dukungan

Pada Gambar 3.5 ditunjukkan *activity diagram* dari proses pemberian komentar & dukungan oleh pengguna, pertama pengguna dapat melihat daftar pengaduan pada halaman utama yang ingin diberikan dukungan lalu pengguna bisa langsung memberikan dukungan apabila sudah verifikasi *e-mail* berupa *upvote* atau menekan tombol komentar lalu memberikan komentar kedalam aduan, setelah itu pengguna dapat menekan tombol kirim komentar namun sistem akan memeriksa apakah komentar sudah ditulis dan apakah pengguna sudah melakukan verifikasi *e-mail* , apabila belum maka sistem akan mengirimkan pesan agar pengguna menuliskan

komentar atau verifikasi *e-mail*. Apabila sudah maka sistem akan memasukkan *input* dari pengguna kedalam basis data, lalu ditampilkan kedalam kolom komentar pada pengaduan yang diberikan komentar.



Gambar 3. 6.Activity diagram proses penindaklanjutan pengaduan

Proses dari penindaklanjutan gambaran dari sistem dapat ditunjukkan pada Gambar 3.6, pertama pemerintah ke halaman utama untuk melihat pengaduan yang diajukan oleh masyarakat lalu menekan salah satu pengaduan yang akan ditindaklanjuti, setelah itu dapat memberikan balasan berupa deskripsi dan bukti berupa foto namun di sini sistem akan memeriksa apakah dari penindaklanjutan yang

dilakukan apakah bukti dan deskripsi sudah dimasukkan. Apabila sudah, maka data akan dimasukkan kedalam basis data lalu ditampilkan ke halaman utama sebagai balasan dari pemerintah. Masyarakat juga akan mendapatkan notifikasi dari tindaklanjut tersebut.

4. Implementasi

Pembangunan sistem pengaduan masyarakat ini dibuat dengan menggunakan bahasa pemrograman *Java* dengan *platform Android Studio*. Layanan basis data yang digunakan adalah *Firebase*.

5. Pengujian

Pada tahapan ini, pengujian aplikasi dilakukan dengan metode *Usability Testing*. *Usability Testing* adalah sebuah fase pengujian perangkat lunak yang dilakukan oleh pengembang dengan melibatkan pengguna secara langsung menggunakan aplikasi dan memeriksa bagaimana pengguna berinteraksi dengan aplikasi yang sedang dibangun (Hema Mitta Kalyani, 2019). Tujuan dari *Usability Testing* yaitu mencari masalah kegunaan, pengumpulan data kualitatif dan kuantitatif, dan sebagai penentuan kepuasan pengguna dengan produk (Dwinawan Hariwijaya, 2017). Pada penelitian ini, dilakukannya *usability testing* berguna untuk menganalisa dan mengetahui tingkat kepuasan pengguna terhadap aplikasi yang dibuat melalui kuesioner yang diberikan setelah melakukan beberapa tugas (*task*) dalam menjalankan alur aplikasi.

Pada pengujian aplikasi, para pengguna memiliki tugas yang dikerjakan pada saat menggunakan aplikasi sebelum mengisi kuesioner. Tugas tersebut antara lain:

- a. Pengguna melakukan pembukaan aplikasi untuk melihat tampilan awal.

- b. Pengguna melakukan registrasi untuk menggunakan fitur pada aplikasi.
- c. Setelah registrasi, pengguna melakukan *login* kedalam aplikasi.
- d. Setelah *login*, pengguna masuk kedalam halaman utama dari aplikasi dan melihat tampilan.
- e. Pada halaman utama, pengguna dapat melihat tampilan dari *posting* pengaduan yang dilaporkan oleh masyarakat, memberikan dukungan maupun komentar serta melihat tampilan pada halaman tersebut.
- f. Pada navigasi kedua, pengguna dapat melihat masalah pengaduan yang sudah terselesaikan oleh pemerintah serta melihat tampilan pada halaman tersebut.
- g. Pada navigasi ketiga ditengah, pengguna dapat membuat sebuah *posting* pengaduan untuk aplikasi agar muncul ke halaman utama serta melihat tampilan pada halaman tersebut.
- h. Pada navigasi keempat bagian notifikasi, pengguna dapat melihat notifikasi terkait pengaduan yang diajukan serta melihat tampilan pada halaman tersebut.
- i. Pada navigasi terakhir pengguna dapat *logout* atau keluar akun dari aplikasi serta melihat tampilan pada halaman tersebut.

Setelah beberapa tugas tersebut dilakukan oleh pengguna, maka selanjutnya dapat dilakukan proses *usability testing* dengan menggunakan pertanyaan kuesioner yang diberikan melalui *Google Form*. Pertanyaan kuesioner untuk melakukan pengujian *usability testing* antara lain terdapat pada Tabel 3.1.

Tabel 3. 1.Tabel pertanyaan kuesioner *usability testing*

No	Pertanyaan	STS	TS	RG	S	SS
1.	Saya berpikir akan menggunakan sistem ini lagi					
2.	Saya merasa sistem ini rumit untuk digunakan					
3.	Saya merasa sistem ini mudah digunakan					
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini					
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya					
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)					
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat					
8.	Saya merasa sistem ini membingungkan					
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini					
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini					

Pada pengujian, dilakukan terhadap jumlah responden sebesar 20 pengguna, karena menurut (Jakob Nielsen, 2012) pengujian *usability testing* membutuhkan 20 pengguna untuk mendapatkan angka yang signifikan pada perhitungan. Perhitungan yang dilakukan untuk menguji *usability testing* pada aplikasi ini menggunakan

persamaan yang dapat dilihat pada Persamaan 3.1, persamaan ini digunakan untuk menghitung *SUS* (*System Usability Scale*).

$$\bar{x} = \frac{\sum x}{n} \quad (3.1),$$

dengan,

\bar{x} = Skor rata-rata,

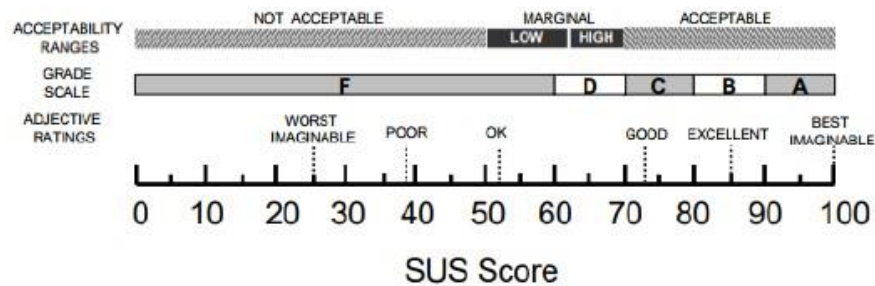
$\sum x$ = Jumlah skor *SUS*,

n = Jumlah responden

Melalui persamaan tersebut, akan dilakukan perhitungan terhadap 10 pertanyaan kuesioner yang memiliki jawaban menggunakan skala Likert. Skala Likert terdiri atas angka 1 sampai 5 yaitu 1 = Sangat Tidak Setuju (STS), 2 = Tidak Setuju (TS), 3 = Ragu-Ragu (RG), 4 = Setuju (S), 5 = Sangat Setuju (SS). Pengujian dilakukan terhadap masyarakat yang berdomisili di Kabupaten Tangerang.

Setelah dilakukan perhitungan dengan menggunakan Persamaan 3.1, akan menghasilkan sebuah skor *SUS* yang memiliki simpulan akhir sebuah angka yang

menunjukkan apabila sistem yang dibuat sudah layak atau belum, pada Gambar 3.7 ditunjukkan skor skala kelayakan sebuah sistem.



Gambar 3. 7.Skor skala kelayakan sebuah sistem

6. Pelaporan

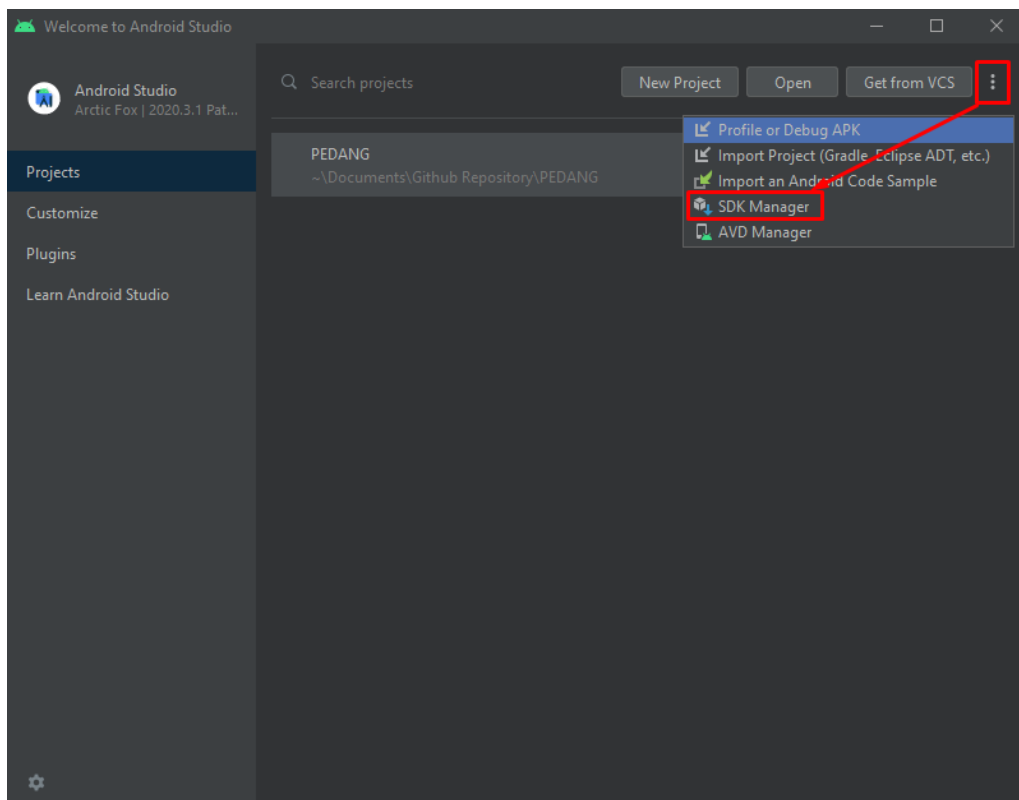
Tahap pelaporan dilakukan dengan penyusunan Buku Tugas Akhir sesuai dengan Panduan Penulisan Buku Tugas Akhir. Pada bagian ini, terdapat seluruh penjelasan secara teknis aplikasi yang dibuat dan juga hasil analisa *usability testing* yang dilakukan.

BAB 4

HASIL DAN PEMBAHASAN

4.1. Hasil & Implementasi

Dalam pembuatan sistem pengaduan ini, dilakukan pembuatan aplikasi menggunakan *platform Android Studio*. Platform diunduh melalui *website* yang beralamat di <https://www.developer.android.com/studio/> lalu melakukan pemasangan platform tersebut. Setelah melakukan pemasangan, selanjutnya pemasangan beberapa hal yang dibutuhkan untuk pembuatan aplikasi pada bagian awal harus melakukan pengaturan *SDK* melalui *SDK Manager* pada platform.



Gambar 4. 1. Menu SDK Manager

Berikut beberapa *SDK* yang dibutuhkan untuk pembuatan sistem pengaduan masyarakat ini:

1. *SDK Platforms*

Dibutuhkan 1 *SDK Platforms* yaitu *Android 10.0 (Q)* yang mencakup beberapa *package* diantaranya adalah:

- a. *Android SDK Platform 29*
- b. *Sources for Android 29*
- c. *Intel x86 Atom_64 System Image*
- d. *Google APIs Intel x86 Atom_64 System Image*
- e. *Google Play Intel x86 Atom System Image*
- f. *Google Play Intel x86 Atom_64 System Image*

2. *SDK Tools*

Dibutuhkan beberapa *SDK Tools* dalam pembuatan aplikasi ini, diantaranya adalah:

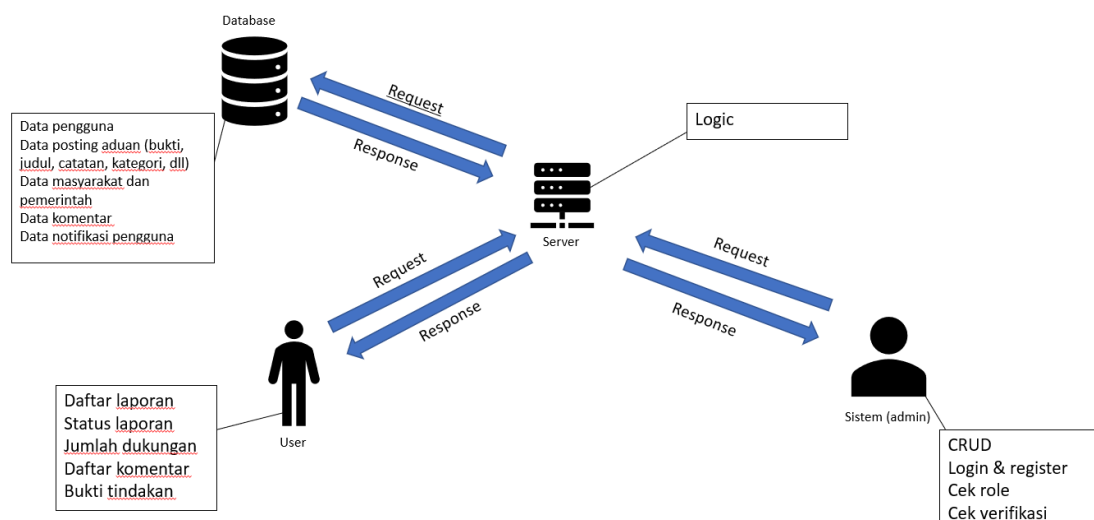
- a. *Android SDK Build-Tools 32* versi 29.0.0.
- b. *Android Emulator*
- c. *Android SDK Platform-Tools*
- d. *Intel x86 Emulator Accelerator (HAXM Installer)*

Setelah melakukan pemasangan beberapa *SDK* yang dibutuhkan tersebut, maka pembuatan sistem sudah dapat dilakukan.

Sistem pengaduan masyarakat Kabupaten Tangerang ini dibuat didasarkan dari beberapa hal, seperti belum adanya sistem pelaporan digital yang dimiliki wilayah Kabupaten Tangerang, banyaknya kerusakan jalan/lingkungan yang terjadi terlalu

lama tanpa adanya tindakan yang dilakukan oleh pemerintah dan sebagainya. Oleh karena itu, dibutuhkan data yang akan dimasukkan oleh pengguna untuk melaporkan sebuah aduan, contohnya seperti bukti berupa foto dari masalah, judul dari aduan, kategori masalah, dan lokasi.

Dalam sistem ini terdapat dua level *user* yang akan menggunakan sistem ini, setiap level memiliki *menu* tampilan dan fungsi yang berbeda-beda. Level yang pertama adalah level “Masyarakat” apabila masyarakat ingin mengajukan pengaduan baru maka masyarakat harus melakukan *login* dan memasukkan *username* sebagai tanda pengenalan pada aplikasi, setelah itu akan diarahkan ke halaman utama. Level kedua yaitu “Pemerintah” sebuah akun yang ingin dijadikan akun pemerintah harus melakukan *request* kepada *administrator* untuk dijadikan pemerintah. Apabila level akun merupakan pemerintah, maka langsung diarahkan ke halaman utama khusus pemerintah. Berikut hasil implementasi Sistem Pengaduan Masyarakat Kabupaten Tangerang.

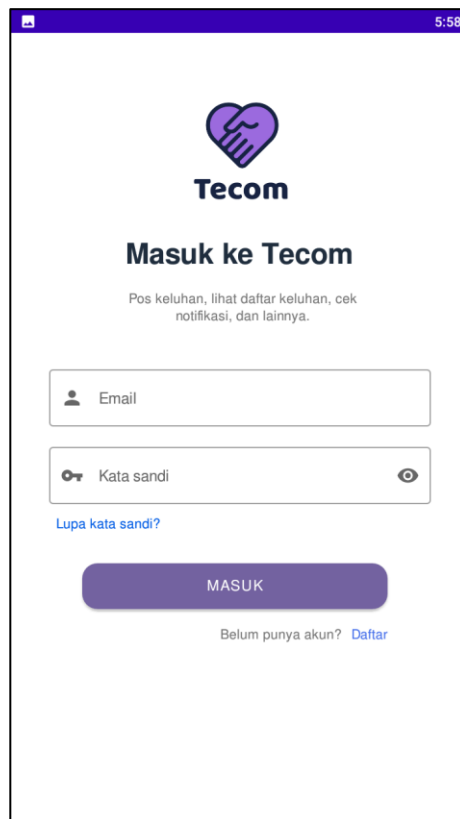


Gambar 4. 2. Gambaran sistem secara keseluruhan

Gambaran keseluruhan pada sistem ditunjukkan pada Gambar 4.2, di mana pengguna dapat melihat data yang diberikan oleh basis data melalui *server* berupa daftar laporan, status laporan, jumlah dukungan, daftar komentar, dan bukti tindakan. Basis data menyimpan beberapa informasi yang diberikan kepada pengguna dan juga kepada *administrator* melalui *server* yaitu data pengguna, data *posting* aduan, data komentar, data notifikasi dari pengguna. Selain itu yang *handle* proses *CRUD*, *login* & *register*, cek role(level), cek verifikasi merupakan sistem.

4.1.1. Halaman *login*

Halaman ini adalah halaman yang muncul pertama kali pada saat pengguna membuka aplikasi. Apabila pengguna sudah memiliki akun maka dapat langsung melakukan *login* pada aplikasi, apabila belum memiliki akun maka pengguna harus menekan tombol “Daftar” untuk membuat akun baru. Halaman *login* ditunjukkan pada Gambar 4.3.



Gambar 4. 3.Halaman *login*

Fungsi dari halaman ini yaitu sebelum pengguna mengajukan pengaduan kepada pemerintah, pengguna harus melakukan *login* terlebih dahulu dengan mengisi *e-mail* dan *password* pengguna. Setelah pengguna sudah dimasukkan dan terdapat data pengguna pada basis data, maka pengguna akan langsung diarahkan ke halaman pertama.

Kode Program 4. 1.Fungsi *login*

```

1. mAuth.signInWithEmailAndPassword(email,
   password).addOnCompleteListener(task -> {
2.     if(task.isSuccessful()){
3.         String currentUserID = mAuth.getCurrentUser().getUid();
4.
5.         DatabaseReference RoleRef =
           FirebaseDatabase.getInstance().getReference().child("Users").child(currentU
           serID);
6.
7.         RoleRef.addValueEventListener(new ValueEventListener()
8.         {
           @Override

```

```

9.         public void onDataChange(@NonNull DataSnapshot
    snapshot) {
10.             String role = (String)
    snapshot.child("role").getValue();
11.
12.             if(role.equals("Pemerintah")){
13.                 SendUserToDashboardPemerintah();
14.             }
15.             else{
16.                 SendUsertoMainActivity();
17.             }
18.         }
19.
20.         @Override
21.         public void onCancelled(@NonNull DatabaseError
    error) {
22.
23.         }
24.     });
25. }
26. else{
27.     String message = task.getException().getMessage();
28.     Toast.makeText(LoginActivity.this, "Terdapat error : "
    + message, Toast.LENGTH_SHORT).show();
29. }
30.     loadingBar.dismiss();
31. });

```

Fungsi *login* yang ditunjukkan pada Kode Program 4.1, di sini apabila pengguna sudah memasukkan *e-mail* dan *password* lalu menekan tombol “Masuk” maka fungsi *mAuth.signInWithEmailAndPassword* akan dijalankan, lalu program akan memeriksa data pada basis data *Firebase Authentication* apabila terdapat data pengguna di dalamnya, maka selanjutnya akan memeriksa *role* atau level dari pengguna, pada baris 12 sampai 17 program melakukan pengecekan apakah level dari pengguna “Pemerintah” atau “Masyarakat”, jika “Pemerintah” maka pengguna akan diarahkan ke halaman utama pemerintah, jika “Masyarakat” maka langsung diarahkan ke halaman utama dari masyarakat. Untuk pemberitahuan apabila ada *error* pada saat menjalankan fungsi, akan dijalankan fungsi *getException().getMessage()* yang ditunjukkan pada baris 27, untuk mendapatkan pesan *error* yang akan ditampilkan kepada pengguna.

4.1.2. Halaman *register*

Halaman ini diperuntukkan bagi semua masyarakat atau pemerintah yang belum memiliki akun. Pada halaman ini pengguna dapat melakukan pendaftaran akun dengan mengisi *e-mail* dan *password* serta konfirmasi *password*.



7:32


Tecom

Daftar ke Tecom

 Email

 Kata sandi 

 Konfirmasi kata sandi 

DAFTAR

Sudah punya akun? [Masuk](#)

Gambar 4. 4.Halaman *register*

Halaman *register* ditunjukkan pada Gambar 4.4 di mana pada saat pengguna membuka aplikasi untuk pertama kalinya akan dimunculkan halaman *login*, namun apabila pengguna belum memiliki akun maka harus menekan tombol “daftar”. Setelah itu maka halaman inilah yang dimunculkan, selanjutnya pengguna dapat langsung memasukkan *e-mail*, *password*, dan konfirmasi *password* yang akan digunakan untuk

login ke halaman utama. *E-mail* yang dibutuhkan merupakan *e-mail* asli yang aktif digunakan oleh pengguna, karena proses berikutnya apabila pengguna ingin mengirim aduan, memberikan dukungan atau komentar pengguna harus verifikasi *e-mail* terlebih dahulu.

Kode Program 4. 2.Fungsi *register*

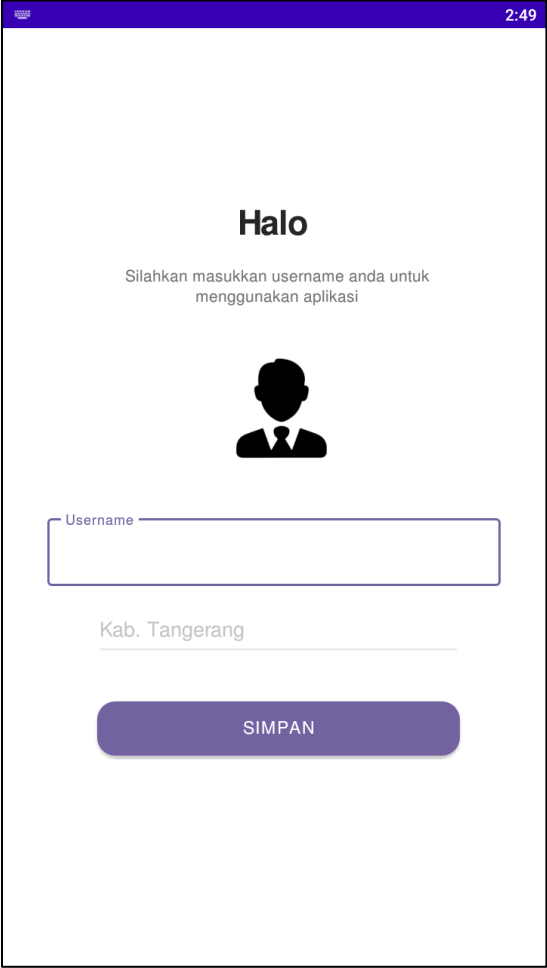
```
1. mAuth.createUserWithEmailAndPassword(email, password)
2.     .addOnCompleteListener(task -> {
3.         if(task.isSuccessful())
4.         {
5.             Toast.makeText(RegisterActivity.this, "Akun
anda telah terdaftar!", Toast.LENGTH_SHORT).show();
6.             loadingBar.dismiss();
7.
8.             SendUsertoEnterUsername();
9.         }
10.        else{
11.            String message =
task.getException().getMessage();
12.            Toast.makeText(RegisterActivity.this, "Terdapat
error : " + message, Toast.LENGTH_SHORT).show();
13.            loadingBar.dismiss();
14.        }
15.    });
```

Fungsi *register* yang ditunjukkan pada Kode Program 4.2, di sini jika pengguna sudah memasukkan informasi *e-mail*, *password* dan konfirmasi *password* maka setelah itu pengguna dapat langsung menekan tombol “Daftar” maka fungsi dari *mAuth.createUserWithEmailAndPassword* pada baris 1 akan dijalankan, lalu program akan memasukkan data ke basis data *Firebase Authentication*.

4.1.3. Halaman memasukkan *username*

Sebelum pengguna dapat masuk ke dalam halaman utama masyarakat, pengguna harus memasukkan *username* sebagai tanda pengenalan di dalam sistem, maka dari itu pada Gambar 4.5 merupakan halaman pada saat pengguna memasukkan

username. *Username* akan muncul pada akun di bagian profil, halaman utama, notifikasi kepada masyarakat lain, dan halaman pemerintah.



Gambar 4. 5. Halaman memasukkan *username*

Pada fungsi Kode Program 4.3 di deklarasikan variabel *HashMap* dengan nama variabel *userMap*, deklarasi ini bertujuan yang digunakan untuk memasukkan data (*value*) ke beberapa *key* pada basis data *Firebase*. Data yang dimasukkan juga akan masuk ke dalam *section Firebase Realtime Database* untuk menyimpan data kustom. Pada Kode Program 4.3 baris 2-4 data yang dimasukkan sebagai tanda pengenal yaitu:

1. key "*username*" dengan data variabel "*username*"
2. key "*domisili*" dengan data variabel "*domisili*"

3. key "role" dengan data variabel "Masyarakat"

Kode Program 4. 3.Fungsi memasukkan *username*

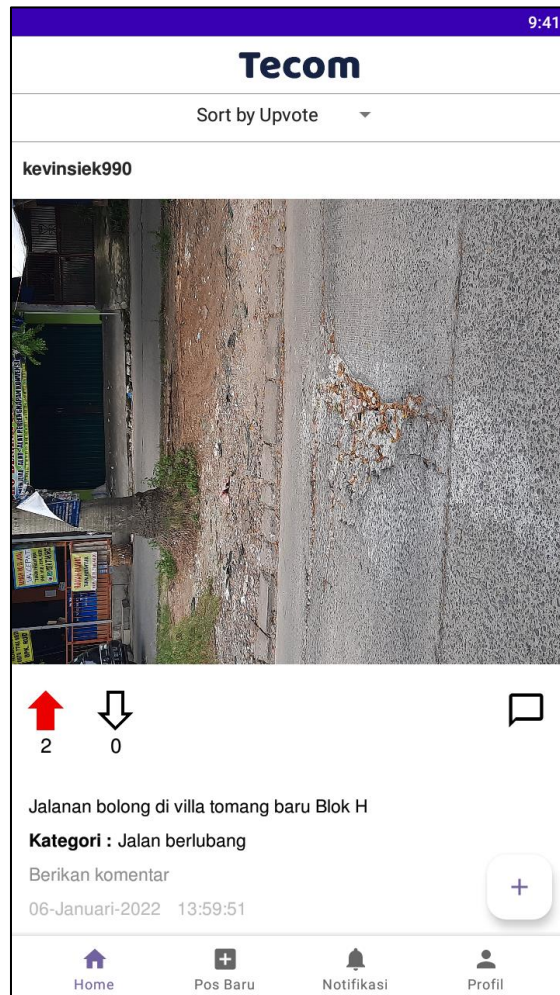
```
1. HashMap<String, Object> userMap = new HashMap<>();
2.     userMap.put("username", username);
3.     userMap.put("domisili", domisili);
4.     userMap.put("role", "Masyarakat");
5.     UsersRef.updateChildren(userMap).addOnCompleteListener(task ->
6.     {
7.         if(task.isSuccessful()){
8.             SendUserToMainActivity();
9.             Toast.makeText(SetupActivity.this, "Sukses membuat
10.     akun.", Toast.LENGTH_LONG).show();
11.         }
12.         else{
13.             String message = task.getException().getMessage();
14.             Toast.makeText(SetupActivity.this, "Terdapat error : "
15.     + message , Toast.LENGTH_SHORT).show();
16.             loadingBar.dismiss();
17.         }
18.     });
```

Pada saat tombol "Simpan" ditekan maka fungsi *updateChildren()* pada baris 5 akan dijalankan, yang berisikan data variabel *userMap*. Setelah data sudah berhasil dimasukkan ke dalam basis data, maka pengguna akan langsung diarahkan ke halaman utama masyarakat. Apabila data gagal untuk dimasukkan maka sistem akan memberikan pesan.

4.1.4. Halaman utama masyarakat

Pada halaman utama masyarakat, terdapat beberapa *menu* yang dapat diakses oleh masyarakat diantaranya adalah *menu home*, pos baru, notifikasi, dan profil. *Menu home* akan berisi semua *posting* pengaduan yang diajukan oleh masyarakat, *menu pos baru* akan menampilkan permintaan bukti aduan berupa foto melalui galeri ataupun melalui kamera secara langsung, *menu notifikasi* berisi semua notifikasi apabila ada yang berkomentar di *posting* yang diajukan oleh pengguna, *menu profil* akan berisi

username dari pengguna, status verifikasi *e-mail* dan *uid* dari pengguna serta tombol *logout*.



Gambar 4. 6.Halaman utama masyarakat

Pada saat *menu home* ditekan maka akan ditampilkan seperti pada Gambar 4.6, terdapat beberapa fungsi yang dapat dilakukan oleh pengguna pada halaman utama masyarakat, seperti memberikan dukungan (*upvote*), memberikan *downvote* dengan menekan tombol pada Gambar 4.7, membuat sebuah *posting* baru apabila pengguna sudah melakukan verifikasi *e-mail* , *sorting* atau mengurutkan *posting* berdasarkan variabel tertentu dan navigasi ke komentar. Berikut beberapa hal yang dapat dilakukan tersebut:

1. Fungsi pemberian dukungan (*upvote*) dan tidak mendukung (*downvote*)

Fungsi pada Kode Program 4.4 merupakan fungsi pemberian dukungan dari masyarakat terhadap aduan yang diajukan oleh masyarakat lain atau aduan sendiri. Pada baris 2 terdapat pengecekan apakah pengguna sudah melakukan verifikasi *e-mail* atau belum, apabila belum maka pengguna akan diberikan pesan untuk verifikasi *e-mail* terlebih dahulu baru bisa memberikan dukungan.



Gambar 4. 7.Tombol *upvote* dan *downvote*

Kode Program 4. 4.Fungsi *Upvote*

```
1. holder.UpvoteButton.setOnClickListener(view -> {
2.     if(!user.isEmailVerified()){
3.         Toast.makeText(MainActivity.this, "Email belum
4.         terverifikasi.", Toast.LENGTH_SHORT).show();
5.     }
6.     else{
7.         UpvoteChecker = true;
8.         UpvoteRef.addValueEventListener(new
9.         ValueEventListener() {
10.             @Override
11.             public void onDataChange(@NonNull DataSnapshot
12.             snapshot) {
13.                 int countUpvotes = (int)
14.                 snapshot.child(PostKey).getChildrenCount();
15.                 if (UpvoteChecker.equals(true)) {
16.                     if(snapshot.child(PostKey).hasChild(currentUserID)){
17.                         UpvoteRef.child(PostKey).child(currentUserID).removeValue();
18.                         PostsRef.child(PostKey).child("upvote").setValue(countUpvotes - 1);
19.                     }
20.                     else{
21.                         UpvoteRef.child(PostKey).child(currentUserID).setValue(true);
22.                         PostsRef.child(PostKey).child("upvote").setValue(countUpvotes + 1);
23.                     }
24.                     UpvoteChecker = false;
25.                 }
26.             }
27.         });
28.     }
29. }
```


Variabel *UpvoteChecker* dengan tipe variabel *boolean* akan memeriksa apabila tombol beri dukungan ditekan maka akan berubah menjadi *true*, setelah itu pada baris 13 program akan memeriksa apakah *UpvoteChecker* bernilai *true* jika ya maka data pada basis data *Firebase* akan bertambah dalam hal ini pada *key* "*upvote*", apabila bernilai *false* maka data pada basis data akan berkurang satu. Untuk fungsi tidak memberikan dukungan atau *Downvote* sama dengan fungsi *upvote*, karena yang berbeda hanya pada pemasukan data pada basis data *Firebase* dimasukkan kedalam *key* "*downvote*" kode program dari fungsi tidak memberi dukungan (*downvote*) dapat dilihat pada Kode Program 4.5.

Kode Program 4. 5.Fungsi *Downvote*

```

1.  if(!user.isEmailVerified()){
2.      Toast.makeText(MainActivity.this, "Email belum
   terverifikasi.", Toast.LENGTH_SHORT).show();
3.  }
4.      else {
5.          DownvoteChecker = true;
6.
7.          DownvoteRef.addValueEventListener(new
   ValueEventListener() {
8.              @Override
9.              public void onDataChange(@NonNull DataSnapshot
   snapshot) {
10.                  int countDownvotes = (int)
   snapshot.child(PostKey).getChildrenCount();
11.                  if (DownvoteChecker.equals(true)) {
12.                      if
   (snapshot.child(PostKey).hasChild(currentUserID)) {
13.                          DownvoteRef.child(PostKey).child(currentUserID).removeValue();
14.                          PostsRef.child(PostKey).child("downvote").setValue(countDownvotes - 1);
15.                      } else {
16.                          DownvoteRef.child(PostKey).child(currentUserID).setValue(true);
17.                          PostsRef.child(PostKey).child("downvote").setValue(countDownvotes + 1);
18.                      }
19.                      DownvoteChecker = false;
20.                  }
21.              }
22.              @Override
23.              public void onCancelled(@NonNull DatabaseError error) {
24.
25.              }
26.          }
27.      });

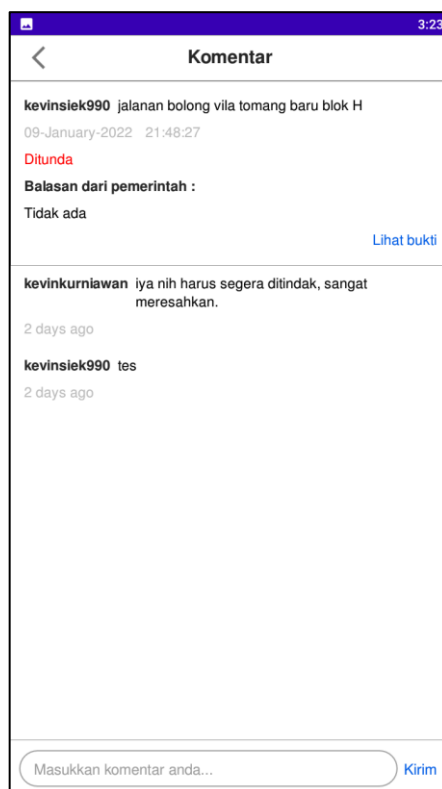
```

2. Fungsi memberikan komentar

Fungsi memberikan komentar dapat dilakukan apabila pengguna sudah melakukan verifikasi *e-mail* dan hanya dapat dilakukan oleh masyarakat, pemerintah tidak bisa melakukan fungsi ini. Apabila pengguna ingin memberikan komentar pada suatu aduan pengguna dapat menekan tombol pada halaman utama seperti yang ditampilkan pada Gambar 4.8.



Gambar 4. 8.Tombol komentar



Gambar 4. 9.Halaman pemberian komentar

Setelah menekan tombol komentar maka selanjutnya pengguna akan diarahkan ke halaman pemberian komentar yang ditampilkan seperti pada Gambar 4.9.

Terdapat beberapa detail dari aduan seperti nama pengirim, judul aduan, tanggal dibuatnya aduan, status, balasan dari pemerintah dan tombol bukti pemerintah. Untuk memberikan komentar pengguna dapat menuliskan komentar pada kotak “masukkan komentar” lalu menekan tombol kirim.

Kode Program 4. 6.Fungsi memberikan komentar

```
1. commentText = commentsInputText.getText().toString();
2.
3.     if(TextUtils.isEmpty(commentText)){
4.
5.     }
6.     else{
7.         //get Tanggal
8.         Calendar calForDate = Calendar.getInstance();
9.         @SuppressWarnings("SimpleDateFormat") SimpleDateFormat currentDate
10.    = new SimpleDateFormat("dd-M-yyyy HH:mm:ss");
11.    final String saveCurrentDate =
12.    currentDate.format(calForDate.getTime());
13.
14.    final String RandomKey = currentUserID + saveCurrentDate;
15.
16.    HashMap<String, Object> commentsMap = new HashMap<>();
17.    commentsMap.put("uid", currentUserID);
18.    commentsMap.put("comment", commentText);
19.    commentsMap.put("date", saveCurrentDate);
20.    commentsMap.put("username", username);
21.    commentsMap.put("sort_descending", countComments);
22.
23.    PostsRefComments.child(RandomKey).updateChildren(commentsMap);
24.
25.    if(currentUserID.equals(UidCommented)){
26.
27.    }else{
28.        AddNotification(username);
29.    }
```

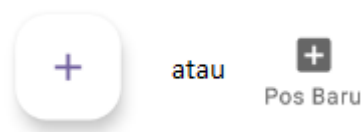
Setelah menekan tombol kirim maka Kode Program 4.6 akan berjalan, di sini terdapat data yang dimasukkan kedalam basis data:

- a. key “uid” dengan data variabel “currentUserID”
- b. key “comment” dengan data variabel “commentText”
- c. key “date” dengan data variabel “saveCurrentDate”
- d. key “username” dengan data variabel “username”
- e. key “sort_descending” dengan data variabel “countComments”

Data *currentUserID* didapatkan melalui fungsi *FirebaseAuth.getInstance().getCurrentUser().getUid()*. Data dari *commentText* didapatkan melalui *value* dari kotak “masukkan komentar”. Data *date* atau tanggal didapatkan melalui fungsi *currentDate.format(calForDate.getTime())* yang ditunjukkan pada baris 10. Data *username* didapatkan melalui variabel *username* yang dilemparkan dari pemanggilan fungsi pemberian komentar. Sedangkan data *countComments* didapatkan dari perhitungan jumlah komentar yang ada pada basis data. Setelah data sudah masuk kedalam basis data, lalu sistem akan menampilkan pada halaman komentar sesuai dengan yang ditunjukkan pada Gambar 4.9.

4.1.5. Fungsi *posting* pengaduan

Fungsi *posting* pengaduan dapat dilakukan oleh pengguna yang sudah melakukan verifikasi *e-mail*. *Posting* pengaduan dapat dilakukan dengan menekan tombol yang ditunjukkan pada Gambar 4.10.



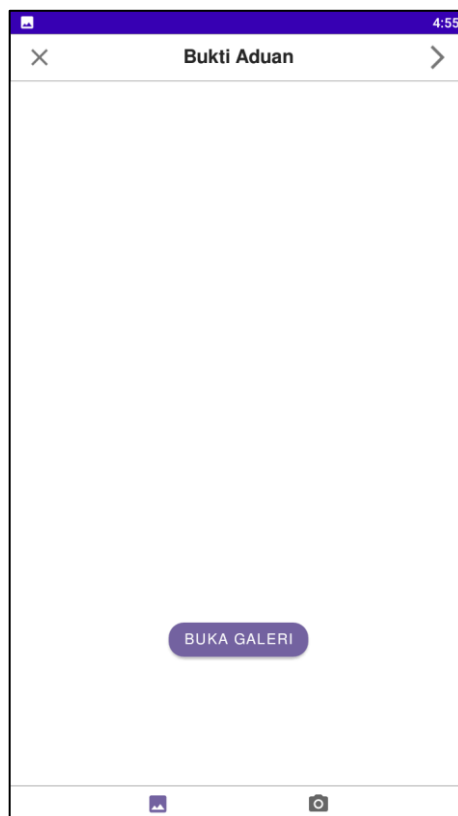
Gambar 4. 10. Tombol *posting* pengaduan

Tahapan dari dilakukannya *posting* pengaduan adalah pengguna harus memberikan bukti berupa foto, dapat dipilih melalui galeri atau mengambil foto secara langsung menggunakan kamera. Selanjutnya adalah pengguna harus memberikan data yang akan dikirimkan seperti judul dari aduan, catatan yang akan diberikan ke pemerintah, kategori dan lokasi dari aduan.

Berikut penjabaran langkah-langkah dari pembuatan *posting* aduan:

1. Pemberian bukti berupa foto

Sebelum memberikan judul, kategori dan lainnya, pengguna harus memberikan bukti berupa foto terhadap aduan yang diajukan. Pemberian bukti dapat dipilih melalui galeri atau pengguna memfoto secara langsung masalah yang terjadi. Apabila pengguna ingin menggunakan galeri maka dapat menekan tombol “Buka Galeri” dan apabila pengguna ingin menggunakan kamera maka dapat menekan tombol kamera di bawah kanan yang bergambarkan sebuah kamera yang ditunjukkan pada Gambar 4.11.



Gambar 4. 11.Halaman pemberian bukti aduan

Setelah menekan tombol buka galeri maka akan terbuka galeri dari pengguna untuk memasukkan foto, fungsi yang dijalankan pada pembukaan galeri ini ditunjukkan pada Kode Program 4.7.

Kode Program 4. 7.Pembukaan galeri

```
1. Intent galleryIntent = new Intent();
2.     galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
3.     galleryIntent.setType("image/*");
4.     startActivityForResult(galleryIntent, Gallery_Pick);
```

Apabila pengguna ingin memasukkan foto melalui kamera, maka fungsi yang akan dijalankan ditunjukkan pada Kode Program 4.8.

Kode Program 4. 8.Pembukaan kamera

```
1. if (checkPermissions(Permissions.CAMERA_AND_WRITE_PERMISSIONS[0])) {
2.     String filename = "photo";
3.     File storageDirectory =
4.         getExternalFilesDir(Environment.DIRECTORY_PICTURES);
5.
6.     try {
7.         File imageFile = File.createTempFile(filename,
8.         ".jpg", storageDirectory);
9.
10.        ImageUri =
11.        FileProvider.getUriForFile(PostActivity.this,
12.        "com.example.pedang.fileprovider", imageFile);
13.
14.        Intent cameraIntent = new
15.        Intent(MediaStore.ACTION_IMAGE_CAPTURE);
16.        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,
17.        ImageUri);
18.        startActivityForResult(cameraIntent,
19.        CAMERA_REQUEST_CODE);
20.    } catch (IOException e) {
21.        e.printStackTrace();
22.    }
23. } else {
24.     Intent intent = new Intent(this, PostActivity.class);
25.     intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
26.     Intent.FLAG_ACTIVITY_CLEAR_TASK);
27.     startActivity(intent);
28. }
```

Pada Kode Program 4.8 terjadi fungsi pembukaan kamera dan pengambilan data dari gambar yang ditangkap. Hal ini ditunjukkan pada baris 12 di mana pada saat menjalankan fungsi kamera akan dibuka, setelah pengguna sudah menangkap

gambar yang masalah yang akan diadukan selanjutnya data dari gambar berupa *URI* akan dikirimkan ke fungsi selanjutnya untuk menampilkan kedalam *imageView*.

Pengambilan data *URI* ditunjukkan pada Kode Program 4.8 baris 11 berupa variabel *ImageUri* yang didapatkan dari fungsi *FileProvider.getUriForFile(PostActivity.this, “com.example.pedang.fileprovider”, imageFile)*; pada saat fungsi ini dijalankan program mengambil data *URI* dari pembuatan *file* sementara yang ditunjukkan pada baris 6.

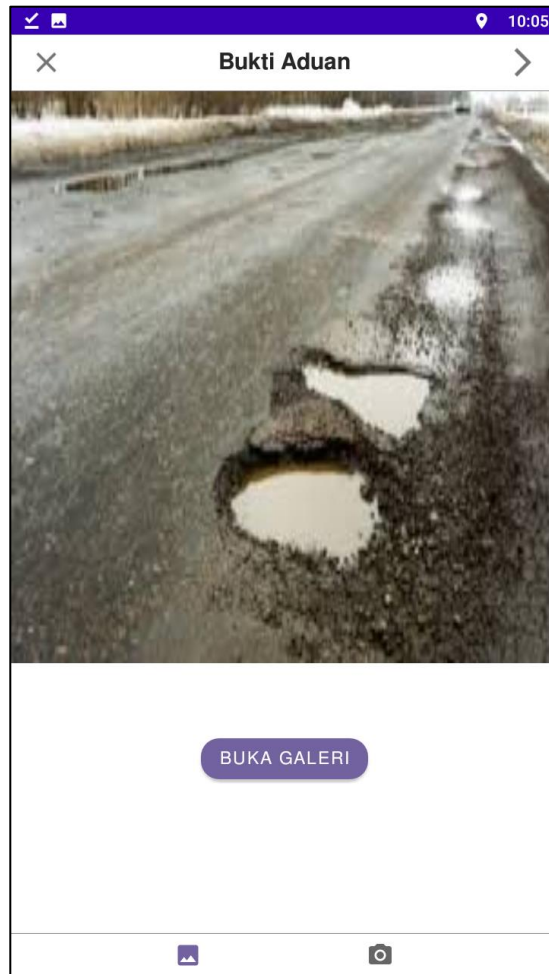
Setelah pengguna melakukan salah satu melalui galeri atau melalui kamera maka baris 4 pada Kode Program 4.7 atau baris 12 pada Kode Program 4.8 akan dijalankan, yang akan memanggil fungsi selanjutnya untuk menampilkan bukti aduan ke *imageView*. Fungsi yang akan dijalankan selanjutnya ditunjukkan pada Kode Program 4.9.

Kode Program 4. 9.Menampilkan bukti aduan ke imageView

```
1.      @Override
2.      protected void onActivityResult(int requestCode, int resultCode,
3.      @Nullable Intent data) {
4.          super.onActivityResult(requestCode, resultCode, data);
5.          switch(requestCode) {
6.              case Gallery_Pick:
7.                  ImageUri = data.getData();
8.                  imageView.setImageURI(ImageUri);
9.                  break;
10.             case CAMERA_REQUEST_CODE:
11.
12.                 imageView.setImageURI(ImageUri);
13.                 imageView.setScaleType(ImageView.ScaleType.FIT_XY);
14.                 break;
15.             }
16.     }
```

Pada Kode Program 4.9 baris 5 ditunjukkan pengambilan data *requestCode* yang dikirimkan pada saat menjalankan fungsi *startActivityForResult(variabel1,*

variabel2). Setelah itu data berupa *URI* akan dipakai untuk menampilkan ke *imageView* dengan hasil pada Gambar 4.12. Setelah gambar ditampilkan, pengguna dapat menekan tombol selanjutnya yang terdapat pada pojok kanan atas dari halaman *posting* pengaduan.

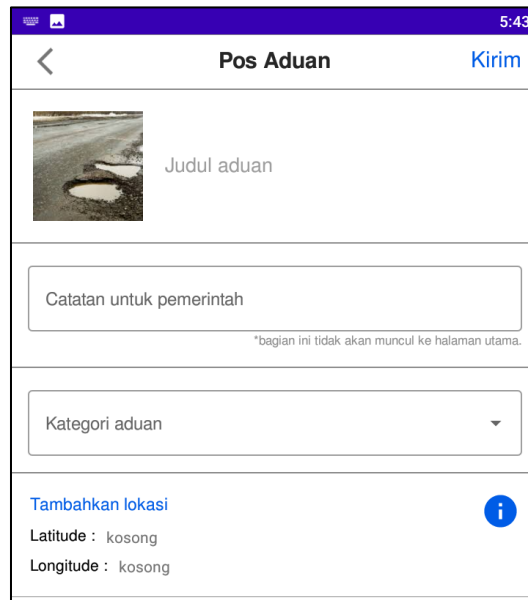


Gambar 4. 12.Menampilkan bukti aduan yang dipilih oleh pengguna melalui galeri/kamera

2. Pemberian judul aduan, catatan, kategori, dan lokasi

Pada tahap ini pengguna harus memasukkan data yang akan diberikan kepada pemerintah berupa judul dari aduan, catatan kepada pemerintah, kategori aduan

dan lokasi di mana masalah itu terjadi. *Field* untuk memasukkan data tersebut ditunjukkan pada Gambar 4.13.



Gambar 4. 13.*Field* untuk memasukkan judul, catatan, kategori dan lokasi aduan

Pada halaman pengisian data yang ditunjukkan di Gambar 4.13, pengguna dapat mengisi beberapa informasi yaitu judul aduan, catatan, kategori dan menekan tombol tambah lokasi. Semua informasi ini harus diisi oleh pengguna sebagai data masalah yang diajukan.

Setelah mengisi beberapa informasi, salah satu hal yang harus dilakukan oleh pengguna adalah menekan tombol Tambah lokasi. Pada saat menekan tombol lokasi, lokasi yang digunakan adalah lokasi di mana letak pengguna berada secara real time dan tidak bisa diubah. Pada saat tombol tambah lokasi ditekan maka fungsi yang berjalan ditunjukkan pada Kode Program 4.10.

Kode Program 4. 10.Mengambil informasi *latitude* dan *longitude* lokasi pengguna

```
1. @SuppressWarnings("SetTextI18n")
2. @Override
3. public void onRequestPermissionsResult(int requestCode, @NonNull
   String[] permissions, @NonNull int[] grantResults) {
4.     super.onRequestPermissionsResult(requestCode, permissions,
      grantResults);
```

```

5.         if (requestCode == LOCATION_REQUEST_CODE) {
6.             if (grantResults.length > 0
7.                 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
8.
9.                 if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
10.                    return;
11.                }
12.
13.                mMap.setMyLocationEnabled(true);
14.
15.                LocationCallback locationCallback = new LocationCallback()
16.                {
17.                    @SuppressWarnings("SetTextI18n")
18.                    @Override
19.                    public void onLocationResult(@NonNull LocationResult
locationResult) {
20.                        super.onLocationResult(locationResult);
21.                        locationResult.getLastLocation();
22.                        mMap.clear();
23.                        MarkerOptions mp = new MarkerOptions();
24.                        mp.position(new
LatLng(locationResult.getLastLocation().getLatitude(),
locationResult.getLastLocation().getLongitude()));
25.                        mp.title("Posisi saat ini");
26.                        mMap.addMarker(mp);
27.                        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(
new
LatLng(locationResult.getLastLocation().getLatitude(),
locationResult.getLastLocation().getLongitude()), 16));
28.
29.                        latitude =
locationResult.getLastLocation().getLatitude();
30.                        longitude =
locationResult.getLastLocation().getLongitude();
31.
32.                        Latitude.setText(Double.toString(latitude));
33.                        Longitude.setText(Double.toString(longitude));
34.                    }
35.                };
36.                LocationRequest locationRequest = new LocationRequest();
37.                locationRequest.setInterval(4000);
38.                locationRequest.setFastestInterval(2000);
39.                locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
40.
41.                LocationServices.getFusedLocationProviderClient(this).requestLocationUpdate
s(locationRequest, locationCallback, Looper.getMainLooper());
42.            }
43.
44.        }
45.    }

```

Pada Kode Program 4.10 ditunjukkan pada saat tombol “Tambahkan Lokasi” ditekan maka program akan mengambil data *latitude* dan *longitude* dari pengguna saat ini, namun sebelum mengambil lokasi saat ini program akan

meminta sebuah *permission* atau persetujuan dari pengguna untuk aplikasi mengetahui lokasi perangkat *smartphone* saat ini. Apabila pengguna sudah menyetujui *permission* tersebut maka aplikasi akan menampilkan lokasi pengguna saat ini dan mengambil data *latitude* dan *longitude* titik lokasi. Hal ini ditunjukkan pada baris 22-33 di mana awalnya sistem menyiapkan sebuah *marker* atau penanda berwarna merah setelah itu mengambil data lokasi saat ini dan menempatkan *marker* tersebut di titik lokasi *smartphone* dengan data *latitude* dan *longitude* yang diambil di mana ini ditunjukkan pada baris 23 dengan fungsi:

1. Untuk mengambil data *latitude* =
`locationResult().getLastLocation().getLatitude()`
2. Untuk mengambil data *longitude* =
`locationResult().getLastLocation().getLongitude()`

Setelah itu, pada baris 32 dan 33 data *latitude* dan *longitude* tadi dimasukkan pada properti *textView* untuk ditampilkan kepada pengguna. Hasil yang didapatkan setelah menekan tombol “Tambahkan lokasi” akan ditunjukkan pada Gambar 4.14.



Gambar 4. 14. Hasil setelah tombol "Tambahkan lokasi" ditekan

3. Menekan tombol kirim

Saat semua informasi dikirim, selanjutnya pengguna dapat menekan tombol kirim untuk mengirim aduan.

Kirim

Gambar 4. 15. Tombol kirim

Beberapa informasi yang dimasukkan kedalam basis data antara lain:

1. key "uid" dengan data dari variabel "current_user_id"
2. key "date" dengan data dari variabel "saveCurrentDate"
3. key "time" dengan data dari variabel "saveCurrentTime"
4. key "judul_post" dengan data dari variabel "judul"

5. key “*catatan*” dengan data dari variabel “*catatan*”
6. key “*kategori*” dengan data dari variabel “*kategori*”
7. key “*postimage*” dengan data dari variabel “*downloadUrl*”
8. key “*username*” dengan data dari variabel “*username*”
9. key “*lat*” dengan data dari variabel “*latitude*”
10. key “*long*” dengan data dari variabel “*longitude*”
11. key “*sort_descending*” dengan data dari variabel “*countPosts*”
12. key “*upvote*” dengan data 0
13. key “*downvote*” dengan data 0
14. key “*status*” dengan data “0”
15. key “*balasan*” dengan data “*Tidak ada*”
16. key “*bukti_pemerintah*” dengan data “*null*”

Data tersebut dikirimkan ke dalam basis data yang ditunjukkan dengan Kode Program 4.11, dengan fungsi *updateChildren(postsMap)*. Apabila data sudah berhasil dimasukkan ke dalam basis data maka pengguna akan langsung diarahkan ke halaman utama kembali. Salah satu contoh dari data yang dimasukkan ke dalam basis data ditunjukkan pada Gambar 4.16.

Kode Program 4. 11.Pengiriman data aduan kedalam basis data

```

1. String catatan = txtCatatan.getText().toString();
2.     if (snapshot.exists()) {
3.         String username =
snapshot.child("username").getValue().toString();
4.
5.         HashMap<String, Object> postsMap = new HashMap<>();
6.         postsMap.put("uid", current_user_id);
7.         postsMap.put("date", saveCurrentDate);
8.         postsMap.put("time", saveCurrentTime);
9.         postsMap.put("judul_post", judul);
10.        postsMap.put("catatan", catatan);
11.        postsMap.put("kategori", kategori);
12.        postsMap.put("postimage", downloadUrl);
13.        postsMap.put("username", username);

```

```

14.         postsMap.put("lat", Double.toString(latitude));
15.         postsMap.put("long", Double.toString(longitude));
16.         postsMap.put("sort_descending", countPosts);
17.         postsMap.put("upvote", 0);
18.         postsMap.put("downvote", 0);
19.         postsMap.put("status", "0");
20.         postsMap.put("balasan", "Tidak ada");
21.         postsMap.put("bukti_pemerintah", "null");
22.
23.         postsRef.child(current_user_id +
postRandomName).updateChildren(postsMap)
24.             .addOnCompleteListener(task -> {
25.                 if (task.isSuccessful()) {
26.                     SendUserToMainActivity();
27.                 } else {
28.                     String message =
task.getException().getMessage();
29.                     Toast.makeText(NextActivity.this,
"Terdaftar error : " + message, Toast.LENGTH_SHORT).show();
30.                 }
31.                 loadingBar.dismiss();
32.             });
33.     }

```



Gambar 4. 16. Contoh data yang dimasukkan setelah menekan tombol "Kirim"

4.1.6. Fungsi notifikasi

Fungsi notifikasi berguna untuk mengetahui apakah pengaduan yang diajukan sudah dibalas oleh pemerintah atau belum, selain itu pada saat pengguna lain berkomentar pada *posting* aduan, pengguna mendapatkan notifikasi. Hasil dari implementasi fungsi notifikasi ditunjukkan pada Gambar 4.17.



Gambar 4. 17.Halaman notifikasi

Pada halaman ini terdapat satu tombol yang difungsikan untuk menghapus notifikasi yang ada. Selain itu pengguna juga dapat menekan notifikasi, apabila notifikasi ditekan maka akan langsung diarahkan pada *posting* pengaduan yang berkaitan. Fungsi pemberian notifikasi dari komentar orang lain & tindaklanjut pemerintah ditunjukkan pada Kode Program 4.12.

Kode Program 4. 12.Fungsi notifikasi

```
1. private void AddNotification(String username) {
2.     //get Tanggal
3.     Calendar calForDate = Calendar.getInstance();
4.     @SuppressWarnings("SimpleDateFormat") SimpleDateFormat currentDate =
new SimpleDateFormat("dd-M-yyyy HH:mm:ss");
5.     final String saveCurrentDate =
currentDate.format(calForDate.getTime());
6.
7.     final String RandomKey = currentUserID + saveCurrentDate;
8.
9.     //getRole
10.    DatabaseReference roleRef =
FirebaseDatabase.getInstance().getReference().child("Users").child(currentU
serID);
11.    roleRef.child("role").get().addOnCompleteListener(new
OnCompleteListener<DataSnapshot>() {
12.        @Override
13.        public void onComplete(@NonNull Task<DataSnapshot> task) {
14.            String role = String.valueOf(task.getResult().getValue());
15.
16.            HashMap<String, Object> notificationsMap = new HashMap<>();
17.            notificationsMap.put("uid", currentUserID);
18.            notificationsMap.put("comment", commentText);
19.            notificationsMap.put("date", saveCurrentDate);
20.            notificationsMap.put("username", username);
21.            notificationsMap.put("postkey", PostKey);
22.            notificationsMap.put("uid_commented", UidCommented);
23.            notificationsMap.put("role", role);
24.
25.            NotificationsRef.child(RandomKey).updateChildren(notificationsMap);
26.        }
27.    });
28.
29. }
```

Data notifikasi akan dimasukkan ke dalam basis data dari pengguna yang mengirimkan *posting* pengaduan, maka dari itu terdapat pelemparan data *username* yang ditunjukkan pada baris 1 untuk mengetahui notifikasi akan dikirimkan ke mana. Lalu apabila sistem sudah mengetahui notifikasi akan dikirimkan ke *username* yang dituju maka akan dijalankan pemasukan data sebagai berikut kedalam basis data(ditunjukkan pada Kode Program 4.12 baris 16-23):

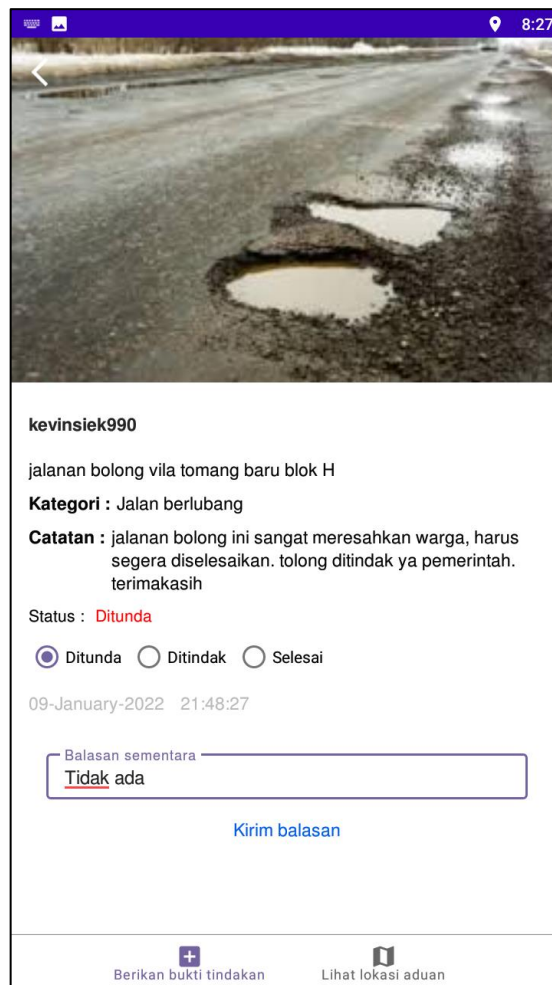
1. key “*uid*” dengan data dari variabel “*currentUserID*” (sebagai tanda pengenalan pengirim komentar)
2. key “*comment*” dengan data dari variabel “*commentText*” (sebagai isi dari notifikasi berupa teks komentar)

3. key “*date*” dengan data dari variabel “*saveCurrentDate*” (sebagai data tanggal pengiriman komentar)
4. key “*username*” dengan data dari variabel “*username*” (sebagai tanda pengenal pengirim komentar yang akan ditampilkan di notifikasi pembuat *posting*)
5. key “*postkey*” dengan data dari variabel “*PostKey*” (sebagai navigasi pembuat *posting* apabila ingin melihat komentar yang diberikan di daftar komentar)
6. key “*uid_commented*” dengan data dari variabel “*UidCommented*” (sebagai tanda pengenal pengguna yang diberikan notifikasi)
7. key “*role*” dengan data dari variabel “*role*” (sebagai pengenal level dari pengirim notifikasi apakah pemerintah atau masyarakat)

Setelah data didapatkan maka akan dimasukkan ke dalam basis data dengan fungsi yang ditunjukkan pada baris 25 yaitu `.child(RandomKey).updateChildren(notificationsMap)`. Setelah itu data akan masuk ke dalam basis data dan ditampilkan ke dalam notifikasi dari pengirim *posting* aduan yang ditunjukkan hasilnya pada Gambar 4.17.

4.1.7. Fungsi perubahan status, balasan sementara dan tindaklanjut

Fungsi ini diperuntukkan kepada pemerintah yang ingin mengubah status dari pengaduan yang diajukan oleh masyarakat, pada halaman utama pemerintah dapat menekan salah satu gambar dari aduan yang diajukan oleh masyarakat lalu akan diberikan detail dari aduan yang diajukan hal ini ditunjukkan pada Gambar 4.18.



Gambar 4. 18.Detail dari aduan yang diajukan masyarakat

Pada Gambar 4.18 pemerintah dapat melihat detail dari aduan berupa *username* dari pengirim aduan, judul dari aduan, kategori, catatan dari masyarakat yang melaporkan, dan status. Pada halaman ini pemerintah dapat langsung mengubah status dari “Ditunda” menjadi “Ditindak” dengan menekan tombol “Ditindak”. Pemerintah tidak dapat mengubah status menjadi selesai secara langsung karena apabila ingin mengubah status menjadi selesai harus memberikan bukti dari tindaklanjut yang dilakukan berupa foto dan deskripsi.

Selain dapat mengubah status, pemerintah dapat memberikan balasan sementara kepada laporan dengan menuliskan di kotak “Balasan sementara” lalu menekan tombol “kirim balasan”. Fungsi dari perubahan status ditunjukkan pada Kode Program 4.13.

Kode Program 4. 13.Fungsi perubahan status

```
1. radio_ditunda.setOnClickListener(view -> {
2.     status = "0";
3.     statusMap.put("status", status);
4.     PostRef.updateChildren(statusMap);
5. });
6.
7. radio_ditindak.setOnClickListener(view -> {
8.     status = "1";
9.     statusMap.put("status", status);
10.    PostRef.updateChildren(statusMap);
11. });
12.
13. radio_selesai.setOnClickListener(view -> {
14.    MaterialAlertDialogBuilder builder = new
MaterialAlertDialogBuilder(ClickPost.this);
15.    builder.setTitle("Informasi status");
16.    builder.setMessage("Status tidak bisa diubah ke 'Selesai'.
Apabila ingin mengubah status maka berikan bukti penindakan dari aduan yang
diajukan.");
17.    builder.setPositiveButton("Ok", (dialogInterface, i) ->
radio_selesai.setChecked(false));
18.    builder.show();
19. });
```

Pada baris satu ditunjukkan apabila *radio button* ditunda ditekan maka data pada basis data akan di *update* atau diubah menjadi “0” untuk status Ditunda. Apabila *radio button* ditindak ditekan maka data pada basis data akan diubah menjadi “1” untuk status Ditindak. Pada *radio button* selesai ditekan maka sistem akan memberikan sebuah pesan bahwa status tidak bisa diubah ke selesai secara langsung, untuk mengubah status dari Ditunda / Ditindak menjadi Selesai maka pemerintah harus menekan tombol “Berikan bukti tindakan” yang ditunjukkan pada Gambar 4.19.



Gambar 4. 19.Tombol berikan bukti tindakan

Setelah pemerintah menekan tombol tersebut maka akan langsung diarahkan ke halaman pemberian bukti tindakan, berikut langkah-langkah pemerintah memberikan bukti tindak lanjut:

1. Pemerintah harus memasukkan foto seperti pada fungsi Kode Program 4.7 atau 4.8 yaitu pengambilan bukti melalui galeri atau melalui kamera secara langsung.
2. Setelah memasukkan foto pemerintah dapat menekan tombol selanjutnya, lalu akan dimunculkan halaman baru yaitu halaman untuk memberikan deskripsi dari tindakan yang ditunjukkan pada Gambar 4.20.
3. Menekan tombol kirim



Gambar 4. 20.Pemberian bukti tindakan dan deskripsi penindakan

Setelah menekan tombol kirim, maka fungsi yang akan dijalankan ditunjukkan pada Kode Program 4.14. Pada fungsi ini, data akan dimasukkan ke dalam basis data berupa bukti *URI* foto bukti dan deskripsi tindakan berupa teks.

Kode Program 4. 14.Pemberian bukti tindakan

```
1. UidCommented = (String) snapshot.child("uid").getValue();  
2.  
3.             HashMap<String, Object> buktiTindakanMap = new HashMap<>();  
4.
```

```

5.         buktiTindakanMap.put("bukti_pemerintah", downloadUrl);
6.         buktiTindakanMap.put("balasan", deskripsi);
7.         buktiTindakanMap.put("status", "2");
8.
9.         postsRef.child(PostKey).updateChildren(buktiTindakanMap)
10.        .addOnCompleteListener(task -> {
11.            if (task.isSuccessful()) {
12.
13.                AddNotification();
14.
15.                SendUserToDashboardPemerintah();
16.            } else {
17.                String message =
18.                task.getException().getMessage();
19.                Toast.makeText(Pemerintah_NextActivity.this, "Terdapat error : " + message,
20.                Toast.LENGTH_SHORT).show();
21.            }
22.            loadingBar.dismiss();
23.        });

```

Data yang dimasukkan ke dalam basis data ditunjukkan pada Kode Program 4.14 baris 5-7 yang berisikan sebagai berikut:

1. key “*bukti_pemerintah*” dengan data dari variabel “*downloadUrl*” (sebagai bukti gambar berupa *link* untuk dilihat sebagai bukti tindakan pada aduan)
2. key “*balasan*” dengan data dari variabel “*deskripsi*” (sebagai deskripsi dari tindakan yang dilakukan berupa teks)
3. key “*status*” dengan data “2” (sebagai perubahan status dari aduan menjadi Selesai)

Setelah data didapatkan, maka selanjutnya akan dimasukkan ke dalam basis data edngan fungsi *.updateChildren(buktiTindakanMap)*. Apabila berhasil maka notifikasi akan masuk kedalam pengirim aduan dengan pemanggilan fungsi *AddNotification()* dan akan pemerintah akan langsung diarahkan kembali ke halaman utama pemerintah. Apabila terjadi *error* atau gagal dalam pengiriman maka akan diberikan pesan oleh sistem yang ditunjukkan pada baris 16-19.

Masyarakat yang sudah ditindak aduannya dapat langsung melihat bukti aduan melalui notifikasi, dengan menekan notifikasi yang diberikan oleh pemerintah setelah

aduannya ditindak maka pengguna dapat langsung menekan tombol “Lihat bukti” untuk melihat bukti dari tindak lanjut yang diajukan oleh pemerintah.

4.2. Hasil Analisis *Usability Testing*

Pengujian dari sistem yang dibuat menggunakan *usability testing* yang dilakukan kepada masyarakat berdomisili di Kabupaten Tangerang. Pertama, responden menggunakan sistem terlebih dahulu dan menguji semua fungsi yang ada pada sistem selama kurun waktu 5-10 menit, setelah itu akan diberikan kuesioner dengan jawaban skala *likert* yang ditunjukkan pada Tabel 3.1. Hasil dari pengujian sistem dibagi menjadi dua, yaitu hasil dari responden individual untuk mendapatkan skor dari masing-masing responden dan hasil menggunakan Persamaan 3.1 untuk mendapatkan skor rata-rata dari kelayakan sistem yang akan ditentukan melalui *system usability scale* di mana ditunjukkan pada Gambar 3.8. Jika skor rata-rata yang didapatkan melebihi angka 70 maka sistem dapat dikatakan layak.

Perhitungan sementara didapatkan dari penjumlahan nilai responden dari Q1-Q10 dikalikan 2,5. Hasil tersebut ditunjukkan pada tabel 4.1.

Tabel 4. 1.Data responden dan jumlah skor *SUS*

Responden	Jumlah Ganjil	Jumlah Genap	Jumlah x 2,5
1	20	14	85
2	20	18	95
3	18	15	82,5
4	19	17	90
5	19	18	92,5
6	20	17	92,5
7	18	13	77,5
8	18	13	77,5
9	17	13	75
10	14	12	65
11	18	14	80
12	17	12	72,5

13	19	16	87,5
14	18	17	87,5
15	20	13	82,5
16	19	13	80
17	20	14	85
18	19	15	85
19	19	13	80
20	20	14	85
Total			1657,5
Rata-rata			82,875

Berdasarkan hasil tersebut rata-rata skor yang didapatkan dari responden individual didapatkan skor di atas 70. Selanjutnya akan dilakukan perhitungan menggunakan persamaan untuk mendapatkan skor rata-rata kelayakan sebuah sistem yaitu dengan Persamaan 3.1 menggunakan data sebagai berikut:

1. Jumlah skor $SUS = 1657,5$
2. Jumlah responden = 20

Skor rata-rata didapatkan dari perhitungan jumlah skor SUS dibagi jumlah responden. Skor rata-rata yang didapatkan dari sistem ini adalah 82,875 dengan *grade scale* B (*Acceptable*). Jadi, sistem yang dibuat pada penelitian ini dapat dikatakan layak. Menurut (Susilo, 2019) sebuah sistem dikatakan layak untuk digunakan apabila mendapatkan skor di atas 70 dengan *grade scale* lebih besar dari C.

BAB 5

SIMPULAN DAN SARAN

5.1. Simpulan

Melalui penelitian analisis dan perancangan Sistem Pengaduan Masyarakat Kabupaten Tangerang ini, didapatkan simpulan sebagai berikut:

1. Aplikasi dapat digunakan oleh masyarakat dan pemerintah melalui *smartphone*.
2. Penelitian yang dilakukan ini dapat digunakan sebagai acuan dari pergantian sistem yang sebelumnya dan dapat menjadi sistem pelaporan baru untuk wilayah Kabupaten Tangerang.

Proses pengujian dilakukan dengan menggunakan *usability testing* kepada masyarakat Kabupaten Tangerang. Hasil dari pengujian sistem ini dapat disimpulkan bahwa sistem ini layak untuk digunakan sesuai dengan jumlah skor *SUS* sebesar 82,875 dan *grade scale* B (*Acceptable*).

5.2. Saran

Saran yang dapat diberikan terhadap sistem yang dibuat ini yaitu dilakukannya pengembangan pada *user interface* dari sistem karena tidak difokuskannya penelitian ini terhadap hal tersebut. Selain itu juga dapat dilakukannya pengembangan terhadap fitur-fitur baru yang dapat membantu penindakan laporan menjadi lebih mudah dari yang saat ini dibuat.

DAFTAR PUSTAKA

- Ahmad Ansori. (2020, March 29). *Pengertian Activity Diagram : Tujuan, Simbol, dan Contohnya*. <https://www.ansoriweb.com/2020/03/pengertian-activity-diagram.html>.
- Ainiyah, N. (2018). Remaja Millenial dan Media Sosial: Media Sosial Sebagai Media Informasi Pendidikan bagi Remaja Millenial. In *JPII* (Vol. 2, Issue 2).
- Alifia Seftin Oktriwina. (2021, April 20). *Apa Itu Class Diagram dan Fungsinya dalam Pemrograman*. <https://glints.com/id/lowongan/class-diagram-adalah/>.
- Ananda, A. N. (2018). *Mobile Commerce Berbasis Android pada PT Mega Mulia Pharma Palembang*. 1–11.
- Andika Wasisto. (2018, March 7). *Belajar Menggunakan Firebase Cloud Messaging*. <https://medium.com/@pplA2EDC/belajar-menggunakan-firebase-cloud-messaging-7f6bf3b42da7>.
- Avin Riyan. (2018, June 7). *Mencoba Firebase Authentication di Android (Email & Password)*. <https://medium.com/@avinriyan/mencoba-firebase-authentication-di-android-email-password-4089bd2dfec0>.
- Deny Suryani. (2018, January 11). *Masalah Lingkungan Butuh Ketegasan Pemerintah*. <https://sumbarprov.go.id/home/news/13134-masalah-lingkungan-butuh-ketegasan-pemerintah>.
- Dicoding Intern. (2020, November 25). *Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya*. <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>.

- Dori, D., Sillitto, H., Griego, R. M., McKinney, D., Arnold, E. P., Godfrey, P., Martin, J., Jackson, S., & Krob, D. (2020). System Definition, System Worldviews, and Systemness Characteristics. *IEEE Systems Journal*, 14(2), 1538–1548.
<https://doi.org/10.1109/JSYST.2019.2904116>
- Dwinawan Hariwijaya. (2017, May 9). *10 Langkah dalam Melakukan Usability Testing untuk Produk Kamu*. <https://id.techinasia.com/talk/langkah-usability-testing-untuk-produk>.
- Fikri Akhdi Maulana. (2014). *Semua Tentang Android*.
- Flin Setyadi. (2021, February 14). *Use Case Diagram Lengkap Dengan Contohnya*. <https://flinsetyadi.com/use-case-diagram/>.
- Hema Mitta Kalyani. (2019, May 1). *Mengenal Usability Testing*. <https://medium.com/learnfazz/mengenal-usability-testing-6ebb9069a2c0>.
- Jakob Nielsen. (2012, June 3). *How Many Test Users in a Usability Study?* <https://www.nngroup.com/articles/how-many-test-users/>.
- Juanda. (2018). *Fenomena Eksploitasi Lingkungan dalam Cerpen Koran Minggu Indonesia Pendekatan Ekokritik*.
- Leslie Hendric Spits Warnars, H. (2017). Warnars-Pemodelan E-Learning Perguruan Tinggi dengan Menggunakan Framework Learning Technology System Architecture (LTSA) dan Unified Modeling Language (UML) Pemodelan E-Learning Perguruan Tinggi dengan Menggunakan Framework Learning Technology System Architecture (LTSA) dan Unified Modeling Language (UML). *Jurnal Ilmiah Teknolog Informasi*, 15(1), 43–55.
- Loki Fadilah. (2019, November 5). *Pengertian Android Studio, Fungsi dan Cara Menggunakannya*. <https://androbuntu.com/2019/11/05/android-studio/>.

- M Prawiro. (2018, November 18). *Pengertian Sistem: Definisi, unsur-unsur, dan jenis-jenis sistem*. <https://www.maxmanroe.com/vid/manajemen/pengertian-sistem.html>.
- M Teguh Prihandoyo. (2018). Unified Modeling Language (UML) Model untuk Pengembangan Sistem Informasi Akademik Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 126–129.
- Michael Kharisma Hutaeruk. (2019, November 26). *UML Diagram: Use Case Diagram*. <https://socs.binus.ac.id/2019/11/26/Uml-Diagram-Use-Case-Diagram/>.
- Muhammad Muslihudin, & Oktafianto. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML* (A. Pramesta, Ed.). CV Andi Offset (Penerbit ANDI).
- Putra. (2019, October 23). *Pengertian Android: Sejarah, Kelebihan & Versi Sistem Operasi Android OS*. <https://salamadian.com/pengertian-android/>.
- Sanad, E. A. W. (2019). Pemanfaatan *Realtime Database* di *Platform Firebase* pada Aplikasi E-Tourism Kabupaten Nabire. *Jurnal Penelitian Enjiniring*, 22(1), 20–26. <https://doi.org/10.25042/jpe.052018.04>
- Selatan, J., Fadallah, M. F., & Rosyida, S. (2018). VII NO.1 Februari 2018] 61 ISSN 2098-8711 | Analisa Penerimaan Teknologi... 1 Program Manajemen Informatika AMIK BSI Bekasi. *Jurnal Sistem Informasi STMIK Antar Bangsa*, 7(1), 61–70.
- Sholihin, M. R., Arianto, W., Khasanah, D. F., Widya, S., & Lumajang, G. (2018). Keunggulan Sosial Media dalam Perkembangan Ekonomi Kreatif Era Digital di

- Indonesia. *Prosiding 4th Seminar Nasional Dan Call for Papers Fakultas Ekonomi Universitas Muhammadiyah Jember*, 149–160.
- Sugiyono. (2015). *Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D)*. Alfabeta.
- Susilo, E. (2019, March 7). *Cara Menggunakan System Usability Scale (SUS) pada Evaluasi Usability*. <https://www.edisusilo.com/cara-menggunakan-system-usability-scale/>.
- T Bayu Kurniawan. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman pada Cafeteria No Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MYSQL. *Jurnal Tikar*, 193–206.
- Utomo, A. P., Amalia, T. R., Iqbal, M., & Narulita, E. (2020). *Android-Based Comic of Biotechnology For Senior High School Students. International Journal of Scientific & Technology Research*, 9(03), 4143–4150. www.ijstr.org
- Wahyu Setia Bintara. (2021, February 27). *Pengertian Android Adalah | Sejarah, Versi, Kelebihan dan Kekurangan*. <https://dianisa.com/pengertian-android/>.
- Yuli Syafitri. (2016). *Analisa dan Perancangan Berbasis UML pada Sistem Informasi Simpan Pinjam Koperasi Swamitra Bandar Lampung*. 22–31.

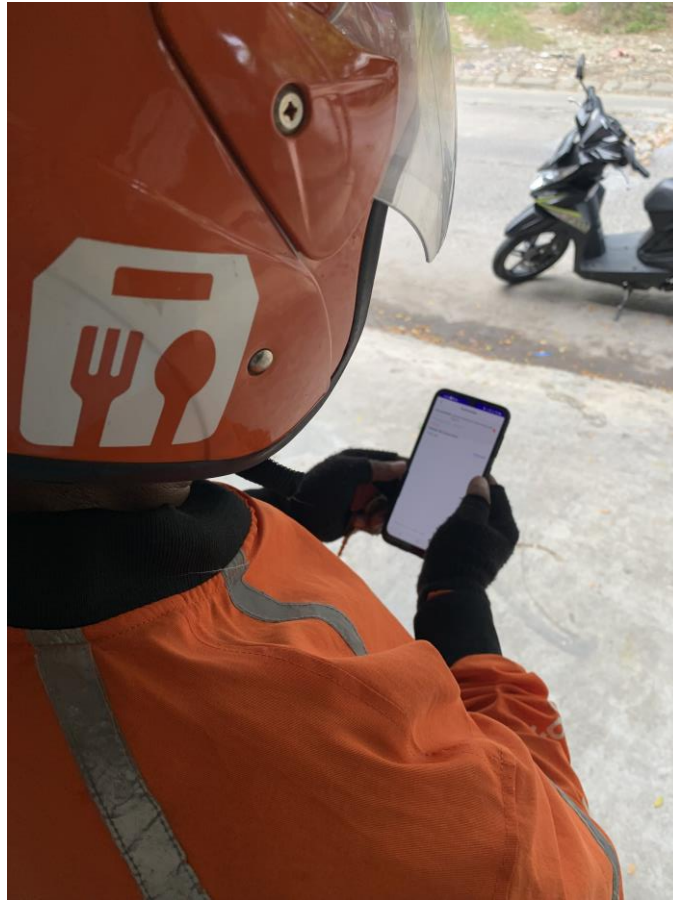
LAMPIRAN



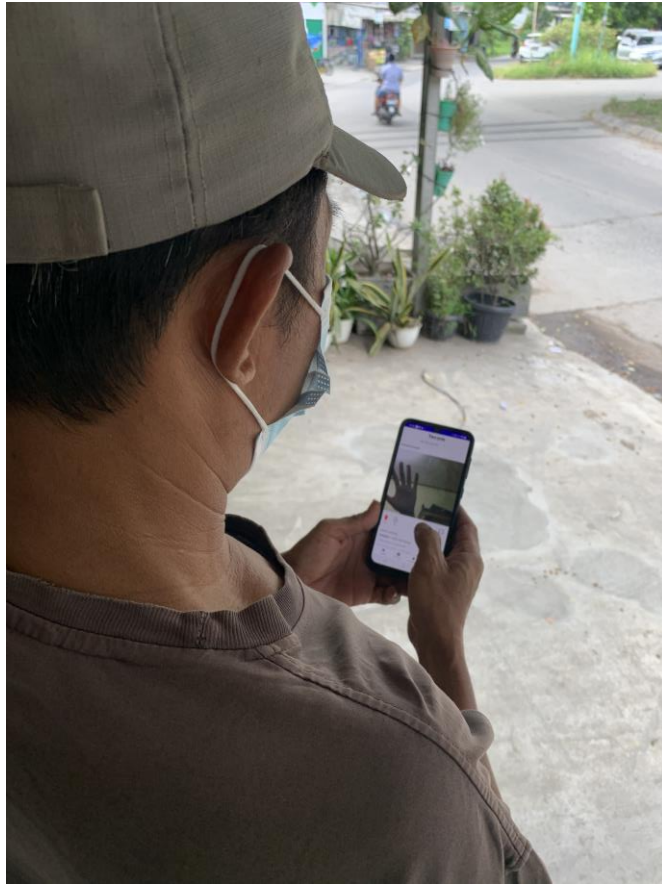
A- 1.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 1



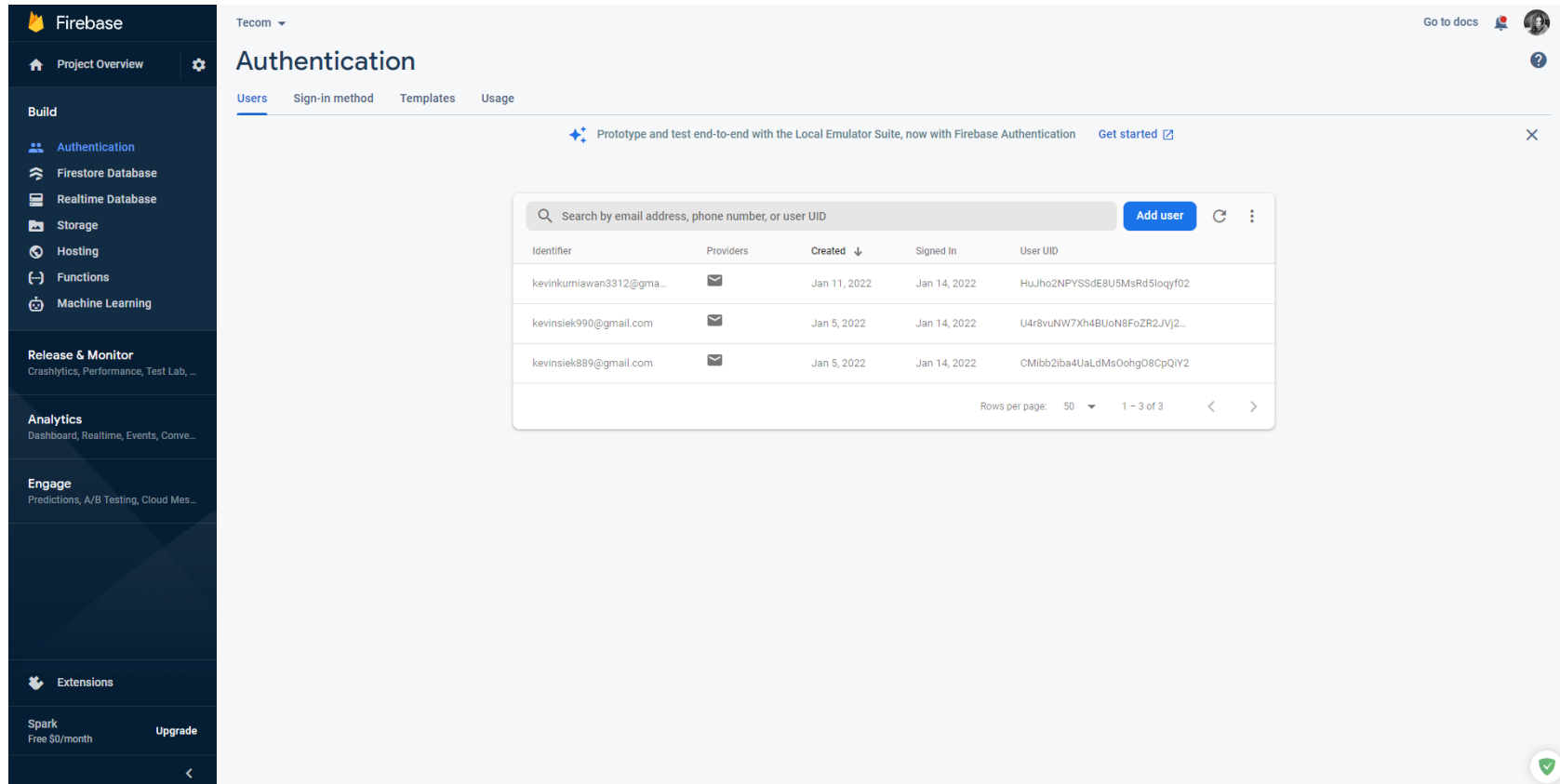
A- 2.Pengujian sistem terhadap masyarakat Kabupten Tangerang 2



A- 3.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 3



A- 4.Pengujian sistem terhadap masyarakat Kabupaten Tangerang 4



The screenshot displays the Firebase Authentication console. On the left is a dark sidebar with the Firebase logo and navigation links: Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor, Analytics, Engage, Extensions, and Spark. The main area is titled 'Authentication' and includes tabs for Users, Sign-in method, Templates, and Usage. A notification banner at the top suggests using the Local Emulator Suite. Below this is a search bar and an 'Add user' button. A table lists three users with columns for Identifier, Providers, Created, Signed In, and User UID. The table shows three users, all created on Jan 5, 2022, and signed in on Jan 14, 2022. The bottom of the console shows a 'Spark' section with a 'Free \$0/month' label and an 'Upgrade' button.

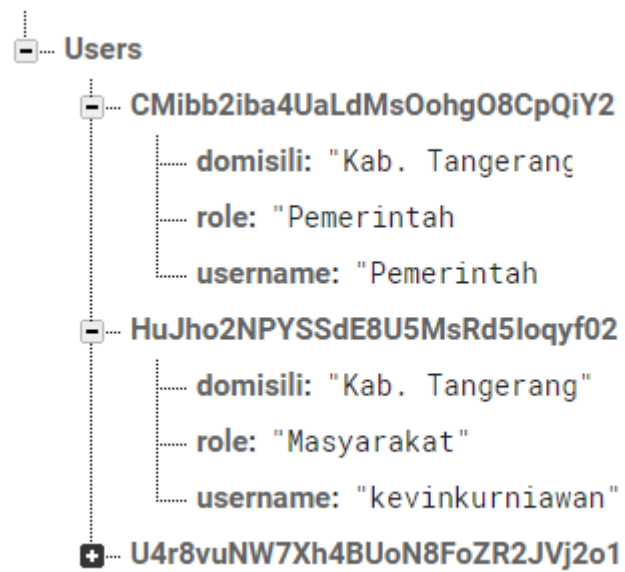
Identifier	Providers	Created ↓	Signed In	User UID
kevinkurniawan3312@gma...	📧	Jan 11, 2022	Jan 14, 2022	HuJho2NPYSSdE8U5MsRd5loqyf02
kevinsiek990@gmail.com	📧	Jan 5, 2022	Jan 14, 2022	U4r8vuNW7Xh4BUoN8FoZR2JVj2...
kevinsiek889@gmail.com	📧	Jan 5, 2022	Jan 14, 2022	CMibb2Iba4UaLdMsOohgO8CpQIY2

B- 1.Data *e-mail* dan *password* pengguna di *Firebase Authentication*

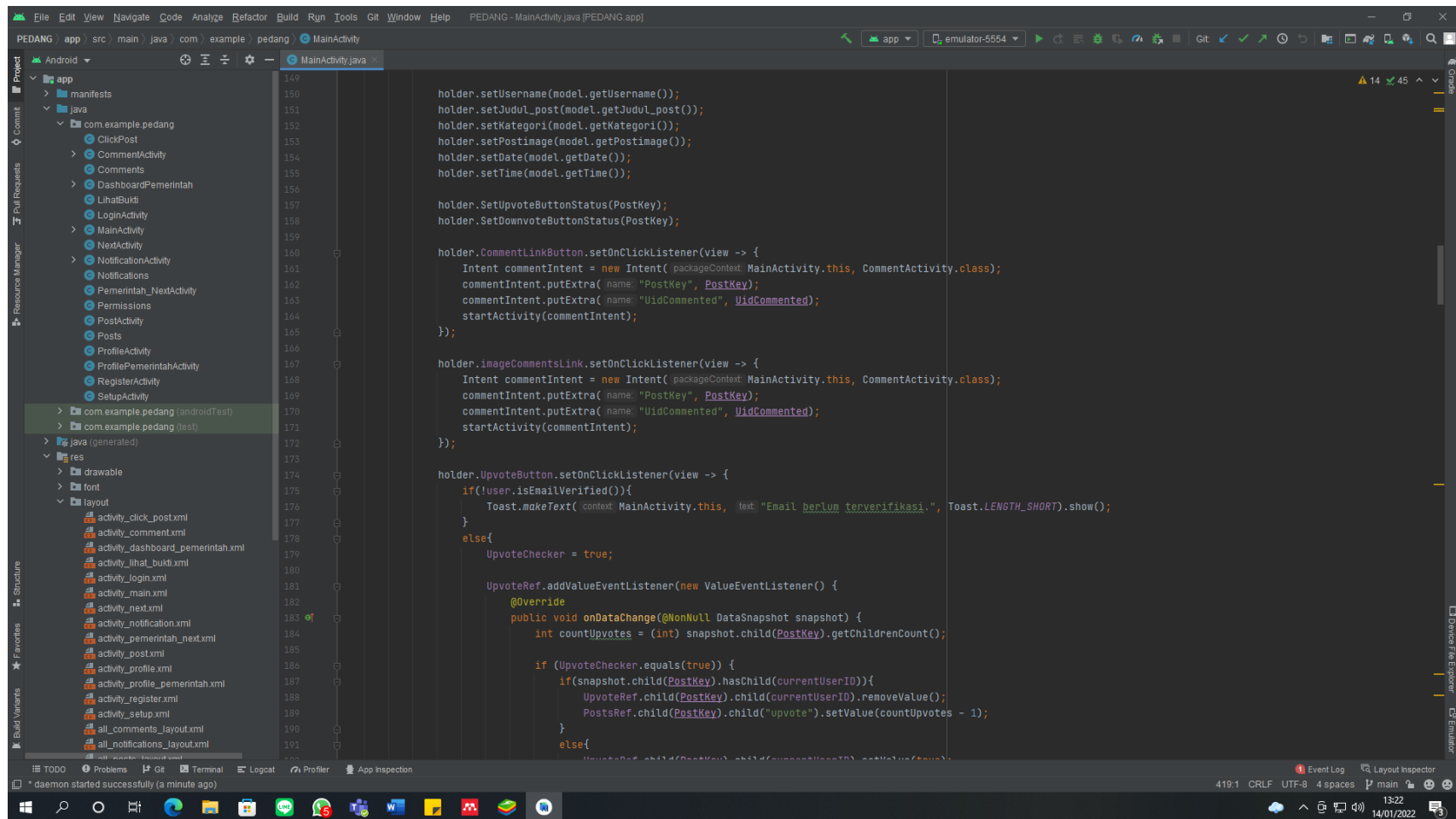
The screenshot displays the Firebase Realtime Database interface. On the left is a dark sidebar with navigation options: Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor, Analytics, Engage, and Extensions. The main area is titled 'Realtime Database' with tabs for Data, Rules, Backups, and Usage. A warning banner at the top states: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The data tree shows a 'posts' node under the project name 'pedang-76e16-default-rtdb'. A specific post is expanded, showing a key-value pair where the key is a long alphanumeric string and the value is a JSON object representing a complaint. The JSON object includes fields for 'balasan', 'bukti_pemerintah', 'catatan', 'date', 'downvote', 'judul_post', 'kategori', 'lat', 'long', 'postimage', 'sort_descending', 'status', 'time', 'uid', 'upvote', and 'username'. The database location is noted as 'Singapore (asia-southeast1)'.

```
pedang-76e16-default-rtdb
├── Posts
│   └── HuJho2NPYSSdE8U5MsRd5Ioqyf0214-January-2022 13:04:42
│       ├── balasan: "Tidak ada"
│       ├── bukti_pemerintah: "null"
│       ├── catatan: "tes"
│       ├── date: "14-January-2022"
│       ├── downvote: 0
│       ├── judul_post: "tes"
│       ├── kategori: "Rapid test"
│       ├── lat: "-6.2087631"
│       ├── long: "106.8455963"
│       ├── postimage: "https://firebasestorage.googleapis.com/v0/b/ped..."
│       ├── sort_descending: 0
│       ├── status: "0"
│       ├── time: " 13:04:42"
│       ├── uid: "HuJho2NPYSSdE8U5MsRd5Ioqyf02"
│       ├── upvote: 0
│       └── username: "kevinkurniawan"
└── Users
```

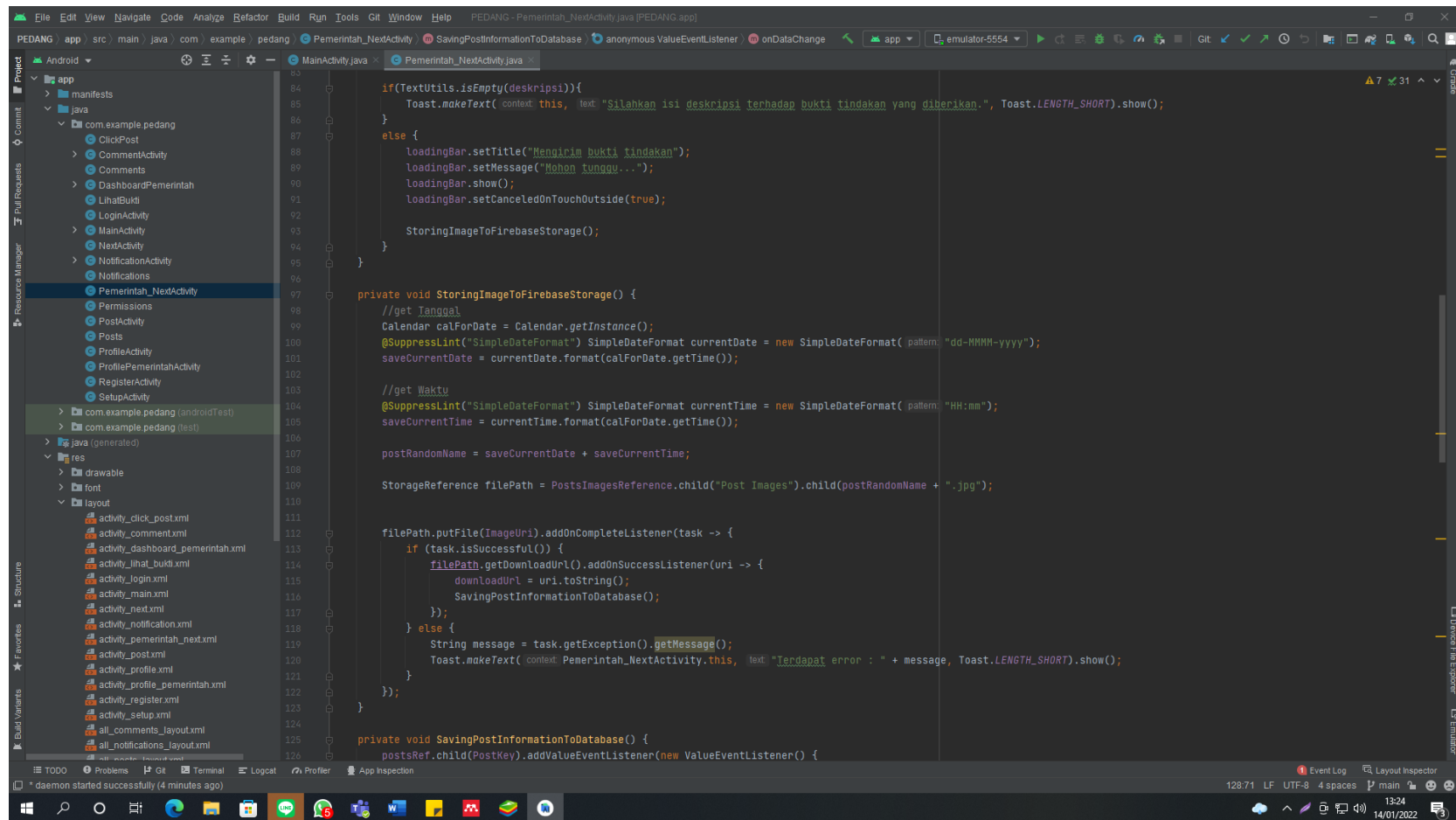
B- 2.Contoh data *posting* aduan pada *Firebase Realtime Database*

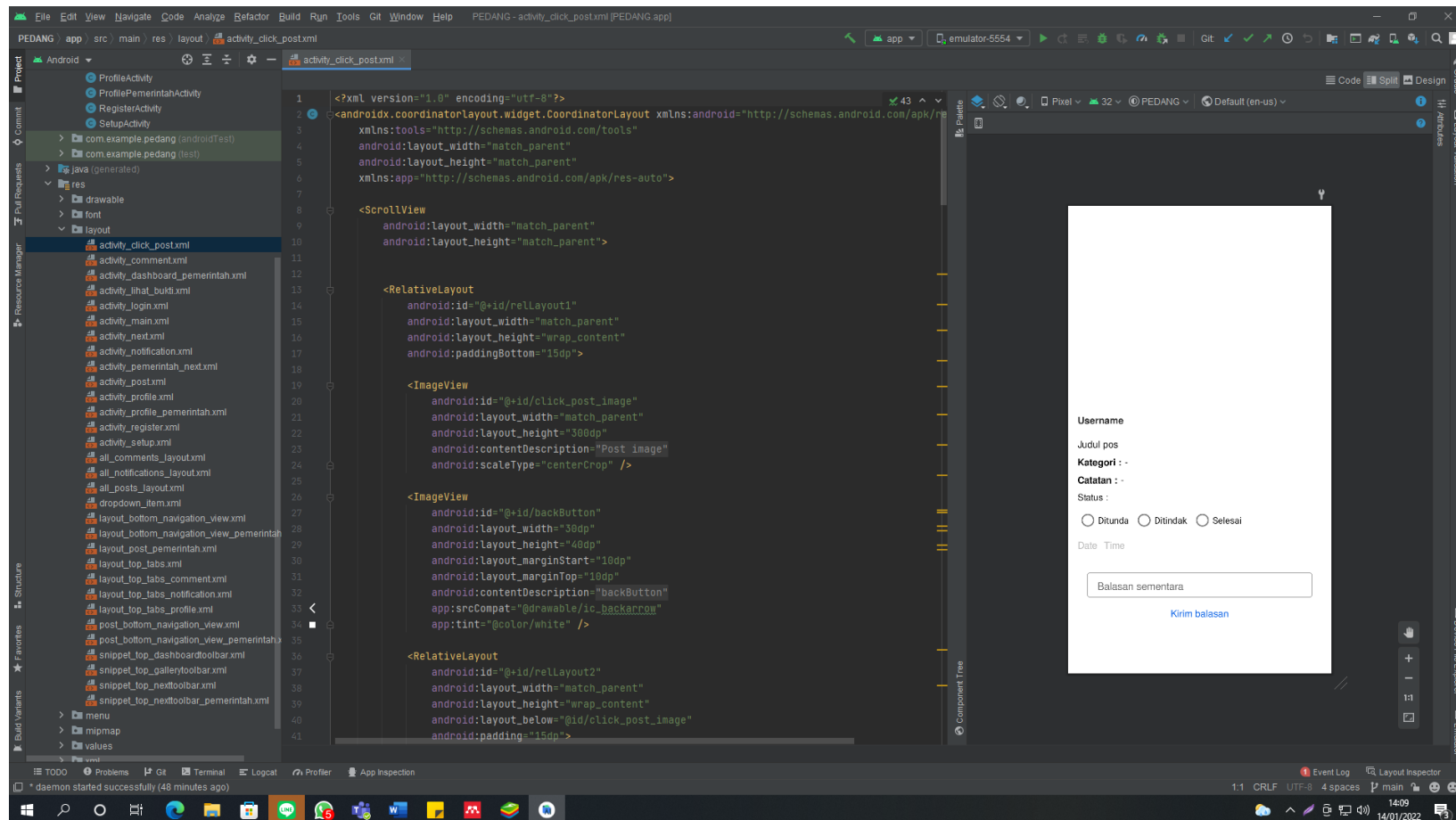


B- 3.Contoh data pemerintah dan masyarakat pada *Firebase Realtime Database*



C- 1.Koding fungsi navigasi komentar dan tombol *upvote* di *Android Studio*

C- 2.Koding fungsi memasukkan foto ke dalam database *Firebase Storage*

C- 3.Koding *xml* untuk membuat *design* sebuah halaman