

**APLIKASI PEMANGGIL PEMADAM KEBAKARAN MENGGUNAKAN  
METODE *LOCATION BASED SERVICE* (LBS)  
(STUDI KASUS : DAMKAR KOTA TEGAL)**



**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi Pada  
Program Studi Teknik Informatika

Oleh :

**Nama : Wahyu Indrawan**  
**NIM : 18090108**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK HARAPAN BERSAMA**  
**TEGAL**  
**2023**

**APLIKASI PEMANGGIL PEMADAM KEBAKARAN MENGGUNAKAN  
METODE *LOCATION BASED SERVICE* (LBS)  
(STUDI KASUS : DAMKAR KOTA TEGAL)**



**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi Pada  
Program Studi Teknik Informatika

Oleh :

**Nama : Wahyu Indrawan**  
**NIM : 18090108**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK HARAPAN BERSAMA**  
**TEGAL**  
**2023**

## HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Wahyu Indrawan

NIM : 18090108

Adalah mahasiswa Program Studi Teknik Informatika Politeknik Harapan Bersama. Dengan ini saya menyatakan bahwa laporan Skripsi yang berjudul:

**“APLIKASI PEMANGGIL PEMADAM KEBAKARAN MENGGUNAKAN  
METODE LOCATION BASED SERVICE (LBS)  
(STUDI KASUS : DAMKAR KOTA TEGAL)”**

Merupakan hasil pemikiran sendiri secara orisinil yang saya susun secara mandiri dengan tidak melanggar kode etik hak karya cipta. Apabila dikemudian hari Laporan Skripsi ini terbukti melanggar kode etik karya cipta, maka saya bersedia untuk melakukan penelitian baru dan menyusun laporannya sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya dan sesungguhnya.

Tegal, 31 Agustus 2023

Yang membuat pernyataan,



Wahyu Indrawan

NIM. 18090108

## HALAMAN REKOMENDASI

Pembimbing Tugas Akhir memberikan rekomendasi kepada :

Nama : Wahyu Indrawan

NIM : 18090108

Program Studi : D IV Teknik Informatika

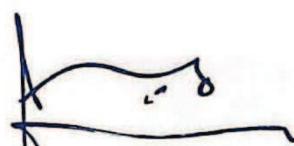
Judul Tugas Akhir : Aplikasi *Location Based Service* (LBS)  
Pemanggil Pemadam Kebakaran (Studi Kasus  
: Kota Tegal)

Untuk mengikuti Ujian Tugas Akhir karena telah memenuhi  
persyaratan yang telah ditentukan.

Tegal, Juli 2023

Pembimbing I,

Pembimbing II



Ginanjar Wiro Sasmito, M.Kom.  
NIPY. 10.007.032



Dega Sugono Wibowo, ST., M.Kom.  
NIPY. 06.014.183

## HALAMAN PENGESAHAN

Nama : Wahyu Indrawan  
NIM : 18090108  
Program Studi : D IV Teknik Informatika  
Judul Skripsi : Aplikasi Pemanggil Pemadam Kebakaran Menggunakan  
Metode Location Based Service (LBS)  
(Studi Kasus : Damkar Kota Tegal)

Dinyatakan LULUS / TIDAK LULUS Ujian Skripsi pada program studi Teknik Informatika Politeknik Harapan Bersama.

Tegal, Agustus 2023

Dewan Pengaji

Nama	Tanda Tangan
1. Ketua : M.Nishom, M.Kom.,	
2. Anggota I : Taufiq Abidin, S.Pd., M.Kom.,	
3. Anggota II : Dega Surono Wibowo, S.T., M.Kom.,	

Mengetahui,

Ketua Program Studi Teknik Informatika



Siagian Wibowo, S.Pd., M.Eng.

NIPY. 08.015.222

## ABSTRAK

Petugas pemadam kebakaran atau biasa disingkat Damkar merupakan pelayanan masyarakat yang bertugas memadamkan api, menyelamatkan, dan mengatasi bencana atau kejadian lainnya. Masyarakat Kota Tegal menghadapi kendala ketika dalam melaporkan situasi, kondisi ataupun keadaan darurat. Banyak masyarakat Kota Tegal yang merasa sulit untuk menghubungi petugas pemadam kebakaran di karenakan ketidaktahuan masyarakat cara menghubungi pemadam kebakaran. Oleh karena itu, membuat aplikasi di wilayah Kota Tegal yang berguna untuk memudahkan dalam menghubungi pemadam kebakaran sehingga dapat langsung membantu keadaan darurat. Aplikasi ini diberi nama “Aplikasi Pemanggil Pemadam Kebakaran Menggunakan Metode *Location Based Service* (LBS) (Studi Kasus : Damkar Kota Tegal)”. Aplikasi pemadam kebakaran ini dibangun menggunakan Android Studio dan menggunakan Google Maps Api untuk menampilkan peta dan rute dari suatu tempat, menggunakan Global Positioning System (GPS) untuk mengetahui posisi pengguna yang melakukan panggilan saat ini.

Kata kunci : Pemadam Kebakaran, LBS, *Location Based Service*, Kota Tegal

## KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah Subhaanahu Wa Ta'aala, Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya hingga terselesaikannya laporan Skripsi dengan judul “Aplikasi Pemanggil Pemadam Kebakaran Menggunakan Metode *Location Based Service* (LBS) (Studi Kasus : Damkar Kota Tegal)”.

Skripsi merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat kelulusan dalam mencapai derajat Sarjana Sains Terapan pada program Studi Teknik Informatika Politeknik Harapan Bersama. Selama melaksanakan penelitian dan kemudian tersusun dalam laporan Skripsi ini, banyak pihak yang telah memberikan bantuan, dukungan dan bimbingan.

Pada kesempatan ini, tak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Agung Hendarto, S.E., M.A. selaku Direktur Politeknik Harapan Bersama Tegal.
2. Slamet Wiyono, S.Pd., M.Eng. selaku Ketua Program Studi Teknik Informatika Politeknik Harapan Bersama Tegal.
3. Ginanjar Wiro Sasmito, M.Kom. selaku pembimbing I.
4. Dega Surono Wibowo, ST., M.Kom. selaku pembimbing II.
5. Bagian Sistem Penjaminan Mutu Internal Program Studi Teknik Informatika Politeknik Harapan Bersama Tegal.
6. Semua pihak yang telah mendukung, membantu serta mendoakan penyelesaian laporan Skripsi ini.

Tegal, Agustus 2023  
Penulis

Wahyu Indrawan

## DAFTAR ISI

<b>HALAMAN SAMPUL .....</b>	<b>i</b>
<b>HALAMAN JUDUL.....</b>	<b>ii</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>iii</b>
<b>HALAMAN REKOMENDASI .....</b>	<b>iv</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>DAFTAR ISI .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR .....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xiii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Tujuan Penelitian .....	2
1.3    Manfaat Penelitian.....	3
1.4    Batasan Penelitian .....	3
1.5    Tinjauan Pustaka .....	3
1.6    Data Penelitian .....	8
<b>BAB II PRODUK .....</b>	<b>9</b>
2.1    Perancangan Sistem.....	9
2.1.1    Desain Sistem .....	9
2.1.2    Pemodelan Sistem UML .....	10
2.1.3    Perancangan Antarmuka .....	38

2.1.4	Implementasi .....	51
2.2	Hasil Pengujian .....	63
2.3	Kesimpulan Dan Saran.....	65
2.3.1	Kesimpulan.....	65
2.3.2	Saran.....	66
<b>BAB III HKI</b>	.....	<b>67</b>
3.1	Proses .....	67
3.2	Identitas HKI.....	67
<b>DAFTAR PUSTAKA</b>	.....	<b>69</b>
<b>LAMPIRAN</b>	.....	<b>72</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>Use Case Diagram</i> .....	11
Gambar 2.2 <i>Activity Warga Login Warga</i> .....	12
Gambar 2.3 <i>Activity Warga Register Warga</i> .....	13
Gambar 2.4 <i>Activity Laporan Warga</i> .....	13
Gambar 2.5 <i>Activity Telpone warga</i> .....	14
<i>Gambar 2.6 Activity History warga</i> .....	14
Gambar 2.7 <i>Activity Detail History warga</i> .....	15
Gambar 2.8 <i>Activity Profile warga</i> .....	15
Gambar 2.9 <i>Activity Logout warga</i> .....	16
Gambar 2.10 <i>Activity Login Petugas</i> .....	16
Gambar 2.11 <i>Activity Laporan Petugas</i> .....	17
Gambar 2.12 <i>Activity Profile Petugas</i> .....	17
Gambar 2.13 <i>Activity Logout Petugas</i> .....	18
Gambar 2.14 <i>Activity Login Admin</i> .....	19
<i>Gambar 2.15 Activity Laporan Masuk Admin</i> .....	19
Gambar 2.16 <i>Activity Warga Admin</i> .....	20
<i>Gambar 2.17 Activity Laporan Admin</i> .....	20
Gambar 2.18 <i>Activity Petugas Admin</i> .....	21
Gambar 2.19 <i>Activity Delete Petugas Admin</i> .....	21
Gambar 2.20 <i>Activity Tambah Alat Admin</i> .....	22
Gambar 2.21 <i>Activity Edit Alat Admin</i> .....	23
Gambar 2.22 <i>Activity Logout Admin</i> .....	23
<i>Gambar 2.23 Sequence Login Warga</i> .....	24
<i>Gambar 2.24 Sequence Laporan Warga</i> .....	25
<i>Gambar 2.25 Sequence Telephone Warga</i> .....	25
<i>Gambar 2.26 Sequence History Warga</i> .....	26
Gambar 2.27 <i>Sequence Detail History Warga</i> .....	27
Gambar 2.28 <i>Sequence Profile Warga</i> .....	27
Gambar 2.29 <i>Sequence Logout warga</i> .....	28
Gambar 2.30 <i>Sequence Login Petugas</i> .....	29
<i>Gambar 2.31 Sequence Laporan Petugas</i> .....	29

Gambar 2.32 <i>Sequence Profile</i> Petugas .....	30
Gambar 2.33 <i>Sequence Logout</i> Petugas .....	31
Gambar 2.34 <i>Sequence Login</i> Admin .....	31
Gambar 2.35 <i>Sequence Konfirmasi Laporan</i> Admin .....	32
Gambar 2.36 <i>Sequence Warga</i> Admin .....	33
Gambar 2.37 <i>Sequence Laporan</i> Admin .....	33
Gambar 2.38 <i>Sequence Petugas</i> Admin .....	34
Gambar 2.39 <i>Sequence Tambah Petugas</i> Admin .....	34
Gambar 2.40 <i>Sequence Delete Petugas</i> Admin.....	35
Gambar 2.41 <i>Sequence Alat Damkar</i> Admin.....	36
Gambar 2.42 <i>Sequence Tambah Alat Damkar</i> Admin.....	36
Gambar 2.43 <i>Sequence Logout</i> Admin .....	37
Gambar 2.44 <i>Class Diagram</i> .....	38
Gambar 2.45 <i>Login</i> Warga.....	39
Gambar 2.46 <i>Register</i> Warga .....	39
Gambar 2.47 <i>Home</i> Warga.....	40
Gambar 2.48 <i>History</i> Warga .....	41
Gambar 2.49 <i>Informasi</i> Warga.....	41
Gambar 2.50 <i>Profile</i> Warga .....	42
Gambar 2.51 <i>Laporan</i> Warga .....	43
Gambar 2.52 <i>Login</i> Petugas .....	44
Gambar 2.53 <i>Home</i> Petugas .....	45
Gambar 2.54 <i>Informasi</i> Petugas .....	45
Gambar 2.55 <i>Profile</i> Petugas .....	46
Gambar 2.56 <i>Login</i> Admin .....	47
Gambar 2.57 <i>Dashboard</i> admin .....	48
Gambar 2.58 <i>Laporan</i> Masuk Admin.....	48
Gambar 2.59 <i>Laporan</i> Selesai .....	49
Gambar 2.60 <i>Warga</i> Admin .....	50
Gambar 2.61 <i>Alat Damkar</i> Admin .....	50
Gambar 2.62 Halaman <i>login</i> aplikasi warga .....	51
Gambar 2.63 Halaman <i>register</i> aplikasi warga.....	52

Gambar 2.64 Halaman home aplikasi warga .....	52
Gambar 2.65 Halaman Laporan aplikasi warga .....	53
Gambar 2.66 Halaman <i>history</i> aplikasi warga .....	54
Gambar 2.67 Halaman detail <i>history</i> aplikasi warga .....	54
Gambar 2.68 Halaman informasi aplikasi warga .....	55
Gambar 2.69 Halaman <i>user</i> aplikasi warga .....	56
Gambar 2.70 Halaman <i>login</i> aplikasi petugas.....	56
Gambar 2.71 Halaman <i>home</i> aplikasi petugas .....	57
Gambar 2.72 Halaman laporan aplikasi petugas.....	57
Gambar 2.73 Halaman informasi aplikasi petugas .....	58
Gambar 2.74 Halaman <i>user</i> aplikasi petugas .....	58
Gambar 2.75 Halaman <i>login</i> admin .....	59
Gambar 2.76 Halaman <i>dashboard</i> admin .....	59
Gambar 2.77 Halaman laporan masuk admin .....	60
Gambar 2.78 Halaman laporan selesai admin.....	60
Gambar 2.79 Halaman warga admin.....	61
Gambar 2.80 Halaman alat-alat damkar admin.....	61
Gambar 2.81 Halaman petugas admin .....	62

## **DAFTAR TABEL**

Tabel 2.1 Pengujian <i>Black Box</i> .....	63
--	----

## **DAFTAR LAMPIRAN**

Lampiran 1. Surat Kesedian Pembimbing 1 .....	A-1
Lampiran 2. Surat Kesedian Bimbingan 2 .....	A-2
Lampiran 4. Lembar Bimbingan Pembimbing 1.....	A-3
Lampiran 4. Lembar Bimbingan Pembimbing 2.....	A-6
Lampiran 5. Surat Pernyataan Pengajuan HKI .....	A-9
Lampiran 6. Surat Pengalihan HKI.....	A-11
Lampiran 7. Data Anggota Damkar Kota Tegal .....	A-12
Lampiran 8. Manual Book .....	A-13
Lampiran 9. Dokument Technical.....	A-14
Lampiran 10. Surat HKI Yang Terbit .....	A-15

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sektor telekomunikasi saat ini berkembang pesat karena perkembangan teknologi global, terutama di bidang telepon seluler dan internet[1]. Salah satunya adalah kemajuan aplikasi *mobile* berbasis Android. Perkembangan aplikasi Android belakangan ini telah membawa perubahan di berbagai bidang, termasuk bidang pelayanan masyarakat. Saat ini, kebanyakan orang sudah memiliki *smartphone* sendiri. Dengan berkembangnya teknologi, mencari tahu lokasi suatu tempat dengan menggunakan *smartphone* bukanlah hal yang mustahil[2].

Petugas pemadam kebakaran atau biasa disingkat Damkar merupakan pelayanan masyarakat yang bertugas memadamkan api, menyelamatkan, dan mengatasi bencana atau kejadian lainnya. Kejadian Seperti kebakaran, bencana alam, hewan liar, dan bantuan pertolongan pada tahun 2022 di Kota Tegal cukup banyak hampir mencapai 20 kasus. Sebagai penolong yang dibutuhkan pada saat terjadi bencana kebakaran, dan lainnya harus selalu siap siaga. namun tidak jarang menghubungi petugas pemadam kebakaran sangat sulit.

Masyarakat Kota Tegal menghadapi kendala ketika dalam melaporkan situasi, kondisi ataupun keadaan darurat[5]. Banyak masyarakat Kota Tegal yang merasa sulit untuk menghubungi petugas pemadam kebakaran di karenakan ketidakfahaman masyarakat cara menghubungi

pemadam kebakaran. Dan tidak jarang juga petugas pemadam kebakaran kesulitan untuk menuju ke titik lokasi karena tidak jelas informasi lokasi yang diberikan. Dengan semangkin pesatnya perkembangan teknologi harus dapat menfaatkanya untuk mempermudah kita dalam melakukan pelaporan keadaan darurat.

Teknologi yang tepat digunakan sebagai media pencari lokasi pada smartphone adalah *Global Positioning System* (GPS) yang sudah dapat digunakan pada *smartphone* dan salah satunya menggunakan platform android[3]. Oleh karena itu, aplikasi pemadam kebakaran yang akan dibuat dilengkapi dengan navigasi *Global Positioning System* (GPS) untuk memudahkan masyarakat dalam melakukan panggilan pemadam kebakaran jika terjadi bencana kebakaran atau kejadian lainnya serta memudahkan petugas pemadam kebakaran untuk sampai ke titik yang dituju dengan cepat dan tepat. Aplikasi ini diberi nama “*Applikasi Location Based Service* (LBS) Pemanggil Pemadam Kebakaran”. Aplikasi pemadam kebakaran ini dibangun menggunakan Android Studio dan menggunakan Google Maps Api untuk menampilkan peta dan rute dari suatu tempat, menggunakan *Global Positioning System* (GPS) untuk mengetahui posisi pengguna yang melakukan panggilan saat ini.

## 1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat aplikasi di wilayah Kota Tegal yang berguna untuk memudahkan dalam menghubungi pemadam kebakaran sehingga dapat langsung membantu keadaan darurat.

### **1.3 Manfaat Penelitian**

#### 1. Peneliti

Menerapkan ilmu yang didapat di Politeknik Harapan Bersama dan mengetahui pengujian kualitas *software* berbasis *website* dan *mobile*.

#### 2. Masyarakat

Mempermudah masyarakat dalam mendapatkan pertolongan secara cepat untuk dievakuasi dengan lokasi terjadinya kejadian gawat darurat.

#### 3. Pemadam Kebakaran

Membantu mewujudkan bentuk implementasi kewajiban pemadam kebakaran dalam memberikan pelayanan kepada masyarakat.

#### 4. Politeknik Harapan Bersama

Menambah kepustakaan sebagai referensi dan pembanding untuk penelitian yang sejenis

### **1.4 Batasan Penelitian**

Dalam pembuatan Aplikasi ini dibatasi masalah pada :

1. Aplikasi ini hanya digunakan untuk keperluan memanggil pemadam kebakaran Kota Tegal.
2. Aplikasi yang dibuat berbasis *mobile* dengan sistem operasi android pada *user* masyarakat dan petugas pemadam kebakaran.
3. Aplikasi yang dibuat ditujukan hanya bagi masyarakat Kota Tegal.

### **1.5 Tinjauan Pustaka**

Penelitian yang sejenis yang di gunakan dalam penelitian yang berjudul “Aplikasi *Location Based Service* (LBS) Pertolongan Cepat Menuju

Rumah Sakit Terdekat (Studi Kasus: Kota Tegal)”. Pertolongan korban yang mengalami kondisi gawat darurat seringkali terhambat dalam proses evakuasi menuju rumah sakit. Faktor yang memperlambat proses evakuasi diantaranya kendaraan yang digunakan untuk mengevakuasi korban yakni kendaraan milik pribadi yang dimintai tolong untuk mengevakuasi korban ke rumah sakit. Ketersediaan kendaraan yang tidak pasti, membuat proses evakuasi lebih lambat. Faktor lain yang menghambat yakni kurangnya pengetahuan pengendara yang mengevakuasi korban tentang lokasi rumah sakit terdekat dari lokasi kejadian gawat darurat. Selain itu, rumah sakit yang terdekat tidak menjamin dapat menerima korban dikarenakan terbatasnya penggunaan IGD (Instalasi Gawat Darurat) dan pelayanan kesehatan yang dapat dilakukan di IGD sesuai kemampuan masing-masing rumah sakit. Dengan adanya Aplikasi *Location Based Service* (LBS) Pertolongan Cepat Menuju Rumah Sakit Terdekat (Studi Kasus : Kota Tegal) sebagai upaya untuk menyediakan aplikasi yang dapat menghubungi rumah sakit terdekat dengan cepat untuk menolong korban kejadian gawat darurat di Kota Tegal. Dengan aplikasi ini dapat meningkatkan kecepatan pelayanan rumah sakit dalam menangani korban kejadian gawat darurat dengan dukungan sistem yang terintegrasi antara warga dan pihak rumah sakit[4].

Penelitian yang sejenis yang di gunakan dalam penelitian yang berjudul “Aplikasi *Frontend* Pada *Emergency Call* Berbasis *Mobile Hybrid* di Kota Tegal”. Masyarakat Kota Tegal menghadapi kendala ketika dalam melaporkan situasi, kondisi ataupun keadaan darurat. Masyarakat kurang

paham terhadap mekanisme pelaporan kejadian darurat. Selain itu tidak semua masyarakat mengetahui nomor telepon untuk menghubungi pihak-pihak instansi terkait dalam melakukan pelaporan kejadian tersebut. Dengan demikian akan berdampak pada keterlambatan penanganan korban yang berujung pada keadaan yang lebih buruk hingga dapat menyebabkan kematian, karena terlambatnya masyarakat dalam melaporkan kejadian darurat. Untuk membantu masyarakat Kota Tegal dalam melakukan kegiatan pelaporan tanggap darurat maka dibutuhkan suatu sistem yang dapat melakukan pelaporan kejadian darurat secara cepat, tepat dan akurat. Aplikasi *Frontend Emergency Call* dibuat dengan menggunakan *Ionic Framework*. Sistem kerja aplikasi dengan cara mengirimkan data pelaporan kejadian berupa gambar dan alamat lokasi pelapor. Hasil pengujian aplikasi dengan metode blackbox diperoleh hasil bahwa tidak ditemukan adanya error, di mana aplikasi dapat menampilkan sesuai dengan fungsi-fungsinya. Dari hasil pengujian tersebut aplikasi siap di implementasikan untuk masyarakat Kota Tegal[5].

Penelitian yang sejenis yang di gunakan dalam penelitian yang berjudul “Aplikasi *Location Based Service* “TOLONG.IN” Pada Platform Android”. Masyarakat membutuhkan satu sama lain sebagai makhluk sosial dimana perilaku menolong merupakan kunci utama dalam membangun sifat kekeluargaan dalam bentuk permasalahan apapun. Sikap tolong-menolong ini mengilhami masyarakat untuk menggunakannya sebagai sumber penghasilan. Masyarakat menyadari bahwa semakin banyak orang yang

membutuhkan pertolongan atau jasa dari orang lain sehingga mereka menggunakan peluang ini untuk meningkatkan perekonomiannya. Berdasarkan hasil dari pembuatan aplikasi berbasis android “TOLONG.IN” dan hasil dari kuesioner penggunaan aplikasi”. Perancangan dan pengembangan aplikasi “TOLONG.IN” dapat membantu user mendapatkan informasi mengenai penyedia jasa yang dibutuhkan oleh user. Aplikasi “TOLONG.IN” dapat memudahkan komunikasi antara penyedia jasa dan pencari jasa dengan adanya fitur chat dan teknologi GPS[6].

Penelitian yang sejenis yang di gunakan dalam penelitian yang berjudul “ Aplikasi Pengaduan *Bullying* dan Kekerasan Anak Serta Perempuan Menggunakan *Location Based Service* “.Menyelesaikan permasalahan Bullying dan kekerasan pada perempuan dan anak-anak yang terjadi disekolah dan lingkungan masyarakat. Permasalahan yang terjadi adanya pemecahan untuk itu dibuat aplikasi pengaduan bullying dan kekerasaan pada kanak-kanak serta perempuan menggunakan Location Based Service yang memungkin si target bisa melakukan pelaporan sendiri melewati prangkat seluler dan otomatis lokasi akan terdeteksi dengan sendirinya sehingga dapat dilakukan tindakan pencegahan untuk mencegah efek yang lebih besar bagi sikorban. Berdasarkan hasil pengujian yang dilakukan oleh user hampir 83% menyatakan aplikasi ini menarik dan dibutuhkan, menggunakan Location Based Service yang memungkin si target bisa melakukan pelaporan sendiri melewati prangkat seluler dan otomatis lokasi akan terdeteksi dengan sendirinya, Kelebihan LBS, rata-rata terjadi delay

sekitar 2 detik dan layanan LBS tetap berfungsi walaupun pengguna berada di dalam gedung dan pengaruh medan elektromagnetik lain yang tidak terlalu besar, kekurangan jangkauan area sangat bergantung pada jangkauan jaringan selular dan aplikasi dapat dijalankan minimal pada versi android Kitkat sedangkan dengan versi di bawah Kitkat aplikasi tidak dapat dijalankan[7].

Penelitian yang sejenis yang di gunakan dalam penelitian yang berjudul “*Android Based Application for Searching Indekos In Pasar Minggu Jakarta South With Location Based Service (LBS) Method*”. Seringkali ketersediaan jumlah kos tidak sebanding dengan banyaknya pencari kos, maka pencari kos harus mencari informasi lebih cepat agar mendapatkan tempat tinggal yang sesuai dengan apa mereka ingin. Pemilik kost mempromosikan kostnya di cara konvensional yaitu dari mulut ke mulut, menyebarkan brosur, atau memasang pengumuman di tempat umum. Berdasarkan permasalahan tersebut, Dibutuhkan aplikasi pencarian kost berbasis android yaitu sebuah aplikasi sistem yang dapat menjadi wadah bagi masyarakat luar kota yang merasa kebingungan dalam mencari kost yang sesuai dengan keinginannya, dan dapat membantu pemilik kost mempromosikan kos-kosan mereka secara online melalui smartphone berbasis android. Tujuan penelitian adalah merancang aplikasi pencarian kost di Pasar Minggu, Jakarta Selatan, metode yang digunakan adalah air terjun metodologi, yaitu metode sistem yang diawali dengan kebutuhan tahap analisis, kemudian perancangan sistem, implementasi aplikasi pengembangan, dan pengguna dilakukan. Sistem aplikasi ini adalah

dikembangkan menggunakan bahasa pemrograman Java dan memanfaatkan Lokasi Layanan Berbasis dan Google Maps. Hasil penelitian tersebut menghasilkan aplikasi pencarian kos yang memudahkan pengguna untuk menemukannya kost sesuai kebutuhan dan juga aplikasi ini bisa memberikan informasi tentang fasilitas kos dan aplikasi ini juga bisa mempermudah pemilik kos untuk mempublikasikan kosnya[8].

### **1.6 Data Penelitian**

Dalam penelitian ini, data dan informasi yang di butuhkan di ambil langsung dari data damkar Kota Tegal. Data berupa data kejadian peristiwa pada tahun 2022 di Kota Tegal dan data anggota pemadam kebakaran Kota Tegal. Data anggota pemadam kebakaran yang berisi Nama, NIP, Tempat, Tanggal Lahir, Pangkat / Gol.Ruang, Pendidikan, Keterangan dan data peristiwa kejadian yang berisi Tanggal, Kategori, Keterangan, dan Lokasi.

## **BAB II**

### **PRODUK**

#### **2.1 Perancangan Sistem**

Agar sistem yang di rancang dapat berjalan dengan baik, maka perlu kebutuhan minimum dilakukan analisis terhadap kebutuhan sistem yang baru, baik berupa kebutuhan perangkat keras (hardware), maupun perangkat lunak (software). Peralatan yang digunakan dalam penelitian ini di spesifikasikan dalam Perangkat Keras ( Hardware) dan Perangkat Lunak (Software) sebagai berikut :

1. Perangkat Keras (Hardware)
  - a. Laptop ASUS X411
  - b. SSD 120
  - c. RAM 4 Giga
  - d. Prosesor AMD A6
2. Perangkat Lunak (Software)
  - a. Google Chrome berfungsi sebagai *web browser*
  - b. Xampp berfungsi sebagai *web server*
  - c. Visual Studio Code berfungsi sebagai *text editor*
  - d. Android Studio berfungsi sebagai *text editor*
  - e. Emulator Android Studio sebagai emulator untuk menjalankan program

##### **2.1.1 Desain Sistem**

Perancangan sistem ini bertujuan untuk mengetahui

hubungan antar komponen pendukung sistem yang akan dirancang untuk memberikan gambaran kepada pengguna sistem tentang informasi yang dihasilkan oleh perancangan sistem. Tata letak ini dirancang dalam *use case diagram, activity diagram, dan sequence diagram*.

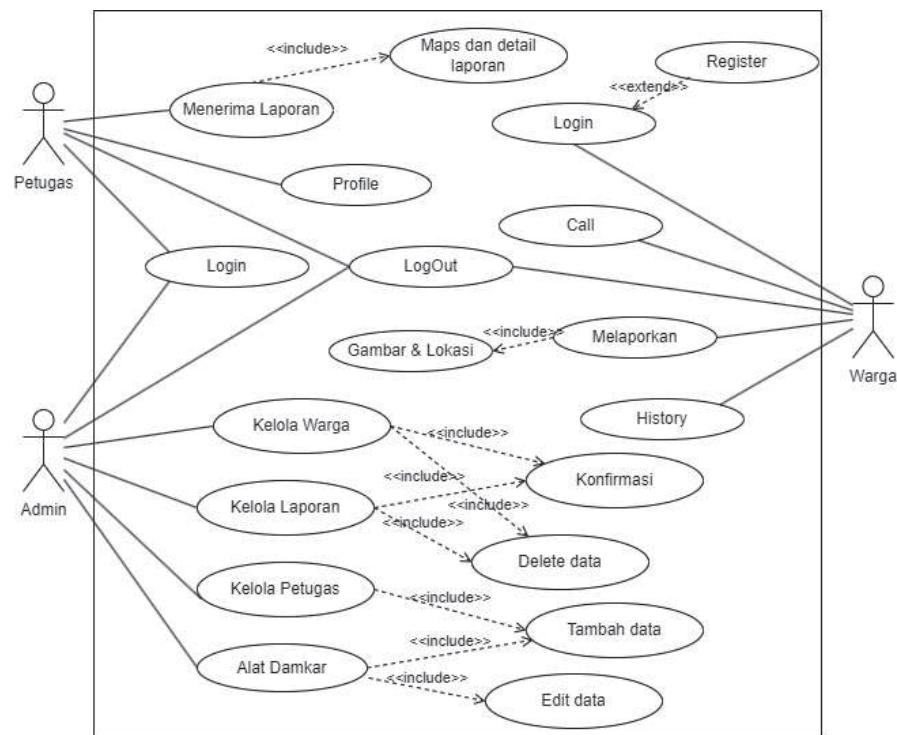
### 2.1.2 Pemodelan Sistem UML

Unified Modelling Language merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [9]. Beberapa liberator menyebutkan UML menyediakan 9 jenis diagram gabungan. Misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Jenis diagram-diagram tersebut adalah diagram kelas (class diagram), diagram paket (package diagram), diagram use-case(use case diagram), diagram interaksi dan sequence(sequence diagram), diagram komunikasi (communication diagram), diagram komponen (componentdiagram), diagram deployment(deployment diagram) [10]. Berikut ini adalah penjelasan dari masing-masing diagram :

#### 2.2.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuakan sistem informasi yang akan dibuat. Use case digunakan untuk

mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [11]. Dalam use case diagram perancangan aplikasi terdapat 3 aktor yaitu masyarakat, petugas, dan Admin. Warga dapat melakukan kegiatan pelaporan kejadian darurat, Petugas Dapat menerima laporan dari admin dan melihat detail laporan yang di laporakan dan melihat rute menuju lokasi kejadian, Admin dapat mengelola data laporan, kelola data petugas dan kelola data warga. Kegiatan dari aktor telah digambarkan ke dalam usecase diagram pada Gambar 2.1 sebagai berikut :



Gambar 2.1 Use Case Diagram

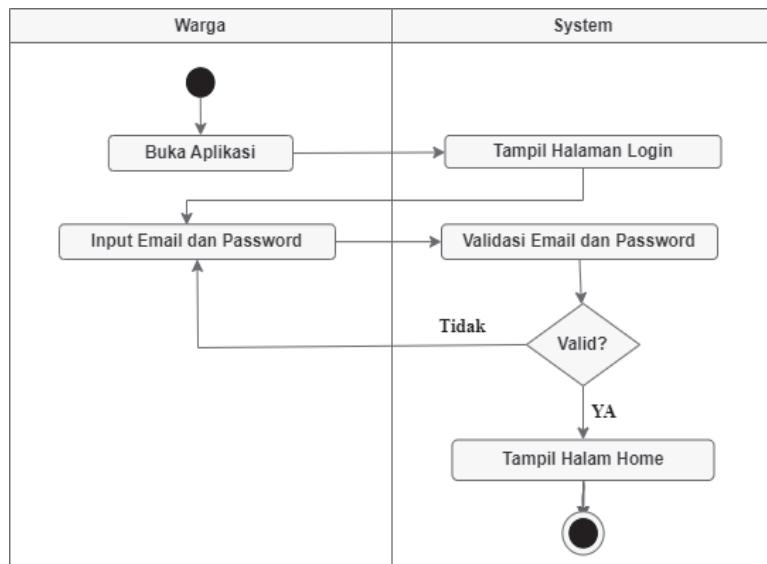
### 2.2.2 Activity Diagram

Activity Diagram digunakan untuk mengetahui sebuah alur

proses aliran kerja dari sebuah sistem yang ada pada perangkat lunak agar dapat direncanakan dengan baik di awal pengembangan sistem. Activity diagram juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi [12]. Berikut activity diagram dalam pengembangan aplikasi ini, sebagai berikut:

### 1. *Activity Warga Login Warga*

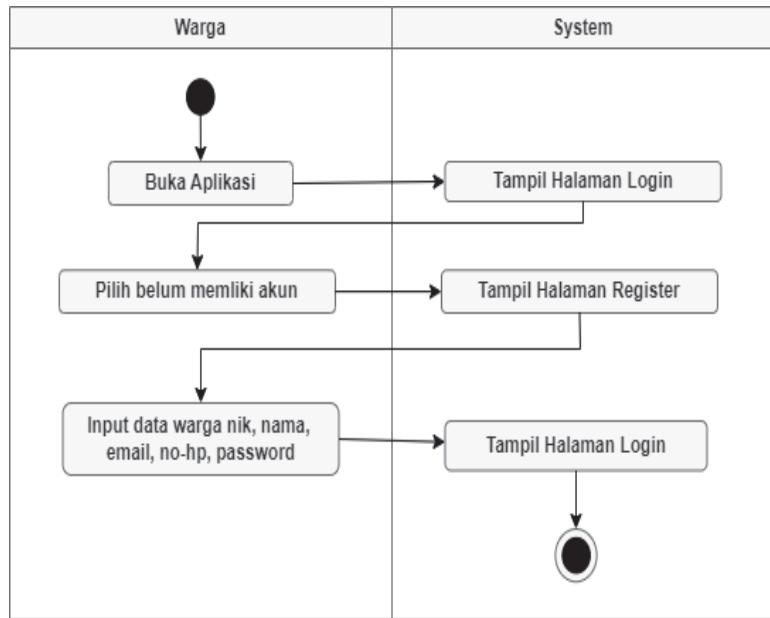
*Activity Warga Login Warga* menjelaskan aktivitas dari proses untuk masuk kedalam aplikasi warga sebelum melakukan pelaporan. *Activity Warga Login Warga* terdapat pada Gambar 2.2.



Gambar 2.2 *Activity Warga Login Warga*

### 2. *Activity Warga Register Warga*

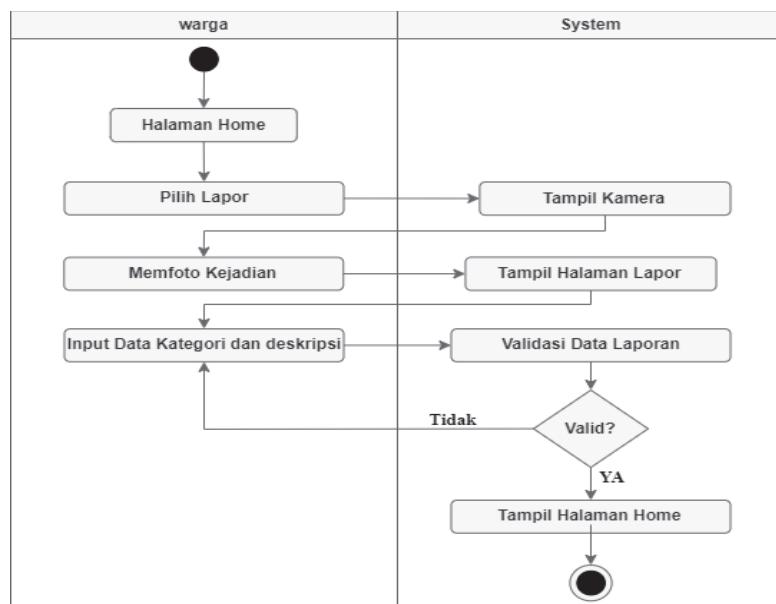
*Activity Warga Register Warga* menjelaskan aktivitas dari proses pembuatan akun atau pembuatan user. *Activity Warga Register Warga* terdapat pada Gambar 2.3.



Gambar 2.3 *Activity* Warga Register Warga

### 3. *Activity* Laporan warga

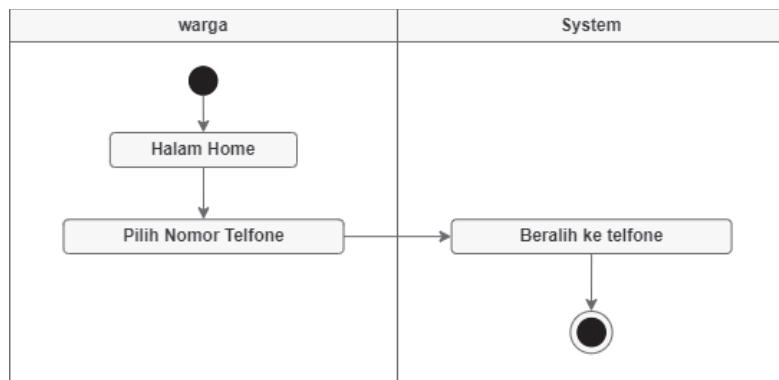
*Activity* Laporan Warga menjelaskan aktivitas dari proses untuk melakukan pelaporan melalui aplikasi. *Activity* Laporan Warga terdapat pada Gambar 2.4.



Gambar 2.4 *Activity* Laporan warga

#### 4. *Activity Telpone warga*

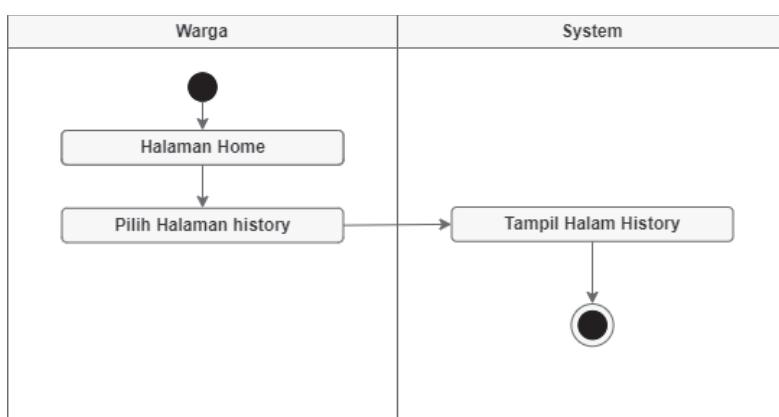
*Activity Telphon Warga* menjelaskan aktivitas dari proses untuk melakukan pelaporan secara telphon seluller. *Activity Telephone Warga* terdapat pada Gambar 2.5.



Gambar 2.5 *Activity Telpone warga*

#### 5. *Activity History warga*

*Activity History Warga* menjelaskan aktivitas dari proses untuk melihat laporan yang sudah pernah kita laporankan. *Activity Histoty Warga* terdapat pada Gambar 2.6.

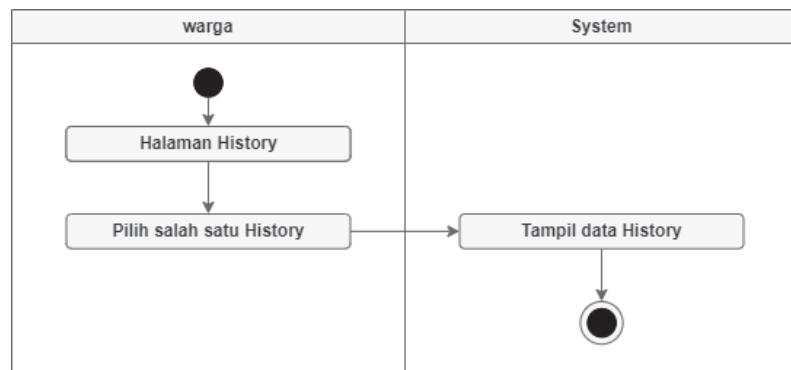


Gambar 2.6 *Activity History warga*

#### 6. *Activity Detail History warga*

*Activity Detail History Warga* menjelaskan aktivitas dari

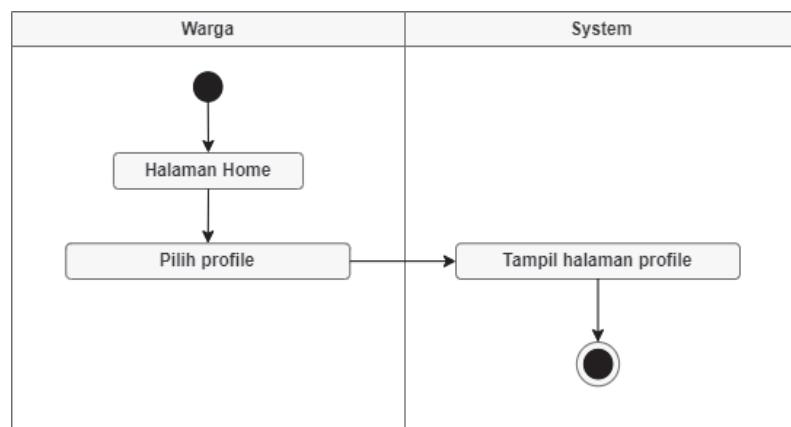
proses untuk melihat detail dari isi laporan yang sudah pernah kita laporankan. *Activity Detail Histoty* Warga terdapat pada Gambar 2.7.



Gambar 2.7 *Activity Detail History* warga

#### 7. *Activity Profile* warga

*Activity Profile* Warga menjelaskan aktivitas dari proses untuk melihat data dari akun atau *user* yang masuk kedalam aplikasi. *Activity Profile* Warga terdapat pada Gambar 2.8.

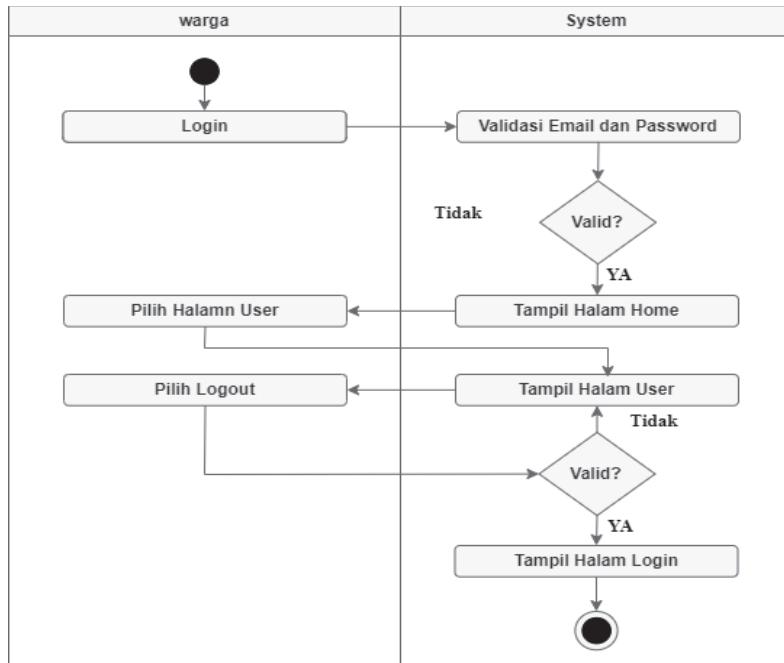


Gambar 2.8 *Activity Profile* warga

#### 8. *Activity Logout* warga

*Activity Logout* Warga menjelaskan aktivitas dari proses untuk keluarkan akun atau *user* dari aplikasi. *Activity Logout*

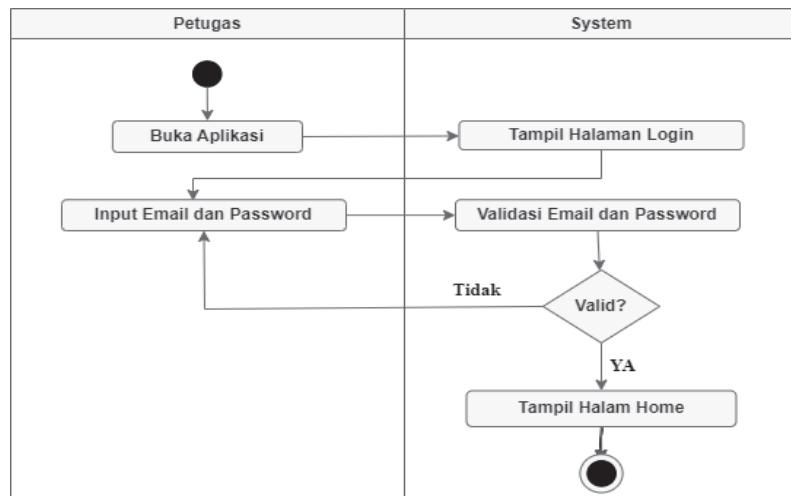
Warga terdapat pada Gambar 2.9.



Gambar 2.9 *Activity Logout* warga

#### 9. *Activity Login* Petugas

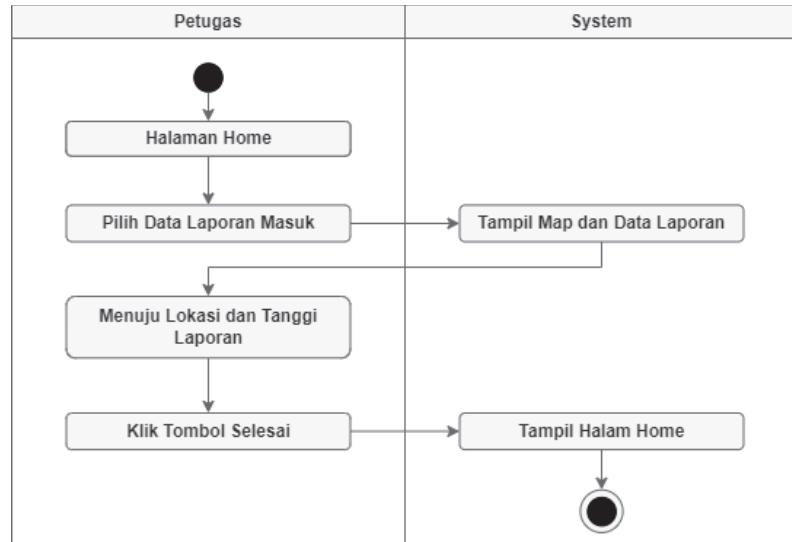
*Activity Login* Petugas menjelaskan aktivitas dari proses untuk masuk kedalam aplikasi petugas. *Activity Login* Petugas terdapat pada Gambar 2.10.



Gambar 2.10 *Activity Login* Petugas

## 10. *Activity* Laporan Petugas

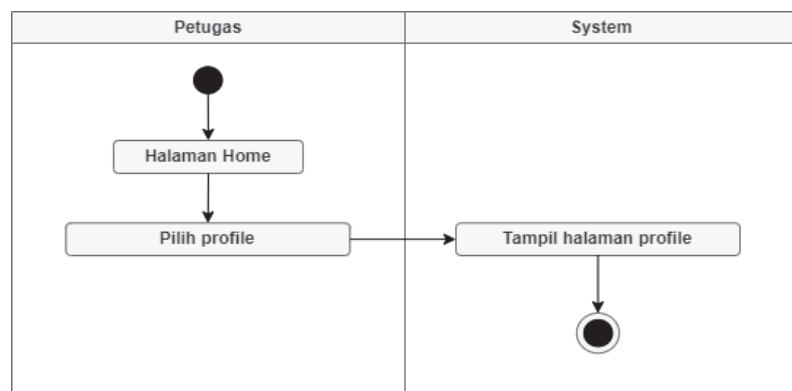
*Activity* Laporan Petugas menjelaskan aktivitas dari proses untuk melakukan penanganan laporan dari warga. *Activity* Laporan Petugas terdapat pada Gambar 2.11.



Gambar 2.11 *Activity* Laporan Petugas

## 11. *Activity Profile* Petugas

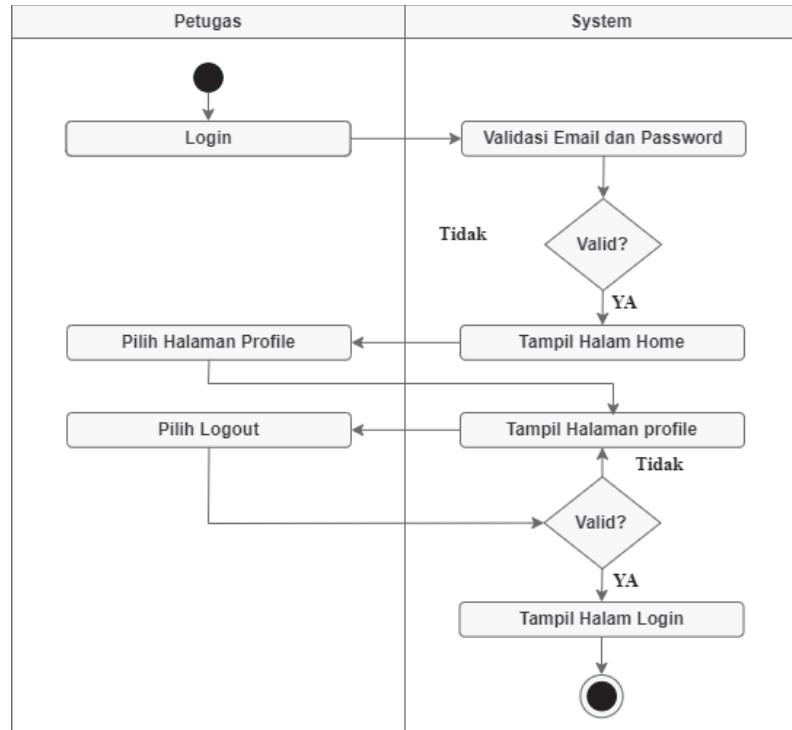
*Activity Profile* Petugas menjelaskan aktivitas dari proses untuk melihat data dari akun atau *user* yang masuk kedalam aplikasi. *Activity Profile* Petugas terdapat pada Gambar 2.12.



Gambar 2.12 *Activity Profile* Petugas

### 12. *Activity Logout* Petugas

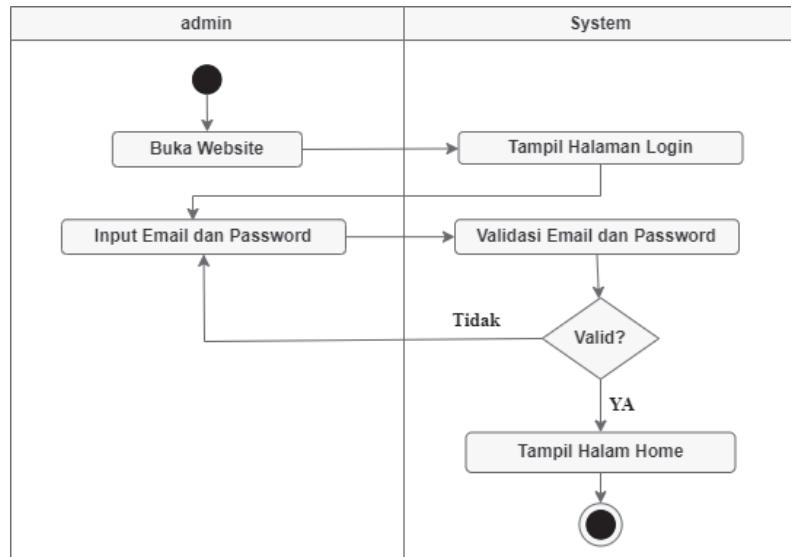
*Activity Logout* Petugas menjelaskan aktivitas dari proses untuk keluarkan akun atau *user* dari aplikasi. *Activity Logout* Petugas terdapat pada Gambar 2.13.



Gambar 2.13 *Activity Logout* Petugas

### 13. *Activity Login* Admin

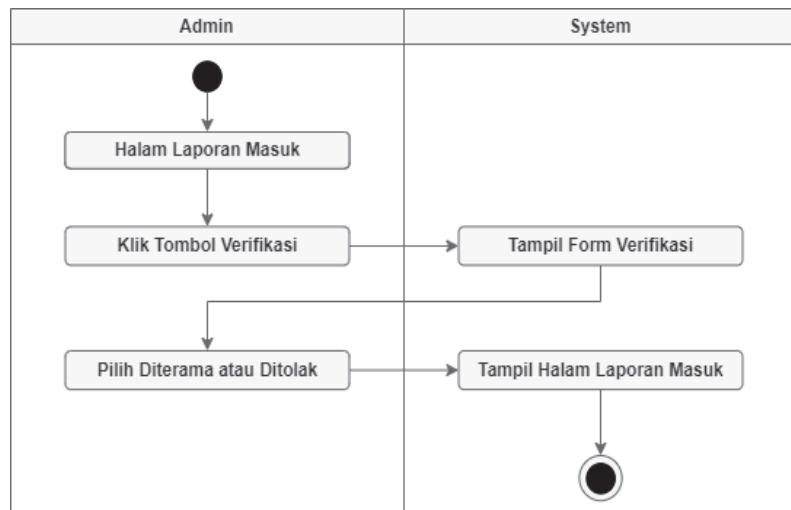
*Activity Login* Admin menjelaskan aktivitas dari proses untuk masuk kedalam *website* Admin. *Activity Login* Admin terdapat pada Gambar 2.14.



Gambar 2.14 *Activity Login Admin*

#### 14. *Activity Laporan Masuk Admin*

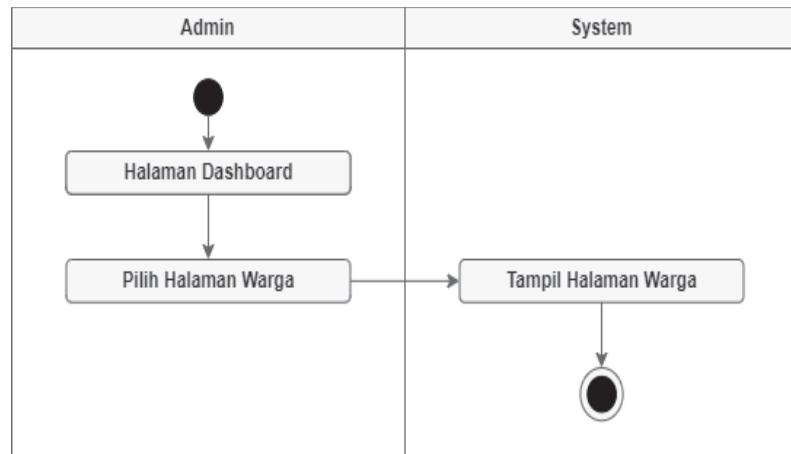
*Activity Laporan Masuk Admin* menjelaskan aktivitas dari proses untuk melakukan konfirmasi laporan yang masuk dari warga. *Activity Laporan Masuk Admin* terdapat pada Gambar 2.15.



Gambar 2.15 *Activity Laporan Masuk Admin*

### 15. *Activity* Warga Admin

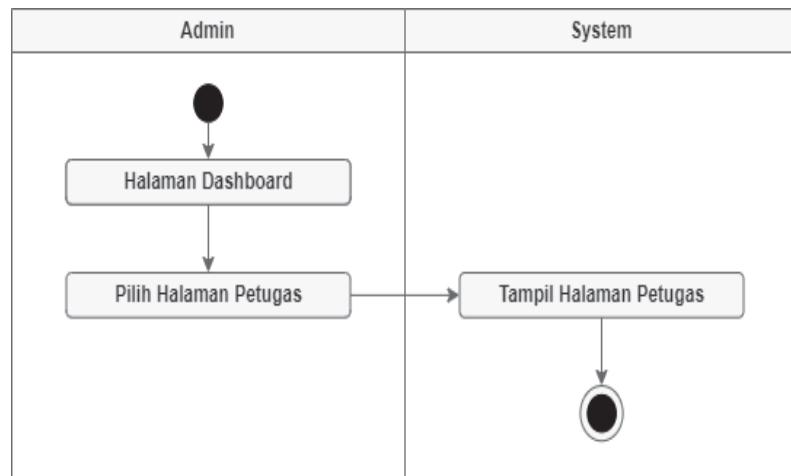
*Activity* Warga Admin menjelaskan aktivitas dari proses untuk melihat data warga yang sudah melakukan *register*. *Activity* Warga Admin terdapat pada Gambar 2.16.



Gambar 2.16 *Activity* Warga Admin

### 16. *Activity* Laporan Admin

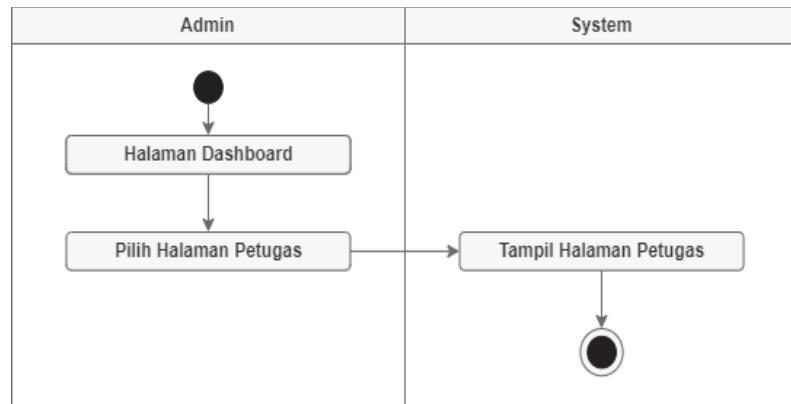
*Activity* Laporan Admin menjelaskan aktivitas dari proses untuk melihat data laporan yang sudah selesai di tangani. *Activity* Laporan Admin terdapat pada Gambar 2.17



Gambar 2.17 *Activity* Laporan Admin

### 17. *Activity Petugas Admin*

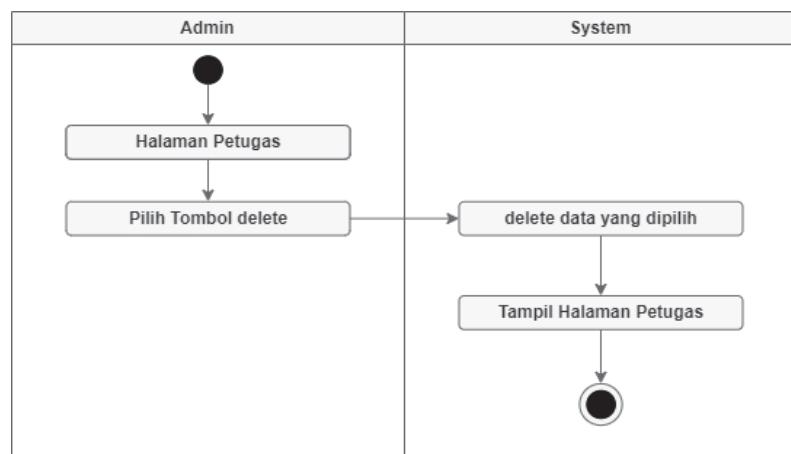
*Activity Petugas Admin* menjelaskan aktivitas dari proses untuk melihat data petugas. *Activity Petugas Admin* terdapat pada Gambar 2.18.



Gambar 2.18 *Activity Petugas Admin*

### 18. *Activity Delete Petugas Admin*

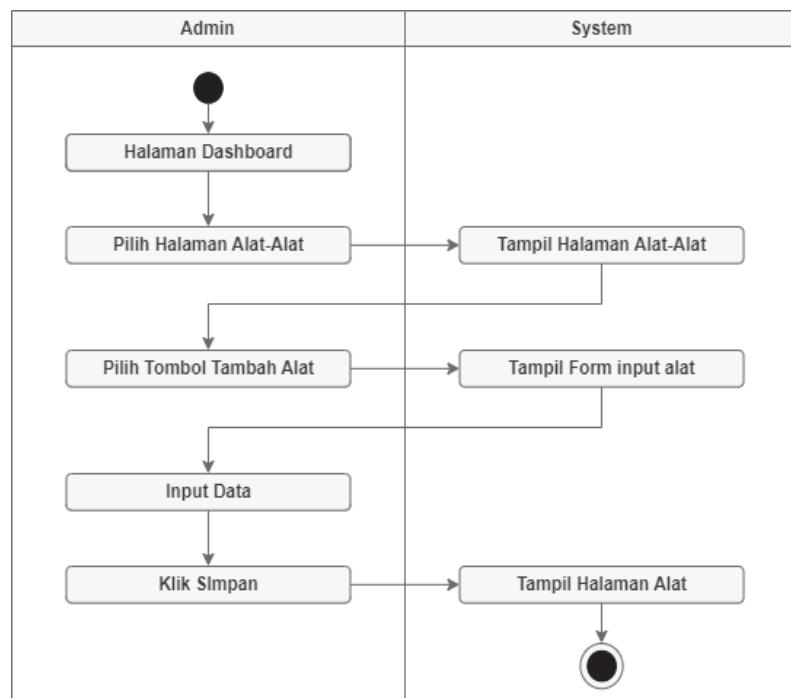
*Activity Delete Petugas Admin* menjelaskan aktivitas dari proses untuk melakukan penghapusan data petugas yang sudah tidak aktif. *Activity Delete Petugas Admin* terdapat pada Gambar 2.19.



Gambar 2.19 *Activity Delete Petugas Admin*

### 19. *Activity* Tambah Alat Admin

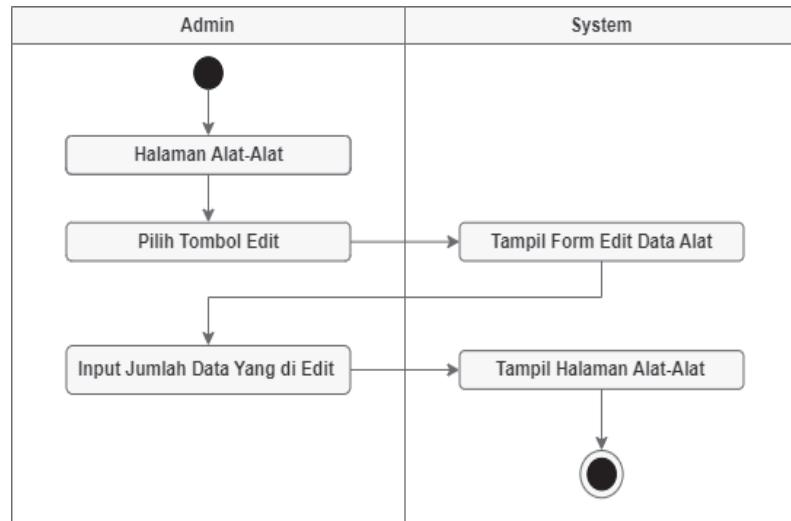
*Activity* Tambah Alat Admin menjelaskan aktivitas dari proses untuk melakukan penambahan data alat yang di miliki pemadam kebakaran. *Activity* Tambah Alat Admin terdapat pada Gambar 2.20.



Gambar 2.20 *Activity* Tambah Alat Admin

### 20. *Activity* Edit Alat Admin

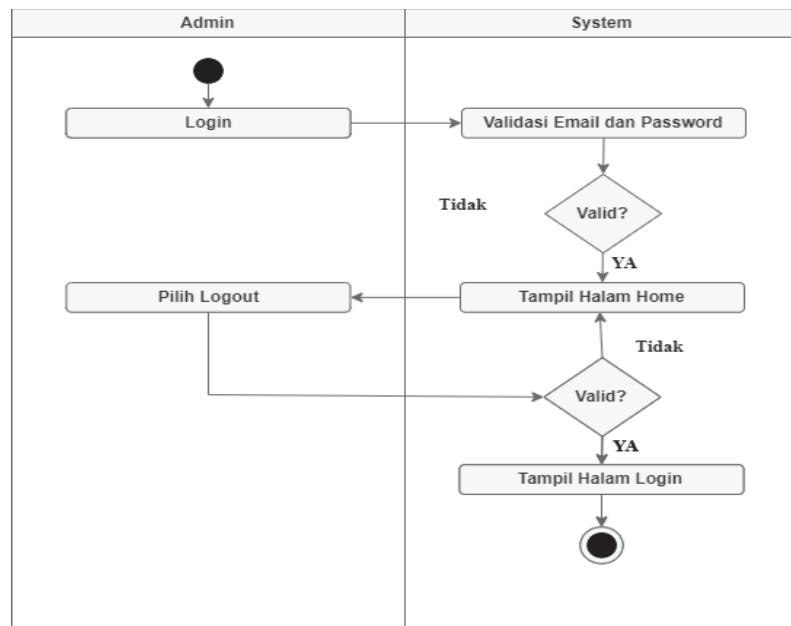
*Activity* Edit Alat Admin menjelaskan aktivitas dari proses untuk melakukan edit jumlah data alat yang di miliki pemadam kebakaran. *Activity* Edit Alat Admin terdapat pada Gambar 2.21.



Gambar 2.21 *Activity Edit Alat Admin*

### 21. *Activity Logout Admin*

*Activity Logout Admin* menjelaskan aktivitas dari proses untuk keluarkan akun atau *user* dari aplikasi. *Activity Logout Admin* terdapat pada Gambar 2.13.



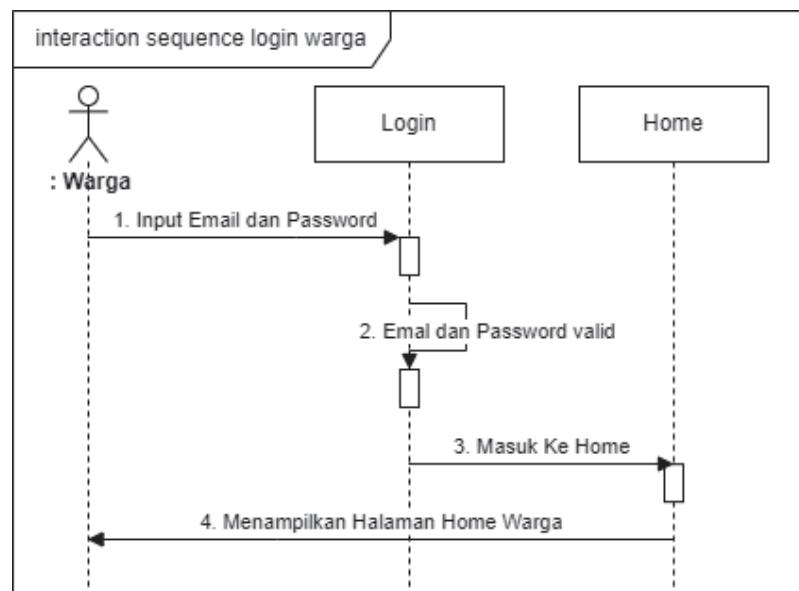
Gambar 2.22 *Activity Logout Admin*

### 2.2.3 Sequence diagram

Sequence diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu [13]. Sequence diagram digunakan untuk menjelaskan dan menampilkan interaksi antar objek dalam sebuah sistem secara terperinci.

#### 1. *Sequence Login Warga*

*Sequence Login Warga* menjelaskan alur aktor ketika masuk kedalam aplikasi. *Sequence Login Warga* terdapat pada Gambar 2.23.

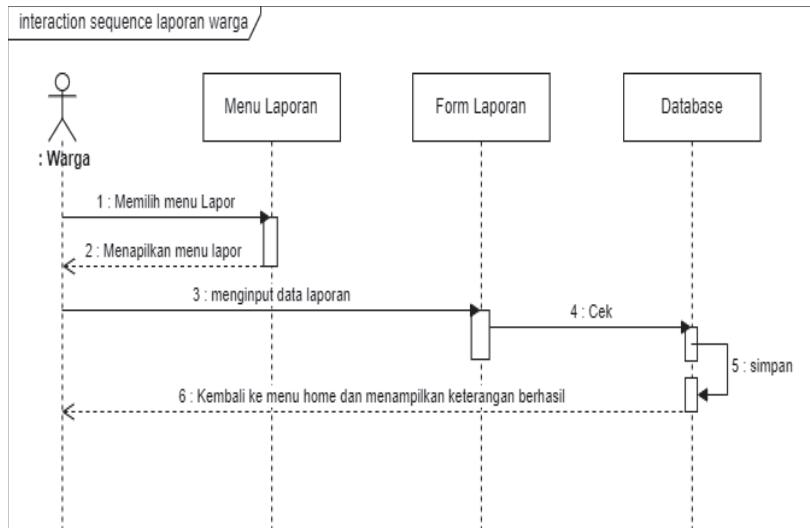


Gambar 2.23 *Sequence Login Warga*

#### 2. *Sequence Laporan Warga*

*Sequence Laporan Warga* menjelaskan alur aktor ketika melakukan pelaporan kejadian melalui aplikasi. *Sequence*

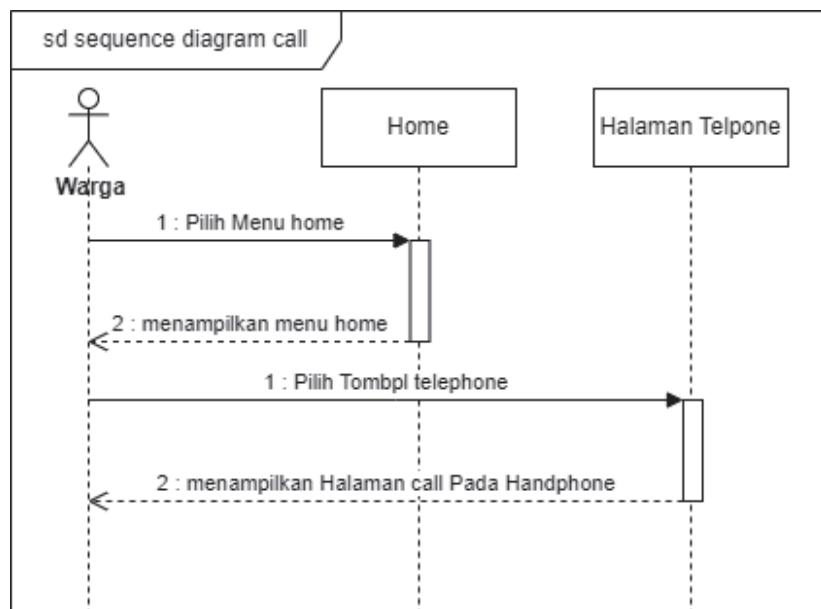
Laporan Warga terdapat pada Gambar 2.24.



Gambar 2.24 *Sequence Laporan Warga*

### 3. *Sequence Telphon Warga*

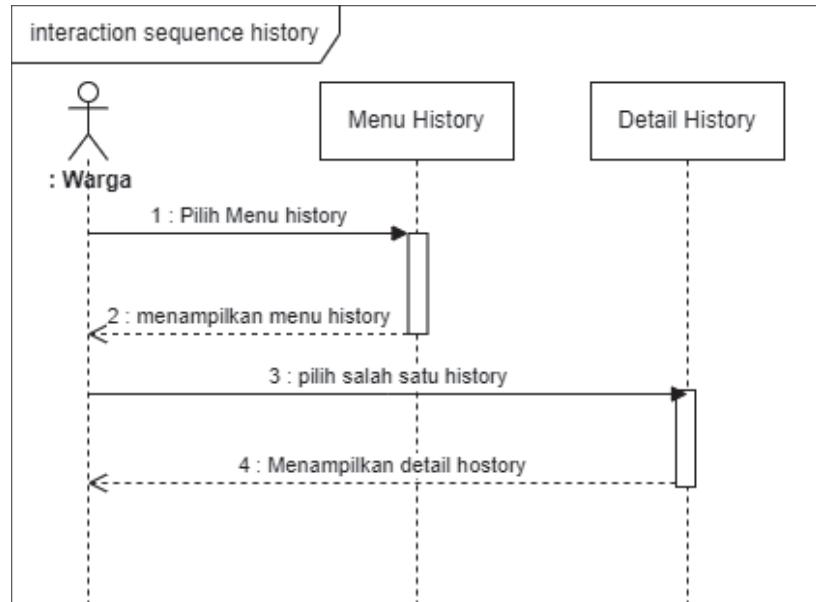
*Sequence Telphon Warga* menjelaskan alur aktor ketika melakukan pelaporan kejadian melalui telephone seluler. *Sequence Telphon Warga* terdapat pada Gambar 2.25.



Gambar 2.25 *Sequence Telphon Warga*

#### 4. *Sequence History* Warga

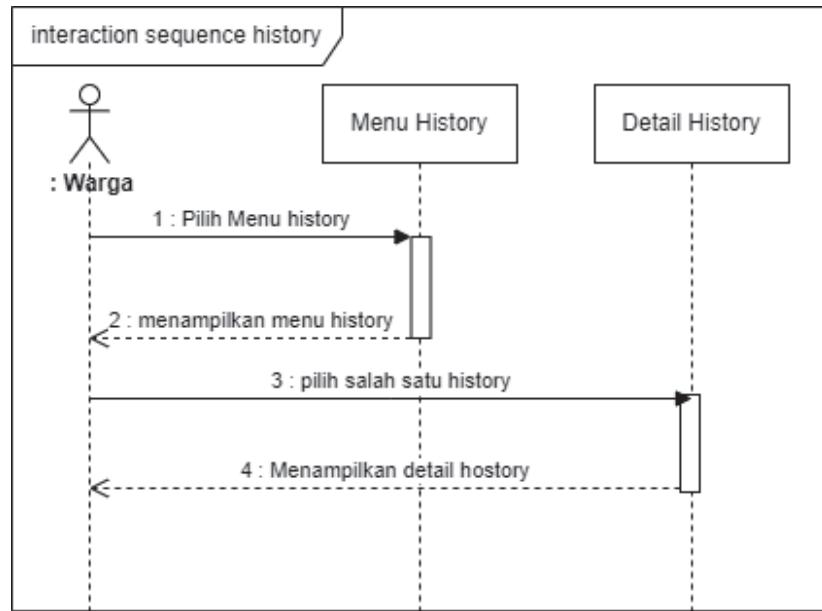
*Sequence History* Warga menjelaskan alur aktor ketika melihat data yang pernah kita laporakan. *Sequence History* Warga terdapat pada Gambar 2.26.



Gambar 2.26 *Sequence History* Warga

#### 5. *Sequence Detail History* Warga

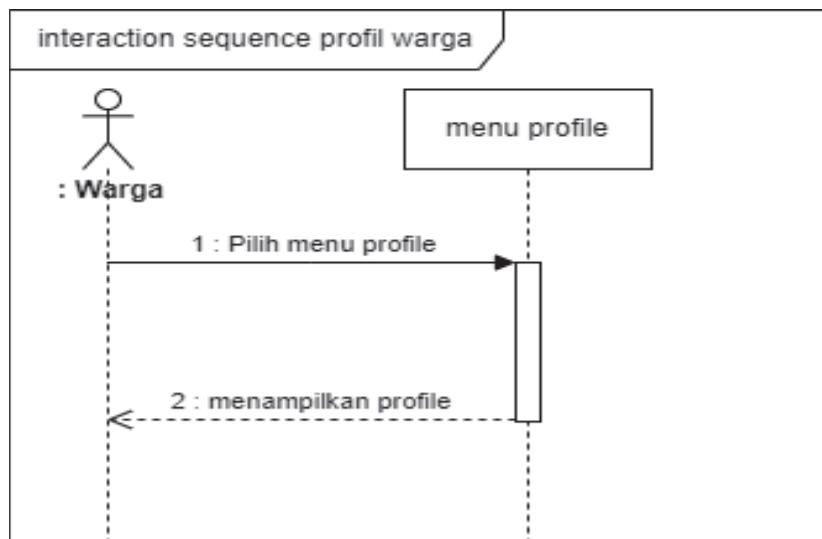
*Sequence Detail History* Warga menjelaskan alur aktor ketika melihat detail data yang pernah kita laporakan. *Sequence Detail History* Warga terdapat pada Gambar 2.27.



Gambar 2.27 *Sequence Detail History Warga*

#### 6. *Sequence Profile Warga*

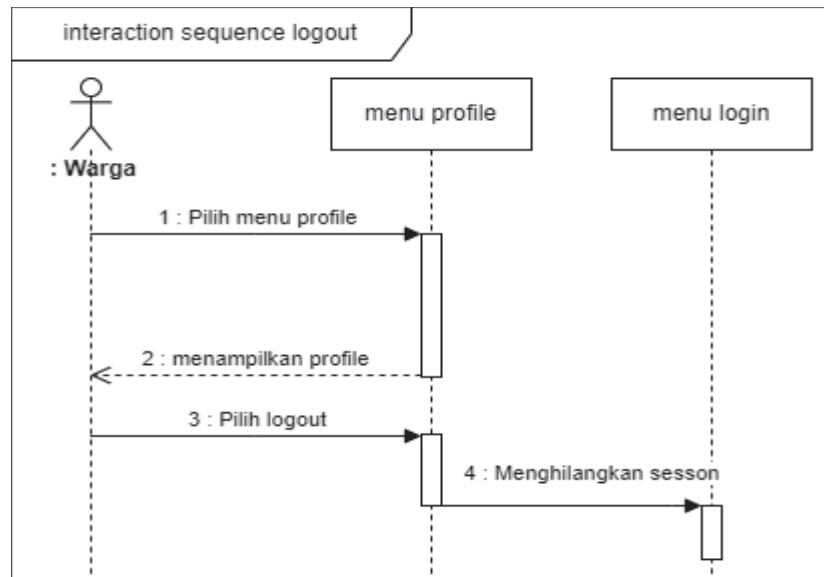
*Sequence Profile Warga* menjelaskan alur aktor ketika melihat data akun atau *user* yang masuk di aplikasi. *Sequence Profile Warga* terdapat pada Gambar 2.28.



Gambar 2.28 *Sequence Profile Warga*

### 7. *Sequence Logout* warga

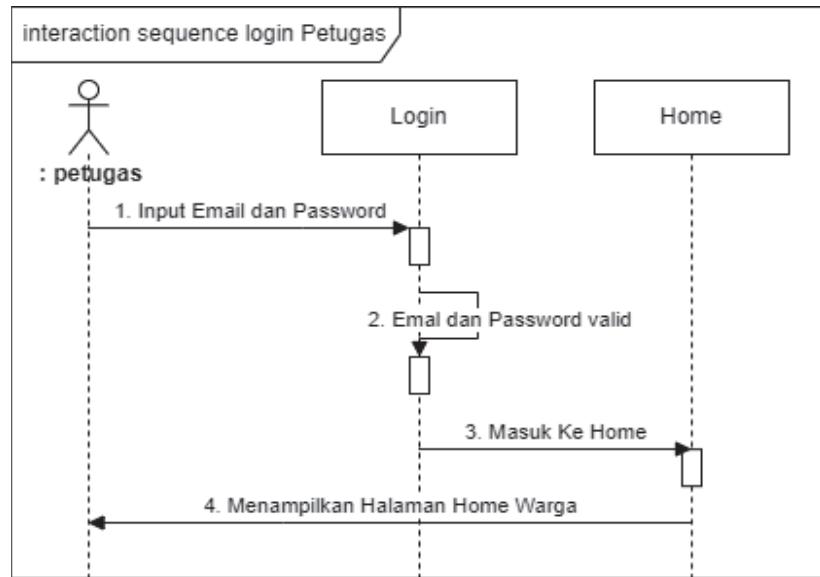
*Sequence Logout* Warga menjelaskan alur aktor ketika keluar dari aplikasi. *Sequence Logout* Warga terdapat pada Gambar 2.29.



Gambar 2.29 *Sequence Logout* warga

### 8. *Sequence Login* Petugas

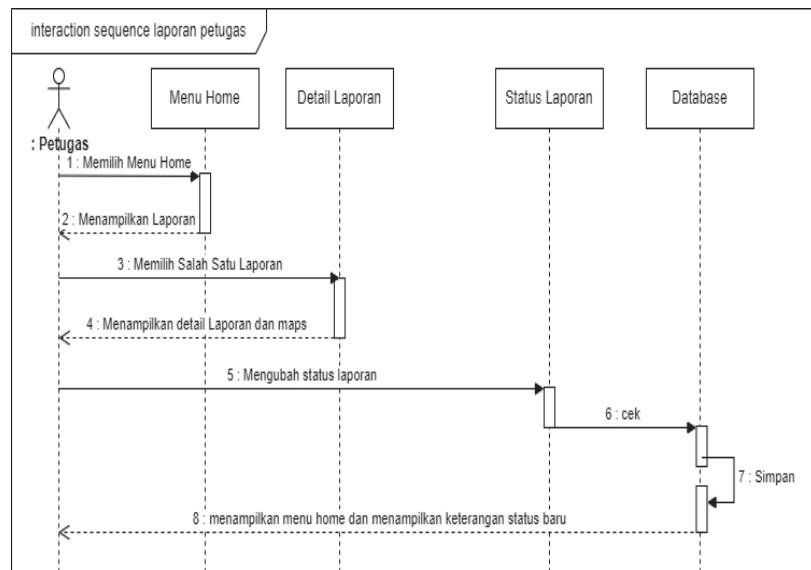
*Sequence Login* Petugas menjelaskan alur aktor ketika masuk kedalam aplikasi. *Sequence Login* Petugas terdapat pada Gambar 2.30.



Gambar 2.30 *Sequence Login Petugas*

#### 9. *Sequence Laporan Petugas*

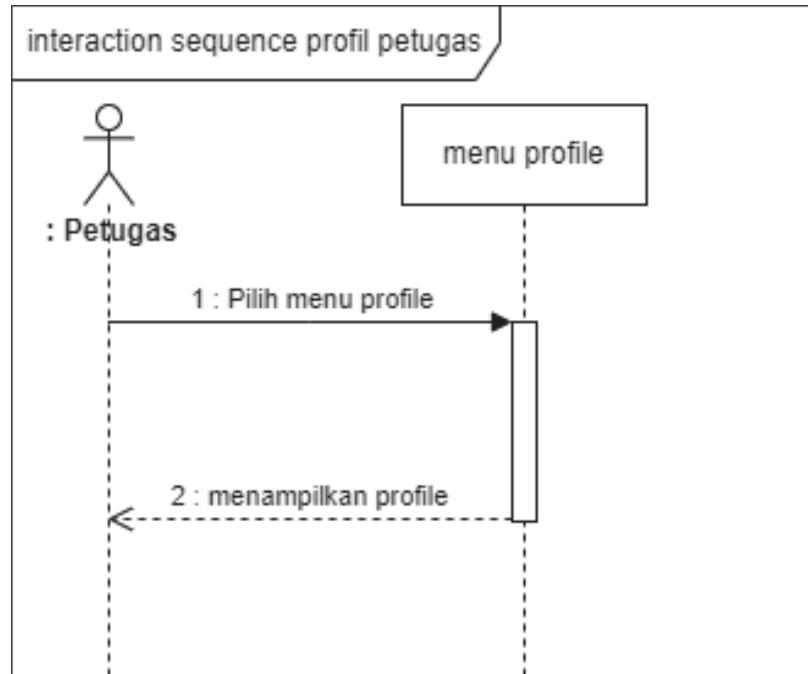
*Sequence Laporan Petugas* menjelaskan alur aktor ketika ada laporan masuk dari admin dan menangani laporan yang masuk. *Sequence Laporan Petugas* terdapat pada Gambar 2.23.



Gambar 2.31 *Sequence Laporan Petugas*

### 10. *Sequence Profile* Petugas

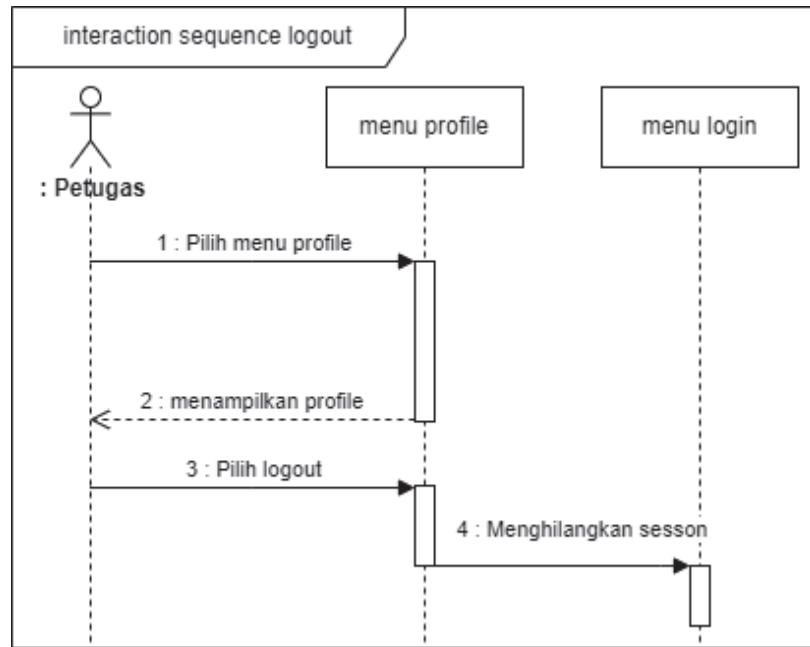
*Sequence Profile* Petugas menjelaskan alur aktor ketika melihat data akun atau *user* yang masuk di aplikasi. *Sequence Profile* Petugas terdapat pada Gambar 2.32.



Gambar 2.32 *Sequence Profile* Petugas

### 11. *Sequence Logout* Petugas

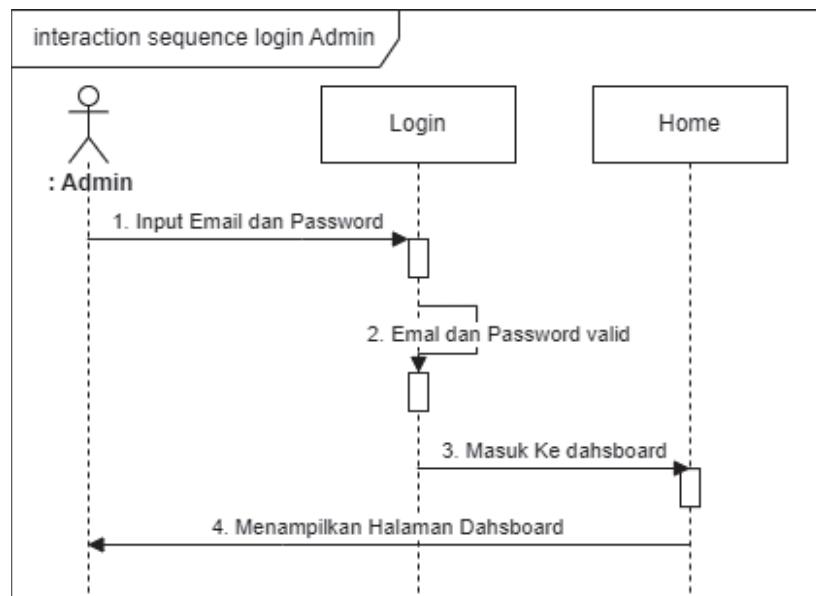
*Sequence Logout* Petugas menjelaskan alur aktor ketika keluar dari aplikasi. *Sequence Logout* Petugas terdapat pada Gambar 2.33.



Gambar 2.33 *Sequence Logout Petugas*

## 12. Sequence Login Admin

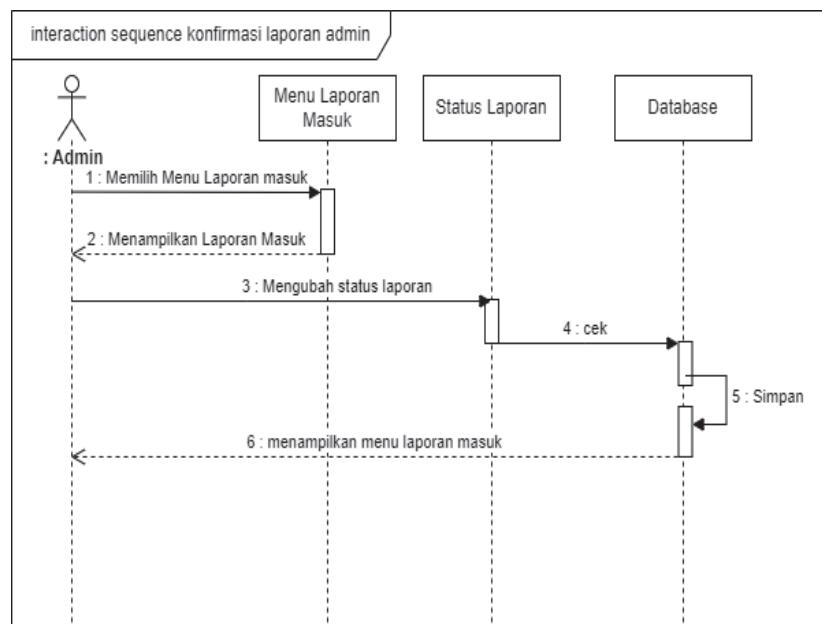
*Sequence Login Admin* menjelaskan alur aktor ketika masuk kedalam aplikasi. *Sequence Login Admin* terdapat pada Gambar 2.34.



Gambar 2.34 *Sequence Login Admin*

13. *Sequence Konfirmasi Laporan Admin*

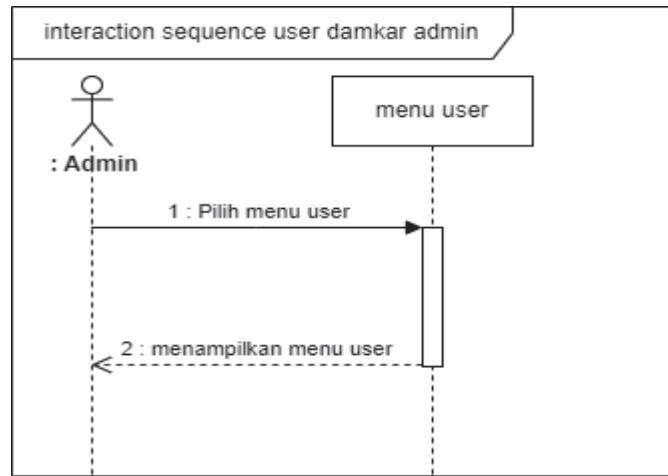
*Sequence Konfirmasi Laporan Admin* menjelaskan alur aktor ketika melakukan konfirmasi laporan yang masuk. *Sequence Konfirmasi Laporan Admin* terdapat pada Gambar 2.35.



Gambar 2.35 *Sequence Konfirmasi Laporan Admin*

14. *Sequence Warga Admin*

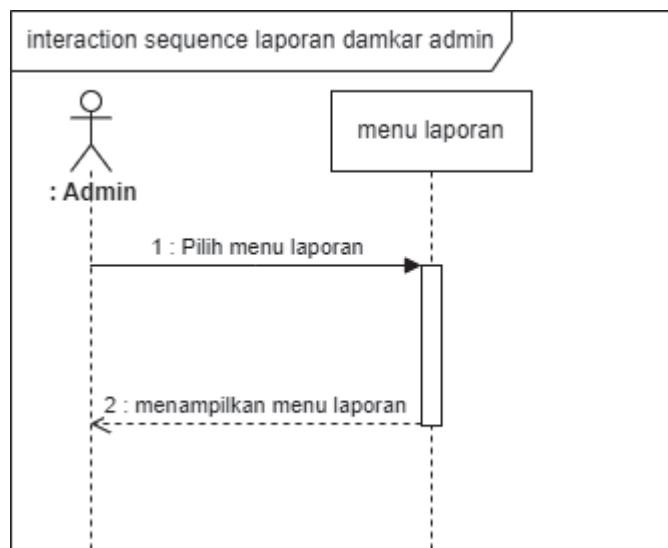
*Sequence Warga Admin* menjelaskan alur aktor ketika melihat data warga yang sudah aktif. *Sequence Warga Admin* terdapat pada Gambar 2.36.



Gambar 2.36 *Sequence* Warga Admin

#### 15. *Sequence* Laporan Admin

*Sequence* Laporan Admin menjelaskan alur aktor ketika melihat data laporan yang sudah selesai. *Sequence* Laporan Admin terdapat pada Gambar 2.37.

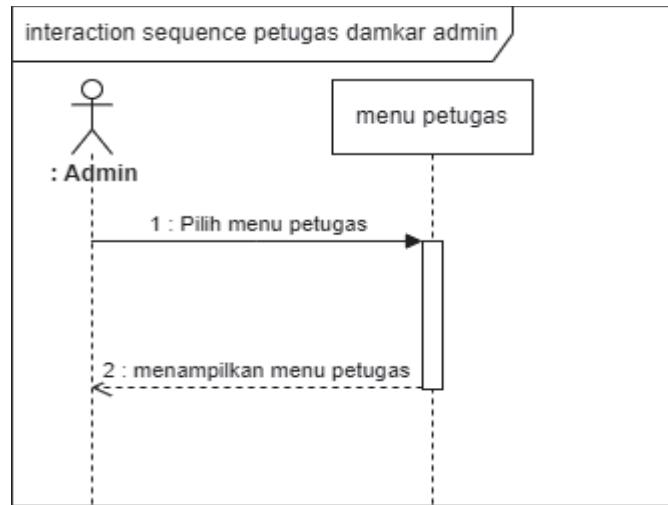


Gambar 2.37 *Sequence* Laporan Admin

#### 16. *Sequence* Petugas Admin

*Sequence* Petugas Admin menjelaskan alur aktor ketika melihat data petugas yang masih aktif. *Sequence* Petugas

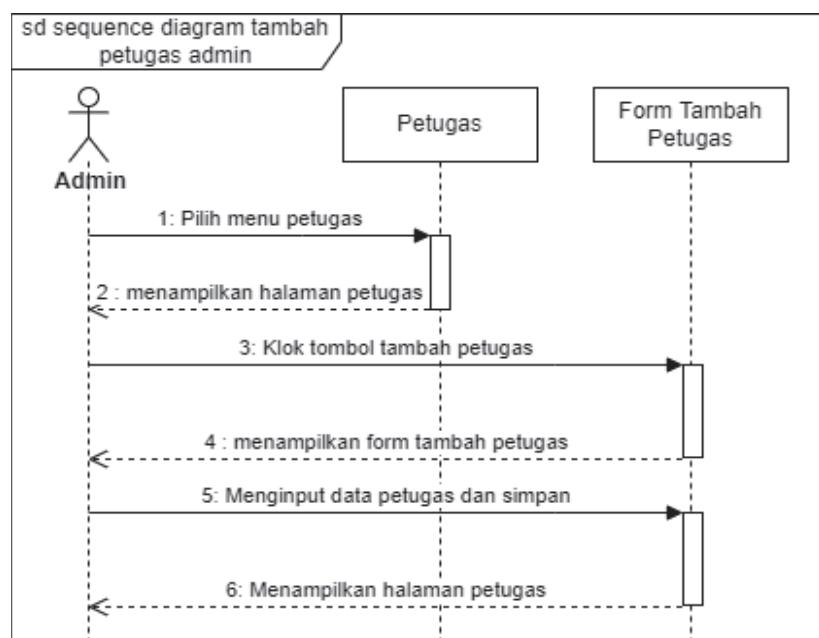
Admin terdapat pada Gambar 2.38.



Gambar 2.38 Sequence Petugas Admin

### 17. Sequence Tambah Petugas Admin

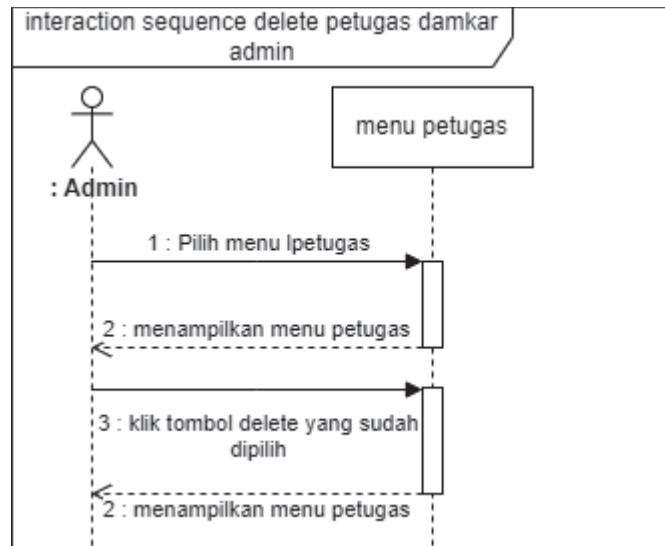
Sequence Tambah Petugas Admin menjelaskan alur aktor ketika melakukan penambahan data petugas. Sequence Tambah Petugas Admin terdapat pada Gambar 2.39.



Gambar 2.39 Sequence Tambah Petugas Admin

### 18. Sequence Delete Petugas Admin

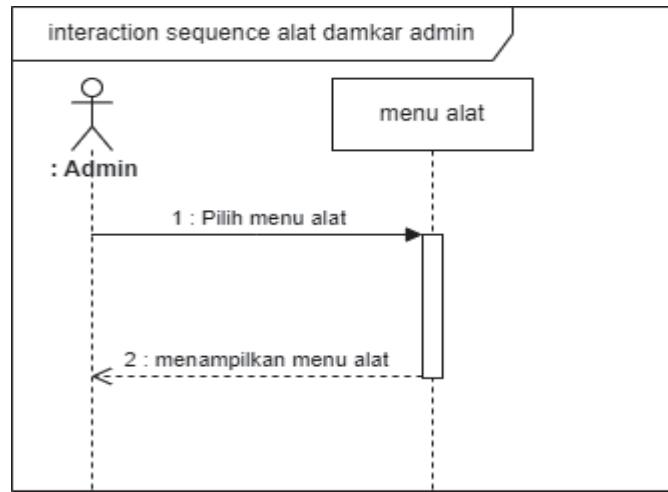
*Sequence Delete Petugas Admin* Admin menjelaskan alur aktor ketika melakukan penghapusan data petugas. *Sequence Delete Petugas Admin* terdapat pada Gambar 2.40.



Gambar 2.40 *Sequence Delete Petugas Admin*

### 19. Sequence Alat Damkar Admin

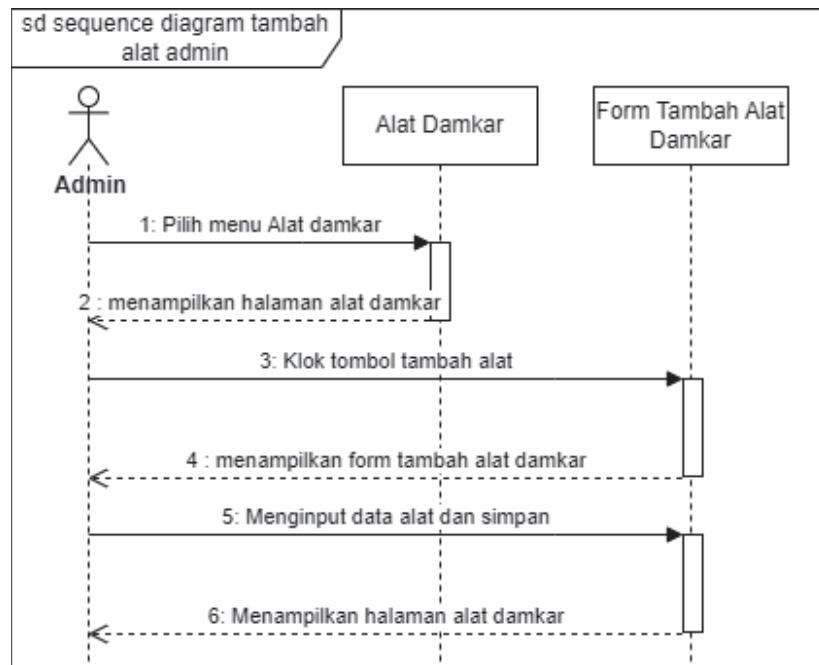
*Sequence Alat Damkar Admin* Admin menjelaskan alur aktor ketika melakukan penambahan alat pemadam kebakaran. *Sequence Alat Damkar Admin* terdapat pada Gambar 2.41.



Gambar 2.41 *Sequence Alat Damkar Admin*

#### 20. *Sequence Tambah Alat Damkar Admin*

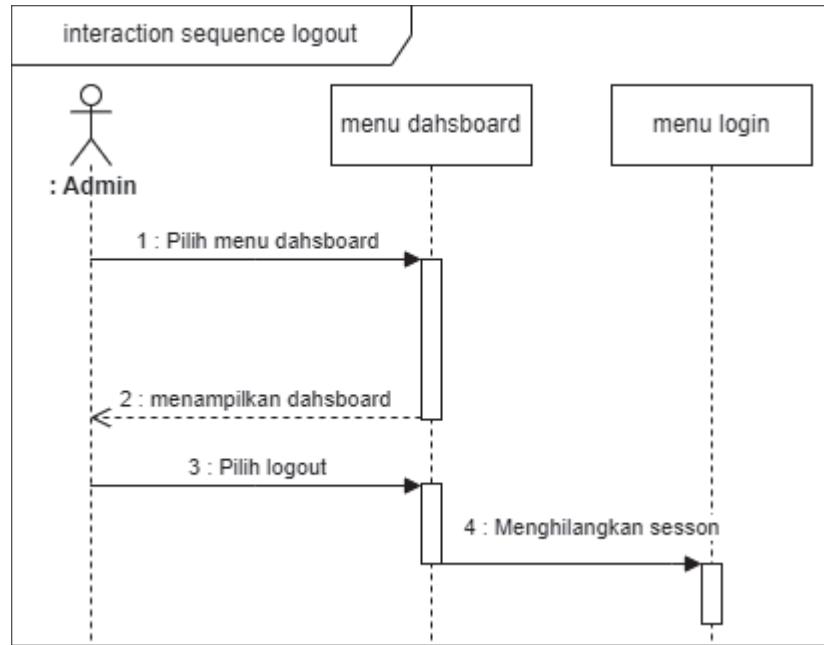
*Sequence Tambah Alat Damkar Admin* menjelaskan alur aktor ketika melakukan penambahan data alat pemadam kebakaran. *Sequence Tambah Alat Damkar Admin* terdapat pada Gambar 2.39.



Gambar 2.42 *Sequence Tambah Alat Damkar Admin*

## 21. Sequence Logout Admin

*Sequence Logout Admin* menjelaskan alur aktor ketika keluar dari aplikasi. *Sequence Logout Admin* terdapat pada Gambar 2.43.

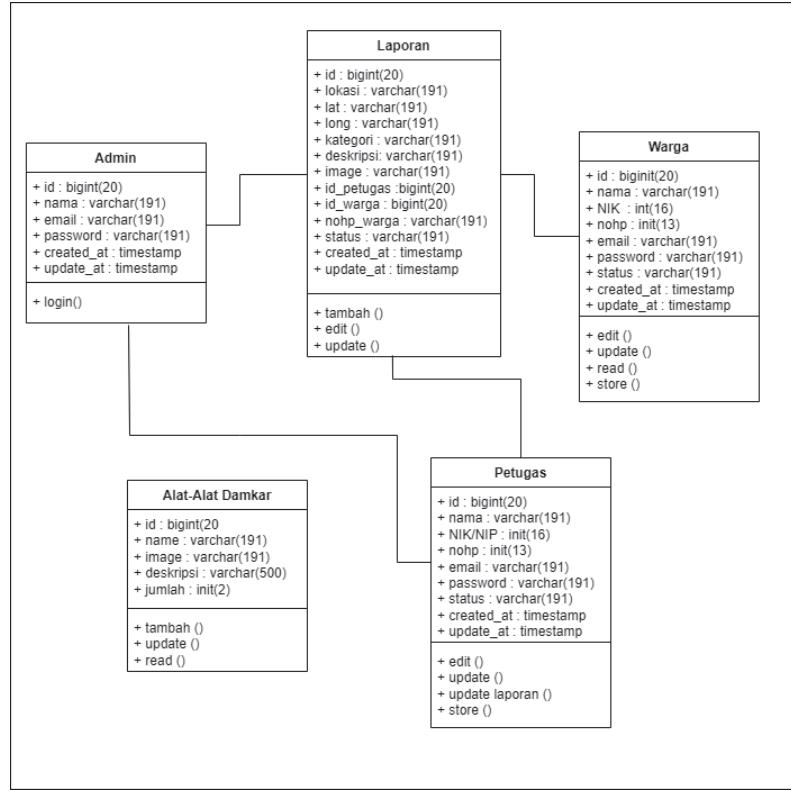


Gambar 2.43 *Sequence Logout Admin*

### 2.2.4 Class Diagram

Class diagram merupakan kumpulan dari beberapa class dan relasinya. Class identik dengan entity yang direpresentasikan dalam bentuk persegi dimana pada bagian atas ditulis nama class, kemudian kebawah ditulis attribute yang terdapat pada class, kemudian ke bawah lagi ditulis metode yang ada pada class [14].

Adapun rancangannya ada pada Gambar 2.44 berikut ini.



Gambar 2.44 Class Diagram

### 2.1.3 Perancangan Antar Muka

Perancangan antarmuka dalam pengembangan sistem ini dapat dilihat pada gambar berikut:

#### 1. Perancangan Halam *Login* Warga

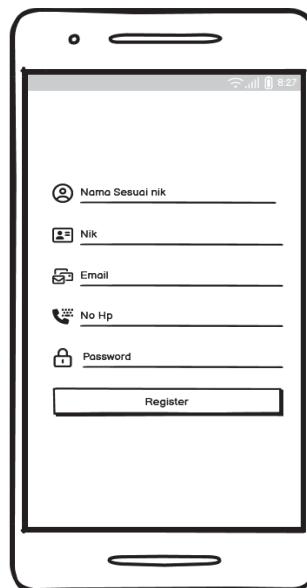
Perancangan Halaman *Login* Warga dibuat untuk sebagai contoh dalam membuat halaman *login* warga. Perancangan Halam *Login* Warga terdapat pada Gambar 2.45.



Gambar 2.45 *Login* Warga

## 2. Perancangan Halaman *Register* Warga

Perancangan Halaman *Register* Warga dibuat untuk sebagai contoh dalam membuat halaman *Register* warga. Perancangan Halam *Register* Warga terdapat pada Gambar 2.46.



Gambar 2.46 *Register* Warga

### 3. Perancangan Halaman *Home* Warga

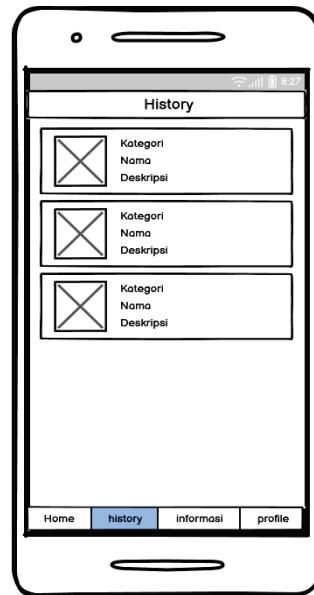
Perancangan Halaman *Home* Warga dibuat untuk sebagai contoh dalam membuat halaman *Home* warga. Perancangan Halam *Home* Warga terdapat pada Gambar 2.47.



Gambar 2.47 *Home* Warga

### 4. Perancangan Halaman *History* Warga

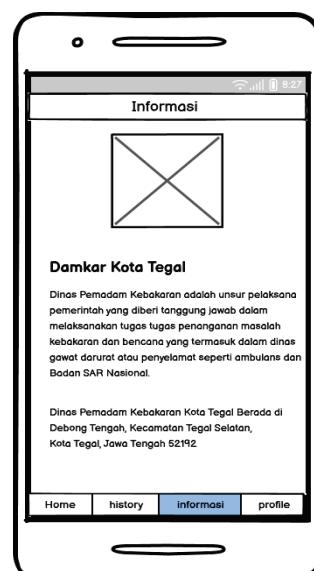
Perancangan Halaman *History* Warga dibuat untuk sebagai contoh dalam membuat halaman *History* warga. Perancangan Halam *History* Warga terdapat pada Gambar 2.48.



Gambar 2.48 *History* Warga

##### 5. Perancangan Halaman Informasi

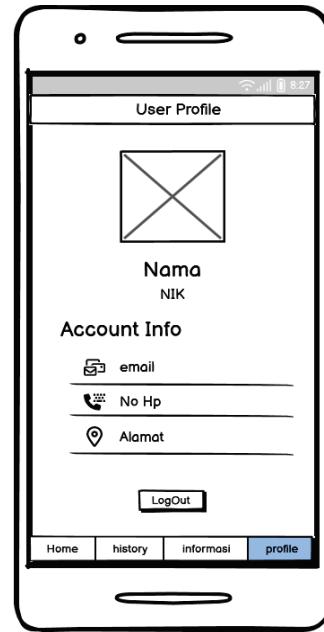
Perancangan Halaman Informasi Warga dibuat untuk sebagai contoh dalam membuat halaman Informasi warga. Perancangan Halam Informasi Warga terdapat pada Gambar 2.49.



Gambar 2.49 Informasi Warga

## 6. Perancangan Halaman *User* Warga

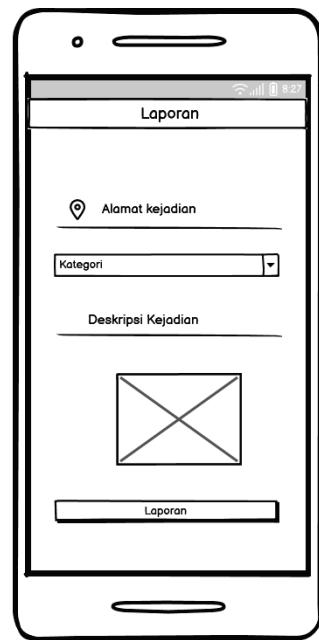
Perancangan Halaman *User* Warga dibuat untuk sebagai contoh dalam membuat halaman *User* warga. Perancangan Halam *User* Warga terdapat pada Gambar 2.50.



Gambar 2.50 *Profile* Warga

## 7. Perancangan Halaman Laporan Warga

Perancangan Halaman Laporan Warga dibuat untuk sebagai contoh dalam membuat halaman Laporan warga. Perancangan Halam Laporan Warga terdapat pada Gambar 2.51.



Gambar 2.51 Laporan Warga

## 8. Perancangan Halaman *Login* Petugas

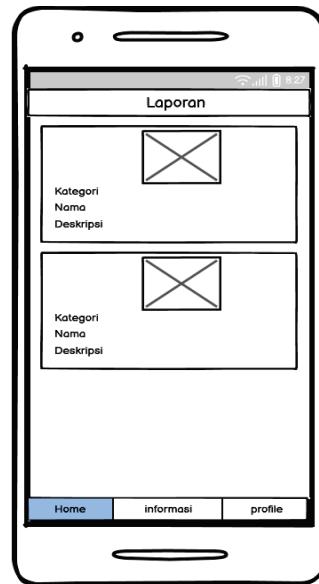
Perancangan Halaman *Login* Petugas dibuat untuk sebagai contoh dalam membuat *Login* Petugas. Perancangan Halam *Login* Petugas terdapat pada Gambar 2.52.



Gambar 2.52 *Login* Petugas

## 9. Perancangan Halaman *Home* Petugas

Perancangan Halaman *Home* Petugas dibuat untuk sebagai contoh dalam membuat *Home* Petugas. Perancangan Halam *Home* Petugas terdapat pada Gambar 2.53.



Gambar 2.53 *Home* Petugas

#### 10. Perancangan Informasi Petugas

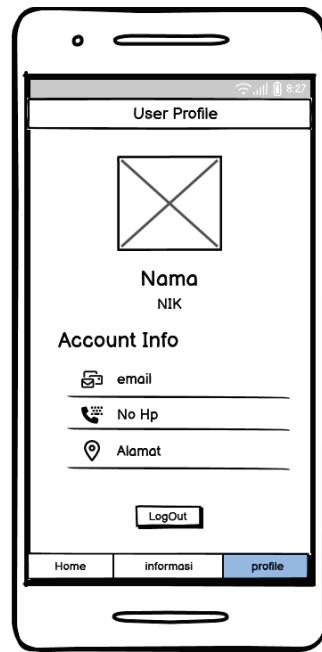
Perancangan Halaman Informasi Petugas dibuat untuk sebagai contoh dalam membuat Informasi Petugas. Perancangan Halam Informasi Petugas terdapat pada Gambar 2.54.



Gambar 2.54 Informasi Petugas

## 11. Perancangan *Profile* Petugas

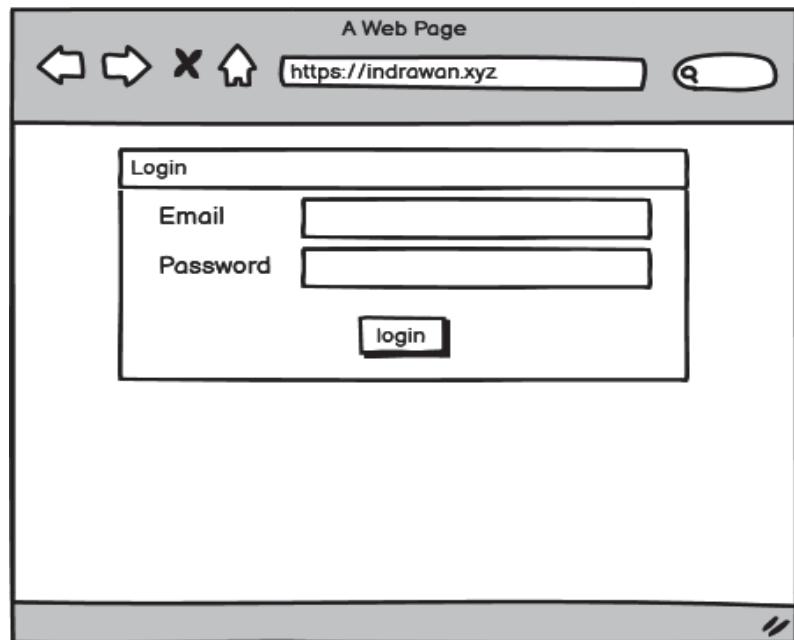
Perancangan Halaman *Profile* Petugas dibuat untuk sebagai contoh dalam membuat *Profile* Petugas. Perancangan Halam *Profile* Petugas terdapat pada Gambar 2.55.



Gambar 2.55 *Profile* Petugas

## 12. Perancangan Halaman *Login* Admin

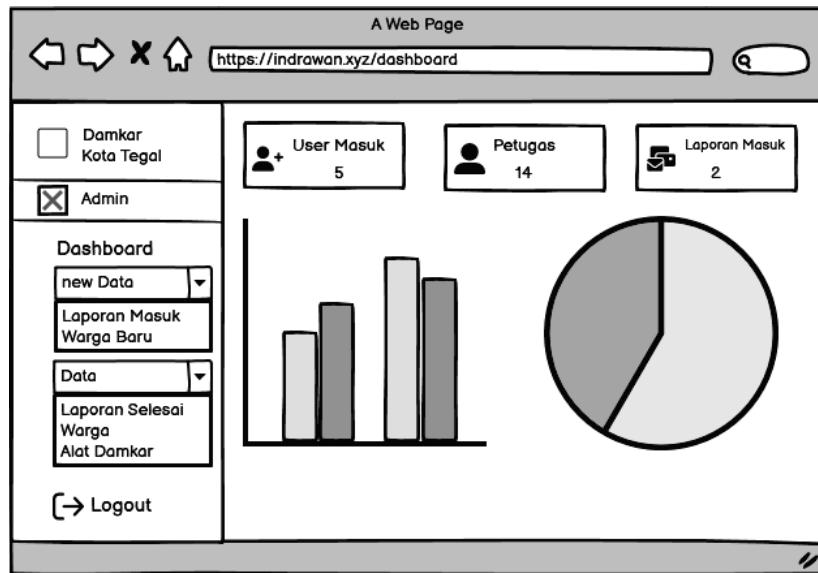
Perancangan Halaman *Login* Admin dibuat untuk sebagai contoh dalam membuat *Login* Admin. Perancangan Halam *Login* Admin terdapat pada Gambar 2.56.



Gambar 2.56 *Login* Admin

## 13. Perancangan Halaman *Dashboard* Admin

Perancangan Halaman *Dashboard* Admin dibuat untuk sebagai contoh dalam membuat *Dashboard* Admin. Perancangan Halam *Dashboard* Admin terdapat pada Gambar 2.57.



Gambar 2.57 Dashboard admin

#### 14. Perancangan Halaman Laporan Masuk Admin

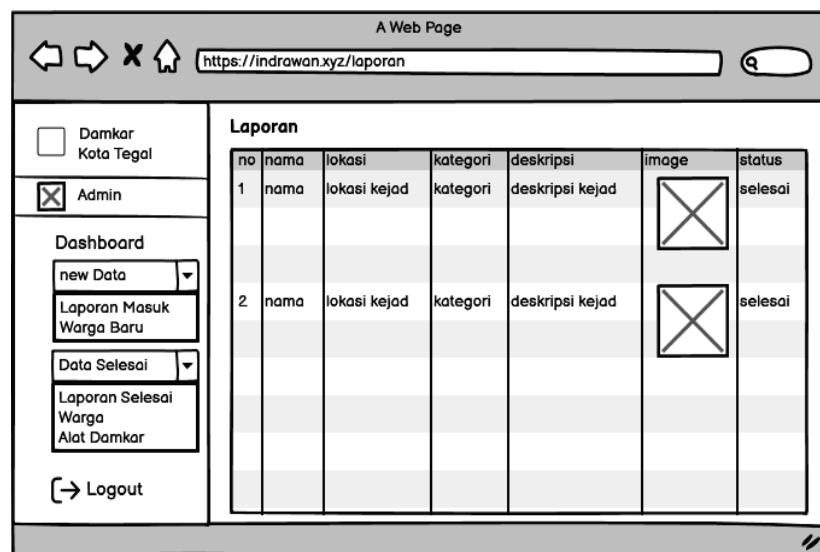
Perancangan Halaman Laporan Masuk Admin dibuat untuk sebagai contoh dalam membuat Laporan Masuk Admin. Perancangan Halam Laporan Masuk Admin terdapat pada Gambar 2.58.

The screenshot shows a web-based page for managing reports. The top navigation bar includes back, forward, and search icons, and the URL https://indrawan.xyz/laporanmasuk. The page header features the text 'A Web Page' and the URL. On the left, a sidebar menu for 'Admin' includes 'Dashboard' with 'new Data' and 'Laporan Masuk Warga Baru' options, and 'Data Selesai' with 'Laporan Selesai Warga' and 'Alat Damkar' options. Below the sidebar is a 'Logout' button. The main content area is titled 'Laporan Masuk' and displays a table with two rows of data. The table columns are: no, nama, lokasi, kategori, deskripsi, gambar, status, and action. The first row has values: 1, nama, lokasi kejad, kategori, deskripsi kejad, a crossed-out image, menunggu, and edit. The second row has values: 2, nama, lokasi kejad, kategori, deskripsi kejad, a crossed-out image, menunggu, and edit.

Gambar 2.58 Laporan Masuk Admin

## 15. Perancangan Halaman Laporan Selesai Admin

Perancangan Halaman Laporan Selesai Admin dibuat untuk sebagai contoh dalam membuat Laporan Selesai Admin. Perancangan Halam Laporan Selesai Admin terdapat pada Gambar 2.59.



no	nama	lokasi	kategori	deskripsi	image	status
1	nama	lokasi kejad				selesai
2	nama	lokasi kejad	kategori	deskripsi kejad		selesai

Gambar 2.59 Laporan Selesai

## 16. Perancangan Halaman Warga Admin

Perancangan Halaman Warga Admin dibuat untuk sebagai contoh dalam membuat Warga Admin. Perancangan Halam Warga Admin terdapat pada Gambar 2.60.

A Web Page				
<a href="https://indrawan.xyz/user">https://indrawan.xyz/user</a>				
<b>Daftar Warga</b>				
no	nama	email	status	
1	warga	warga@gmail.com	Aktif	
2	warga2	warga2@gmail.com	Aktif	

Gambar 2.60 Warga Admin

#### 17. Perancangan Halaman Alat-Alat Damkar Admin

Perancangan Halaman Alat-Alat Damkar Admin dibuat untuk sebagai contoh dalam membuat Alat-Alat Damkar Admin. Perancangan Halam Alat-Alat Damkar Admin terdapat pada Gambar 2.61.

A Web Page				
<a href="https://indrawan.xyz/alatdamkar">https://indrawan.xyz/alatdamkar</a>				
<b>Alat-Alat Damkar</b>				
Tambah Data	no	nama	gambar	desripsi
	1	baju tahan api		baju tahan api.....
	2	baju tahan panas		baju tahan panas.....

Gambar 2.61 Alat Damkar Admin

#### 2.1.4 Implementasi

Proses implementasi dilakukan untuk membuat aplikasi sesuai dengan perancangan yang telah dirancang sebelumnya, sebagai berikut:

1. Halaman *Login* Aplikasi Warga

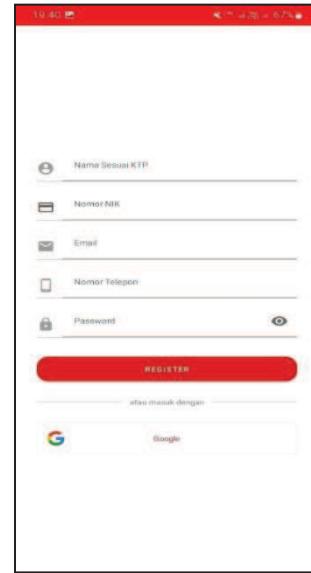
Merupakan tampilan halaman masyarakat *login* terlebih dahulu agar dapat menggunakan aplikasi Damkar.



Gambar 2.62 Halaman *login* aplikasi warga

2. Halaman *Register* Aplikasi Warga

Halaman *register* merupakan halaman untuk menginput data warga agar dapat login dan menggunakan aplikasi Damkar.



Gambar 2.63 Halaman *register* aplikasi warga

### 3. Halaman *Home* Aplikasi Warga

Halaman *home* merupakan halaman utama dalam aplikasi Damkar terdapat tombol untuk leporan kejadian dan telfon secara langsung.



Gambar 2.64 Halaman *home* aplikasi warga

#### 4. Halaman Laporan Aplikasi Warga

Halaman laporan warga merupakan halaman untuk warga melapor menggunakan aplikasi Damkar,



Gambar 2.65 Halaman Laporan aplikasi warga

#### 5. Halaman History Aplikasi Warga

Halaman history berisi semua data yang pernah di laporkan oleh *user*.



Gambar 2.66 Halaman *history* aplikasi warga

#### 6. Halaman Detail History Aplikasi Warga

Halaman history berisi semua data yang pernah di laporkan oleh *user*.



Gambar 2.67 Halaman detail *history* aplikasi warga

## 7. Halaman Informasi Aplikasi Warga

Halaman informasi merupakan halaman tentang damkar dan tentang aplikasi Damkar.



Gambar 2.68 Halaman informasi aplikasi warga

## 8. Halaman *User* Aplikasi Warga

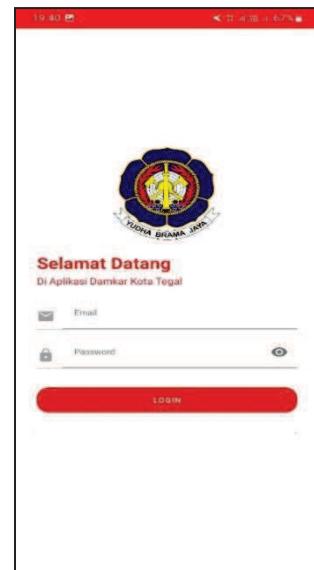
Halaman *user* adalah halaman yang berisi data *user* atau data warga yang login pada aplikasi.



Gambar 2.69 Halaman *user* aplikasi warga

#### 9. Halaman *Login* Aplikasi Petugas

Merupakan tampilan halaman petugas *login* terlebih dahulu agar dapat menggunakan aplikasi Damkar.



Gambar 2.70 Halaman *login* aplikasi petugas

## 10. Halaman *Home* Aplikasi Petugas

Merupakan tampilan halaman berisi semua laporan yang telah diverifikasi oleh admin.



Gambar 2.71 Halaman *home* aplikasi petugas

## 11. Halaman Laporan Aplikasi Petugas

Merupakan tampilan halaman berisi maps dan data laporan.



Gambar 2.72 Halaman laporan aplikasi petugas

## 12. Halaman Informasi Aplikasi Petugas

Halaman informasi merupakan halaman tentang damkar dan tentang aplikasi Damkar.



Gambar 2.73 Halaman informasi aplikasi petugas

## 13. Halaman *User* Aplikasi Petugas

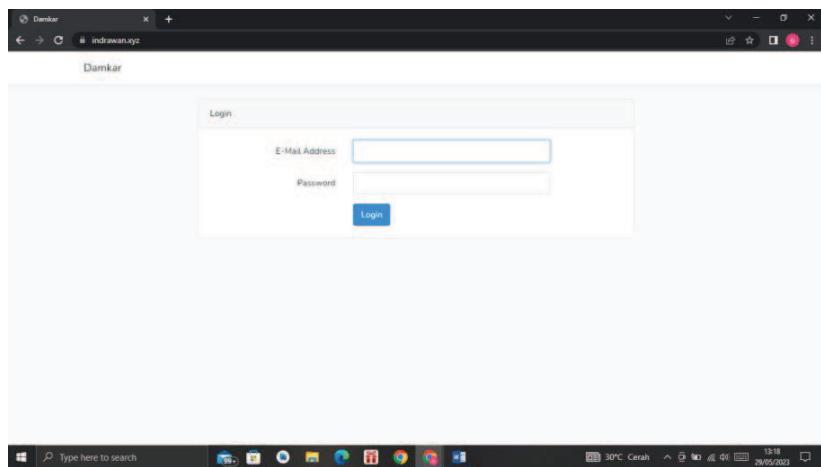
Halaman *user* adalah halaman yang berisi data user atau data petugas yang login pada aplikasi.



Gambar 2.74 Halaman *user* aplikasi petugas

#### 14. Halaman *Login* Admin

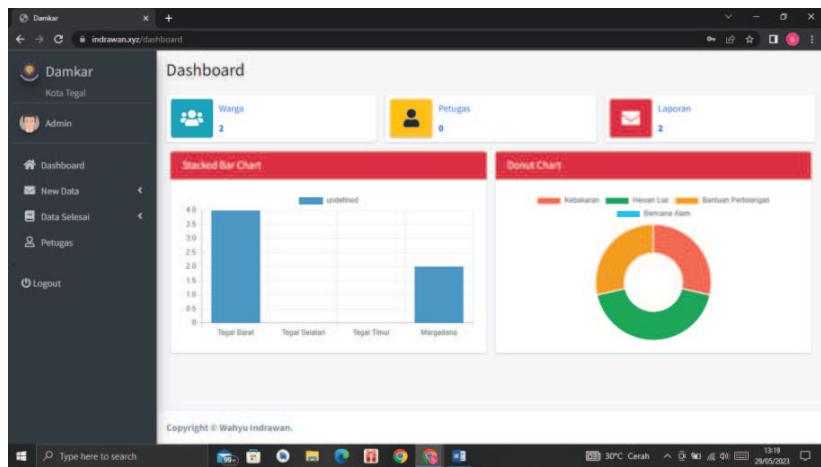
Merupakan halaman untuk login agar bisa masuk ke aplikasi admin damkar.



Gambar 2.75 Halaman *login* admin

#### 15. Halaman *Dashboard* Admin

Merupakan halaman utama dalam website di admin.



Gambar 2.76 Halaman *dashboard* admin

#### 16. Halaman Laporan Masuk Admin

Halaman laporan masuk merupakan halaman untuk admin *verifikasi* laporan yang masuk.

Tabel Laporan Masuk							
NO	Nama	lokasi	kategori	Deskripsi	Image	Status	Action
29	wain3	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	Kebakaran	bsbsb		Menunggu	<button>EDIT</button>
31	warga	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	Kebakaran	gujuf		Menunggu	<button>EDIT</button>

Gambar 2.77 Halaman laporan masuk admin

### 17. Halaman Laporan Selesai Admin

Merupakan halaman yang berisi laporan yang telah selesai di tangani petugas.

Tabel Laporan							
NO	Nama	lokasi	kategori	Deskripsi	Image	Status	
19	warga@gmail.com	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	Hewan liar	hsbs		Selesai	
20	warga	FPIV(C+5)5, Batusari, Kec. Dawuan, Kabupaten Subang, Jawa Barat 41271, Indonesia	bantuan pertolongan	hssjka		Selesai	
21	wain	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	bantuan pertolongan	leptop masuk tas		Selesai	
24	wain3	Gg. Muara No.11, RW.01, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	hewan liar	ular masuk rumah		Selesai	

Gambar 2.78 Halaman laporan selesai admin

### 18. Halaman Warga Admin

Merupakan halaman yang berisi warga warga yang aktif akunya.

NO	Nama	Email	Status
2	warga	warga@gmail.com	Aktif
6	wain3	wain3@gmail.com	Aktif
7	wain13	wain4@gmail.com	Aktif
8	wain10	wain10@gmail.com	Aktif

Gambar 2.79 Halaman warga admin

### 19. Halaman Alat-Alat Damkar Admin

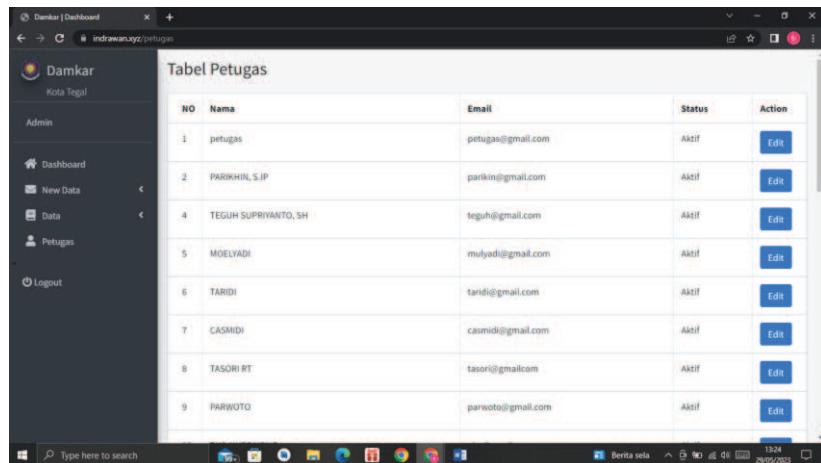
Merupakan halaman yang berisi data alat damkar yang dimiliki damkar Kota Tegal.

NO	Nama	Image	Deskripsi	Jumlah	Action
1	Baju Tahan Api		Baju pemadam kebakaran tahan api digunakan secara lengkap bersama dengan celana dari bahan aluminium foil yang terbal. Selain itu, baju tahan api juga dilengkapi dengan helm atau penutup kepala yang diberi lapisan kaca bening, sepatu, senter, dan sarung tangan tahan api. Perlakuan pemadam kebakaran akan melindungi pengguna dari kobaran api.	5	<button>edit</button>
2	Baju Tahan Panas		Baju tahan panas dirancang secara khusus menggunakan bahan dari serat akrilik yang dipadukan dengan aluminium dan bahan rang fiberglass. Dengan bahan tersebut, baju tahan panas ini mampu melawan radiasi, sehingga memberikan perlindungan pada tubuh pengguna dari panasnya suhu kobaran api yang menyala saat menjalankan tugasnya.	17	<button>edit</button>
3	Helm pemadam kebakaran		Helm yang digunakan oleh petugas pemadam kebakaran terbuat dari material yang anti benturan dan anti panas. Dengan menggunakan helm tersebut, maka petugas akan aman dan terlindungi dari panas maupun benturan saat melaksanakan tugasnya. Helm pemadam kebakaran juga disertai dengan tali pengikat dan neck protection yang tahan panas.	15	<button>edit</button>
4	Sarung Tangan		Peralatan keselamatan pemadam kebakaran ini dibuat dari bahan utama	17	<button>edit</button>

Gambar 2.80 Halaman alat-alat damkar admin

## 20. Halaman Petugas Admin

Merupakan halaman yang berisi petugas yang aktif.



The screenshot shows a web-based application interface for managing staff. The left sidebar is titled 'Damkar' and includes 'Dashboard', 'New Data', 'Data', and 'Petugas' (selected). The main content area is titled 'Tabel Petugas' and displays a table with the following data:

NO	Nama	Email	Status	Action
1	petugas	petugas@gmail.com	Aktif	<button>Edit</button>
2	PARIKHIN, S.I.P	parkin@gmail.com	Aktif	<button>Edit</button>
4	TEGUH SUPRIYANTO, SH	teguh@gmail.com	Aktif	<button>Edit</button>
5	MOELYADI	mulyadi@gmail.com	Aktif	<button>Edit</button>
6	TARIDI	taridi@gmail.com	Aktif	<button>Edit</button>
7	CASMIDI	casmidi@gmail.com	Aktif	<button>Edit</button>
8	TASORI RT	tasori@gmail.com	Aktif	<button>Edit</button>
9	PARWOTO	parwoto@gmail.com	Aktif	<button>Edit</button>

Gambar 2.81 Halaman petugas admin

## 2.2 Hasil Pengujian

### A. Metode *Black Box*

Pengujian sistem dilakukan bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Pengujian yang dilakukan yaitu pengujian *alpha*. Pengujian *alpha* yang digunakan adalah metode *black-box*. Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Pengujian *alpha* dilakukan dengan menggunakan metode *black box*. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data). Kesalahan performasi, kelasahan inisialisasi dan akhir program[15]. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Rencana pengujian yang akan dilakukan dapat dilihat pada tabel berikut :

Tabel 2.1 Pengujian *Black Box*

Hasil Pengujian Menggunakan Pengujian Black Box				
No	Kasus	Uji Sezenario	Hasil yang Disingginkan	Hasil Pengujian
1	<i>Login</i> Aplikasi Warga	Halaman <i>Login</i>	Jika berhasil Login makan	[ ✓]Berhasil

			akan menampilkan halaman home aplikasi	[ <input type="checkbox"/> ]Tidak Berhasil
2	<i>Registrasi</i> Aplikasi Warga	Halaman <i>Register</i>	Jika berhasil melakukan register maka akun sudah aktif dan dapat di gunakan untuk <i>login</i>	[ <input checked="" type="checkbox"/> ]Berhasil [ <input type="checkbox"/> ]Tidak Berhasil
3	<i>Login</i> Aplikasi Petugas	Halaman <i>Login</i>	Jika berhasil menampilkan halaman home aplikasi petugas	[ <input checked="" type="checkbox"/> ]Berhasil [ <input type="checkbox"/> ]Tidak Berhasil
4	<i>Login</i> Admin	Halaman <i>Login</i> Admin	Jika berhasil menampilkan halaman <i>dashboard</i> aplikasi petugas	[ <input checked="" type="checkbox"/> ]Berhasil [ <input type="checkbox"/> ]Tidak Berhasil
5	Menu Laporan pada aplikasi warga	Memilih Fitur laporan	Menampilkan halaman laporan dan memperoleh	[ <input checked="" type="checkbox"/> ]Berhasil [ <input type="checkbox"/> ]Tidak Berhasil

		menggunakan aplikasi	lokasi otomatis setelah foto	
6	Menu Laporan Aplikasi Petugas	Memilih salah satu laporan yang di terima	Menampilkan rute lokasi dan detail laporan	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Tidak Berhasil

## 2.3 Kesimpilan dan Saran

### 2.3.1 Kesimpulan

Berdasarkan hasil pembahasan dan penelitian Aplikasi *Location Based Service* (LBS) Pemanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal) maka dapat di tarik kesimpulan bahwa:

1. Dengan adanya Aplikasi *Location Based Service* (LBS) Pemanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal) sebagai upaya untuk menyediakan aplikasi yang dapat menghubungi damkar dengan cepat untuk menolong korban kejadian di Kota Tegal. Dengan aplikasi ini dapat meningkatkan kecepatan pelayanan damkar dalam menangani korban kejadian dengan dukukungan sistem yang terintegrasi warga dan petugas Damkar
2. Berdasarkan pengujian sistem menggunakan *Black-Box Testing*, diperoleh hasil *output* sesuai dengan target berdasarkan input yang diberikan.

### 2.3.2 Saran

Berikut adalah saran-saran untuk pengembangan lebih lanjut terhadap aplikasi *Location Based Service* (LBS) Pemanggil Pemadam Kebakaran (Studi kasus : Kota Tegal) dalam upaya pengembangan sistem agar dapat menjadi lebih baik sesuai yang diharapkan dalam pengembangan selanjutnya sistem dapat lebih dikembangkan lagi secara materi maupun tampilan agar lebih menarik dan *user* mudah untuk menggunakanya.

## **BAB III**

### **HAK KEKAYAAN INTELEKTUAL (HKI)**

#### **3.1 Proses**

Pendaftaran Hak Kekayaan Intelektual (HKI) dapat diproses setelah mengumpulkan dokumen persyaratan seperti manual book atau dokumen technical, KTP pemohon, dan surat pernyataan atau surat pengalihan yang sebelumnya telah disetujui dan ditanda tangani oleh Dosen Pembimbing I dan II serta di berikan materai Sepuluh Ribu pada TTD mahasiswa.

#### **3.2 Identitas HKI**

Nomor	: EC00202352474
Tanggal Dikeluarkan	: 6 Juli 2023
Nama Pencipta	: 1. Wahyu Indrawan 2. Ginanjar Wiro Sasmito, M.Kom 3. Dega Surono Wibowo, S.T., M.Kom.
Nama Pemegang Hak	: Pusat Penelitian dan Pengabdian
Cipta	Masyarakat : (P3M) Politeknik Harapan Bersama
Jenis Ciptaan	: Program Komputer

Judul Ciptaan : Location Based Service Pertolongan  
Memanggil Pemadam Kebakaran (Studi  
Kasus : Kota Tegal)

URL Bukti : <https://e-hakcipta.dgip.go.id/index.php/c?code=MDFmMzU5ZDcxYjY5ZmM5NDIzZDViOGQxZGI1NTFINjUK>

## DAFTAR PUSTAKA

- [1] M. Kay, J. Santos, and M. Takane, “mHealth: New Horizons for Health through Mobile Technologies,” World Heal. Organ., 2011.
- [2] D. Ben-Zeev, S. M. Schueller, M. Begale, J. Duffecy, J. M. Kane, and D. C. Mohr, “Strategies for mHealth Research: Lessons from 3 Mobile Intervention Studies,” Adm. Policy Ment. Heal. Ment. Heal. Serv. Res., 2015.
- [3] Akmal Eddy Madda, Moh. Ramlil, Ahmad Yani, "Aplikasi mHealth “SIGA” Untuk Pelayanan Kesehatan Masyarakat Umum Berbasis Android," osf.io ,2019
- [4] Apriyanto, Fredy Nur. 2020. “Aplikasi *Location Based Service* (LBS) Pertolongan Cepat Menuju Rumah Sakit Terdekat (Studi Kasus: Kota Tegal)”. Dari Perpustakaan Politeknik Harapan Bersama.
- [5] Handoyo, Hasan dkk. 2017. “Aplikasi *Frontend* Pada *Emergency Call* Berbasis *Mobile Hybrid* di Kota Tegal”. Dari Perpustakaan Politeknik Harapan Bersama.
- [6] Irene Anindaputri Iswanto, Kevin Wahyu Triwaldi, Bosta Harry Hagata G, Dhana Firmansyah dkk. 2016. “Aplikasi *Location Based Service* “TOLONG.IN” Pada Platform Android”. Jurnal Infotech. Vol. 4 No.2
- [7] Veronika, Nuri David Maria., Saroni, Saparudin., Muntahanah. 2022. “Aplikasi Pengaduan *Bullying* dan Kekerasan Anak Serta Perempuan Menggunakan *Location Based Service*”. Jurnal Pseudocode, Volume 9 Nomor 2.
- [8] Fauzan, Farhan., NM, Faizah., Koryanto, Lucky. 2022. “*Android Based*

*Application for Searching Indekos In Pasar Minggu Jakarta South With Location Based Service (LBS) Method*’. Journal Mobile Technologies.

- [9] Trise Putra, Dede W., & Rahmi, Andriani. 2019. Unified Modelling Language (UML)dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. Jurnal TEKNOIF.
- [10] Sukamto, R. A., dan Shalahuddin, M. 2018. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi. Bandung: Informatika.
- [11] Hendini A. Pemodelan UML sistem informasi monitoring penjualan dan stok barang (studi kasus: distro zhezha pontianak). Jurnal Khatulistiwa Informatika. vol 4, no.2, 2016, doi : <https://doi.org/10.31294/jki.v4i2.1262>
- [12] Kurniawan, T.B., 2020. Perancangan sistem aplikasi pemesanan makanan dan minuman pada cafetaria no caffe di Tanjung Balai Karimun menggunakan bahasa pemograman PHP Dan MySQL. Jurnal Tikar, 1(2), pp.192-206.
- [13] Meilinda, Eva. 2016. "Perancangan Aplikasi Kearsipan Surat Menyurat Pada Badan Pemerintahan (Studi Kasus: Badan Pemberdayaan Perempuan Perlindungan Anak Dan Keluarga Berencana Pontianak)." Jurnal Khatulistiwa Informatika, vol. 4, no. 2. doi: <https://doi.org/10.31294/jki.v4i2.1266>.
- [14] N. Rizkita, E. Rosely, and H. Nugroho., 2018. “Aplikasi Pendaftaran dan Transaksi Pasien Klinik Hewan di Bandung Berbasis Web,” eProceedings Appl. Sci., vol. 4, no. 3, pp. 1512–1520.

[15] Nindra, Srinivas dkk. 2012, “Black Box And White Box Testing” international Journal of Embeded system and aplikcation, Jawaharlal Nehru Technology University.(Vol.2, No.2, June 2012).

# LAMPIRAN

## Lampiran 1. Surat Kesedian Pembimbing 1

### **SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR**

Kami yang bertanda tangan di bawah ini:

Pihak Pertama

Nama : Wahyu Indrawan  
NIM : 18090108  
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Ginanjar Wiro Sasmito, M.Kom.  
Status : Dosen Tetap  
NIDN : 0613028601  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : III/d

Menyatakan sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Tugas Akhir Pihak Pertama dengan syarat berjanji menyelesaikan tepat waktu.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Tegal, Maret 2022

Pihak Pertama



Wahyu Indrawan  
NIM. 18090108

Pihak Kedua

  
Ginanjar Wiro Sasmito, M.Kom.  
NIP. 10.007.032

Mengetahui,  
Ketua Program Studi Sarjana Terapan Teknik  
Informatika



Slamet Wirabowo, S.Pd., M.Eng.  
NIP. 08.015.222

## Lampiran 2. Surat Kesedian Bimbingan 2

### **SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR**

Kami yang bertanda tangan di bawah ini:

**Pihak Pertama**

Nama : Wahyu Indrawan  
NIM : 18090108  
Program Studi : DIV Teknik Informatika

**Pihak Kedua**

Nama : Dega Surono Wibowo, M.Kom.  
Status : Dosen Tetap  
NIDN : 0607108202  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : III/c

menyatakan sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir Pihak Pertama dengan syarat melakukan bimbingan 1x dalam seminggu atau setidak-tidaknya 3x bimbingan dalam 1 bulan, apabila saya tidak memenuhi syarat tersebut, maka saya tidak berhak meminta surat rekomendasi mengikuti sidang Tugas Akhir, dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan tepat waktu.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

**Pihak Pertama**

  
Wahyu Indrawan  
NIM. 18090108

Tegal, Maret 2022

**Pihak Kedua**

  
Dega Surono Wibowo, M.Kom.  
NIPY. 06.014.183

Mengetahui,  
Ketua Program Studi Teknik Informatika



Lampiran 3. Lembar Bimbingan Pembimbing 1



**D-IV TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

**LEMBAR BIMBINGAN TUGAS AKHIR**

**Nama** : Wahyu Indrawan  
**NIM** : 18090108  
**No. Ponsel** : 085326699718  
**Judul TA** : Aplikasi *Location Based Service*(LBS) Pertolongan  
Memanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal)  
**Dosen Pembimbing I** : Ginanjar Wiro Sasmito, M.Kom

No	Tanggal	Pemeriksaan	Perbaikan yang Perlu Dilakukan	Paraf Pembimbing
1	21/06/2022	- Use case Diagram	- Silahkan lanjut membuat MOCK UP	f.
2	29/06/2022	- MOCK UP Aplikasi	- di lanjut membuat Aplikasi	f.
3	09/07/2022	- Aplikasi	- Baru 75% lanjutkan sampai selesai	f.

4	10/2023 03	Aplikasi	<ul style="list-style-type: none"> <li>- menambahkan fitur :</li> <li>- fitur notifikasi</li> <li>- button selesai nyala jika di dekat kejadian</li> </ul>	f.
5	5/2023 04	Aplikasi	<ul style="list-style-type: none"> <li>- menambahkan fitur grafik</li> <li>- membuat HKI</li> </ul>	f.
6	5/2023 06	Laporan HKI Aplikasi	<ul style="list-style-type: none"> <li>- selesaikan HKI</li> <li>- Aplikasi selesai</li> </ul>	f.
67	11/2023 07	Laporan HKI	<ul style="list-style-type: none"> <li>- refisi kembali format penulisan</li> </ul>	f.
8.	19/23 /9	Laporan HKI	Oke	f.

	19/2023 /9	Ace	siap ujian	f.
--	---------------	-----	------------	----

Tegal, 2022  
Dosen Pembimbing,



Gimanjar Wiro Sasmito, M.Kom  
NIP.Y. 10.007.032

Lampiran 4. Lembar Bimbingan Pembimbing 2



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

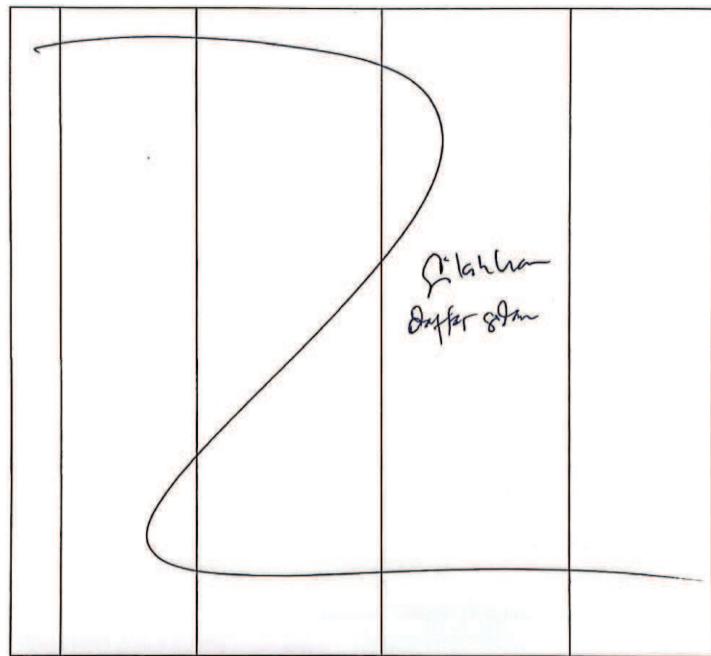
**LEMBAR BIMBINGAN TUGAS AKHIR**

**Nama** : Wahyu Indrawan  
**NIM** : 18090108  
**No. Ponsel** : 085326699718  
**Judul TA** : Aplikasi *Location Based Service*(LBS) Pertolongan Memanggil Pemadam Kebakaran (Studi Kasus: Kota Tegal)

**Dosen Pembimbing II** : Dega Surono Wibowo, S.T., M.Kom.

NO	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	9/5/22		<ul style="list-style-type: none"> <li>- Laporan motif dulu</li> <li>- buat use case komplit</li> <li>- Saya Nggak nunggu dulu</li> </ul>	<i>f</i>
	6/4/22		<ul style="list-style-type: none"> <li>- URL sesuai dengan ketentuan yg berlaku</li> <li>- buat mock up</li> </ul>	<i>f</i>
	4/4/22		<ul style="list-style-type: none"> <li>- Usecase - fix</li> <li>kehaluan ada tambahan silahkan ditambahkan</li> <li>- pilih verifikasi penerima</li> <li>Caranya di tiap personil / di admin -</li> <li>- Cari tau setting ada kejadian yg berangkat berapa personel</li> </ul>	<i>f</i>

NO	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
	3/6 2022	aplikasi dilanjutkan	- kalau bisa sudah dibuild di android	X -
	10/6 2022	aplikasi diteruskan		X
	22/6 2022	aplikasi diteruskan	- admin harus bisa memastikan bahwa user terejeksi asli (paling tidak ada pembanding laporan)	X
	13/6 2023	laporan tambah pencarian tgl kejadian	bantuan manual buat dr teknikal	X -
	23/6 2023	Silahkan datter HCU, apke diupload ke play store		X -
	13/7 2023	Sistematisasi penulisan diperbaiki		X
	14/7 2023	Laporan selesai		X



Tegal, 13 -7- 2023  
Dosen Pembimbing,

  
Dega Surono Wibowo, S.T., M.Kom  
NIPY. 06.014.183

## Lampiran 5. Surat Pernyataan Pengajuan HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta :

- |                 |   |  |
|-----------------|---|--|
| 1. N a m a      | : | Wahyu Indrawan   |
| Kewarganegaraan | : | Indonesia  |
| Alamat          | : | Jalan Brawijaya, RT 04 RW 01, Kelurahan Muarareja, Kecamatan Tegal Barat, Kota Tegal, Provinsi Jawa Tengah |
| 2. N a m a      | : | Ginanjar Wiro Sasmoro, M.Kom.  |
| Kewarganegaraan | : | Indonesia  |
| Alamat          | : | Jl. Raya Kluwut Timur No 24, RT 03 RW 02, Kluwut Kec. Bulakamba, Kab.Brebes                                |
| 3. N a m a      | : | Dega Surono Wibowo, M.Kom.   |
| Kewarganegaraan | : | Indonesia  |
| Alamat          | : | Perumahan Sapphire Regency Blok H No 1, RT 004 RW 001, Kelurahan Pulosari, Kecamatan Brebes                |

Dengan ini menyatakan bahwa:

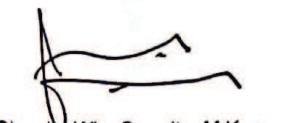
1. Karya Cipta yang saya mohonkan:  
Berupa : Program Komputer  
Berjudul : Aplikasi *Location Based Service (IBS)* Pertolongan Memanggil Pemadam Kebakaran  
(Studi Kasus : Kota Tegal)
  - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
  - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
  - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
  - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
  - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
  - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
  - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
  - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
  - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 27 Juni 2023



Wahyu Indrawan  
Pemegang Hak Cipta\*



Ginanjar Wiro Sasmito, M.Kom.  
Pemegang Hak Cipta\*



Dega Surono Wibowo, M.Kom.  
Pemegang Hak Cipta\*

Pemegang Hak Cipta\*

\* Semua pemegang hak cipta agar menandatangani di atas materai.

## Lampiran 6. Surat Pengalihan HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

- 1 Nama : Wahyu Indrawan  
Kewarganegaraan : Indonesia  
Alamat : Jalan Brawijaya, RT 04 RW 01, Kelurahan Muarareja, Kecamatan Tegal Barat, Kota Tegal, Provinsi Jawa Tengah
- 2 Nama : Ginanjar Wiro Sasmito, M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Raya Kluwul Timur No 24, RT 03 RW 02, Kluwul Kec. Bulakamba, Kab. Brebes
- 3 Nama : Dega Surono Wibowo, M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Perumahan Sepphire Regency Blok H No 1, RT 004 RW 001, Kelurahan Pulosari, Kecamatan Brebes

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)  
Politeknik Harapan Bersama  
Alamat : Jl. Mataram N0.9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Aplikasi *Location Based Service* (IBS) Pertolongan Memanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal) untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 27 Juni 2023

Pemegang Hak Cipta  
  
(Dr. Ajdi Budi Riyanta, S.Si, M.T.)

  
Wahyu Indrawan  
  
(Ginanjar Wiro Sasmito, M.Kom.)  
  
(Dega Surono Wibowo, M.Kom.)

Lampiran 7. Data Anggota Damkar Kota Tegal

**DARSTAR NAMA ANGGOTA PEMADAM KEBAKARAN  
SATUAN POLISI PAMONG PRAJA KOTA TEGAL**

NO	NAMA	NIP	TEMPAT, TANGGAL LAHIR	PANGKAT / GOL. RUANG	PENDIDIKAN	KETERANGAN
1	PARIKHIN, S.I.P	2	3	4	5	6
1	PARIKHIN, S.I.P	19650117 199103 1 005	Tegal, 17-01-1965	Penata Tk. I (III/d)	SI	7
2	TEGJUH SUPRIYANTO, SH	19660228 199303 1 007	Pati, 28-02-1966	Penata Tk. I (III/d)	SI	-
3	MOELVADI	19640917 198807 1 001	Tegal, 17-09-1964	Penata Muda Tk. I (III/b)	SLTA	-
4	TARIDI	19640917 198807 1 007	Tegal, 18-07-1973	Penata Muda Tk. I (III/b)	SMK	-
5	CASMIDI	19650818 199303 1 009	Tegal, 18-08-1965	Penata Muda Tk. I (III/b)	SLTA	-
6	TASORI RT	19650802 198803 1 002	Tegal, 02-08-1965	Pengatur Tk. I (IV/d)	SLTA	-
7	PARIWOTO	19670208 198803 1 007	Tegal, 08-02-1967	Pengatur Tk. I (IV/d)	SLTA	-
8	FAKHRRUDIN	19740317 200701 1 010	Tegal, 17-03-1974	Pengatur Tk. I (IV/d)	SMK	-
9	EKO NIURDIVONO	19750127 200801 1 004	Tegal, 27-01-1975	Pengatur Tk. I (IV/d)	SMK	-
10	YULIUS ANDI RINJANI	19730525 200801 1 012	Tegal, 25-05-1973	Pengatur Tk. I (IV/d)	SLTA	-
11	SLAMET MATORI	19611126 199003 1 011	Jepara, 26-11-1961	Pengatur Muda (IV/a)	SD	-
12	ADE SUWARNO	19811105 201001 1 002	Tegal, 05-11-1981	Pengatur Muda (IV/a)	SLTP	-
13	RUDI HARSONO	19840710 201001 1 003	Tegal, 10-07-1984	Pengatur Muda (IV/a)	SLTP	-
14	DARYONO	19860725 201001 1 001	Tegal, 25-07-1986	Pengatur Muda Tk. I (IV/b)	SLTA	-
15	KARYO	19650305 200604 1 011	Tegal, 05-03-1965	Pengatur Muda (IV/a)	SLTP	-
16	MOH. SLAMET	19640908 200801 1 011	Tegal, 09-08-1964	Pengatur Muda Tk. I (IV/b)	SLTP	-
17	BAGUS SULISTIYAWAN	19750316 200801 1 004	Tegal, 16-03-1975	Pengatur Tk. I (IV/d)	SLTA	-
18.	SOUKHIN	19850917 201001 1 001	Tegal, 19-09-1985	Juru Tk. I (V/c)	SD	-
19.	WAKRAD ARDIYANTO	19810521 201001 1 002	Tegal, 21-05-1981	Juru Tk. I (V/d)	SD	-
20.	FITRIAN PANU FEARI	-	Tegal, 14-02-1997	-	SMK	Bampol Satpol PP
21.	FURQON ERHA SETIAWAN	-	Tegal, 12-12-1997	-	S1	Bampol Satpol PP
22.	MOHAMAD IMRON	-	Tegal, 24-03-1997	-	SMK	Bampol Satpol PP
23.	MUHAMMAD KHAMDAN A	-	Tegal, 19-07-1997	-	S1	Bampol Satpol PP
24.	ANISA SEPTIANI WIDORETNO	-	Tegal, 24-09-1993	-	D3	Staf Administrasi
25.	MILLENIARDO MARHAPUTRA P	-	Tegal, 12-06-2000	-	SMA	Staf Administrasi

a.n Kepala Satuan Polisi Pamong Praja Kota Tegal  
Kepala Bidang Pelayanan dan Pencegahan Kebakaran



## LAMPIRAN 8. Manual Book

**MANUAL BOOK**

**APLIKASI LOCATION BASED SERVICE (LBS)**

**PERTOLONGAN MEMANGGIL PEMADAM**

**KEBAKARAN (STUDI KASUS : KOTA TEGAL)**

Oleh :

Wahyu Indrawan

Ginanjar Wiro Sasmito, M.Kom.

Dega Surono Wibowo, S.T., M.Kom.

## **1. Pendahuluan**

### **1.1.Tujuan Pembuatan Dokumen**

Dokumen *user manual* sistem Aplikasi *Location Based Service* (LBS)

Pertolongan Memanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal) Berbasis *Mobile*” ini dibuat untuk tujuan sebagai berikut :

1. Tujuan dari membuat aplikasi ini untuk membantu waraga segera menghubungi pemadam kebakaran sehingga dapat langsung membantu keadaan darurat.

### **1.2.Deskripsi Umum Sistem**

#### **1.2.1. Deskripsi Umum Aplikasi**

Deskripsi umum aplikasi meliputi deskripsi umum Aplikasi untuk membantu waraga menghubungi atau melaporakan kejadian darurat ke pemdam kebakaran.

#### **1.2.2. Deskripsi Umum Kebutuhan Aplikasi**

Deskripsi umum kebutuhan aplikasi yang akan diimplementasikan meliputi semua informasi yang bersifat teknis yang menjadi acuan dalam pengembangan aplikasi.

#### **1.2.3. Deskripsi Dokumen**

Dokumen ini dibuat untuk memberikan panduan penggunaan Aplikasi *Location Based Service* (LBS) Pertolongan Memanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal). Dokumen ini berisikan informasi sebagai berikut :

## **1. BAB I**

Berisi informasi umum yang merupakan bagian pendahuluan, yang didalamnya meliputi pembuatan dokumen, deskripsi umum sistem serta deskripsi dokumen.

## **2. BAB II**

Berisi perangkat yang dibutuhkan untuk penggunaan sistem prediksi penjualan produk meliputi perangkat lunak dan perangkat keras.

## **3. BAB III**

Aplikasi *Location Based Service* (LBS) Pertolongan Memanggil Pemadam Kebakaran (Studi Kasus : Kota Tegal).

### **2. Perangkat yang dibutuhkan**

#### **2.1. Perangkat Lunak**

Perangkat lunak yang dibutuhkan adalah :

1. Windows 10 sebagai *Operating System*
2. *Google Chrome* sebagai *Web Browser*

#### **2.2. Perangkat Keras**

Perangkat keras yang digunakan adalah :

1. Laptop ASUS processor AMD A6 RAM 4GB HDD 1TB SSD 120GB
2. Keyboard sebagai peralatan antarmuka.
3. Monitor sebagai peralatan antarmuka.
4. Smartphone dengan Sistem Operasi Android

### **3. MENU DAN CARA PENGGUNAAN**

#### **3.1. Struktur Menu**

Struktur halaman pada sistem “Aplikasi Monitoring dan Rekomendasi Tempat Praktik Kerja Industri Terdekat Menggunakan Metode *Haversine Formula*” adalah sebagai berikut :

1. Halaman Menu Warga
  - a. Menu *Login*
  - b. Menu Register
  - c. Menu Home
  - d. Menu Informasi
  - e. Menu History
  - f. Menu Users
2. Halaman Menu Petugas
  - a. Menu Login
  - b. Menu Home
  - c. Informasi
  - d. Menu Profil
3. Halaman Menu Admin Prakerin
  - a. Menu Login
  - b. Menu Dashboard
  - c. Menu Laporan Masuk
  - d. Menu User Baru
  - e. Menu Laporan Selesai

- f. Menu Warga
- g. Menu Petugas
- h. Menu Alat-Alat Pemadam

### **3.2.Pengguna**

Pada bagian ini akan dijelaskan menu-menu yang dapat digunakan oleh pengguna

#### **3.2.1. Cara membuka aplikasi**

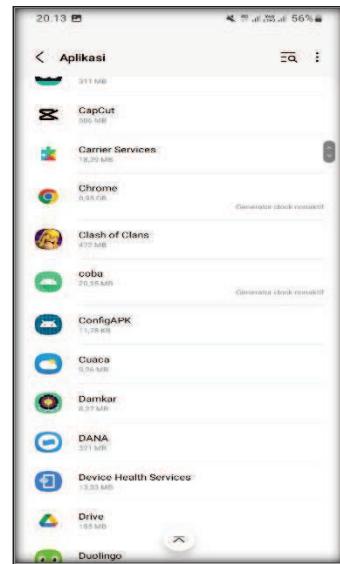
Untuk memulai menggunakan aplikasi prediksi penjualan produk asalah sebagai berikut :

1. Buka folder yang file APK dan install APK.



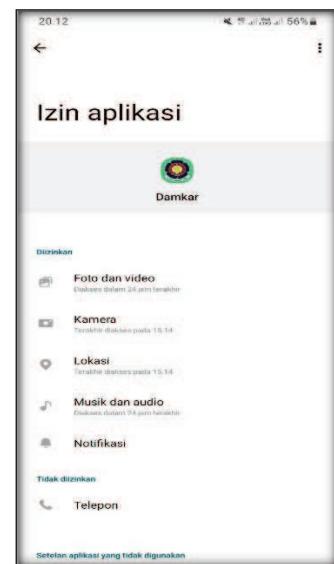
Gambar 3.1 Tampilan *Login*

2. Kemudian setelah berhasil diinstall, buka pengaturan pada pilih aplikasi.



Gambar 3.2. Tampilan Sesudah Install Aplikasi

3. Pilih izin aplikasi.



Gambar 3.3. Tampilan Pilih Izin

A. Panduan Untuk Warga

a. Halaman *Login*

Membuka aplikasi dan pada tampilan pertama terdapat tampilan *Login*, bagi warga yang sudah memiliki akun dapat

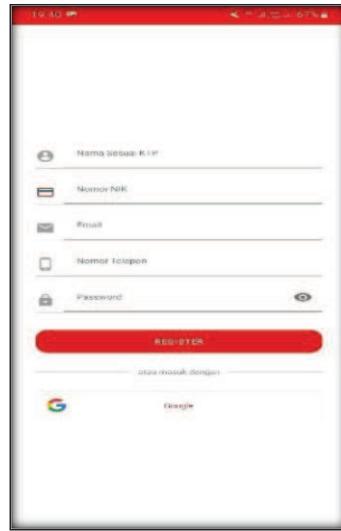
*login* terlebih dahulu agar bisa menjalankan aplikasi dan dapat melaporkan kejadian menggunakan aplikasi.



Gambar 3.4. Tampilan *Login*

b. Halaman Register

Jika warga belum memiliki akun maka pada halaman *login* pilih register untuk melakukan registrasi. Pada halaman register warga diminta menginputkan nama sesuai KTP, NIK, Email, No Hp, dan Password untuk dapat *login* di aplikasi.



Gambar 3.5. Halaman Register

c. Halaman *Home*

Pada halaman *home* warga dapat memilih metode pelaporan ke pemadam kebakaran yaitu laporan melalui aplikasi atau telfon secara langsung. Laporan secara aplikasi maka warga akan masuk ke halaman laporan dan jika warga memilih telfon langsung maka warga akan pergi ke panggilan.



Gmabar 3.6. Halaman *Home*

d. Halaman Kirim Laporan

Sebelum masuk Halaman Kirim Laporan warga langsung diminta mefoto kejadian sebagai bukti adanya kejadian setelah menfoto langsung masuk ke halaman kejadian dan langsung terisi alamat kejadian. Pada halaman kirim laporan terdapat alamat kejadian yang terisi otomatis, kategori yang harus di pilih dan mengisi deskripsi kejadian setelah itu warga dapat kirim laporan kejadian.



Gambar 3.7. Halaman Kirim Laporan

e. Halaman *History*

Halaman *history* terdapat laporan yang pernah di laporkan oleh warga itu sendiri.



Gambar 3.8. Halaman *History*

f. Halaman Detail *History*

Pada halaman detail *history* berisi detail dari history kejadian yang pernah di laporkan oleh warga tersebut.



Gambar 3.9. Halaman Detail *History*

g. Halaman Informasi

Pada halaman informasi terdapat informasi mengenai

pemadam kebakaran Kota Tegal dan lokasi pemdam kebakaran Kota Tegal.



Gambar 3.10. Halaman Informasi

h. Halaman *User Profile*

Halaman User berisi tentang data dari user yang login di aplikasi ini berisi informasi nama, NIK, email, no Hp, dan alamat.



Gambar 3.11. Halaman *User Profile*

## B. Panduan Untuk Petugas

### a. Halaman *Login*

Petugas harus login terlebih dahulu agar bisa menangani kasus menggunakan aplikasi, pada halaman *login* terdapat dua inputan yaitu *email* dan *password*, untuk *email* dan *password* yang di daftarkan oleh admin untuk masing masing petugas.



Gambar 3.12. Halaman *Login*

### b. Halaman *Home*

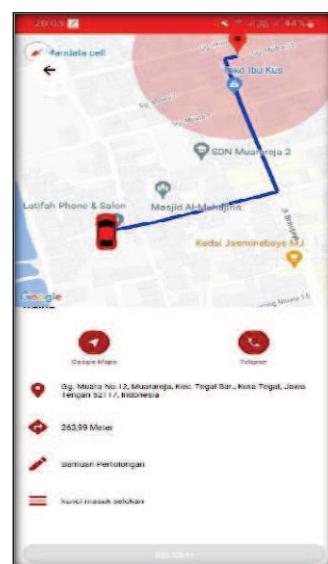
Pada halaman *home* terdapat laporan yang telah di terima oleh admin maka akan masuk ke aplikasi petugas dan tampil pada menu *home*.



Gambar 3.13. Halaman Home

### c. Halaman Laporan

Halaman Laporan berisi detail laporan yang sudah diterima admin, terdapat maps menuju lokasi, alamat lokasi, jarak lokasi kejadian, kategori, dan deskripsi kejadian.



Gambar 3.14. Halaman Laporan

d. Halaman Informasi

Pada halaman informasi terdapat informasi mengenai pemadam kebakaran Kota Tegal dan lokasi pemdam kebakaran Kota Tegal.



Gambar 3.15. Halaman Informasi

e. Halaman *User Profile*

Halaman User berisi tentang data dari user yang login di aplikasi ini berisi informasi nama, NIK, email, no Hp, dan alamat.

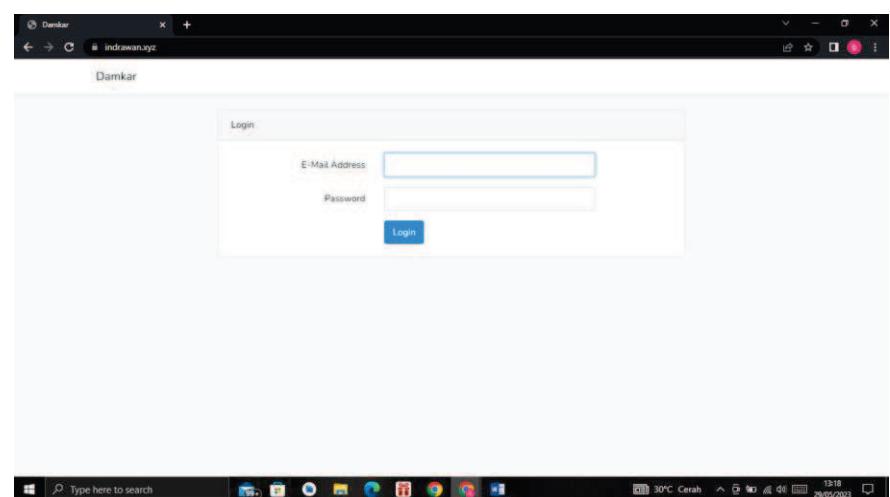


Gambar 3.16. Halaman *User Profile*

### C. Panduan Untuk Admin

#### a. Halaman *Login*

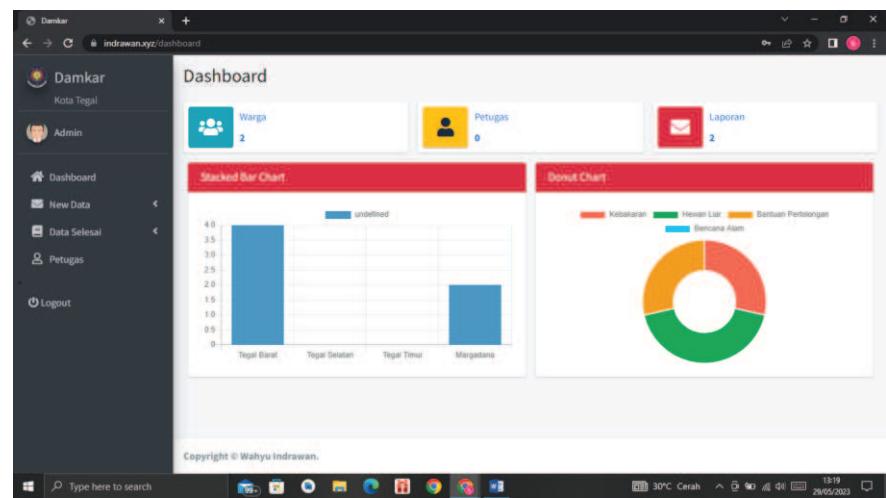
Pada tampilan ini pertama dari website admin adalah *login*, admin harus *login* terlebih dahulu sebelum masuk ke website. Admin harus menginputkan nama dan email admin.



Gambar 3.17. Halaman *Login*

## b. Halaman Dashboard

Halaman dashboard adalah tampilan utama setelah login, pada halaman ini terdapat jumlah laporan masuk, jumlah warga baru yang *registrasi*, dan jumlah petugas. Pada tampilan ini juga terdapat grafik laporan setiap kecamatan di Kota Tegal.



Gambar 3.18. Halaman Dashboard

## c. Halaman Laporan Masuk

Pada tampilan ini berisi laporan kejadian yang di kirim oleh warga dan masuk ke admin untuk di verifikasi di terima atau di tolak laporan tersebut.

Damkar | Dashboard

Kota Tegal

Admin

Dashboard

New Data

Data

Petugas

Logout

Tabel Laporan Masuk

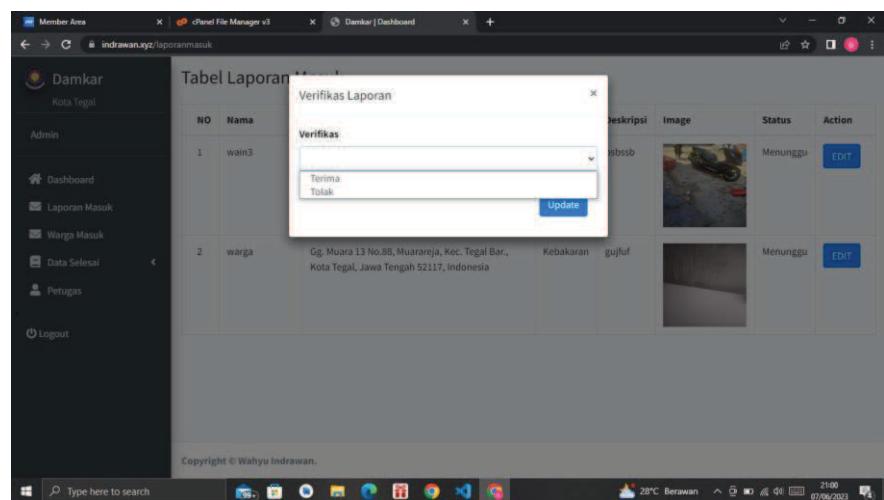
NO	Nama	lokasi	kategori	Deskripsi	Image	Status	Action
29	wain3	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	Kebakaran	bbbbbb		Menunggu	<button>EDIT</button>
31	warga	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	Kebakaran	gujuf		Menunggu	<button>EDIT</button>

Copyright © Wahyu Indrawan.

Gambar 3.19. Halaman Laporan Masuk

d. *Form Verifikasi Laporan*

Pada tampilan *form verifikasi* laporan admin menuntukan laporan itu diterima atau tidak, dengan cara admin mengeklik *buttom edit* dan *form verifikasi* laporan akan muncul.



Gambar 3.20. *Form Verifikasi Laporan*

e. Halaman Warga Baru

Pada tampilan ini berisi table warga yang telah melakukan *registrasi* dan harus di verifikasi oleh admin.

NO	Nama	Email	Status	Action
4	warga2	warga02@gmail.com	Menunggu	<button>VERIFIKASI</button>
5	warga03	warga03@gmail.com	Menunggu	<button>VERIFIKASI</button>

Gambar 3.21. Halaman Warga Baru

f. *Form Verifikasi* warga

Pada tampilan *form verifikasi* warga baru admin menuntukan warga itu diterima atau tidak, dengan cara admin mengeklik *button* verifikasi dan *form verifikasi* warga baru akan muncul.

Verifikas

Status

Menunggu

Action

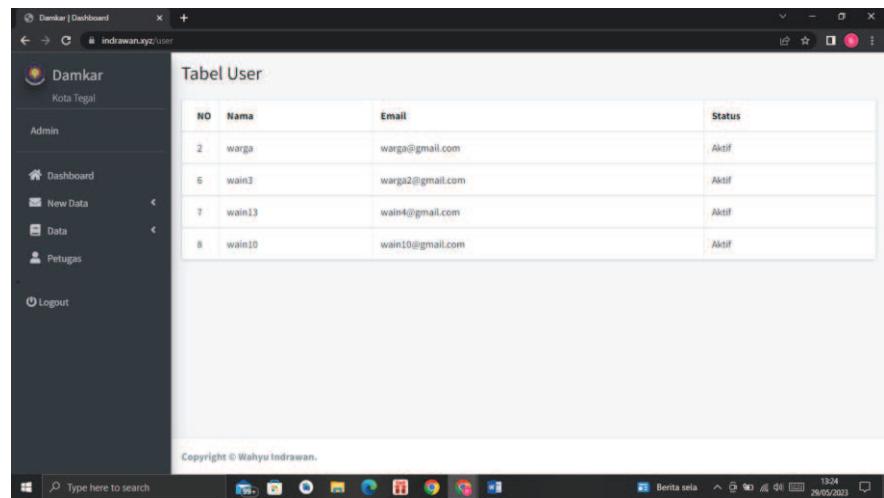
VERIFIKASI

Gambar 3.22. *Form Verifikasi* warga

g. *Halaman* warga Aktif

Pada halaman ini berisi warga yang akunya telah aktif dan

telah di verifikasi oleh admin.



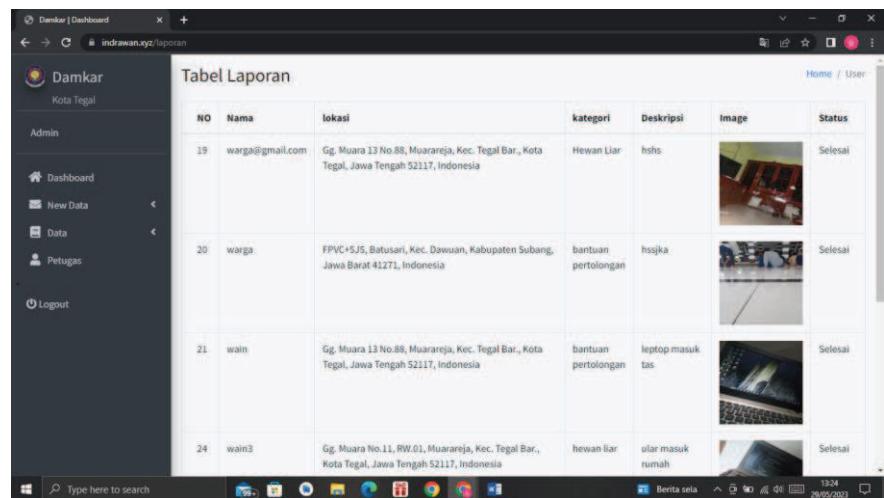
The screenshot shows a web application interface for 'Damkar Kota Tegal'. The left sidebar has a 'Admin' section with 'Dashboard', 'New Data', 'Data', and 'Petugas' buttons, and a 'Logout' button. The main content area is titled 'Tabel User' and contains a table with the following data:

NO	Nama	Email	Status
2	warga	warga@gmail.com	Aktif
6	wain3	warga2@gmail.com	Aktif
7	wain13	wain4@gmail.com	Aktif
8	wain10	wain10@gmail.com	Aktif

Gambar 3.23. Halaman Warga Baru

#### h. Halaman Laporan Selesai

Pada halaman laporan selesai berisi tabel semua laporan yang telah di tangan oleh petugas damkar Kota Tegal.



The screenshot shows a web application interface for 'Damkar Kota Tegal'. The left sidebar has a 'Admin' section with 'Dashboard', 'New Data', 'Data', and 'Petugas' buttons, and a 'Logout' button. The main content area is titled 'Tabel Laporan' and contains a table with the following data:

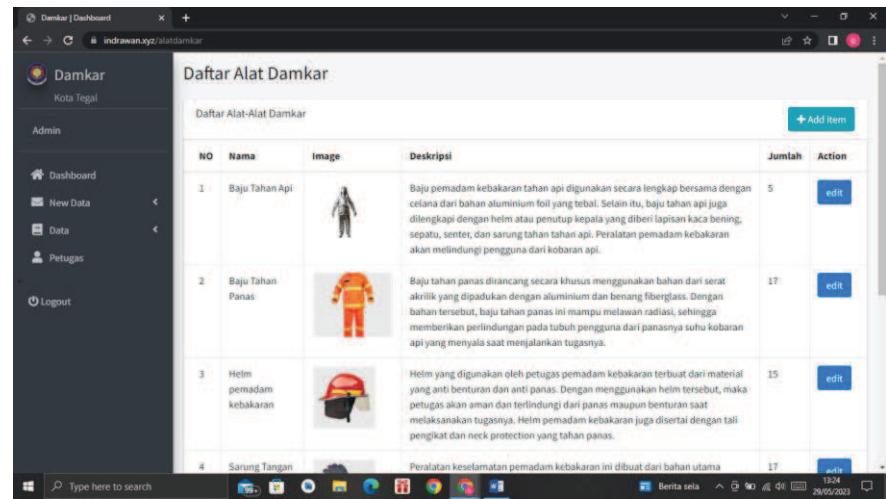
NO	Nama	lokasi	kategori	Deskripsi	Image	Status
19	warga@gmail.com	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	hewan liar	hsbs		Selesai
20	warga	FPVC+5J5, Batusari, Kec. Dawuan, Kabupaten Subang, Jawa Barat 41271, Indonesia	bantuan pertolongan	hssjka		Selesai
21	wain	Gg. Muara 13 No.88, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	bantuan pertolongan	laptop masuk tas		Selesai
24	wain3	Gg. Muara No.11, RW.01, Muarareja, Kec. Tegal Bar., Kota Tegal, Jawa Tengah 52117, Indonesia	hewan liar	ular masuk rumah		Selesai

Gambar 3.24. Halaman Laporan Selesai

#### i. Halaman Alat-Alat Damkar

Pada tampilan ini berisi tabel alat alat yang dimiliki pemadam kebakaran Kota Tegal dan jumlah yang di miliki.

Pada tampilan ini juga terdapat tombol untuk menambah alat-alat dan tombol untuk mengedit jumlah alat pemadam kebakaran.

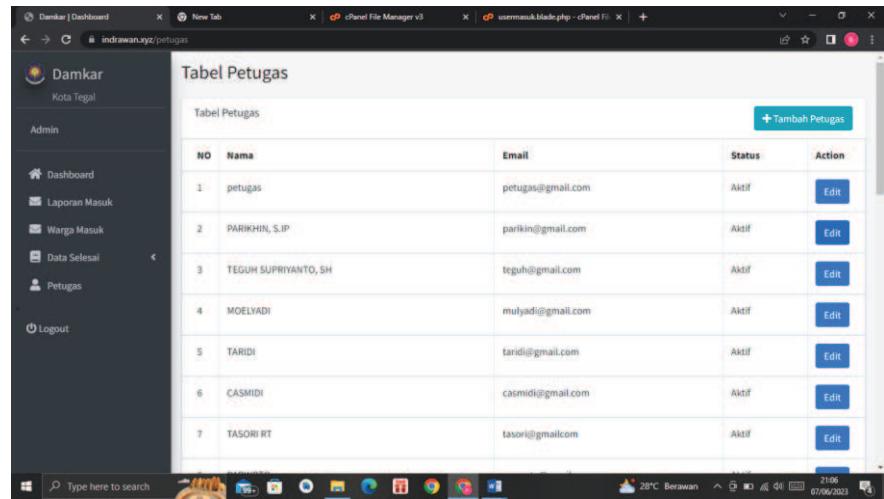


NO	Nama	Image	Deskripsi	Jumlah	Action
1	Baju Tahan Api		Baju pemadam kebakaran tahan api digunakan secara lengkap bersama dengan celana dari bahan aluminium foil yang tebal. Selain itu, baju tahan api juga dilengkapi dengan helm atau penutup kepala yang diberi lapisan kaca bening, sepatu, senter, dan sarung tangan tahan api. Peralatan pemadam kebakaran akan melindungi pengguna dari kobaran api.	5	<button>edit</button>
2	Baju Tahan Panas		Baju tahan panas dirancang secara khusus menggunakan bahan dari serat akrilik yang dipadukan dengan aluminium dan benang fiberglass. Dengan bahan tersebut, baju tahan panas ini mampu melawan radiasi, sehingga memberikan perlindungan pada tubuh pengguna dari panasnya suhu kobaran api yang menyala saat menjalankan tugasnya.	17	<button>edit</button>
3	Helm pemadam kebakaran		Helm yang digunakan oleh petugas pemadam kebakaran terbuat dari material yang anti benturan dan anti panas. Dengan menggunakan helm tersebut, maka petugas akan aman dan terlindungi dari panas maupun benturan saat melaksanakan tugasnya. Helm pemadam kebakaran juga disertai dengan tali pengikat dan neck protection yang tahan panas.	15	<button>edit</button>
4	Sarung Tangan		Peralatan keselamatan pemadam kebakaran ini dibuat dari bahan utama	17	<button>edit</button>

Gambar 3.25. Halaman Alat-Alat Damkar

#### j. Halaman Petugas

Pada halaman ini berisi table petugas yang aktif. Dan terdapat *bottom* tambah petugas untuk menginputkan petugas yang baru, dan terdapat *bottom* edit untuk mengedit status petugas masih aktif atau tidak aktif.

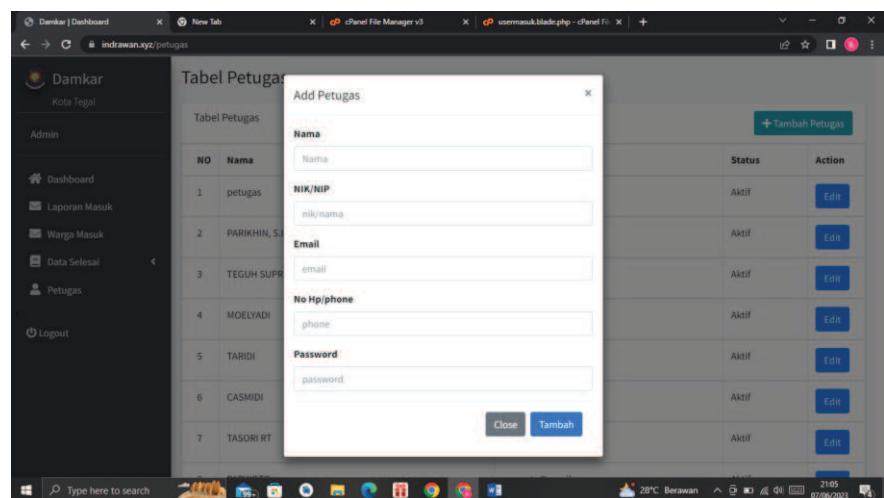


NO	Nama	Email	Status	Action
1	petugas	petugas@gmail.com	Aktif	<button>Edit</button>
2	PARIKHIN, S.I.P	parikhin@gmail.com	Aktif	<button>Edit</button>
3	TEGUH SUPRIYANTO, SH	teguh@gmail.com	Aktif	<button>Edit</button>
4	MOELYADI	mulyadi@gmail.com	Aktif	<button>Edit</button>
5	TARIDI	taridi@gmail.com	Aktif	<button>Edit</button>
6	CASMIDI	casmidi@gmail.com	Aktif	<button>Edit</button>
7	TASORI RT	tasori@gmail.com	Aktif	<button>Edit</button>

Gambar 3.26. Halaman Petugas

#### k. Form Tambah Petugas

Form tambah petugas untuk menambahkan petugas yang baru, form ini akan muncul ketika admin mengeklik button tambah petugas. Pada form ini admin harus mengisi nama, NIK atau NIP, no Hp, dan Password



The modal window is titled 'Add Petugas'. It contains the following fields:

- Nama:**
- NIK/NIP:**
- Email:**
- No Hp/phone:**
- Password:**

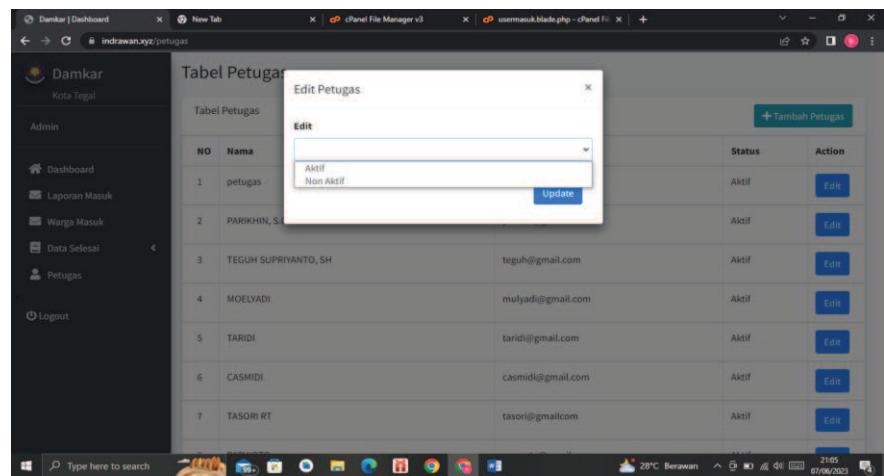
At the bottom of the modal are two buttons: 'Close' and 'Tambah'.

Gambar 3.27. Form Tambah Petugas

#### l. Form Edit Status Petugas

Tampilan *form* edit status petugas untuk mengedit status dari

petugas apakah masih aktif atau tidak, caranya dengan admin mengeklik *button* edit pada table petugas dan pilih aktif atau non aktif.



Gambar 2.28. Form Edit Status Petugas

# LAMPIRAN 9.

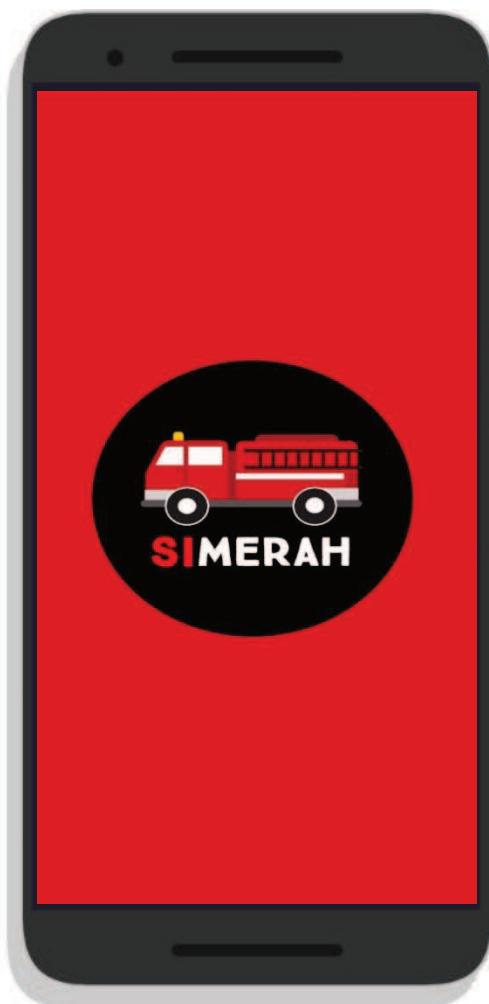
## Dokumen Technical

**DOKUMEN TECHNICAL**

**APLIKASI LOCATION BASED SERVICE (LBS) PERTOLONGAN**

**MEMANGGIL PEMADAM KEBAKARAN (STUDI KASUS : KOTA**

**TEGAL)**



## **Profil**

Pengenalan Aplikasi Locatin Based Service (LBS) Pertolongan Memanggil Pemadam Kebakaran (Studi Kasusu : Kota Tegal). Diharapkan dengan adanya aplikasi ini memanggil pemadam kebaran lebih mudah dan lebih cepat lagi mengetahu lokasi kejadian. Aplikasi ini dibuat berbasis android untuk warga dan petugas dan web untuk admin yang mana ketika aplikasi ini dipublikasi secara luas bisa digunakan oleh warga Kota Tegal yang ingin memanggil pemadam kebakaran. Aplikasi ini masih jauh dari kata sempurna diharapkan dikemudian hari aplikasi bisa dikembangkan dengan lebih baik agar tampilanya lebih bagus dan menarik lagi

## **Latar Belakang**

Sektor telekomunikasi saat ini berkembang pesat karena perkembangan teknologi global, terutama di bidang telepon seluler dan internet. Salah satunya adalah kemajuan aplikasi mobile berbasis Android. Perkembangan aplikasi Android belakangan ini telah membawa perubahan di berbagai bidang, termasuk bidang pelayanan masyarakat. Saat ini, kebanyakan orang sudah memiliki smartphone sendiri Dengan berkembangnya teknologi, mencari tahu lokasi suatu tempat dengan menggunakan smartphone bukanlah hal yang mustahil.

Petugas pemadam kebakaran atau biasa disingkat Damkar merupakan pelayanan masyarakat yang bertugas memadamkan api, menyelamatkan, dan mengatasi bencana atau kejadian lainnya. Sebagai penolong yang dibutuhkan pada saat terjadi bencana kebakaran, namun tidak jarang menghubungi petugas pemadam kebakaran sangat sulit. Banyak orang yang merasa sulit untuk menghubungi petugas pemadam kebakaran. Dan tidak jarang juga petugas

pemadam kebakaran kesulitan untuk menuju ke titik lokasi karena tidak jelas informasi lokasi yang diberikan.

Teknologi yang tepat digunakan sebagai media pencari lokasi pada smartphone adalah Global Positioning System (GPS) yang sudah dapat digunakan pada smartphone dan salah satunya menggunakan platform android[3]. Oleh karena itu, aplikasi pemadam kebakaran yang akan dibuat saya dilengkapi dengan navigasi Global Positioning System (GPS) untuk memudahkan masyarakat dalam melakukan panggilan pemadam kebakaran jika terjadi bencana kebakaran atau kejadian lainnya serta memudahkan petugas pemadam kebakaran untuk sampai ke titik yang dituju dengan cepat dan tepat. Aplikasi ini diberi nama Aplikasi Location Based Service (LBS) Pertolongan Memanggil Pemadam Kebakaran. Aplikasi pemadam kebakaran ini dibangun menggunakan Android Studio dan menggunakan Google Maps Api untuk menampilkan peta dan rute dari suatu tempat, menggunakan Global Positioning System (GPS) untuk mengetahui posisi pengguna yang melakukan panggilan saat ini.

### **Manfaat**

Memudahkan dalam menghubungi pemadam kebakaran di wilayah Kota Tegal sehingga dapat langsung membantu atau menangani keadaan darurat.

### **Spesifikasi Teknis**

Berikut spesifikasi untuk instalasi aplikasi :

1. *Xampp v3.2.2*
2. *MySQL Database*
3. *PHP v.7.4*

4. *Ms. Visual Studio Code*

5. *Google Chrome*

6. *Android Studio*

## **Rancangan Basis Data**

### **1. Tabel Users**

Column	Typ	Indexes
	e	
Id	int(11) Auto Increment	primary
name	Varchar(191)	
nik	Varchar(191)	unique
email	Varchar(191)	unique
phone	Varchar(191)	unique
email_verified_at	timestamp	
password	Varchar(191)	
Remember_token	Varchar(191)	
level	Varchar(191)	
created_at	timestamp	
update_at	timestamp	

### **2. Tabel Laporans**

Column	Type	Indexes

Id	int(11) Auto Increment	primary
name	varchar(191)	
lokasi	varchar(191)	
lat	varchar(191)	
long	varchar(191)	
kategori	varchar(191)	
deskripsi	varchar(191)	
image	varchar(191)	
petugas	varchar(191)	
status	varchar(191)	
user_id	int(10)	unique
hp	varchar(15)	
created_at	timestamp	
updated_at	timestamp	

### 3. Tabel Petugass

Column	Type	Indexes
Id	int(11) Auto Increment	primary
name	varchar(191)	
nik	varchar(191)	
email	varchar(191)	
phone	varchar(15)	
email_verified_at	timestamp	
password	varchar(191)	
Remember_token	varchar(191)	
level	varchar(191)	
created_at	timestamp	
updated_at	timestamp	

### 4. Tabel Alats

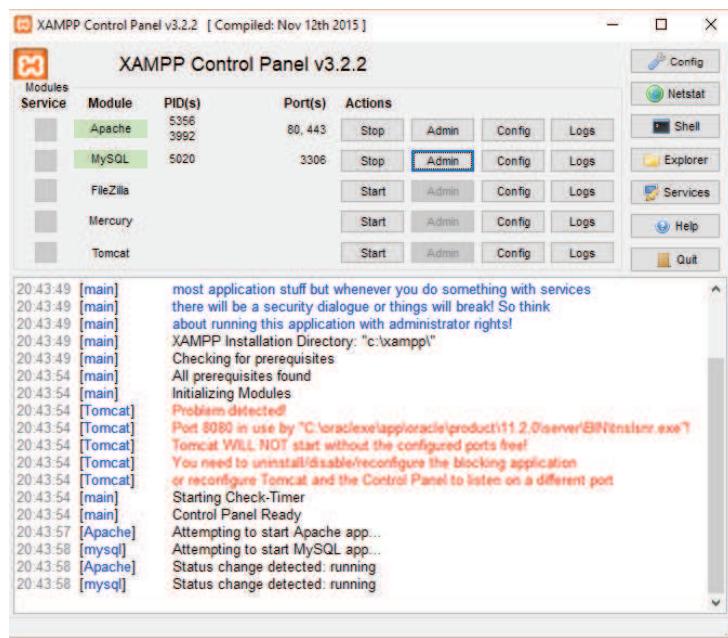
Column	Type	Indexes
id	int(11) Auto Increment	primary
name	varchar(191)	
image	varchar(191)	
deskripsi	varchar(5000)	
jumlah	varchar(191)	
status_akun	timestamp	

last_login	timestamp	
------------	-----------	--

## Instalasi dan konfigurasi

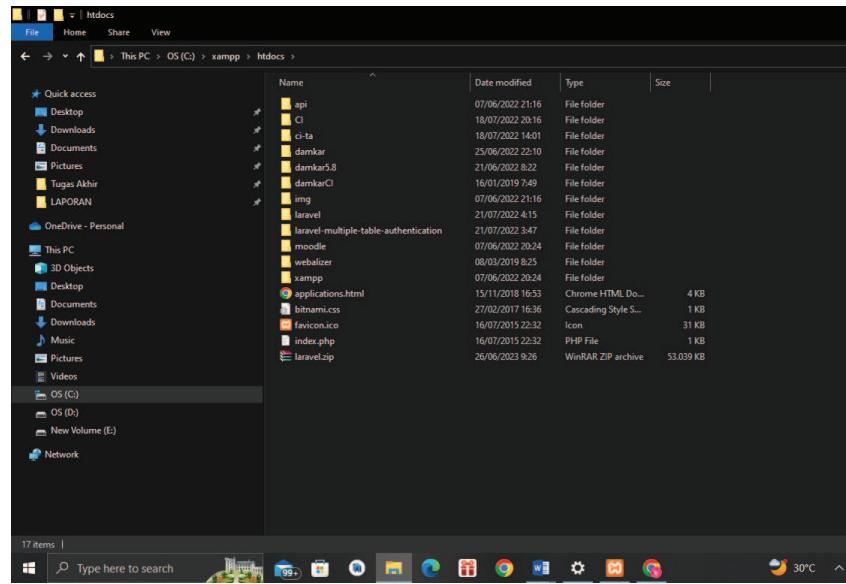
### 1. Konfigurasi Web Server Xampp

- Buka *Control Panel Xampp*, lalu klik tombol *start* pada *Apache* dan *MySQL*.



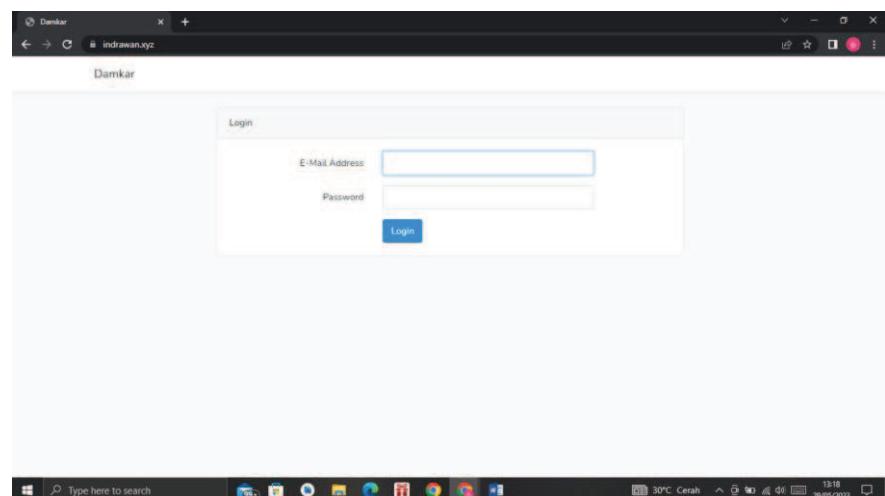
Gambar 1. Xampp Control Panel

- Buka *file project codeigniter* yang telah dibuat pada direktori C/Xampp/htdocs.



Gambar 2. File project

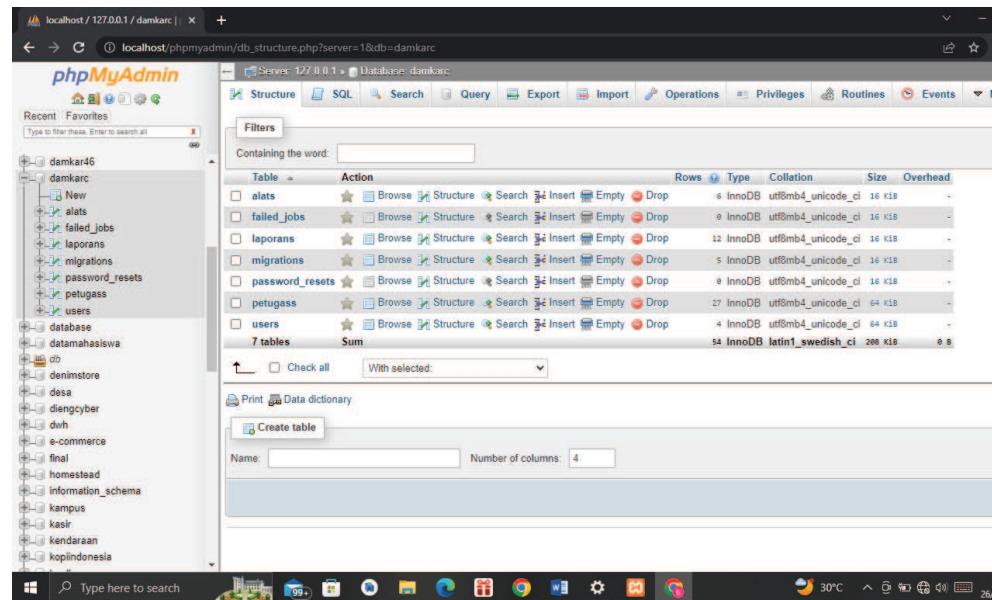
- c. Buka browser ketikan `http://localhost/(nama folder project)` untuk menampilkan projek.



Gambar 3. Menampilkan Projek Website

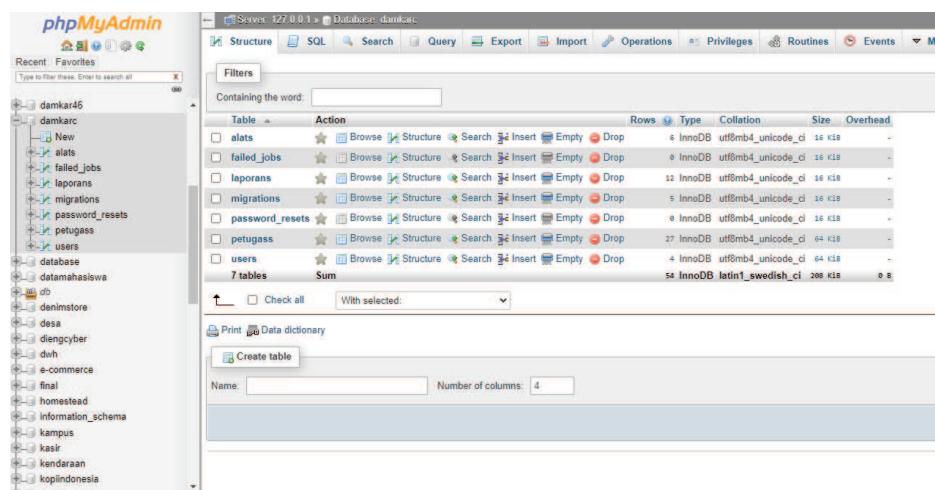
## 2. Konfigurasi Database

- Buka browser dan ketikan <http://localhost/phpmyadmin/> untuk mengakses database MySQL.



Gambar 4. phpMyAdmin

- Buat database baru dengan nama disesuaikan dan import file database yang sudah ada. Seperti pada Gambar

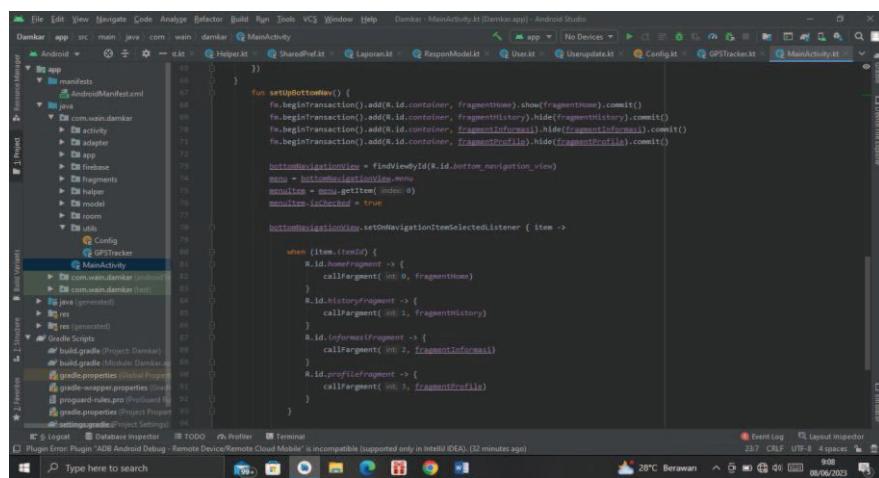


Gambar 5. Database Mysql

### 3. Warga

#### a. MainActivity.kt

Fungsi dari *source code* ini adalah untuk membuat tampilan menu bar di bawah agar bisa bepindah fragmen.



```
fun setupBottomNavigationView() {
    val transaction = fm.beginTransaction()
    transaction.add(R.id.container, fragmentHome).show(fragmentHome).commit()
    transaction.add(R.id.container, fragmentHistory).hide(fragmentHistory).commit()
    transaction.add(R.id.container, fragmentProfile).hide(fragmentProfile).commit()
    transaction.add(R.id.container, fragmentProfile).show(fragmentProfile).commit()
}

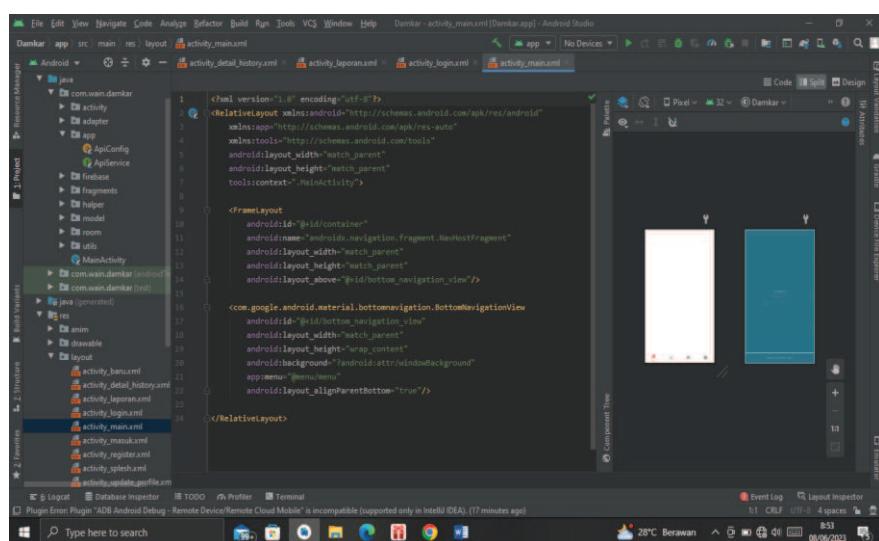
val bottomNavigationView = findViewById(R.id.bottom_navigation_view)
val menu = bottomNavigationView.menu
val menuItems = menu.getItem(0)
menuItems.isChecked = true

bottomNavigationView.setOnNavigationItemSelected { item ->
    when (item.itemId) {
        R.id.homeFragment -> {
            callFragment(0, fragmentHome)
        }
        R.id.historyFragment -> {
            callFragment(1, fragmentHistory)
        }
        R.id.infoFragment -> {
            callFragment(2, fragmentInfo)
        }
        R.id.profileFragment -> {
            callFragment(3, fragmentProfile)
        }
    }
}
```

Gambar 6. MainActivity.kt

#### b. activity\_main.xml

Fungsi dari *source code* ini adalah untuk membuat tampilan dari MainActivity.kt.



Gambar 7. activity\_main.xml

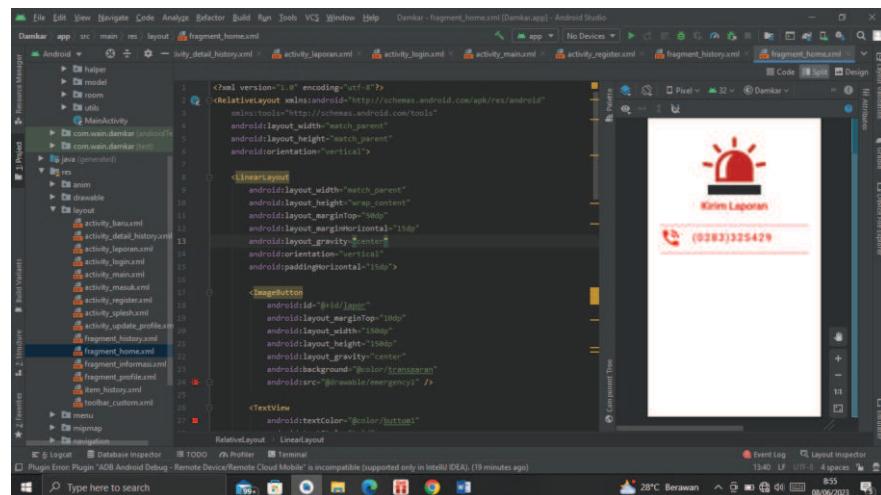
### c. HomeFragment.kt

Fungsi dari *source code* ini untuk bisa menjalankan pada tampilan home, agar semua tombol berfungsi yaitu tombol *call* dan tombol kirim laporan.

Gambar 8. HomeFragment.kt

d. fragment\_home.xml

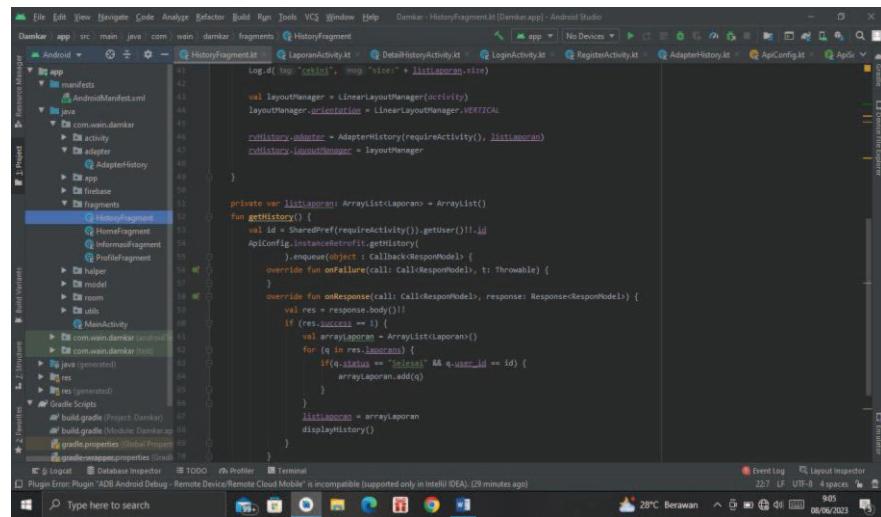
Fungsi dari *source code* ini adalah untuk membuat tampilan dari HomeFragment.kt



Gambar 9. fragment home.xml

### e. HistoryFragment.kt

Fungsi dari *source code* ini digunakan untuk menampilkan history laporan user yang di panggil dari *database*.



```
class HistoryFragment : Fragment() {
    private var listLaporan: ArrayList<Laporan> = ArrayList()
    private var listAdapter: AdapterHistory? = null

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.fragment_history, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        listAdapter = AdapterHistory(requireActivity(), listLaporan)
        val layoutManager = LinearLayoutManager(activity)
        layoutManager.orientation = LinearLayoutManager.VERTICAL
        rvHistory.adapter = listAdapter
        rvHistory.layoutManager = layoutManager
    }

    private var listLaporan: ArrayList<Laporan> = ArrayList()
    private var idUser: String? = null

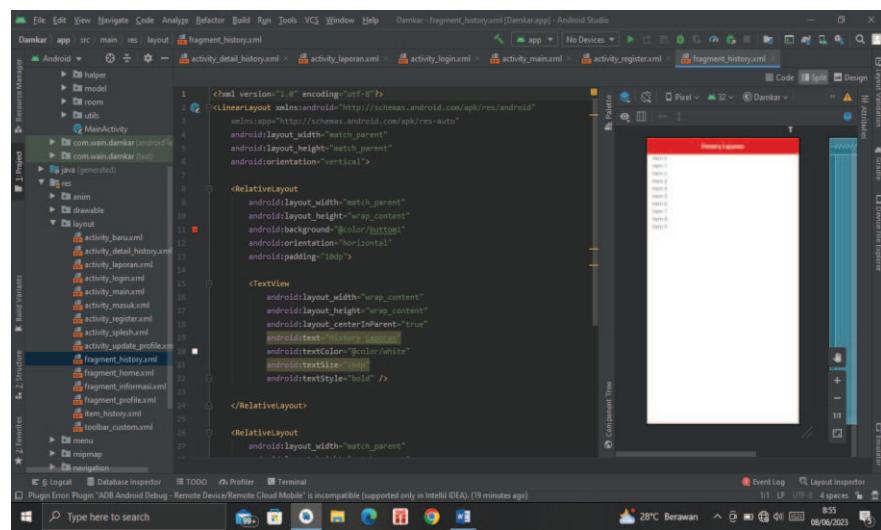
    fun getHistory() {
        val id = SharedPref(requireActivity()).getUserId()!!
        ApiConfig.instance.enqueueCall(CallBackResponseModel {
            override fun onResponse(call: Call<Response<ResponseModel>>, t: Throwable) {
                val res = response(call).body()!!
                if (res.access == 1) {
                    val arrayLaporan = ArrayList<Laporan>()
                    for (e in res.laporan) {
                        if (e.status == "selesai" && e.user_id == id) {
                            arrayLaporan.add(e)
                        }
                    }
                    listLaporan = arrayLaporan
                    displayHistory()
                }
            }
        })
    }

    private fun displayHistory() {
        listAdapter?.notifyDataSetChanged()
    }
}
```

Gambar 10. HistoryFragment.kt

### f. fragment\_history.xml

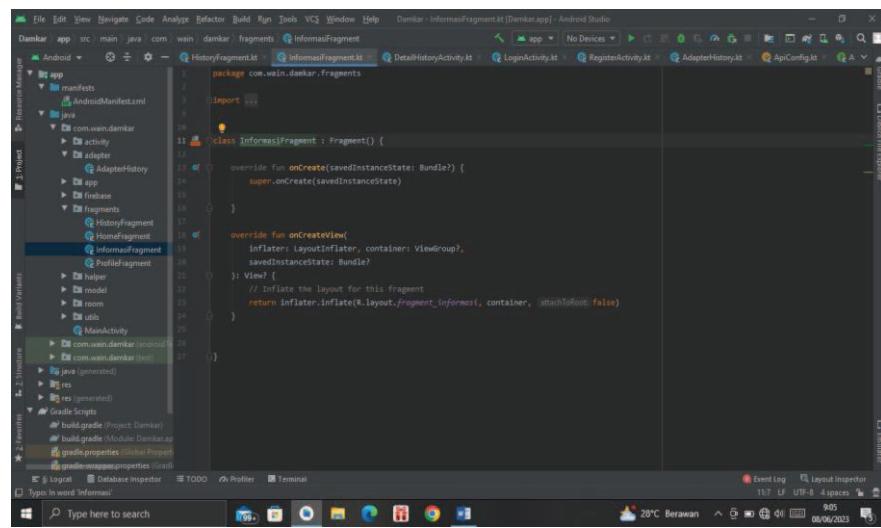
Fungsi dari *source code* ini adalah untuk membuat tampilan dari HistoryFragment.kt



Gambar 11. fragment\_history.xml

### g. InformasiFragment.kt

Fungsi dari *source code* ini untuk menampilkan inforamsi yang ada pada tampilan.



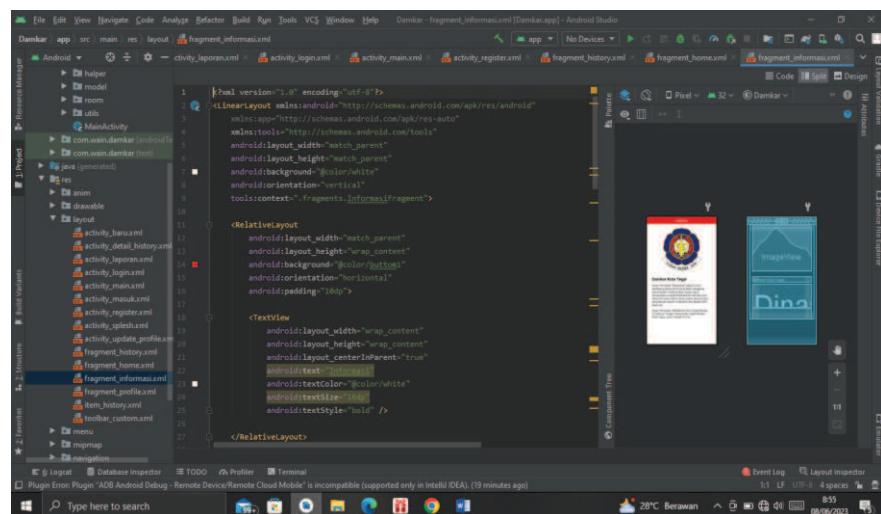
```
package com.wain.damkar.fragments

class InformasiFragment : Fragment() {
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment.
        return inflater.inflate(R.layout.fragment_informasi, container, false)
    }
}
```

Gambar 12. InformasiFragment.kt

### h. fragment\_informasi.xml

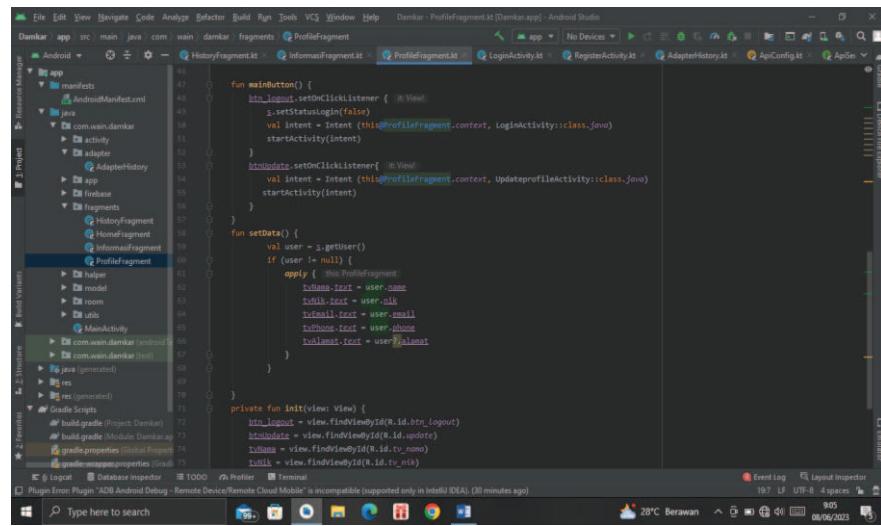
Fungsi dari *source code* ini adalah untuk membuat tampilan InformasiFragment.kt.



Gambar 13. fragment\_informasi.xml

### i. ProfileFragment.kt

Fungsi dari *source code* ini untuk memanggil data *user* yang *login* ke aplikasi dari *database*.



```
fun mainbutton() {
    btn_logout.setOnClickListener { it: View ->
        it.setTranslationY(10f)
        val intent = Intent(this@ProfileFragment.context, LoginActivity::class.java)
        startActivity(intent)
    }
    btnupdate.setOnClickListener { it: View ->
        val intent = Intent(this@ProfileFragment.context, UpdateprofileActivity::class.java)
        startActivity(intent)
    }
}

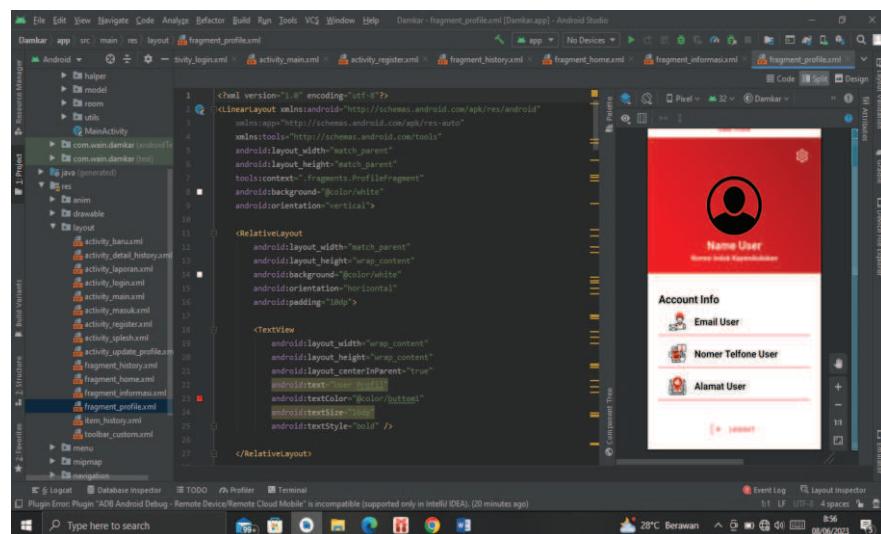
fun setData() {
    val user = i.getuser()
    if (user != null) {
        apply {
            this@ProfileFragment
            txtName.text = user.name
            txtNik.text = user.nik
            txtEmail.text = user.email
            txtPhone.text = user.phone
            txtAlamat.text = user.alamat
        }
    }
}

private fun init(view: View) {
    btn_logout = view.findViewById(R.id.btn_logout)
    btnupdate = view.findViewById(R.id.btn_update)
    txtName = view.findViewById(R.id.tv_nama)
    txtNik = view.findViewById(R.id.tv_nik)
}
```

Gambar 14. ProfileFragment.kt

### j. fragment\_profile.xml

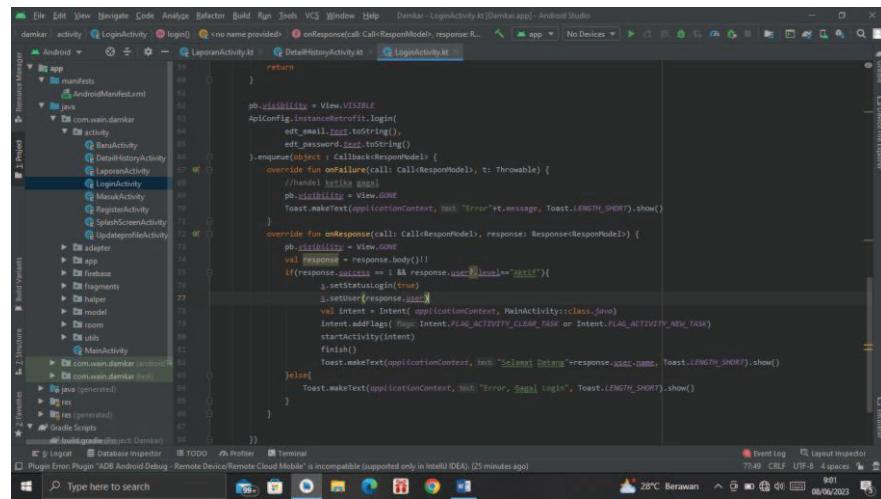
Fungsi dari *source code* ini adalah untuk menampilkan data *user* yang di panggil pada ProfileFragment.kt.



Gambar 15. fragment\_profile.xml

## k. LoginActivity.kt

Fungsi dari *source code* ini untuk memanggil data dari *database* untuk login user, yaitu dengan menyamakan data email dan password.

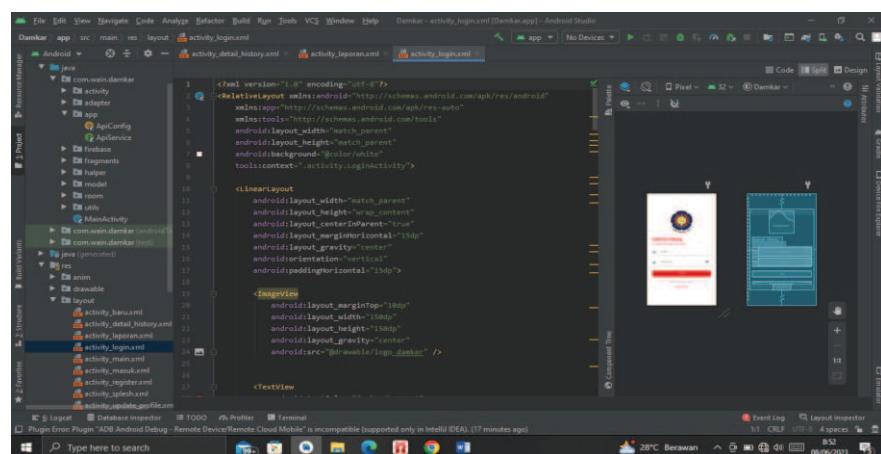


```
private fun login() {
    val email = edt_email.text.toString()
    val password = edt_password.text.toString()
    ApiConfig.instance.retrofit.login(
        edt_email.text.toString(),
        edt_password.text.toString()
    ).enqueue(object : Callback<ResponseModel>, t: Throwable) {
        override fun onFailure(call: Call<ResponseModel>, t: Throwable) {
            pb.visibility = View.GONE
            pb.visibility = View.GONE
            val response = response.body()!!
            if(response.success == 1 && response.user.level == "admin") {
                g.setstatusLogin(true)
                g.setuser(response.user)
                val intent = Intent(applicationContext, MainActivity::class.java)
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)
                startActivity(intent)
                finish()
                Toast.makeText(applicationContext, "Selamat Datang "+response.user.name, Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(applicationContext, "Error, Gagal Login", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Gambar 16. LoginActivity.kt

## 1. activity\_login.xml

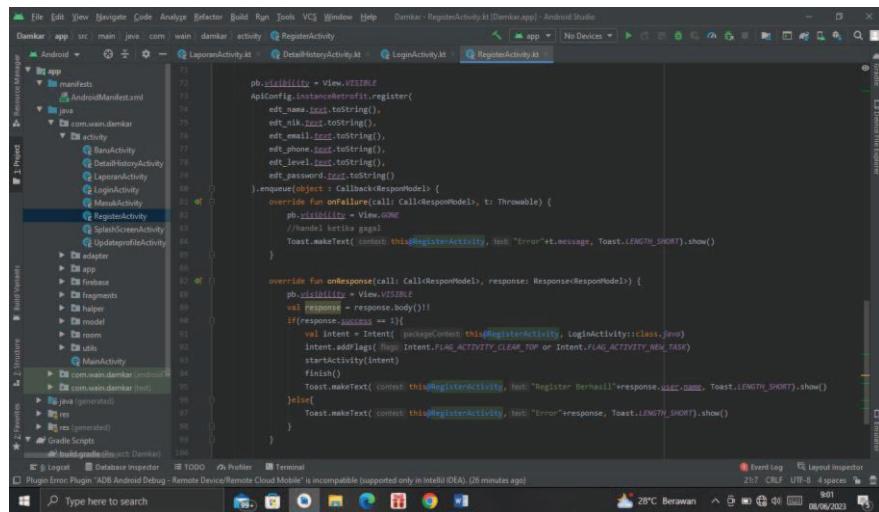
Fungsi dari *source code* ini digunakan untuk membuat tampilan login dengan adanya dua inputan yaitu email dan *password*. Terdapat juga *button* *login* dan *register*.



Gambar 17. activity\_login.xml

## m. RegisterActivity.kt

Fungsi dari *source code* ini adalah untuk mengirim data yang sudah di inputkan *user* ke *database* agar bisa *login*.



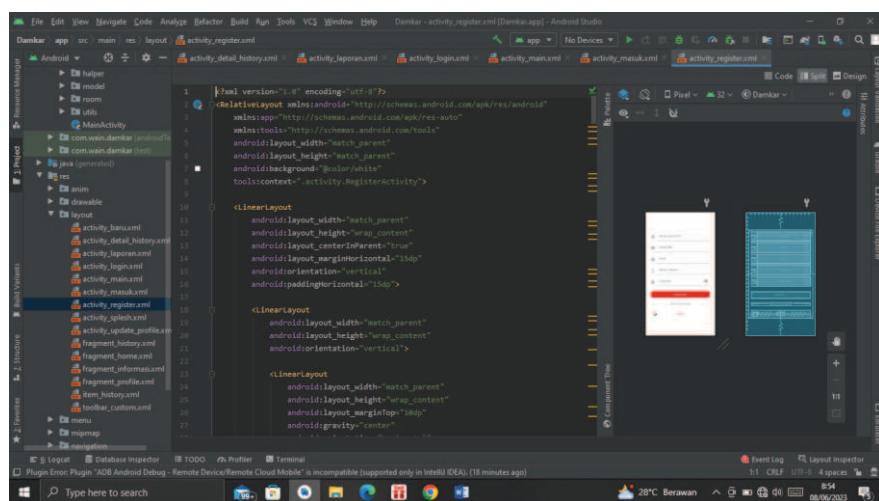
```
pb.setVisibility(View.VISIBLE)
ApiConfig.instance.register(
    edt_nama.text.toString(),
    edt_nik.text.toString(),
    edt_email.text.toString(),
    edt_phone.text.toString(),
    edt_level.text.toString(),
    edt_password.text.toString()
).enqueue(object : Callback<ResponseModel> {
    override fun onFailure(call: Call<ResponseModel>, t: Throwable) {
        pb.visibility = View.GONE
        //handle ketika gagal
        Toast.makeText(context, this@RegisterActivity, R.string.error, Toast.LENGTH_SHORT).show()
    }

    override fun onResponse(call: Call<ResponseModel>, response: Response<ResponseModel>) {
        pb.visibility = View.VISIBLE
        val response = response.body()!!
        if(response.success == 1){
            val intent = Intent(context, LoginActivity::class.java)
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP or Intent.FLAG_ACTIVITY_NEW_TASK)
            startActivity(intent)
            finish()
            Toast.makeText(context, this@RegisterActivity, R.string.register_berhasil, Toast.LENGTH_SHORT).show()
        }else{
            Toast.makeText(context, this@RegisterActivity, R.string.error, Toast.LENGTH_SHORT).show()
        }
    }
})
```

Gambar 18. RegisterActivity.kt

## n. activity\_register.xml

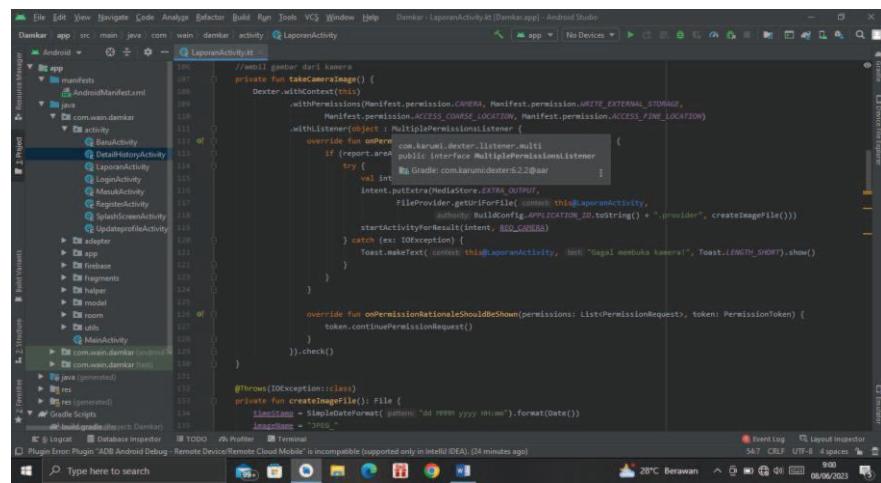
Fungsi dari *source code* ini untuk membuat tampilan *register* dengan adanya inputan nama, nik, email, no hp, dan password dan juga *button register*.



Gambar 19. activity\_register.xml

## o. LaporanActivity.kt

Fungsi dari *source code* ini untuk mengirim laporan melalui aplikasi, Pada *source code* ini terdapat *code* untuk menggunakan kamera, dan mendapatkan lokasi saat pengambilan gambar.



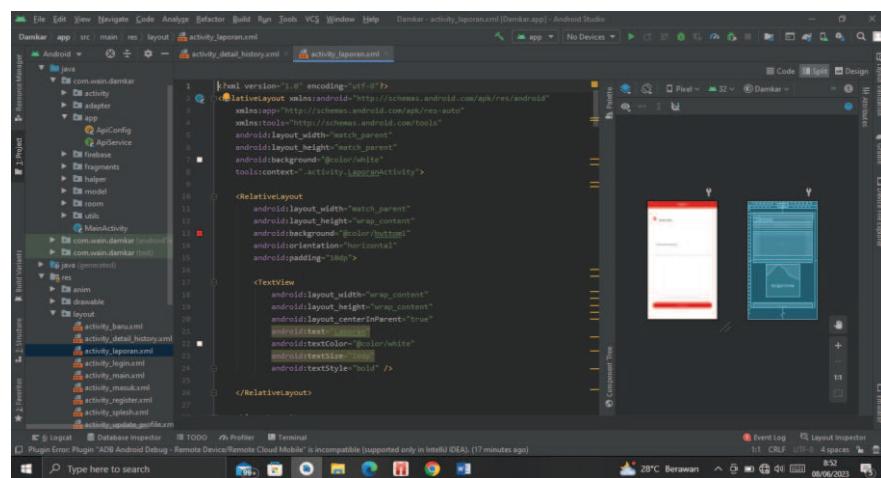
```
private fun takeCameraImage() {
    Dexter.withContext(this)
        .withPermissions(Manifest.permission.CAMERA, Manifest.permission.WRITE_EXTERNAL_STORAGE,
            Manifest.permission.ACCESS_COARSE_LOCATION, Manifest.permission.ACCESS_FINE_LOCATION)
        .withListener(object : MultiplePermissionsListener {
            override fun onPermissionsGranted() {
                if (reporter != null) {
                    reporter.onSuccess("MultiplePermissionsListener")
                }
            }
            override fun onPermissionsDenied(permissions: List<PermissionRequest>, token: PermissionToken) {
                reporter.onFailure("MultiplePermissionsListener", token)
            }
        }).check()
}

@Throws(IOException::class)
private fun createImageFile(): File {
    timestamp = SimpleDateFormat("dd MMMM yyyy HHmm").format(Date())
    imageFile = File(timestamp)
}
```

Gambar 20. LaporanActivity.kt

## p. activity\_laporan.xml

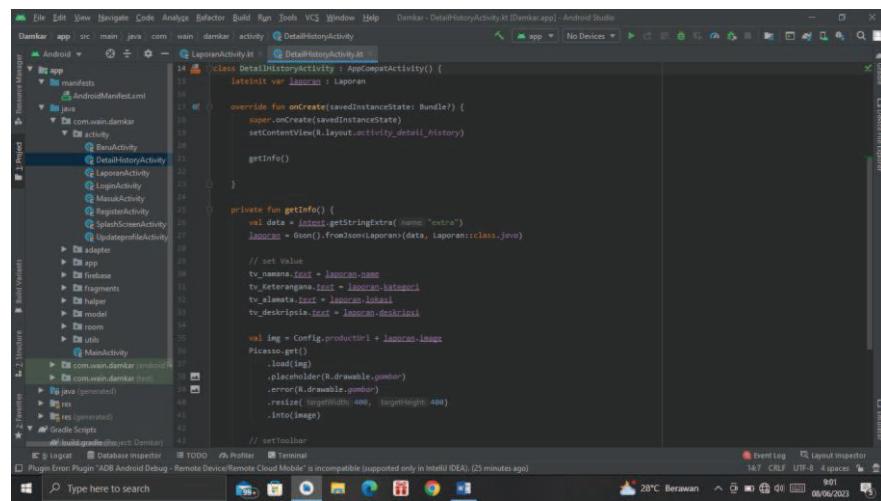
Fungsi dari *source code* ini adalah untuk membuat tampilan pada LaporanActivity.kt. Pada *source code* ini terdapat inputan dan spinner untuk mengisi data laporan.



Gambar 21. activity\_laporan.xml

#### q. DetailHistoryActivity.kt

Fungsi dari *source code* ini adalah untuk memanggil data laporan yang berstatus ‘’selesai’’ pada *database* sesuai dengan *user* yang manggil.



```
class DetailHistoryActivity : AppCompatActivity() {
    lateinit var laporan : Laporan

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail_history)

        getInfo()
    }

    private fun getInfo() {
        val data = intent.getStringExtra("extra")
        laporan = Gson().fromJson(data, Laporan::class.java)

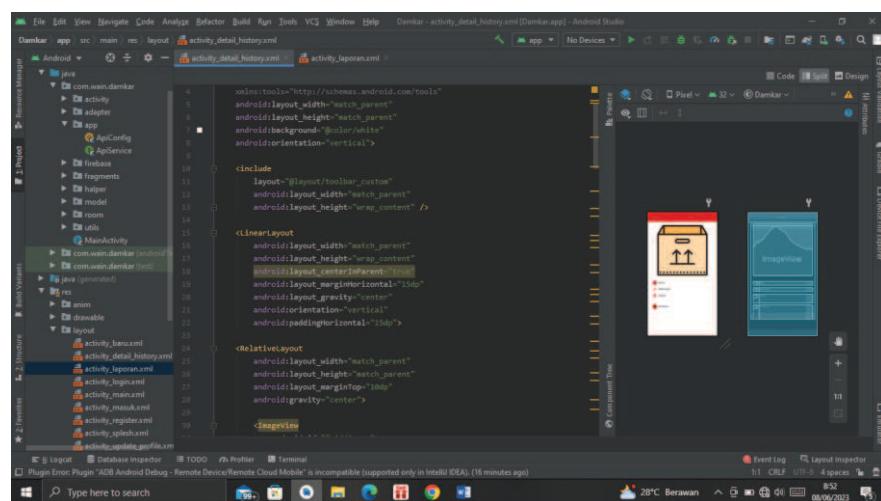
        // set Value
        tv_nama.text = laporan.name
        tv_keterangan.text = laporan.kategori
        tv_elama.text = laporan.lokal
        tv_deskripsi.text = laporan.deskripsi

        val img = Config.productUrl + laporan.image
        Picasso.get()
            .load(img)
            .placeholder(R.drawable.placeholder)
            .error(R.drawable.placeholder)
            .resize(targetWidth = 400, targetHeight = 400)
            .into(image)
    }
}
```

Gambar 22. DetailHistoryActivity.kt

#### r. activity\_detail\_history.xml

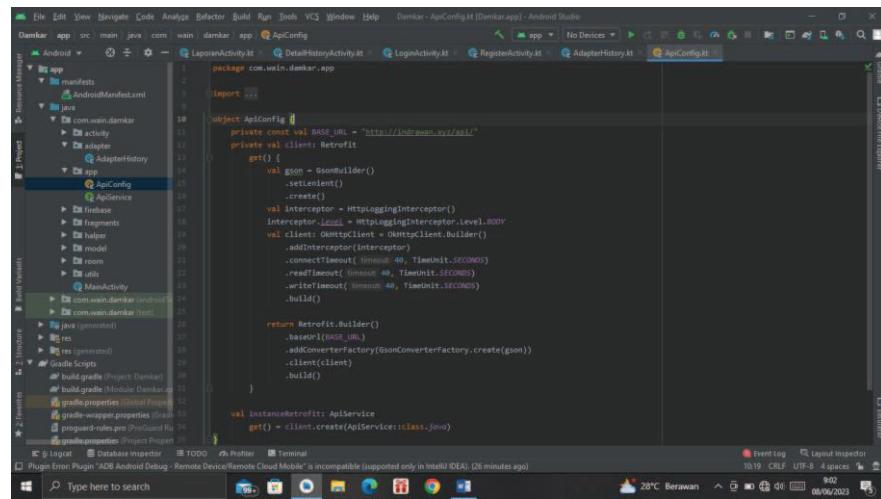
Fungsi dari *source code* ini adalah untuk membuat tampilan pada DetailHistoryActivity.kt.



Gambar 23. activity\_detail\_history.xml

## s. ApiConfig.kt

Fungsi dari *source code* ini adalah untuk mengkoneksikan ke *server* kita agar bisa di jalankan dan bisa mengelola *database*.



```
package com.wein.damkar.app

import ...

object ApiConfig {
    private const val BASE_URL = "https://indrawan.ay/2021"
    private val client: Retrofit
        get() {
            val gson = GsonBuilder()
                .setLenient()
                .create()
            val interceptor = HttpLoggingInterceptor()
            interceptor.level = HttpLoggingInterceptor.Level.BODY
            val client: OkHttpClient = OkHttpClient.Builder()
                .addInterceptor(interceptor)
                .connectTimeout(Timeout(40, TimeUnit.SECONDS))
                .readTimeout(Timeout(40, TimeUnit.SECONDS))
                .writeTimeout(Timeout(40, TimeUnit.SECONDS))
                .build()

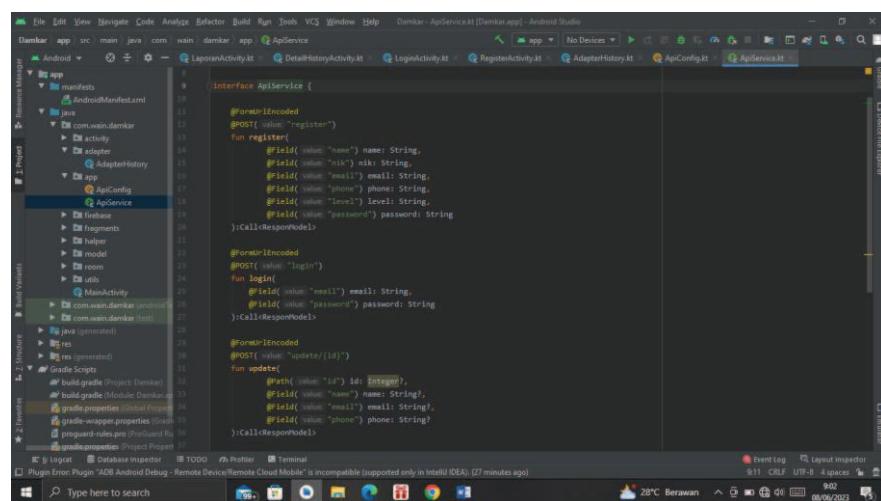
            return Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(JsonConverterFactory.create(gson))
                .client(client)
                .build()
        }

    val instanceRetrofit: ApiService
        get() = client.create(ApiService::class.java)
}
```

Gambar 24. ApiConfig.kt

## t. ApiService.kt

Fungsi dari *source code* ini untuk memanggil *method* dari *server api* ke aplikasi *android*. Pada *source code* ini terdapat beberapa *method* yang di gunakan untuk mengirim dan memanggil datatabse.



```
interface ApiService {
    @FormUrlEncoded
    @POST("user/register")
    fun register(
        @Field("name") name: String,
        @Field("email") email: String,
        @Field("phone") phone: String,
        @Field("level") level: String,
        @Field("password") password: String
    ): Call<ResponseModel>

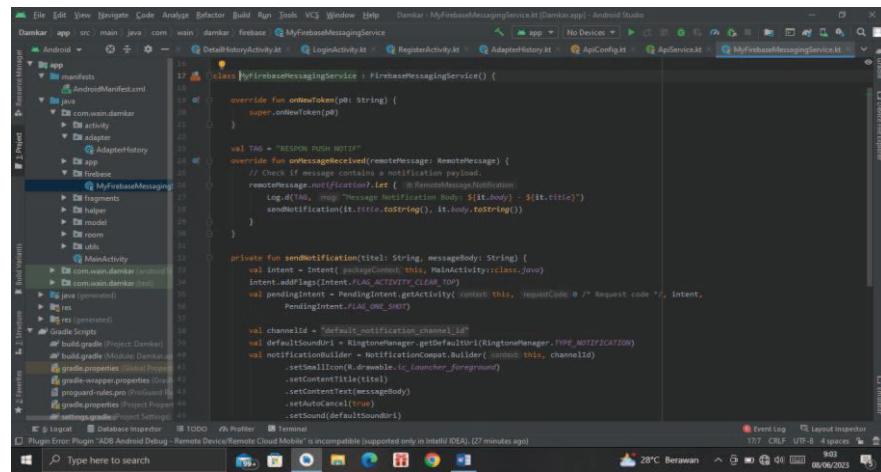
    @FormUrlEncoded
    @POST("user/login")
    fun login(
        @Field("email") email: String,
        @Field("password") password: String
    ): Call<ResponseModel>

    @FormUrlEncoded
    @POST("user/update/{id}")
    fun update(
        @Path("id") id: Integer,
        @Field("name") name: String,
        @Field("email") email: String,
        @Field("phone") phone: String
    ): Call<ResponseModel>
}
```

Gambar 25. ApiService.kt

## u. MyFirebaseManager.kt

Fungsi dari *source code* ini adalah membuat *notifikasi* jika ada data yang masuk ke dalam *system*. Pada *system* ini digunakan jika laporan di terima maka akan muncul notifikasi pada hp *user*.



```
class MyFirebaseMessagingService : FirebaseMessagingService() {
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    val TAG = "RESPONSE PUSH NOTIFICATION"
    override fun onMessageReceived(remoteMessage: RemoteMessage) {
        // Check if message contains a notification payload.
        remoteMessage.notification?.let {
            // Get notification body
            Log.d(TAG, "Message notification body: ${it.body} - ${it.title}")
            sendNotification(it.title.toString(), it.body.toString())
        }
    }

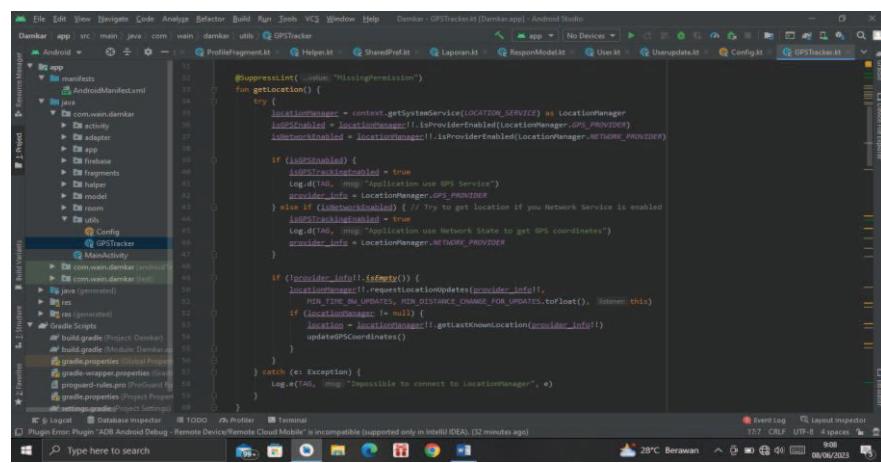
    private fun sendNotification(title: String, messageBody: String) {
        val intent = Intent(packageName, this).apply {
            MainActivitity.class
            flags = Intent.FLAG_ACTIVITY_CLEAR_TOP
        }
        val pendingIntent = PendingIntent.getActivity(context, requestCode, intent, PendingIntent.FLAG_UPDATE_CURRENT)

        val channelId = "default_notification_channel_id"
        val defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION)
        val notificationBuilder = NotificationCompat.Builder(context, channelId)
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle(title)
            .setContentText(messageBody)
            .setAutoCancel(true)
            .setSound(defaultSoundUri)
    }
}
```

Gambar 26. MyFirebaseManager.kt

## v. GPSTracker.kt

Fungsi dari *source code* ini untuk mendapatkan lokasi saat aplikasi di gunakan atau saat laporan. *Source code* ini di panggil pada class LaporanActivity.kt untuk mendapatkan lokasi terkini.



```
class GPSTracker : LocationListener {
    var locationManager: LocationManager? = null
    var lastKnownLocation: Location? = null
    var latitude: Double? = null
    var longitude: Double? = null
    var isGPSTrackingEnabled: Boolean = false
    var providerInfo: LocationManager? = null
    var isNetworkEnabled: Boolean = false

    @SuppressLint("MissingPermission")
    fun getLocation() {
        try {
            locationManager = context.getSystemService(LOCATION_SERVICE) as LocationManager
            isGPSTrackingEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
            isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)

            if (isGPSTrackingEnabled) {
                isGPSTrackingEnabled = true
                Log.d("TAG", "Application use GPS Service")
                providerInfo = locationManager.GPS_PROVIDER
            } else if (isNetworkEnabled) { // Try to get location if you Network Service is enabled
                isGPSTrackingEnabled = true
                Log.d("TAG", "Application use Network Service to get GPS coordinates")
                providerInfo = locationManager.NETWORK_PROVIDER
            }

            if (providerInfo != null) {
                locationManager!!.requestLocationUpdates(providerInfo, 1000, 0.0f, this)
                if (locationManager != null) {
                    location = locationManager!!.getLastKnownLocation(providerInfo)
                    updatedGSCoordinates()
                }
            }
        } catch (e: Exception) {
            Log.e("TAG", "Impossible to connect to locationManager", e)
        }
    }

    if (providerInfo != null) {
        locationManager!!.requestLocationUpdates(providerInfo, 1000, 0.0f, this)
        if (locationManager != null) {
            location = locationManager!!.getLastKnownLocation(providerInfo)
            updatedGSCoordinates()
        }
    }
}
```

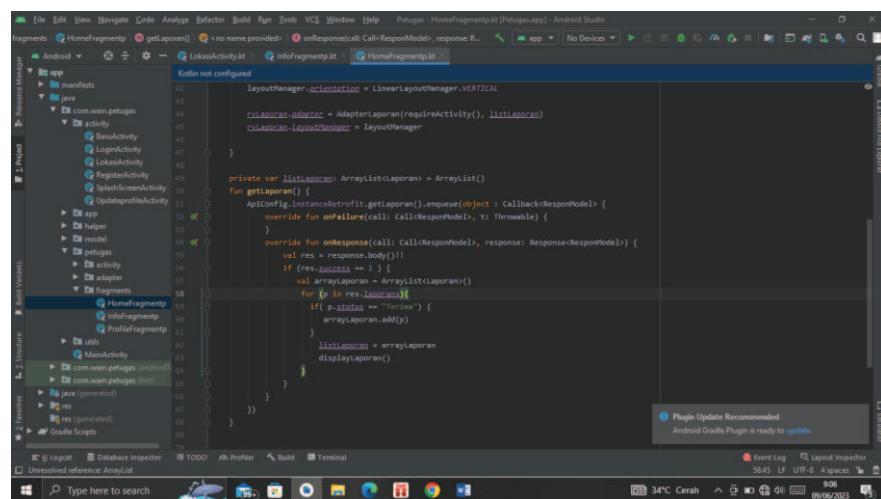
Gambar 27. GPSTracker.kt

#### 4. Petugas

Source code pada petugas hamper sama dengan source code pada aplikasi warga seperti pada tampilan InformasiFragment.kt, fragment\_info.xml, ProfileFragment.kt, fragment\_profile. Ada yang membedakan pada aplikasi petugas yaitu pada source code HomeFragmentp.kt dan fragment\_home.xml seperti berikut :

##### a. HomeFragmentp.kt

Fungsi dari *source code* ini adalah untuk memanggil laporan yang sudah di *verifikasi* oleh admin.



```
package com.wan.petugas

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.wan.petugas.adapter.AdapterLaporan
import com.wan.petugas.model.Laporan
import com.wan.petugas.model.ResponseResponModel
import com.wan.petugas.utils.ApiConfig
import com.wan.petugas.utils.CallBackResponModel
import com.wan.petugas.utils.ResponseResponModel
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class HomeFragmentp : Fragment() {
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_home, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        val rvLaporan: RecyclerView = view.findViewById(R.id.rvLaporan)
        val layoutManager: LinearLayoutManager = LinearLayoutManager(context)
        rvLaporan.layoutManager = layoutManager
        rvLaporan.adapter = AdapterLaporan(requireActivity(), listLaporan)
        rvLaporan.layoutManager = layoutManager
    }

    private var listLaporan: ArrayList<Laporan> = ArrayList()
    fun getLaporan() {
        ApiConfig.instanceRetrofit.getLaporan().enqueue(object : Callback<ResponseResponModel> {
            override fun onFailure(call: Call<ResponseResponModel>, t: Throwable) {
            }

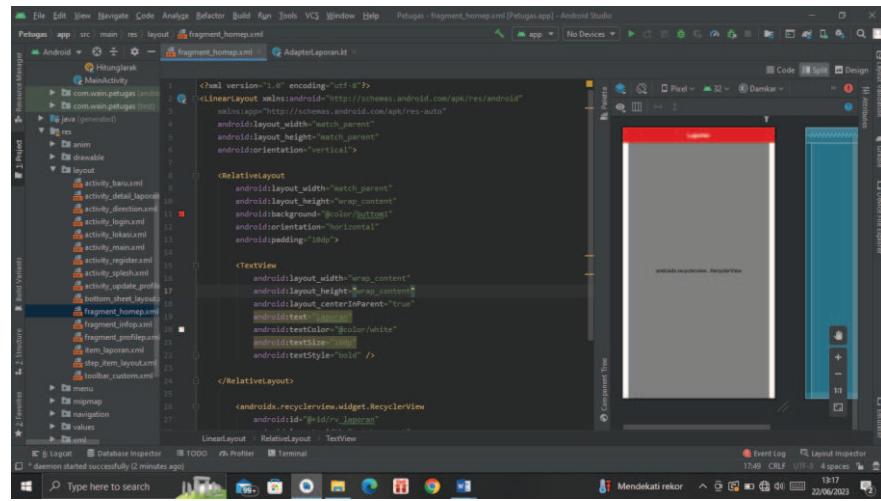
            override fun onResponse(call: Call<ResponseResponModel>, response: Response<ResponseResponModel>) {
                val res = response.body()!!
                if (res.access == 1) {
                    val arrayLaporan: ArrayList<Laporan> = ArrayList()
                    for (p in res.laporan) {
                        if (p.status == "Terima") {
                            arrayLaporan.add(p)
                        }
                    }
                    listLaporan = arrayLaporan
                    displayLaporan()
                }
            }
        })
    }

    private fun displayLaporan() {
        val adapterLaporan = AdapterLaporan(requireActivity(), listLaporan)
        rvLaporan.adapter = adapterLaporan
    }
}
```

Gambar 28. HomeFragmentp.kt

##### b. fragment\_home.xml

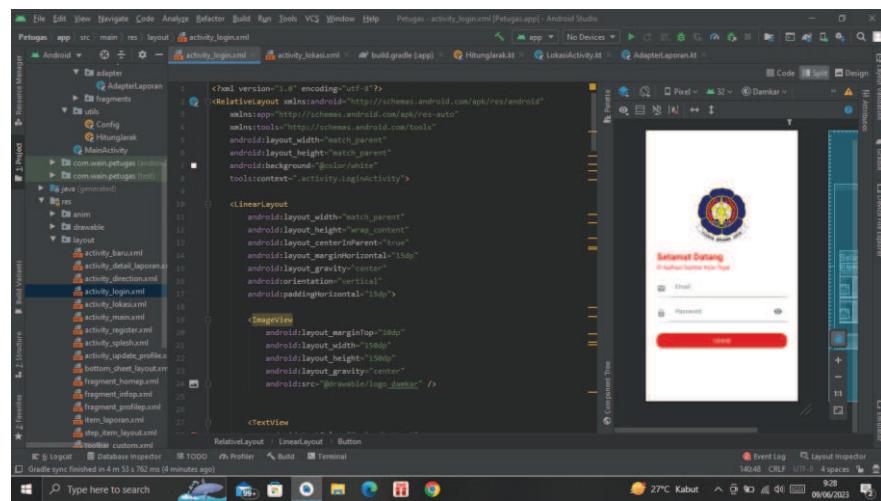
Fungsi dari *source code* ini adalah *layout* untuk menampilkan semua data yang di panggil pada *source code* HomeFragmentp.kt.



Gambar 29. fragment\_home.xml

c. activity\_login.xml

Berbeda dengan yang ada pada aplikasi warga pada aplikasi petugas pada activity\_login.xml tidak ada tombol unruk register karena semua data petugas.



Gambar 30. activity\_login.xml

d. LokasiActivity.kt

Fungsi dari *source code* ini adalah untuk melihat detail laporan dan menampilkan maps route untuk menuju lokasi kejadian.

```

override fun onMapReady(googleMap: GoogleMap) {
    this.googleMap = googleMap
    mapViva = googleMap
    mapViva = googleMap!!
    val YOUR_API_KEY = "AIzaSyDkZBkjWm7Xuv0BhWsd2xF83IwOpu"
    // Sample coordinates
    val latlngOrigin = LatLng(strCurrentLatitude, strCurrentLongitude) // Saya
    val latlngDestination = LatLng(strLatitude, strLongitude) // tujuan
    this.googleMap!!.addMarker(MarkerOptions()
        .position(latlngOrigin)
        .snippet("Saya")
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.redcar)))
    rotation
    this.googleMap!!.addMarker(MarkerOptions().position(latlngDestination).title("Lokasi"))
    this.googleMap!!.moveCamera(CameraUpdateFactory.newLatLngZoom(latlngOrigin, 20f))
    val path: MutableList<LatLng> = ArrayList()
    val urlDirections = "https://maps.googleapis.com/maps/api/directions/json?origin=" + strCurrentLatitude + "," + strCurrentLongitude + "&destination=" + strLatitude + "," + strLongitude + "&key=" + YOUR_API_KEY
    val directionsRequest = object : StringRequest(
        Request.Method.GET,
        urlDirections,
        Response.Listener<String> { response ->
            // parse JSON
            val routes = jsonResponse.getJSONObject("routes")
            val legs = routes.getJSONObject(0).getJSONArray("legs")
            val steps = legs.getJSONObject(0).getJSONArray("steps")
            for (i in 0 until steps.length()) {
                val step = steps.getJSONObject(i)
                val start = step.getJSONObject("start_location")
                val end = step.getJSONObject("end_location")
            }
        },
        Response.ErrorListener { error ->
            Log.e("Error", "Error getting directions: " + error.message)
        }
    )
    Volley.newRequestQueue(this).add(directionsRequest)
}

```

Gambar 31. LokasiActivity.kt

e. activity\_lokasi.xml

Fungsi dari *source code* ini adalah layout untuk menampilkan maps dan data yang di panggil oleh *source code* LokasiActivity.kt

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/edit_lokasi"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:id="@+id/edit_lokasi"
        android:inputType="text" />
    <EditText
        android:id="@+id/edit_lokasi"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:id="@+id/edit_lokasi"
        android:inputType="text" />
    <TextView
        android:id="@+id/tv_lokasi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:text="Masukkan tujuan"
        android:textColor="#000000"
        android:textSize="20sp"
        android:textStyle="bold" />

```

Gambar 32. activity\_lokasi.xml

5. Admin

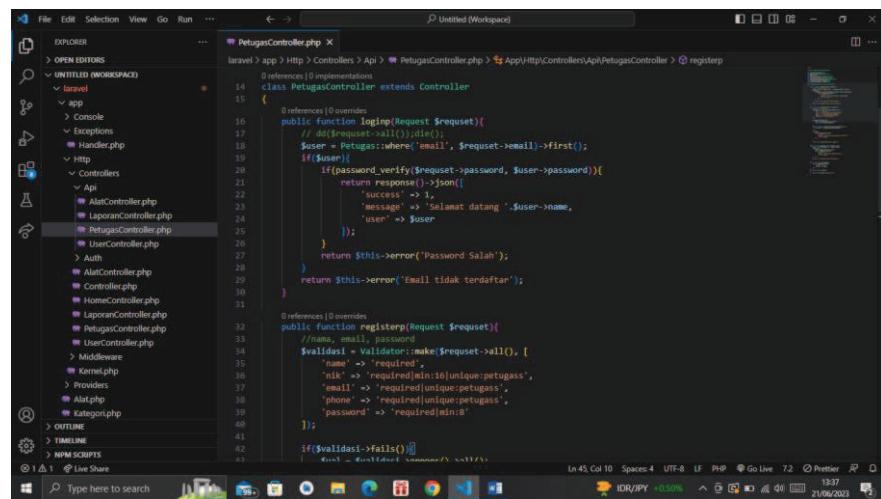
Pada *source code* admin menggunakan website yang dibangun menggunakan *framework* laravel berikut adalah *source*

code-nya:

## 1. Api Controller

### a. LaporanController.php

Fungsi dari *source code* ini adalah untuk menghubungngkan aplikasi *mobile* milik Petugas ke *database* melalui api *server*, untuk mengolah data yaitu mengirim data ke *database* memanggil data dari *database* dan meng-*update* data di *database*. Pada source code ini terdapat beberapa *function* yaitu: ‘*public function loginp*’, dan ‘*public function registerp*’.



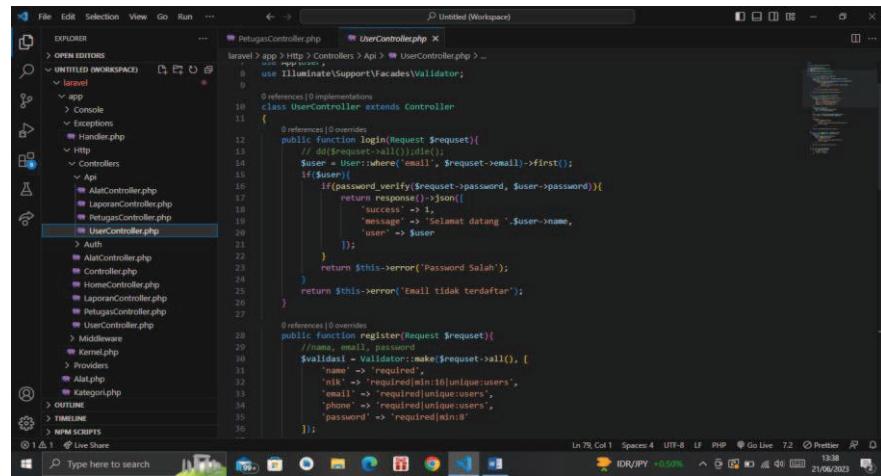
```
class PetugasController extends Controller
{
    public function loginp(Request $request)
    {
        $user = Petugas::where('email', $request->email)->first();
        if($user)
            if(password_verify($request->password, $user->password))
                return response()->json([
                    'name' => $user->name,
                    'message' => 'Selamat datang ' . $user->name,
                    'user' => $user
                ]);
            else
                return $this->error('Password Salah');
        else
            return $this->error('Email tidak terdaftar');
    }

    public function registerp(Request $request)
    {
        //name, email, password
        $validasi = Validator::make($request->all(), [
            'name' => 'required',
            'nik' => 'required|alpha|exists:petugass',
            'email' => 'required|unique:petugass',
            'phone' => 'required|unique:petugass',
            'password' => 'required|min:8'
        ]);
        if($validasi->fails())
            $this->error('Validation Error');
        else
            $user = Petugas::create($request->all());
            return response()->json([
                'name' => $user->name,
                'message' => 'User berhasil dibuat'
            ]);
    }
}
```

Gambar 33. LaporanController.php

### b. UserController.php

Fungsi dari *source code* ini adalah untuk menghubungngkan aplikasi *mobile* milik Petugas ke *database* melalui api *server*, untuk mengolah user seperti *login* dan *register*. Pada *source code* ini terdapat beberapa *function* yaitu : *public function login*, dan *public function register*.



```

class UserController extends Controller
{
    public function login(Request $request)
    {
        $user = User::where('email', $request->email)->first();
        if($user)
            if(password_verify($request->password, $user->password))
                return response()->json([
                    'success' => true,
                    'message' => 'Selamat datang ' . $user->name,
                    'user' => $user
                ]);
            else
                return $this->error('Password Salah');
        else
            return $this->error('Email tidak terdaftar');
    }

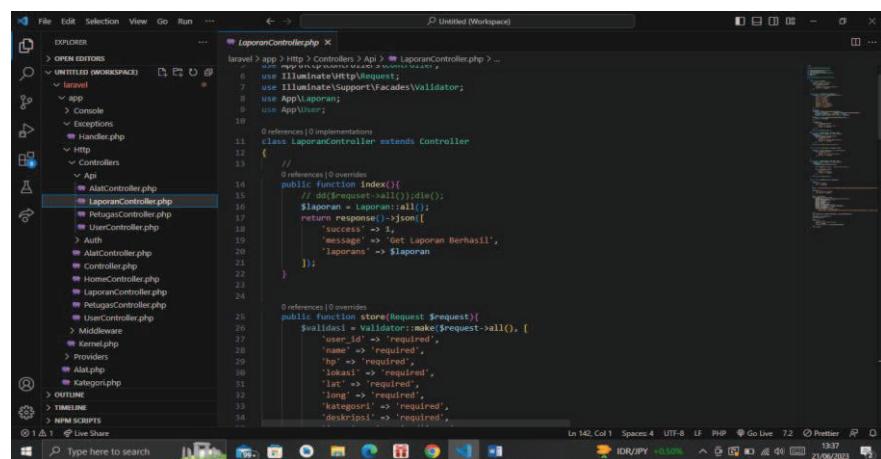
    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required',
            'nik' => 'required|min:16|unique:users',
            'email' => 'required|email|unique:users',
            'phone' => 'required|unique:users',
            'password' => 'required|alpha'
        ]);
    }
}

```

Gambar 34. UserController.php

### c. LaporanController.php

Fungsi dari *source code* ini adalah untuk menghubungngkan aplikasi mobile *milik* warga ke *database* melalui *api server*, untuk mengolah data yaitu mengirim data ke *database* memanggil data dari *database* dan meng-*update* data di *database*. Pada ‘Laporancontroller’ terdapat beberapa *function* yaitu : Public function store, Public function history, Public function update, Public function notification.



```

class LaporanController extends Controller
{
    public function index()
    {
        $laporan = Laporan::all();
        return response()->json([
            'success' => true,
            'message' => 'Get Laporan berhasil',
            'laporan' => $laporan
        ]);
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required',
            'hp' => 'required',
            'lokasi' => 'required',
            'lat' => 'required',
            'long' => 'required',
            'kategori' => 'required',
            'deskripsi' => 'required'
        ]);
    }

    public function update(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required',
            'hp' => 'required',
            'lokasi' => 'required',
            'lat' => 'required',
            'long' => 'required',
            'kategori' => 'required',
            'deskripsi' => 'required'
        ]);
    }
}

```

Gambar 35. LaporanController.php

## 2. Web Controller

a. AlatController.php

Fungsi dari *source code* ini untuk menghubungan model Alat.php dengan *view* alat.blade.php. Pada *source code* ini terdapat beberapa fungsi untuk mengolah data seperti ‘public function index’ untuk memanggil data dari *database*, public function stor untuk meng-*input* data ke *database*, dan public function update untuk meng-*update* data.

Gambar 36. AlatController.php

b. LaporanController.php

Fungsi dari *source code* ini untuk menghubungkan model Laporan.php dengan *view* laporan.blade.php. Pada *source code* ini terdapat beberapa fungsi untuk mengolah data seperti public function index, public function cetakLaporan, public function googlemap, public function Laporanmasuk, public function dashboard, public function update, dan public function pushNotif.

```

1 laravel > app > Http > Controllers > LaporanController.php > App\Http\Controllers\LaporanController > googlemap
2
3 16
4 17 0 references | 0 overrides
5 18  public function index()
6 19  {
7 20  $laporan = $laporan::where('status', '=', 'Selesai')->simplePaginate(4);
8 21  return view('laporan', compact('laporan'));
9 22
10 0 references | 0 overrides
11 13
12 14  public function cetaklaporan($tglawal, $tglakhir)
13 15  {
14 16  //dd(["Tanggal Awal : ", $tglawal, "Tanggal Akhir : ", $tglakhir]);
15 17  $laporan = $laporan::where('status', '=', 'Selesai')->whereBetween('created_at', [$tglawal, $tglakhir]);
16 18  return view('cetak-laporan', compact('laporan'));
17 19
18 20 0 references | 0 overrides
19 21  public function googlemap($id)
20 22  {
21 23  $laporan = $laporan::findOrFail($id);
22 24  return view('googlemap', compact('laporan'));
23 25
24 26 0 references | 0 overrides
25 27  public function listUser()
26 28  {
27 29  $user = 'listUser' = $user::where('status', '=', 'menunggu')->get();
28 30  return view('laporanmasuk')->with($user);
29 31
30 32  /**
31  * Show the form for creating a new resource.
32  *
33  * @return \Illuminate\Http\Response
34  */
35
36
37
38
39
39
40
41
42
43
44

```

Gambar 37. LaporanController.php

### c. PetugasController.php

Fungsi dari *source code* ini untuk menghubungan model Petugas.php dengan *view* petugas.blade.php. Pada *sorce code* ini terdapat beberapa fungsi untuk mengolah data seperti public function index untuk memanggil data dari *database*, public function update untuk *update* data di *database*, public function store untuk menambahkan petugas dan *input* data, dan public function destroy untuk menghapus data di database.

```

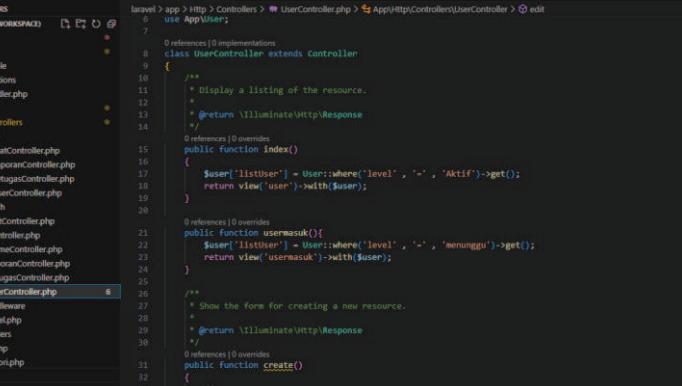
1 laravel > app > Http > Controllers > PetugasController.php > ...
2
3 1 namespace App\Http\Controllers;
4 2 use Illuminate\Http\Request;
5 3 use App\Petugas;
6 4
7 0 references | 0 implementations
8 5 class PetugasController extends Controller
9 6 {
10 7
11 8 0 references | 0 overrides
12 9  public function index()
13 10  {
14 11  $petugas['petugas'] = Petugas::all();
15 12  return view('petugas')->with($petugas);
16 13
17 14 0 references | 0 overrides
18 15  public function update(Request $request)
19 16  {
20 17  $petugas = Petugas::findOrFail($request->id);
21 18  $petugas->level = $request->input('level');
22 19  $petugas->update();
23 20  return redirect('petugas');
24 21
25 22 0 references | 0 overrides
26 23  public function store(Request $request)
27 24  {
28 25  $petugas = Petugas::create($request->all());
29 26  return view('petugas')->with($petugas);
30 27
31
32
33
34
35
36
37
38
39
39
40
41
42
43
44

```

Gambar 38.PetugasController.php

#### d. UserController.php

Fungsi dari *source code* ini untuk menghubungan model User.php dengan *view* user.blade.php. Pada *sorce code* ini terdapa beberapa fungsi untuk mengolah data sepeti function index untuk memanggil data dari *database*, public function update untuk *update* data di *database*, public function usermasuk untuk memanggil data berdasarkan status menunggu, , dan public function destroy untuk menghapus data di databse.



The screenshot shows the Laravel IDE interface with the following details:

- File Explorer:** Shows the project structure under `UNLISTED (WORKSPACE)`. The `UserController.php` file is selected.
- Code Editor:** Displays the `UserController.php` file content. The code is as follows:

```
UserController.php 6

laravel> app > Http > Controllers > UserController.php > App\Http\Controllers\UserController > edit
6 use App\User;
7
8 References (0) implementations
9 class UserController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15     */
16     public function index()
17     {
18         '$user["listUser"]' = User::where("level", '=', 'Aktif')->get();
19         return view('user')->with($user);
20     }
21
22     References (0) overrides
23     public function usermasuk()
24     {
25         '$user["listUser"]' = User::where("level", '=', 'menunggu')->get();
26         return view('usermasuk')->with($user);
27     }
28
29     /**
30      * Show the form for creating a new resource.
31      *
32      * @return \Illuminate\Http\Response
33     */
34     public function create()
35     {
36         //
37     }
38 }
```

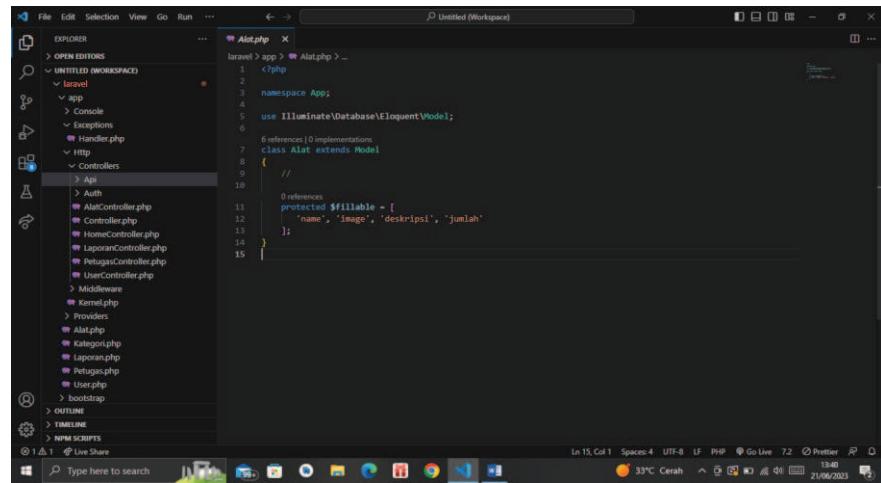
The code implements a `UserController` that extends the `Controller` class. It contains three methods: `index`, `usermasuk`, and `create`. The `index` method retrieves users with the level 'Aktif' and returns a view with the data. The `usermasuk` method retrieves users with the level 'menunggu' and returns a view with the data. The `create` method is currently empty.

### Gambar 39. UserController.php

### 3. Model

a. *Model Alat.php*

*Source code* ini berfungsi membantu mengolah data untuk dimasukan ke *database*, pada *source code* ini beriri *variable* yaitu name, image, deskripsi, dan jumlah.

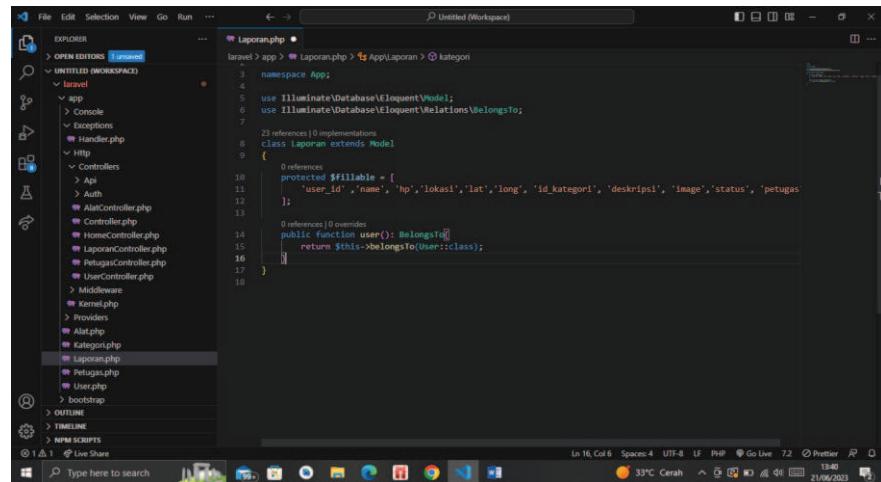


```
Alat.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Alat extends Model
8 {
9
10
11     protected $fillable = [
12         'name', 'image', 'deskripsi', 'jumlah'
13     ];
14 }
15
```

Gambar 40. Modal Alat.php

### b. Model Laporan.php

*Souce code* ini berfungsi membantu mengolah data untuk dimasukan ke *database*, pada *source code* ini beriri *variable* yaitu user\_id , name, hp,lokasi, lat, long, id\_kategori, deskripsi, image, status, petugas.



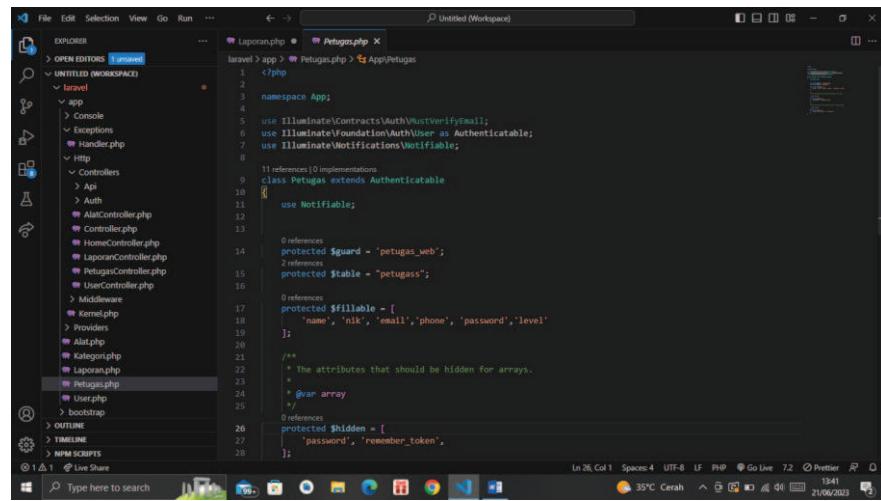
```
Laporan.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Illuminate\Database\Eloquent\Relations\BelongsTo;
7
8 class Laporan extends Model
9 {
10
11     protected $fillable = [
12         'user_id', 'name', 'hp', 'lokasi', 'lat', 'long', 'id_kategori', 'deskripsi', 'image', 'status', 'petugas'
13     ];
14
15     public function user(): BelongsTo
16     {
17         return $this->belongsTo(User::class);
18     }
19 }
```

Gambar 41. Modal Laporan.php

### c. Model Petugas.php

*Souce code* ini berfungsi membantu mengolah data untuk

dimasukan ke *database*, pada *source code* ini beriri *variable* yaitu name, nik, email, phone, password, fcm.



```

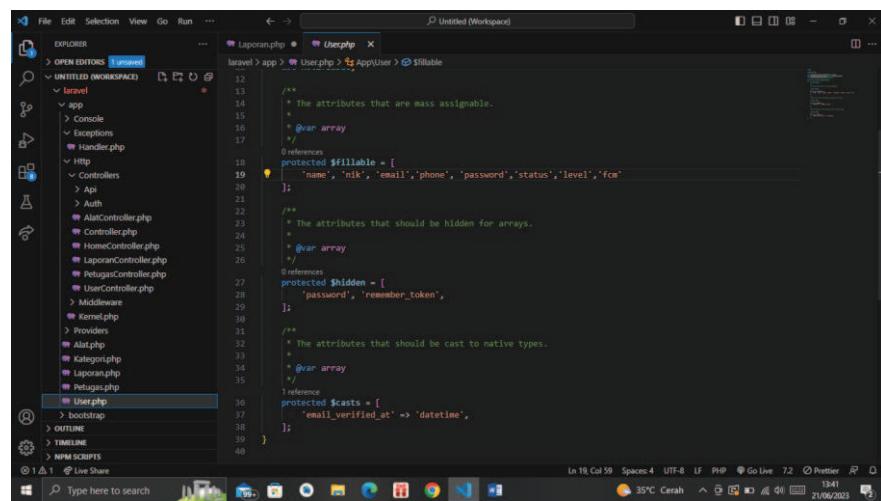
1 <?php
2
3 namespace App;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8
9
10 class Petugas extends Authenticatable
11 {
12     use Notifiable;
13
14     protected $guard = 'petugas_web';
15     protected $table = 'petugass';
16
17     protected $fillable = [
18         'name', 'nik', 'email', 'phone', 'password', 'level'
19     ];
20
21     /**
22      * The attributes that should be hidden for arrays.
23      *
24      * @var array
25     */
26     protected $hidden = [
27         'password', 'remember_token',
28     ];
29
30     /**
31      * The attributes that should be cast to native types.
32      *
33      * @var array
34     */
35     protected $casts = [
36         'email_verified_at' => 'datetime',
37     ];
38 }
39

```

Gambar 42. Modal Petugas.php

#### d. Model User.php

*Source code* ini berfungsi membantu mengolah data untuk dimasukan ke *database*, pada *source code* ini beriri *variable* yaitu name, nik, email, phone, password, status, level, fcm.



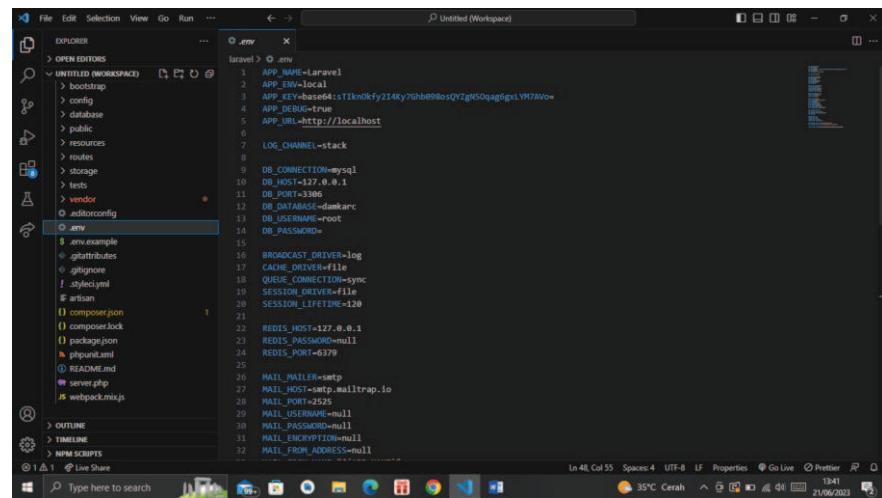
```

12 /**
13  * The attributes that are mass assignable.
14  *
15  * @var array
16  */
17 protected $fillable = [
18     'name', 'nik', 'email', 'phone', 'password', 'status', 'level', 'fcm'
19 ];
20
21 /**
22  * The attributes that should be hidden for arrays.
23  *
24  * @var array
25 */
26 protected $hidden = [
27     'password', 'remember_token',
28 ];
29
30 /**
31  * The attributes that should be cast to native types.
32  *
33  * @var array
34 */
35 protected $casts = [
36     'email_verified_at' => 'datetime',
37 ];
38

```

Gambar 43. Modal User.php

#### 4. .env



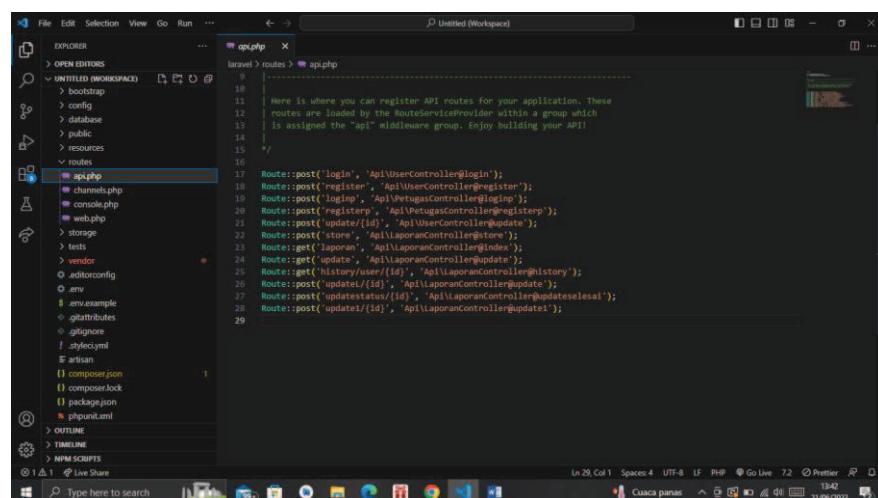
```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:CL1kn0Rfy214Ky70h090sQV2gN50qag6gxLW7AVo-
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtp
MAIL_HOST=satp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME=null
```

Gambar 44. .env

#### 5. Router

##### a. Api Router

Berfungsi untuk menangani *request* dari aplikasi *mobile* untuk memanggil halaman atau *resource* *api controller* sesuai dari *request*.

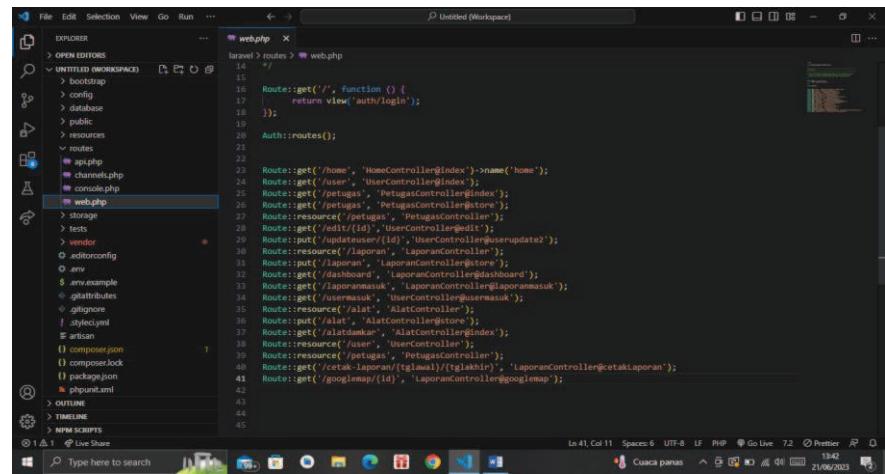


```
Route::post('login', 'ApiUserController@login');
Route::post('register', 'ApiUserController@register');
Route::post('logout', 'ApiUserController@logout');
Route::post('profile', 'ApiUserController@profile');
Route::post('update/{id}', 'ApiUserController@update');
Route::post('store', 'ApiLaporanController@store');
Route::get('laporan', 'ApiLaporanController@index');
Route::get('update', 'ApiLaporanController@update');
Route::get('history/user/{id}', 'ApiLaporanController@history');
Route::post('update/{id}', 'ApiLaporanController@update');
Route::post('updatestatus/{id}', 'ApiLaporanController@updatestatus');
Route::post('updatei/{id}', 'ApiLaporanController@updatei');
```

Gambar 45. Api Router

### b. Web Router

Berfungsi untuk menangani *request* pengguna di *url* kemudian mengarahkan aplikasi untuk memanggil halaman atau *resource* tertentu.



```
File Edit Selection View Go Run ... Untitled (Workspace)
EXPLORER
> OPEN EDITORS
> UNTITLED WORKSPACE
> bootstrap
> config
> database
> public
> resources
> routes
> api.php
> channels.php
> console.php
> web.php
laravel > routes > web.php
14 /**
15  Route::get('/', function () {
16      return view('auth/login');
17  });
18
19
20  Auth::routes();
21
22
23  Route::get('/home', 'HomeController@index')->name('home');
24  Route::get('/user', 'UserController@index');
25  Route::get('/petugas', 'PetugasController@index');
26  Route::get('/petugas', 'PetugasController@store');
27  Route::resource('/petugas', 'PetugasController');
28  Route::get('/edit/{id}', 'UserController@getedit');
29  Route::put('/updateuser/{id}', 'UserController@update2');
30
31  Route::put('/laporan', 'LaporanController@store');
32  Route::get('/dashboard', 'LaporanController@dashboard');
33  Route::get('/laporanmasuk', 'LaporanController@laporanmasuk');
34  Route::get('/usermasuk', 'UserController@usermasuk');
35  Route::put('/usermasuk', 'UserController@updateuser');
36  Route::put('/alat', 'AlatController@store');
37  Route::get('/alatdaftar', 'AlatController@index');
38  Route::resource('/user', 'UserController');
39  Route::resource('/petugas', 'PetugasController');
40  Route::get('/cetak-laporan/{tglaaw}/{tglaakhir}', 'LaporanController@cetakLaporan');
41  Route::get('/googlemap/{id}', 'LaporanController@googlemap');
42
43
44
```

Gambar 46. Web Route

### 6. Layout

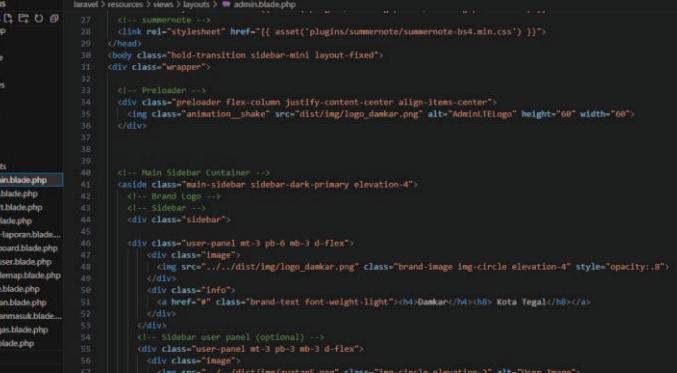
#### a. login.blade.php

*Source code* ini berfungsi untuk membuat tampilan *login* yang berisi beberapa inputan untuk *login* seperti *input email* dan *password*.

Gambar 47. login.blade.php

b. admin.blade.php

Source code ini berfungsi untuk membuat tampilan *template* dari website seperti tampilan menu pada website yang dapat dipanggil pada *source code* yang lain



```
admin.blade.php
laravel resources views layouts > admin.blade.php
27 <!-- Summernote -->
28 <link rel="stylesheet" href="{{ asset('plugins/summernote/summernote-bs4.min.css') }}"/>
29 </head>
30 <body class="holo-transition sidebar-min layout-fixed">
31 <div class="wrapper">
32
33 <!-- Preloader -->
34 <div class="preloader flex-column justify-content-center align-items-center">
35 
36 </div>
37
38 <!-- Main Sidebar Contains -->
39 <aside class="main-sidebar sidebar-dark-primary elevation-4">
40 <!-- Brand Logo -->
41 <!-- Sidebar -->
42 <div class="sidebar">
43
44 <div class="user-panel mt-3 pb-6 mb-3 d-flex">
45 <div class="image">
46 
47 </div>
48 <div class="info">
49 <a href="#" class="brand-text font-weight-light">DAMKAR</a>
50 <small>Kota Tegal</small>
51 </a>
52 </div>
53 </div>
54 <!-- Sidebar user panel (optional) -->
55 <div class="user-panel mt-3 pb-3 mb-3 d-flex">
56 <div class="image">
57 
58 </div>
```

Gambar 48. admin.blade.php

c. alat.blade.php

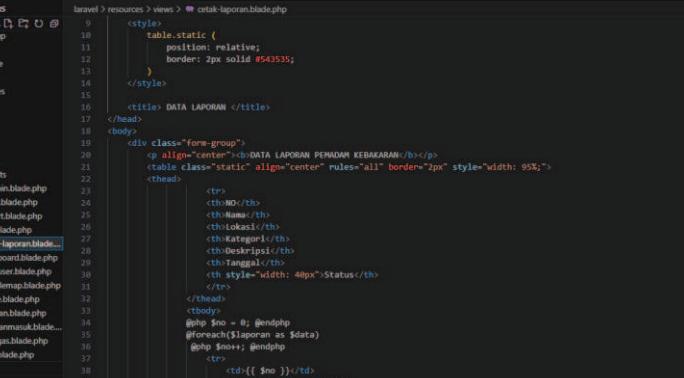
*Source code ini berfungsi untuk membuat tampilan tabel*

yang berisi alat-alat pemadam kebaran dan berisi tombol untuk tambah data dan *update* data.

Gambar 49. alat.blade.php

d. cetak.laporan.blade

Fungsi dari *source code* ini adalah untuk *print* atau *menyimpan* data yang suda di tentukan bulan dan tanggal ke dalam komputer dalam bentuk pdf.



The screenshot shows a Laravel application running in a browser. The URL is `http://127.0.0.1:8000/laporan`. The page displays a table with the following data:

No	Nama	Lokasi	Kategori	Deskripsi	Tanggal	Status
1	John Doe	123 Main St	Category A	Report description	2023-10-01	Approved
2	Jane Smith	456 Elm St	Category B	Report description	2023-10-02	Pending
3	Bob Johnson	789 Oak St	Category C	Report description	2023-10-03	Approved
4	Sarah Davis	210 Pine St	Category D	Report description	2023-10-04	Pending
5	David Wilson	321 Cedar St	Category E	Report description	2023-10-05	Approved

The browser's address bar shows `http://127.0.0.1:8000/laporan`. The page title is "Untitled (Workspace)". The developer tools are open, showing the DOM structure and network requests.

Gambar 50. cetak.blade.php

e. dashboard.blade.php

*Source code* ini berfungsi untuk membuat tampilan tabel yang berisi grafik dari laporan yang sudah di tanggani berdasarkan lokasi kecamatan di kota tegal.

Gambar 51. dashboard.blade.php

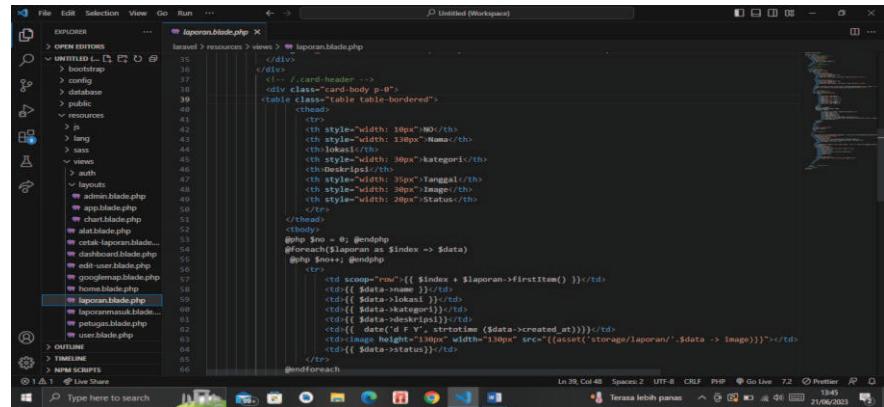
f. googlemap.blade.php

*Source code* ini berfungsi untuk menampilkan maps lokasi dari laporan warga agar tidak ada laporan fiktif atau laporan palsu.

Gambar 52. googlemap.blade.php

### g. laporan.blade.php

*Source code* ini berfungsi untuk membuat tampilan tabel yang berisi tabel semua laporan yang sudah selesai di tangani oleh pemdam kebakaran.



```
laravel > resources > views > laporan.blade.php
35 </div>
36 </div>
37 </div>
38 </div>
39 <div class="card-header p-0">
40   <table class="table table-bordered">
41     <thead>
42       <tr>
43         <th style="width: 10px;">No</th>
44         <th style="width: 130px;">Nama</th>
45         <th>lokasi</th>
46         <th style="width: 30px;">kategori</th>
47         <th>Deskripsi</th>
48         <th style="width: 35px;">Tanggal</th>
49         <th style="width: 30px;">Image</th>
50         <th style="width: 20px;">Status</th>
51     </tr>
52   </thead>
53   <tbody>
54     @php $no = 0; @endphp
55     @foreach($laporan as $index => $data)
56     @php $no++; @endphp
57     <tr>
58       <td scope="row">{{ $index + $laporan->firstItem() }}</td>
59       <td>{{ $data->name }}</td>
60       <td>{{ $data->lokasi }}</td>
61       <td>{{ $data->kategori }}</td>
62       <td>{{ $data->deskripsi }}</td>
63       <td>{{ date('d F Y', strtotime($data->created_at)) }}</td>
64       <td>
23   <thead>
24     <tr>
25       <th style="width: 10px;">No</th>
26       <th style="width: 130px;">Nama</th>
27       <th style="width: 30px;">kategori</th>
28       <th>Deskripsi</th>
29       <th style="width: 30px;">Image</th>
30       <th style="width: 20px;">Status</th>
31       <th>Action</th>
32     </tr>
33   </thead>
34   <tbody>
35     @php $no = 0; @endphp
36     @foreach($listUser as $data)
37     @php $no++; @endphp
38     <tr>
39       <td>{{ $no }}</td>
40       <td>{{ $data->name }}</td>
41       <td>{{ $data->lokasi }}</td>
42       <td>{{ $data->kategori }}</td>
43       <td>{{ $data->deskripsi }}</td>
44       <td>id }} " data-
48         >Verifikasi</button>
49         <button type="button" class="btn btn-block btn-warning btn-sm" data-id="{{ $data->id }} " data-
50         >Lihat</button>
51       </td>
52     </tr>
53   @endforeach
54 </tbody>
55 </table>
56 </div>
```

Gambar 54. laporanmasuk.blade.php

i. petugas.blade.php

*Source code* ini berfungsi untuk membuat tampilan tabel petugas, dan terdapat tombol tambah petugas untuk menambahkan petugas dan tombol edit untuk mengedit data petugas.



```
PUTRI
putri@email.com
Admin

Method: POST Action: {{ route('petugas.update', 'test') }} Role: Form enctype: multipart/form-data
{{ csrf_field() }} {{ method_field('PUT') }}
```

```
petugas.blade.php
laravel > resources > views > petugas.blade.php
39 > @php $no = 0; @endphp
40 > @endphp @foreach($data as $data)
41 > @php $no++; @endphp
42 > <tbody>
43 > <tr>
44 > <td>{{ $no }}</td>
45 > <td>{{ $data->name }}</td>
46 > <td>{{ $data->email }}</td>
47 > <td>{{ $data->level }}</td>
48 > <td>
49 > <button type="button" class="btn btn-primary" data-id="{{ $data->id }}" data-status="{{ $data->status }}>Edit</button>
50 > </td>
51 > </tr>
52 > @endif
53 > </tbody>
54 > </table>
55 > </div>
56 > <!-- /.card-body -->
57 > </div>
58 > </div>
59 > <div class="modal fade" id="exampleModal13" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel">
60 > <div class="modal-dialog" role="document">
61 > <div class="modal-content">
62 > <div class="modal-header">
63 > <h4 class="modal-title" id="exampleModalLabel">Edit Petugas</h4>
64 > <button type="button" class="close" data-dismiss="modal" aria-label="Close">
65 > <span aria-hidden="true">&amptimes</span>
66 > </button>
67 > </div>
68 > <div class="modal-body">
69 > <form method="POST" action="{{ route('petugas.update', 'test') }}" role="form" enctype="multipart/form-data">
70 > {{ csrf_field() }} {{ method_field('PUT') }}
```

Gambar 55. petugas.blade.php

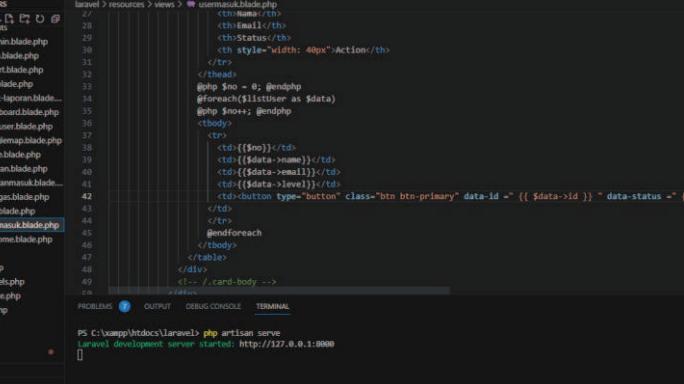
j. user.blade.php

*Source code* ini berfungsi untuk membuat tampilan tabel yang berisi tabel *user* yang telah aktif dan tombol *delete* untuk menghapus user yang sudah tidak aktif.

Gambar 56. user.blade.php

k. usermasuk.blade.php

*Source code* ini berfungsi untuk membuat tampilan tabel yang berisi tabel *user* yang baru melakukan *registrasi* dan terdapat tombol untuk admin dapat *verifikasi user* yang sudah *registrasi*.



The screenshot shows the Visual Studio Code interface with a Laravel project open. The left sidebar displays the project structure under 'OPEN EDITORS' and 'ROUTINES'. The main editor area shows a blade.php file named 'usermask.blade.php' with the following code:

```
laravel / resources / views > usermask.blade.php
1 laravel / resources / views > usermask.blade.php
2 <?php
3 <?php $no = 0; ?>
4 <?php foreach($listUser as $data) ?>
5 <?php $no++; ?>
6 <tbody>
7 <tr>
8 <td>{{ $no }}</td>
9 <td>{{ $data->name }}</td>
10 <td>{{ $data->email }}</td>
11 <td>{{ $data->level }}</td>
12 <td><button type="button" class="btn btn-primary" data-id="{{ $data->id }} " data-status="{{ $data->status }}>Edit</button></td>
13 </tr>
14 </tbody>
15 </table>
16 </div>
17 <!-- ./card-body -->
18 </div>
```

The bottom status bar shows the command: `PS C:\wamp\www\lara> php artisan serve` and the output: `Laravel development server started: http://127.0.0.1:8000`. The bottom right corner shows the date and time: `18/03 21/06/2023`.

## Gambar 57. usermasuk.blade.php

## Lampiran 10. Surat HKI Yang Terbit



**LAMPIRAN PENCIPTA**

No	Nama	Alamat
1	Wahyu Indrawan	Jalan Brawijaya, Rt 04 Rw 01, Kelurahan Muarareja, Kecamatan Tegal Barat
2	Ginanjar Wiro Sasmito	Jl. Raya Kluwut Timur No 24, RT 03 RW 02, Kluwut, Kecamatan Bulakamba, Kabupaten Brebes, Jawa Tengah, 52253
3	Dega Surono Wibowo	Perumahan Sapphire Regency Blok H No 1, RT 004 RW 001, Kelurahan Pulosari, Kecamatan Brebes, Kabupaten Brebes, Jawa Tengah, 52213



