

**RONI ANDARSYAH
DERISKA FADILLA MUSDALIFA**



TUNTUNAN PRAKTIS



**MEMBUAT PEMETAAN
LOKASI PENDUDUK
MENGUNAKAN GPS
BERBASIS ANDROID**

2 0 2 2

**TUNTUNAN PRAKTIS
MEMBUAT PEMETAAN LOKASI
PENDUDUK MENGGUNAKAN
GPS BERBASIS ANDROID**

TUNTUNAN PRAKTIS MEMBUAT PEMETAAN LOKASI PENDUDUK MENGGUNAKAN GPS BERBASIS ANDROID

**RONI ANDARSYAH
DERISKA FADILLA MUSDALIFA**



TUNTUNAN PRAKTIS MEMBUAT PEMETAAN LOKASI PENDUDUK MENGGUNAKAN GPS BERBASIS ANDROID

©BUKU PEDIA

Penulis:

Roni Andarsyah
Deriska Fadilla Musdalifa

Editor:

(Nama Penguji Sidang)

Cetakan Pertama: Isi dengan Bulan saat upload buku

Cover: Tim Penyusun

Tata Letak: Tim Kreatif Penerbit

Hak Cipta 2023, pada Penulis. Diterbitkan pertama kali oleh:

ISI NAMA PENERBIT
ISI ALAMAT PENERBIT

Website: **WEBSITE PENERBIT**
E-mail: **EMAIL PENERBIT**

Copyright © 2023 by **NAMA PENERBIT**
All Right Reserved

- Cet. I –: **NAMA PENERBIT, TAHUN TERBIT**
Dimensi : 14,8 x 21 cm
ISBN: **KOSONGKAN DULU**

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta
Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

KATA PENGANTAR

Dengan memanjatkan Puji Syukur ke hadirat Allah SWT karena atas berkah, rahmat, dan hidayahnya penulis dapat menyelesaikan buku yang berjudul “*Tuntunan Praktis membuat Pemetaan Lokasi Penduduk menggunakan GPS berbasis Android*” secara baik dan juga tepat waktu.

Buku ini disusun dengan harapan dapat memberikan manfaat bagi siapa saja yang membacanya. Penulis menyadari bahwasanya buku ini dapat disusun sedemikian rupa tidak terlepas dari dukungan dan bantuan moral, moril, serta ilmu berbagai pihak. Untuk itu penulis ingin menyampaikan ucapan terima kasih terutama kepada :

1. Tuhan Yang Maha Esa atas segala rahmatnya sehingga penulis dapat menyusun laporan secara baik;
2. Kedua orang tua serta rekan-rekan yang selalu memberikan doa dan dukungannya;
3. Bapak ARDI FRIYATNA, S.Kom selaku pembimbing eksternal di pemerintahan kota X;
4. Pihak pemerintahan kota X yang telah memberikan saya kesempatan untuk melaksanakan magang di instansi;
5. Bapak Cahyo Prianto, S.Pd., M.T. selaku koordinator;
6. Dosen yang telah membimbing saya sehingga saya dapat menyusun buku ini.

Akhir kata dari saya semoga Allah SWT membalas segala kebaikan semua pihak yang terlibat dalam membantu penyusunan proposal. Semoga proposal internship ini membawa manfaat bagi perkembangan ilmu pengetahuan dan teknologi di masa yang akan datang.

Bandung, 10 Januari 2023

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	1
DAFTAR TABEL	3
BAB 1	4
Pendahuluan	4
1.1 Latar Belakang	4
1.2 Tujuan Pemetaan Penduduk	5
1.3 Ruang Lingkup	5
BAB 2.....	7
Studi Literature.....	7
2.1 Teori	7
2.1.1 Kependudukan	7
2.1.2 GIS.....	7
2.1.3 Lokasi	8
2.1.4 Google Maps API	8
2.1.5 Android	9
2.1.6 Java	9
2.1.7 Database	9
2.1.8 MySQL.....	10
2.2 Agile Development.....	11
2.2.1 Tujuan Agile development.....	11
2.2.2 Jenis Agile development	13
2.2.3 Kelebihan Agile development	16

2.2.4	Kekurangan Agile development	16
BAB 3	17
Alur Aplikasi Pemetaan	17
3.1	Menentukan Metode yang digunakan	17
3.2	Perancangan UI Aplikasi	17
3.2.1	User Interface Splash Screen	17
3.2.2	User Interface Login	18
3.2.3	User Interface Register	19
3.2.4	User Interface Main Menu	20
3.2.5	User Interface Menu Bantuan	23
BAB 4	25
Tutorial Membuat Aplikasi	25
4.1	Alur Perancangan	25
4.1.1	Analisis Sistem Berjalan	25
4.1.2	Analisis Sistem yang akan dibangun	27
4.2	Lingkungan Implementasi	50
4.2.1	Implementasi Perangkat Keras	51
4.2.2	Implementasi Perangkat Lunak	51
4.3	Langkah Pembuatan	51
4.3.1	Persiapan Key Maps	51
4.3.2	Persiapan Database	54
4.3.3	Memulai membuat Aplikasi	54
4.3.4	Mempersiapkan Gradle dan manifest	56
4.3.5	Membuat Activity	57
4.3.6	Hasil Uji Coba	77
BAB 5	80
Kesimpulan dan Saran	80
5.1	Kesimpulan	80

5.2	Saran.....	80
DAFTAR	PUSTAKA.....	81

DAFTAR GAMBAR

Gambar 2. 1 Skema model XP	14
Gambar 3. 1 UI Splash Screen	18
Gambar 3. 2 UI Login Admin.....	19
Gambar 3. 3 UI Login User	19
Gambar 3. 4 UI Register	20
Gambar 3. 5 UI Register Success	20
Gambar 3. 6 Menu User.....	21
Gambar 3. 7 Menu Admin	22
Gambar 3. 8 Tambah Penduduk.....	23
Gambar 3. 9 Tambah Penduduk Success.....	23
Gambar 3. 10 Edit Data.....	23
Gambar 3. 11 Hapus Data	23
Gambar 3. 12 Bantuan	24
Gambar 4. 1 Analisis Prosedur yang Berjalan.....	26
Gambar 4. 2 Flowmap akan dibangun.....	29
Gambar 4. 3 Flowmap Aplikasi yang dibangun	30
Gambar 4. 4 Usecase Diagram	31
Gambar 4. 5 Register.....	38
Gambar 4. 6 Login User.....	40
Gambar 4. 7 Read User	41
Gambar 4. 8 Update User	42
Gambar 4. 9 User Cari	43
Gambar 4. 10 Login Admin	44
Gambar 4. 11 Read Admin	45
Gambar 4. 12 Tambah Admin.....	46
Gambar 4. 13 Update Admin	47
Gambar 4. 14 Admin Hapus.....	48
Gambar 4. 15 Bantuan	49
Gambar 4. 16 Keluar.....	50
Gambar 4. 17 Cloud google.....	52
Gambar 4. 18 Navbar Cloud	53
Gambar 4. 19 Create Credential.....	53
Gambar 4. 20 APIKey Create	53
Gambar 4. 21 Edit APIKey	54

Gambar 4. 22 Permulaan 1.....	55
Gambar 4. 23 Permulaan 2.....	55
Gambar 4. 24 Permulaan 3.....	56

DAFTAR TABEL

Tabel 4 1 Dokumen yang berjalan	27
Tabel 4 2 Definisi Aktor	32
Tabel 4 3 Usecase Halaman Login User	33
Tabel 4 4 Usecase Halaman Utama User	33
Tabel 4 5 Usecase Register User	34
Tabel 4 6 Usecase Halaman Login admin	35
Tabel 4 7 Usecase Halaman Utama Admin	36
Tabel 4 8 Usecase Halaman Bantuan	36
Tabel 4 9 Usecase Logout	37
Tabel 4 10 hasil uji coba	79

BAB 1

Pendahuluan

1.1 Latar Belakang

Di era modern ini, peran teknologi tidak lepas dari berbagai aktivitas yang dilakukan oleh masyarakat. Hal tersebut berlaku di semua bidang. Teknologi digunakan untuk mempermudah kegiatan yang dilakukan oleh manusia, sehingga teknologi semakin berkembang pesat dari tahun ke tahun. Perkembangan modernisasi komunikasi saat ini berkembang pesat hingga sekarang teknologi yang sedang berkembang adalah teknologi mobile pada perangkat telepon pintar smartphone. Hal ini faktanya bisa dibuktikan dengan berkembangnya berbagai jenis aplikasi smartphone yang dibuat terdepan vendor ponsel di dunia. Baik secara langsung maupun tidak langsung mempengaruhi pembangunan.

Kota X merupakan salah satu kota yang terletak di Provinsi X. Pada saat ini proses pencarian informasi lokasi penduduk secara detail masih sulit diketahui oleh petugas survei. Oleh karena itu sangat dibutuhkan suatu sistem informasi yang dapat menyajikan informasi detail mengenai alamat dan lokasi masyarakat yaitu Sistem Informasi Geografis (SIG) berbasis mobile android untuk memetakan lokasi masyarakat kota x. Aplikasi SIG dibangun

dengan Bahasa Java, Google Maps API untuk peta lokasi masyarakat terkini dan MySQL untuk membuat basis data. Aplikasi yang digunakan untuk membuat sistem aplikasi ini adalah Android Studio. Untuk menjalankan aplikasi ini membutuhkan aktivasi GPS di ponsel. Untuk itulah penulis mencoba membuat sistem pemetaan lokasi penduduk yang dapat dijalankan pada Smartphone.

Pembuatan aplikasi survei berbasis Android dapat membantu memetakan hasil survei berdasarkan lokasi, serta menyiapkan pertanyaan dan mendistribusikan survei dari perangkat seluler. Pemetaan pengukuran berbasis lokasi dimungkinkan dengan memanfaatkan fungsi GPS pada smartphone. Metode pengembangan sistem aplikasi yang digunakan yaitu metode Agile Development.

1.2 Tujuan Pemetaan Penduduk

Tujuan yang ingin dicapai dalam penelitian ini adalah membangun sistem yang menampilkan peta lokasi penduduk untuk membantu mempermudah petugas survei lapangan dalam mencari lokasi penduduk berdasarkan alamat penduduk dengan berbasis mobile android.

1.3 Ruang Lingkup

Dalam penelitian ini terdapat Batasan masalah sebagai berikut:

1. Aplikasi ini memanfaatkan Maps API dari google yang digunakan untuk menentukan lokasi penduduk;
2. Aplikasi ini dibangun dengan Bahasa java;
3. Aplikasi hanya diperuntukkan smartphone ber-platform android;
4. Data lokasi penduduk akan disimpan dalam database MySQL.

BAB 2

Studi Literature

2.1 Teori

2.1.1 Kependudukan

Penduduk adalah orang dalam matranya sebagai pribadi, anggota keluarga, anggota masyarakat, warga negara dan himpunan kuantitas yang ertempat tinggal di suatu tempat dalam batas wilayah tertentu (Mantra, 2009).

Pengertian kependudukan menurut (Firmansyah, Desember 2018) adalah hal ihwal yang berkaitan dengan jumlah, struktur, umur, jenis kelamin, agama, kelahiran, perkawinan, kehamilan, kematian, persebaran, mobilitas dan kualitas serta ketahanannya yang menyangkut politik, ekonomi, sosial, dan budaya.

2.1.2 GIS

Menurut (Irwansyah, 2013) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis.

GIS menurut (Syam'ani, 2016) merupakan sebuah system yang digunakan untuk menginput, menyimpan, memanggil kemnali, menganalisis, dan memvisualisasikan data atau informasi spasial.

GIS menurut (Awangga, 2019) merupakan sebuah computer

yang berbasis sistem informasi digunakan untuk memberikan informasi bentuk digital dan analisis terhadap permukaan geografi bumi.

2.1.3 Lokasi

(Alma, 2003) mengemukakan bahwa " Lokasi adalah tempat perusahaan beroperasi atau tempat perusahaan melakukan kegiatan untuk menghasilkan barang dan jasa yang mementingkan segi ekonominya".

Menurut (Suwarman, 2004) "lokasi merupakan tempat usaha yang sangat mempengaruhi keinginan seseorang konsumen untuk datang dan berbelanja".

Sedangkan pengertian lokasi menurut (Kasmir, 2009) yaitu tempat melayani konsumen, dapat pula diartikan sebagai tempat untuk memajangkan barang- barang dagangannya.

2.1.4 Google Maps API

Google Maps API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan Google Map di suatu aplikasi yang tidak dibuat oleh Google. Google Maps API adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan (Elian & H, 2012)

Google Maps API memiliki kemampuan untuk mengambil data gambar peta statis dan memberikan penuntun arah yang bersifat gratis. Kekurangannya harus tersedia layanan internet pada

perangkat jika ingin melakukan akses.

2.1.5 Android

Menurut (Murya, 2014) Android adalah “sistem operasi berbasis linux yang di gunakan untuk telepon seluler (mobile) seperti telepon pintar (smartphone) dan komputer tablet (PDA).”

Menurut Safaat dalam (A, 2018) “Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi utama mobile”.

Menurut Li dalam (A, 2018) “Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (touch screen) yang berbasis linux”.

2.1.6 Java

Menurut (Mardison, Agustus 2017) Java merupakan pemrograman yang populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangat luas, mulai dari komputer hingga smartphone. Bahasa pemrograman Java dikenal dengan pemrograman OOP-Object Oriented Programming.

Menurut definisi Sun Microsystem, di dalam buku (Shalahuddin & Rosa, 2010) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan.

2.1.7 Database

Menurut (Kustiyaningsih, 2011) Database adalah struktur penyimpanan data. Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah database komputer, diperlukan system manajemen database seperti MySQL Server.

Menurut (Yakub, 2011) Basis data (database) diartikan sebagai markas atau gudang data, tempat bersarang atau berkumpul data. Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Adapun tujuan basis data diantaranya sebagai efisisensi yang meliputi speed, space&Accurancy, menangani data dalam jumlah besar, kebersamaan pemakaian, dan meniadakan duplikasi.

2.1.8 MySQL

Menurut (Kadir, 2008) “MySQL adalah sebuah software opensource yang digunakan untuk membuat sebuah database.”

Menurut (Raharjo, 2011) “MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user”.

Menurut (Sibero, 2013) “MySQL atau dibaca “My Sekuel” dengan adalah suatu RDBMS (Relational Database Management System) yaitu aplikasi sistem yang mejalankan fungsi pengolahan data”.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu software atau program yang digunakan untuk membuat sebuah database yang bersifat open source.

2.2 Agile Development

Menurut (Pressman, 2010) Agile Software Development adalah sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, di mana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir. Istilah ini diciptakan pada tahun 2001 ketika Agile Manifesto dirumuskan.

Agile development sendiri merupakan model pengembangan perangkat lunak jangka pendek. Kemudian perlu beradaptasi dengan cepat untuk menghadapi perubahan apapun. Nilai terpenting dari pengembangan agile adalah memungkinkan tim membuat keputusan dengan cepat, dengan kualitas dan prediksi yang baik, serta memiliki potensi yang baik untuk menangani setiap perubahan.

2.2.1 Tujuan Agile development

2.2.1.1 *High - value & working App System*

Tujuan yang pertama, untuk menghasilkan sebuah perangkat lunak dengan nilai jual tinggi serta dapat menekan biaya pembuatan. Dan yang terpenting adalah dapat menghasilkan produk dengan kualitas yang baik.

2.2.1.2 *Iterative, Incremental, Evolutionary*

Agile adalah model pengembangan berulang, berulang, bereksperimen dengan perubahan jika diperlukan. Dapat dikatakan bahwa metode ini cukup fleksibel dan dapat digunakan untuk proyek

pembangunan jangka pendek.

2.2.1.3 *Cost Control & value - driven development*

Proses pengembangan perangkat lunak dapat disesuaikan dengan kebutuhan pengguna. Tim pengembangan kemudian dapat mengontrol biaya dan waktu yang diperlukan untuk proses pengembangan perangkat lunak sesuai kebutuhan.

2.2.1.4 *High - quality production*

Bahkan dengan biaya dan waktu yang relatif sedikit, kualitas produk perangkat lunak tetap terjaga dengan baik.

2.2.1.5 *Flexible & Risk Management*

Fleksibel di sini berarti dapat bertemu dengan klien setiap saat sehingga fungsionalitas perangkat lunak dapat dipertahankan. Yang terpenting, meminimalisir terjadinya error pada program dan produk sebelum proses deployment aplikasi berlangsung.

2.2.1.6 *Collaboration*

Proses kolaboratif di sini dilakukan oleh masing-masing tim pengembangan untuk membahas umpan balik yang diberikan oleh pelanggan. Sehingga perlu adanya komunikasi dan koordinasi yang baik antar tim pengembang.

2.2.1.7 *Self - organizing, Self - Managing Teams*

Tujuan akhir dari pendekatan tangkas ini adalah untuk memungkinkan pengembang mengelola urusan pengembangan perangkat lunak mereka sendiri. Tugas manajer adalah menjadi penghubung antara pengembang dan pelanggan untuk mengurangi miskomunikasi.

2.2.2 Jenis Agile development

4.3.1.1 *Scrum Methodology*

Metode Scrum sendiri lebih menekankan pada kolaborasi antar tim dalam pengembangan perangkat lunak.

4.3.1.2 *Scaled Agile Framework (SAFe)*

SAFe merupakan solusi yang lebih fleksibel dan efisien karena adanya kerjasama antar departemen saat proyek dikerjakan.

4.3.1.3 *Lean Software Development (LSD)*

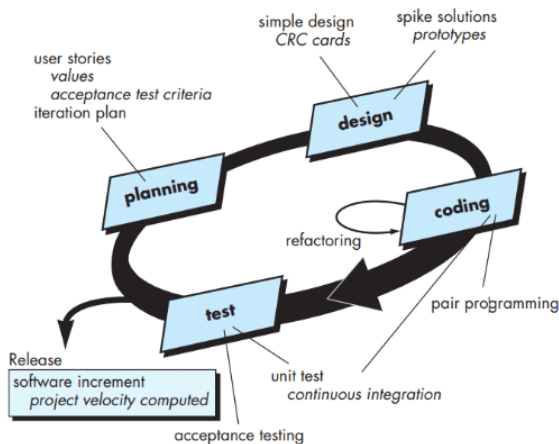
Adalah metodologi tangkas yang ditujukan untuk mengembangkan perangkat lunak dengan sumber daya manusia minimal. LSD dicapai dengan meluncurkan produk dengan fungsi terbatas atau Minimum Viable Product (MVP) berdasarkan permintaan konsumen.

4.3.1.4 *Kanban*

Kanban adalah metodologi tangkas di mana proses dilakukan secara visual untuk memantau aktivitas alur kerja dengan lebih baik.

4.3.1.5 *Extreme Programming (XP)*

Extreme Programming (XP) adalah metodologi tangkas yang berfokus terutama pada aspek teknis pengembangan untuk menghasilkan perangkat lunak berkualitas tinggi. Oleh karena itu, potensi tim pengembangan akan sangat meningkat karena upaya ekstra yang telah mereka lakukan.



Gambar 2. 1 Skema model XP

Tahapan-tahapan dari Extreme Programming terdiri dari:

- planning seperti memahami standar pengguna dan mengembangkan rencana

- design seperti prototipe desain dan presentasi
- coding termasuk pengintegrasian,
- dan yang terakhir adalah testing.

Unsur-unsur lain dari Extreme Programming meliputi paired programming pada tahapan coding, unit testing pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara programmer dan pelanggan ketika terjadi perubahan kebutuhan pelanggan seiring berlalunya waktu berlalu.

4.3.1.6 *Crystal Methodology*

Metodologi tangkas yang lebih berfokus pada kondisi kerja tim daripada alat atau proses. Fokus pada komunikasi tim, interaksi, dokumentasi, dan umpan balik. Dengan demikian, hasil pengembangan perangkat lunak dapat dimaksimalkan.

4.3.1.7 *Dynamic Systems Development Method (DSDM)*

Dynamic Systems Development Method (DSDM) adalah pendekatan gesit yang berfokus pada keterlibatan berkelanjutan dari semua anggota tim untuk membawa perangkat lunak yang benar-benar berguna ke dalam dunia bisnis.

4.3.1.8 *Feature Driven Development (FDD)*

Adalah pendekatan agile yang berfokus pada penyelesaian satu fitur. Mirip dengan Scrum, tetapi hanya 2-20 hari per iterasi. Oleh karena itu, fungsi FDD memiliki skala yang lebih spesifik dan dapat diselesaikan tepat waktu.

2.2.3 Kelebihan Agile development

- Proses pengembangan perangkat lunak membutuhkan waktu yang relatif singkat dan tidak membutuhkan banyak sumber daya.
- Client dapat dengan cepat diproses sesuai dengan kebutuhan pelanggan.
- Client dapat memberikan feedback kepada tim pengembang dalam proses pengembangan program.

2.2.4 Kekurangan Agile development

- Agile tidak cocok jika dikerjakan oleh tim yang tidak berkomitmen untuk menyelesaikan proyek secara bersama-sama.
- Metode Agile ini tidak cocok untuk tim lebih dari 20.
- Tim pengembangan harus siap setiap saat, karena perubahan dapat terjadi kapan saja.

BAB 3

Alur Aplikasi Pemetaan

3.1 Menentukan Metode yang digunakan

Pada pembuatan aplikasi berbasis mobile ini adalah dengan menggunakan metode Agile Development Extreme Programming (XP) untuk pengembangan aplikasi supaya aplikasi dapat dirancang dengan baik dan memiliki kualitas yang baik. Definisi dari metode XP sudah dijelaskan pada bab sebelumnya.

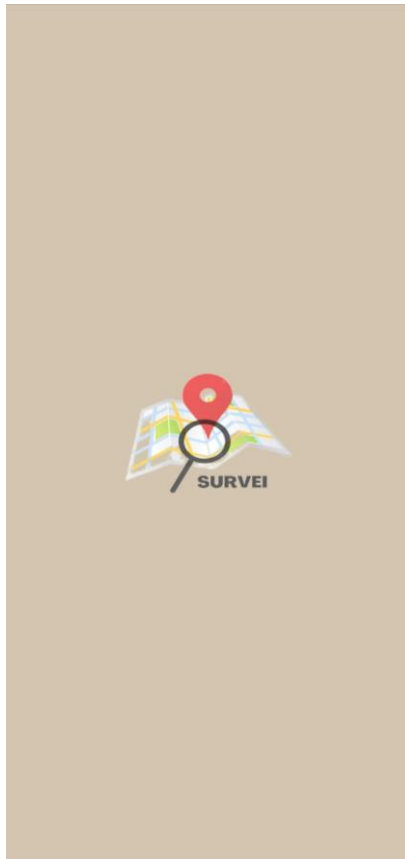
3.2 Perancangan UI Aplikasi

Perancangan UI bertujuan untuk pengguna lebih memahami system yang akan dibangun. Tahapan perancangan UI Aplikasi adalah merancang atau membuat tampilan aplikasi yang meliputi Splash screen, login user, login penduduk, menu user, menu penduduk, menu tentang aplikasi.

Pada perancangan UI aplikasi ini menggunakan *software* Android Studio. Penggunaan aplikasi ini memudahkan *developer* untuk merancang system aplikasi

3.2.1 User Interface Splash Screen

Splash screen merupakan tampilan utama ketika membuka sebuah aplikasi, biasanya hanya terdapat logo dari aplikasi tersebut seperti berikut ini



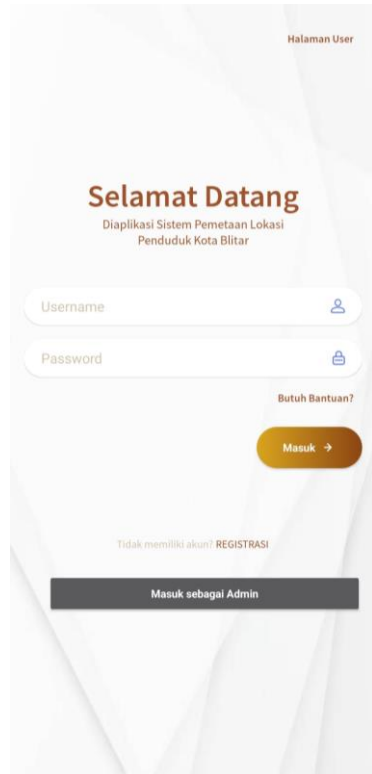
Gambar 3. 1 UI Splash Screen

3.2.2 User Interface Login

Pada user interface login ada 2 pengguna yaitu admin dan user, untuk setiap UI seperti berikut:



Gambar 3. 2 UI Login Admin



Gambar 3. 3 UI Login User

3.2.3 User Interface Register

Pada UI register hanya diperuntukkan user untuk login ke dalam aplikasi dengan syarat NIK harus sudah terdaftar pada database, jika pengguna sudah berhasil register akan muncul halaman baru yaitu register sukses.

Gambar 3. 4 UI Register



Gambar 3. 5 UI Register Success

3.2.4 User Interface Main Menu

3.2.4.1 *Main Menu User*

Pada main menu user berisi data penduduk yang diperoleh dari database MySQL yang berisikan NIK, Nama, TTL, Jenis Kelamin, Agama. Kemudian untuk alamat penduduk perlu menginputkan secara detail sesuai pada KTP, selanjutnya terdapat fragment maps untuk mengetahui lokasi terkini dari seorang penduduk yang menampilkan letak latitude, longitude, dan altitude. Setelah penduduk menyimpan lokasi tersebut

maka akan masuk ke dalam database MySQL.

12:00

PEMETAAN LOKASI PENDUDUK

PEMETAAN LOKASI
PENDUDUK
KOTA BLITAR

NIK
[nik_user]
Name
[nama_user]
Tempat, Tanggal Lahir
[ttl_user]
Jenis Kelamin
[jk_user]
Agama
[agama_user]

Alamat Masukkan Alamat Lengkap anda

fragment

Latitude
Longitude
Altitude
Akurasi

Cari

Simpan Lokasi

Gambar 3. 6 Menu User

3.2.4.2 *Main Menu Admin*

Pada main menu admin menampilkan *list data* penduduk menggunakan *recyclerview* kemudian pada bagian bawah halaman terdapat *button* tambah, ubah, dan delete data untuk memodifikasi data penduduk yang berada pada database MySQL. Kemudian pada bagian pojok atas terdapat sebuah menu untuk masuk ke halaman bantuan dan untuk logout dari akun.



Gambar 3. 7 Menu Admin

Selanjutnya ketika button tambah data di klik, maka akan muncul kolom untuk diisi sesuai data asli, selesai menambahkan data maka akan muncul halaman baru yang menyatakan tambah data berhasil.

PEMETAAN LOKASI PENDUDUK

Tambah Data Penduduk

isi data sesuai dengan data as.

NIK KTP

Nama

Tempat Lahir

Tanggal Lahir (dd-mm-yyyy)

Tambah Data

Kembali

Gambar 3. 8 Tambah Penduduk

PEMETAAN LOKASI PENDUDUK

Selamat!

Akun anda telah berhasil dibuat. Sekarang anda bisa masuk untuk mengakses akun anda!

Go to Halaman Admin

Gambar 3. 9 Tambah Penduduk Success

Kemudian pada *button* edit dan delete data menggunakan sebuah *dialog* yang berisi seperti pada gambar.

id

Nama

Alamat

Gambar 3. 10 Edit Data

nik

Gambar 3. 11 Hapus Data

3.2.5 User Interface Menu Bantuan

| 23

UI Menu bantuan berisikan kontak admin dari pembuat aplikasi itu sendiri.



Gambar 3. 12 Bantuan

BAB 4

Tutorial Membuat Aplikasi

4.1 Alur Perancangan

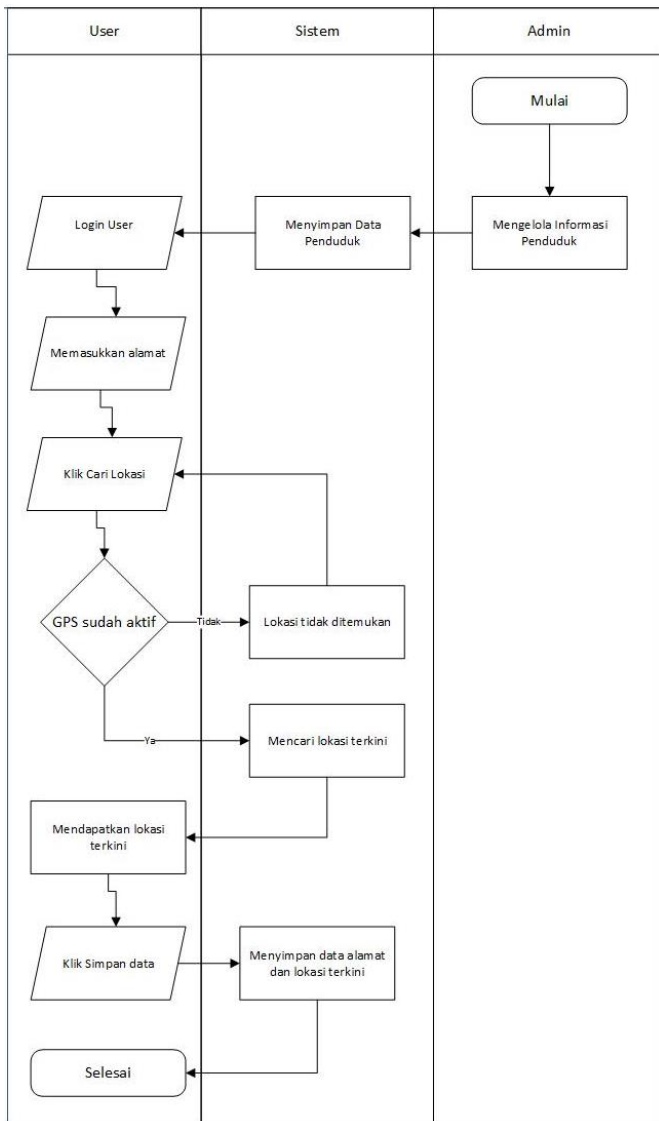
Dalam membangun sebuah aplikasi memerlukan perancangan. Mulai dari flowchart untuk setiap aktivitas pada aplikasi hingga pembuatan diagram-diagram seperti class diagram, sequence diagram, activity diagram serta use case diagram.

4.1.1 Analisis Sistem Berjalan

Pada tahapan melakukan analisis ini, Langkah pertama adalah membuat analisis sistem berjalan, yaitu analisis prosedur yang berjalan dan langkah kedua adalah membuat analisis dokumen yang berjalan

4.1.2.1 Analisis Prosedur yang Berjalan

Berikut merupakan analisis prosedur yang berjalan.



Gambar 4. 1 Analisis Prosedur yang Berjalan

Pada gambar diatas, sistem yang berjalan melibatkan user dan admin. Berikut ini adalah deskripsi

tahapan dari analisis prosedur yang berjalan.

- a. Admin mengelola data penduduk dan admin dapat melakukan CRUD.
- b. Data yang sudah dikelola akan disimpan pada sistem untuk membangun sistem aplikasi.
- c. User menjalankan aplikasi dengan melihat dan mengubah data personal.
- d. User klik simpan.
- e. System menyimpan data yang di kelola user.

4.1.2.2 Analisis Dokumen yang Berjalan

Berikut adalah analisis dokumen yang berjalan.

Dibuat Oleh	Admin
Dibuat Untuk	User
Isi	Berupa informasi data penduduk
Frekuensi	Dibuat sesuai data yang masuk
Tujuan	Menyimpan informasi lokasi data penduduk

Tabel 4 1 Dokumen yang berjalan

4.1.2 Analisis Sistem yang akan dibangun

Setelah melakukan analisis sistem yang berjalan, selanjutnya adalah melakukan analisis sistem yang akan dibangun. Berikut ini adalah analisis sistem yang akan dibangun, yaitu prosedur yang akan dibangun.

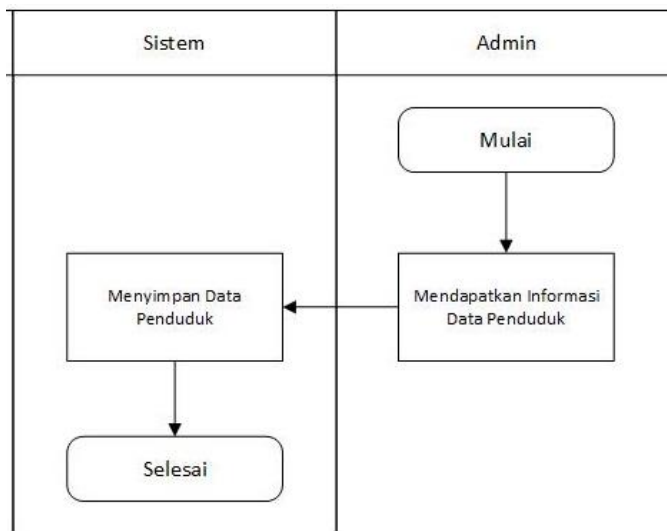
4.1.2.1 Analisis Prosedur yang akan dibangun

Pada tahapan ini menjelaskan analisis prosedur yang akan dibangun. Berikut ini adalah analisis prosedur yang akan dibangun

4.1.2.1.1 Analisis Prosedur yang akan dibangun pada data penduduk

Berikut ini adalah proses mengelola data penduduk yang akan dibangun pada sistem.

- a. Admin akan mengumpulkan data penduduk.
- b. Setelah mendapatkan data, data akan dimasukkan ke dalam database kemudian diimplementasikan pada aplikasi.

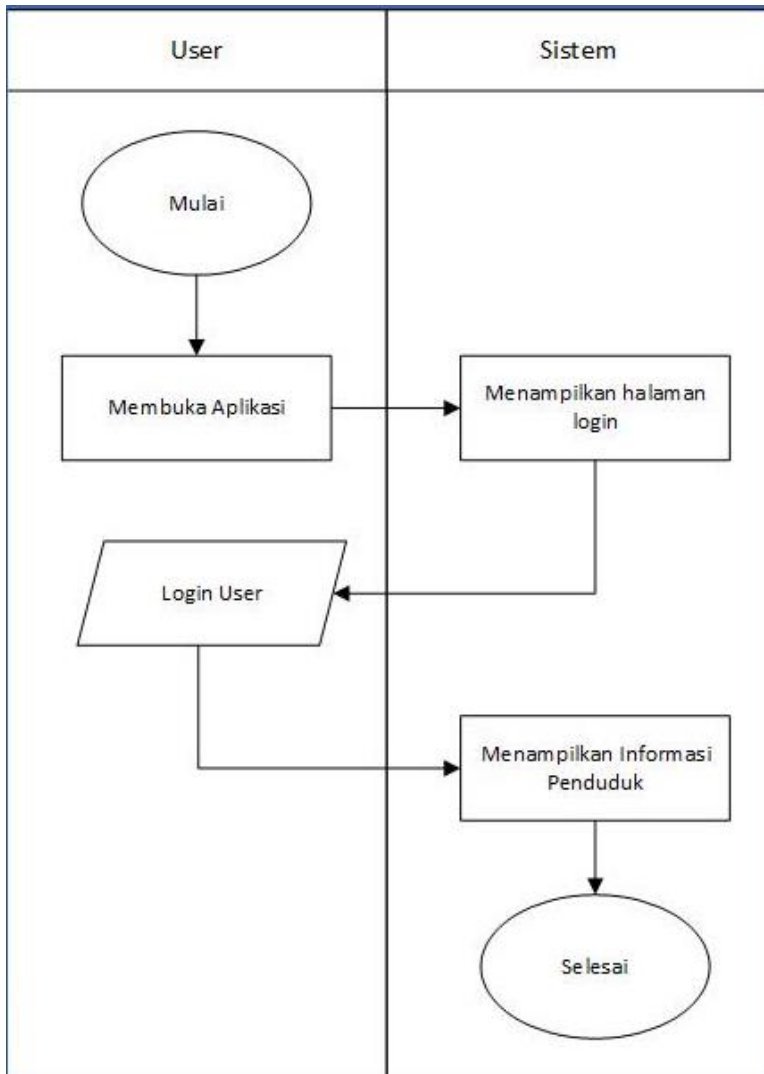


Gambar 4. 2 Flowmap akan dibangun

4.1.2.1 Analisis Prosedur yang akan dibangun pada Aplikasi

Berikut ini adalah proses user menjalankan aplikasi yang akan dibangun pada sistem.

- Tahapan dimulai dari user membuka aplikasi.
- Selanjutnya system akan menampilkan sebuah halaman login.
- Lalu, user melakukan login kemudian system akan menampilkan halaman utama yang berisi informasi penduduk.



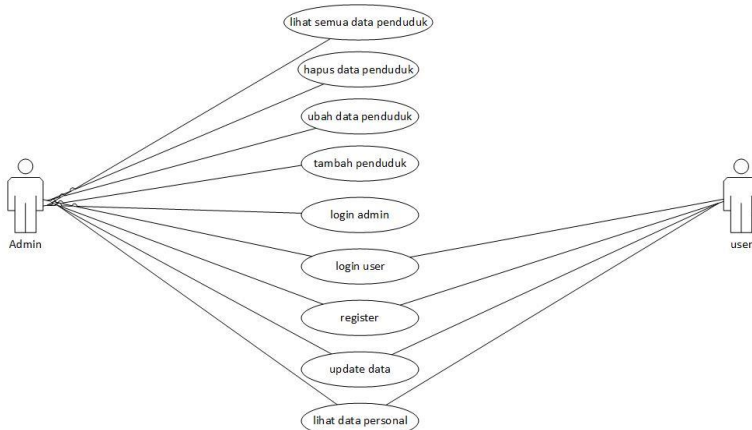
Gambar 4. 3 Flowmap Aplikasi yang dibangun

4.1.2.2 ***UML Diagram***

Setelah membuat analisis prosedur yang akan dibangun, kita akan merancang UML Diagram untuk membangun AR ini. UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

4.1.2.2.1 Usecase Diagram

Pada *Use Case Diagram* yang merupakan sebuah interaksi antara satu atau lebih *actor* terhadap system yang dibuat. *Use Case Diagram* merupakan gambaran umum tentang fungsi apa saja yang terdapat pada system dan siapa saja yang dapat melakukan hal tersebut. Pada aplikasi yang akan dibangun terdapat 2 *actor* yaitu user dan admin yang memiliki fungsi yang berbeda seperti pada *use case diagram* dibawah ini:



Gambar 4. 4 Usecase Diagram

4.1.2.2.1.1 Definisi Aktor

No.	Aktor	Keterangan
1.	<i>User</i>	<i>User</i> memiliki aktifitas untuk membuka aplikasi, membaca informasi dan mengedit informasi pribadi.
2.	Admin	Admin mempunyai aktifitas untuk mengelola aplikasi mulai dari tambah data, edit data, memlihat data keseluruhan hingga menghapus data

Tabel 4 2 Definisi Aktor

4.1.2.2.1.2 Skenario *Usecase*

Identifikasi	
Nama	Halaman Login User
Tujuan	Melakukan aktifitas login
Deskripsi	Melakukan login untuk melihat informasi
Aktor	User
Skenario	
Kondisi Awal	Membuka Aplikasi
Aksi Aktor	Reaksi Sistem
1. User masuk ke Aplikasi	2. Menampilkan Splash Screen
3. Menunggu Splash	4. Menampilkan halaman login user

Screen berganti halaman	
-------------------------	--

Tabel 4 3 Usecase Halaman Login User

Identifikasi	
Nama	Halaman Utama User
Tujuan	Melihat informasi
Deskripsi	Menampilkan informasi dan mengedit informasi yang diperlukan
Aktor	User
Skenario	
Kondisi Awal	Menampilkan halaman login user
Aksi Aktor	Reaksi Sistem
1. User login menggunakan username dan password yang terdaftar	2. Menampilkan halaman utama user
3. melihat dan mengedit data sesuai kebutuhan	4. mengupdate data yang telah disimpan

Tabel 4 4 Usecase Halaman Utama User

Identifikasi	
Nama	Register user

Tujuan	Mendaftarkan akun untuk login
Deskripsi	Register untuk login ke dalam aplikasi sesuai NIK dayng terdaftar
Aktor	User
Skenario	
Kondisi Awal	Menampilkan halaman login user
Aksi Aktor	Reaksi Sistem
2. User klik register pada halaman login	1. Menampilkan halaman login user
4. mengisi halaman register dan simpan	3. menampilkan halaman register

Tabel 4 5 Usecase Register User

Identifikasi	
Nama	Halaman Login admin
Tujuan	Melakukan aktifitas login admin
Deskripsi	login admin menggunakan username dan password yang terdaftar
Aktor	Admin
Skenario	
Kondisi Awal	Menampilkan halaman login user
Aksi Aktor	Reaksi Sistem

1. admin klik button login sebagai admin pada halaman login user	2. Menampilkan halaman login admin
3. admin login menggunakan username dan password yang terdaftar	4. menampilkan halaman utama admin

Tabel 4 6 Usecase Halaman Login admin

Identifikasi	
Nama	Halaman Utama Admin
Tujuan	Mengelola semua informasi
Deskripsi	Melakukan CRUD pada informasi
Aktor	Admin
Skenario	
Kondisi Awal	Menampilkan halaman login admin
Aksi Aktor	Reaksi Sistem
2. admin dapat membaca, membuat mengubah, menghapus data	1. Menampilkan list data informasi penduduk

berupa list data array	
3. melihat informasi yang telah terupdate	3. mengupdate data yang telah dikelola

Tabel 4 7 Usecase Halaman Utama Admin

Identifikasi	
Nama	Halaman Bantuan
Tujuan	Membantu user/admin
Deskripsi	Jika terdapat kesalahan dapat menghubungi admin yang tertera
Aktor	Admin dan User
Skenario	
Kondisi Awal	Menampilkan halaman utama user/admin
Aksi Aktor	Reaksi Sistem
1. aktor masuk ke menu utama	2. Menampilkan menu utama
3. aplikasi tidak berjalan dengan baik, klik 3 titik di atas pilih menu bantuan	3. Menampilkan halaman bantuan

Tabel 4 8 Usecase Halaman Bantuan

Identifikasi	
Nama	Dialog logout
Tujuan	Keluar dari aplikasi
Deskripsi	Logout setelah mengelola data baik admin maupun user
Aktor	Admin dan user
Skenario	
Kondisi Awal	Menampilkan halaman utama user/admin
Aksi Aktor	Reaksi Sistem
1. actor masuk ke halaman utama	2. Menampilkan halaman utama
3. klik 3 titik di bagian atas pilih menu keluar	4. Kembali ke halaman login user/admin

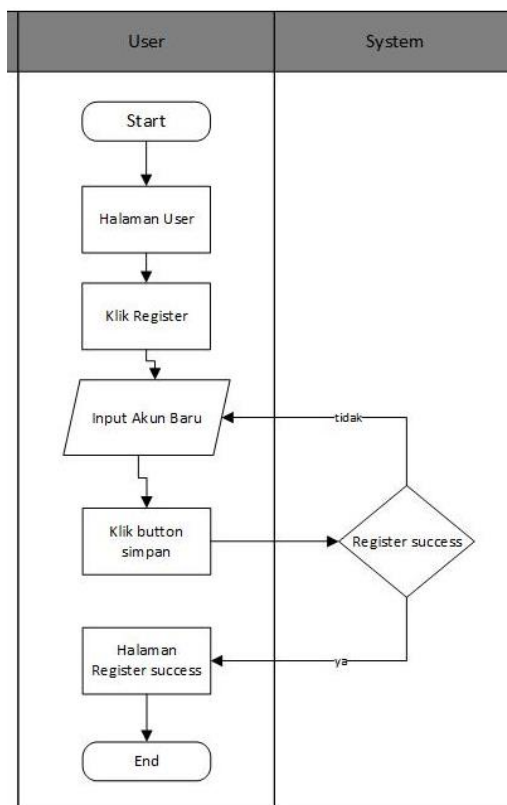
Tabel 4 9 Usecase Logout

4.1.2.2.2 Activity Diagram

Setelah Use Case Diagram terdapat Activity Diagram untuk menggambarkan sebuah aktivitas yang akan dilakukan pada sebuah aplikasi. Activity diagram merupakan suatu diagram yang bertujuan untuk menjelaskan secara prosedural alur proses dari sebuah sistem yang akan dibangun. Berikut ini adalah activity diagram dari Aplikasi Pemetaan.

4.1.2.2.1 *Activity Diagram Register*

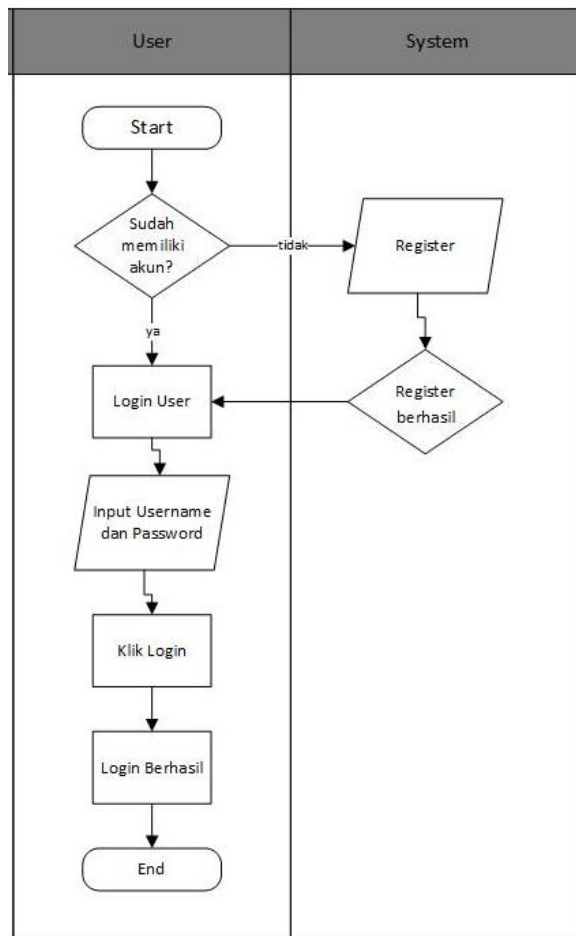
Pada sebuah aplikasi *android* biasanya mempunyai halaman menu untuk login dan register, pada aplikasi pemetaan disini juga membuat sebuah halaman register untuk penduduk dengan syarat NIK telah terdaftar pada database MySQL. Activity diagram ini menjelaskan tentang bagaimana alur untuk registrasi akun penduduk. Untuk alurnya seperti pada gambar 4.1.



Gambar 4. 5 Register

4.1.2.2.2.2 *Activity Diagram User Login*

Pada halaman login untuk user wajib memasukkan username dan password yang telah didaftarkan pada halaman register sebelumnya. Jika belum mempunyai akun untuk login maka akan diarahkan pada halaman register seperti alur dibawah.

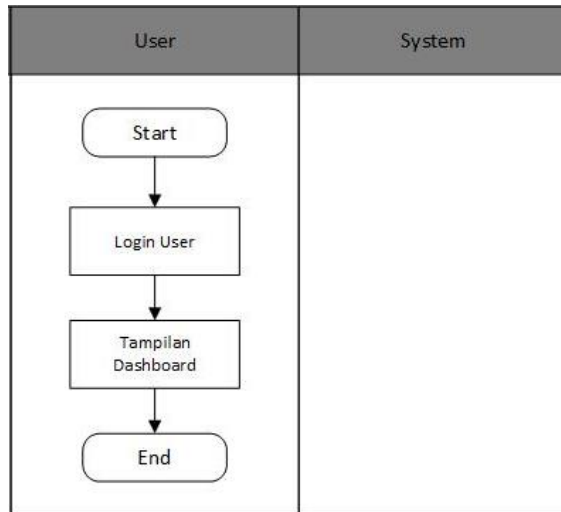


Gambar 4. 6 Login User

4.1.2.2.3 Activity Diagram User Read

Di halaman utama untuk user alurnya lebih sederhana daripada yang lainnya. Dapat dilihat bahwa user hanya dapat edit dan lihat data personal sesuai dengan akun login yang

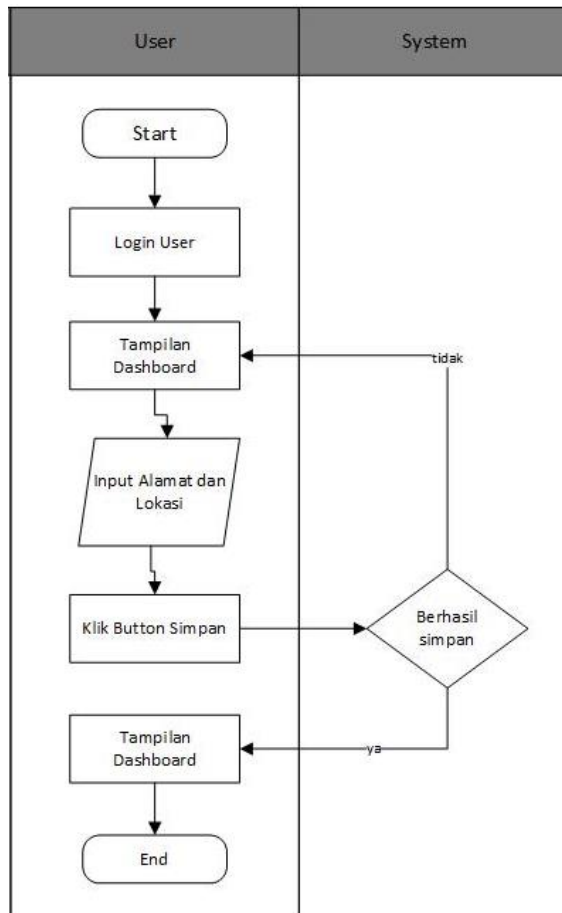
dipautkan dengan data penduduk.



Gambar 4. 7 Read User

4.1.2.2.2.4 Activity Diagram User Update

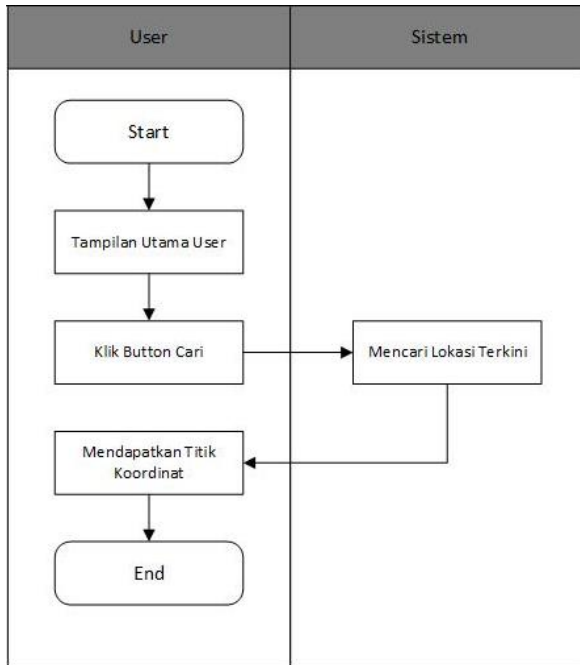
Berdasarkan pada gambar, dapat dilihat bahwa terdapat button untuk menyimpan data yang merupakan update data, user harus melakukan simpan data agar data yang diupdate bisa tersimpan dalam database MySQL.



Gambar 4. 8 Update User

4.1.2.2.2.5 *Activity Diagram User Cari*

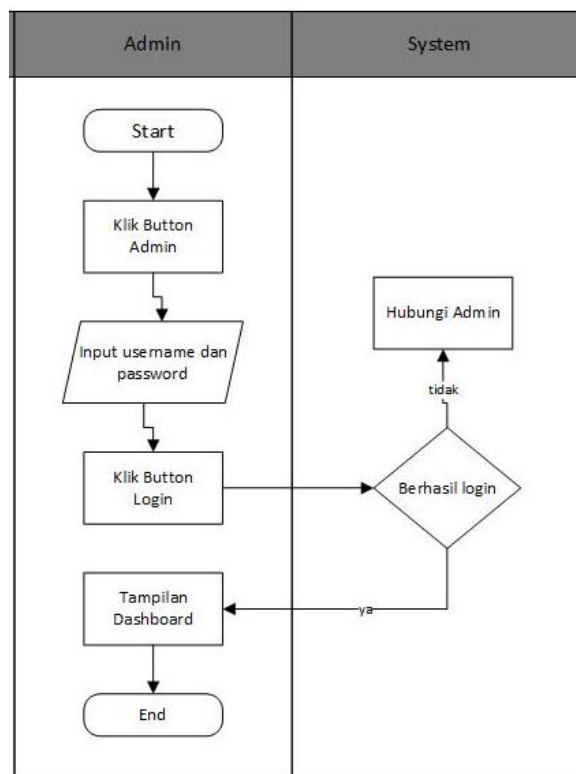
Pada activity diagram untuk user mencari data lokasi terkini yaitu seperti alur dibawah



Gambar 4. 9 User Cari

4.1.2.2.2.6 *Activity Diagram* Admin Login

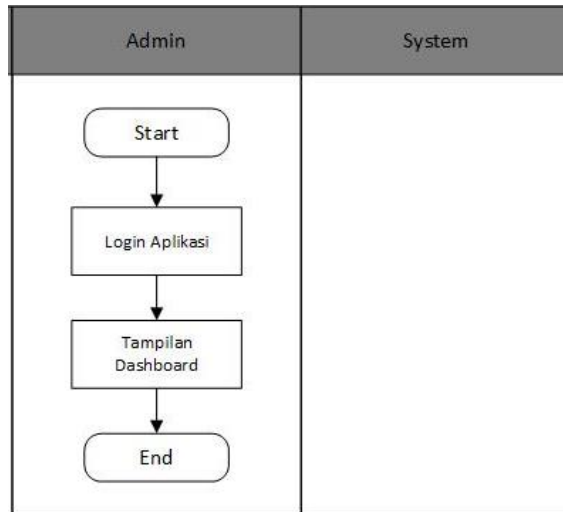
Login untuk admin hanya 1 akun saja, admin dapat melihat, mengedit, menambahkan, serta menghapus data penduduk. Jika admin tidak dapat login maka harus menghubungi administrator sebagai pembuat aplikasi, untuk menghubunginya terdapat *contact person* pada halaman bantuan.



Gambar 4. 10 Login Admin

4.1.2.2.2.7 *Activity Diagram Admin Read*

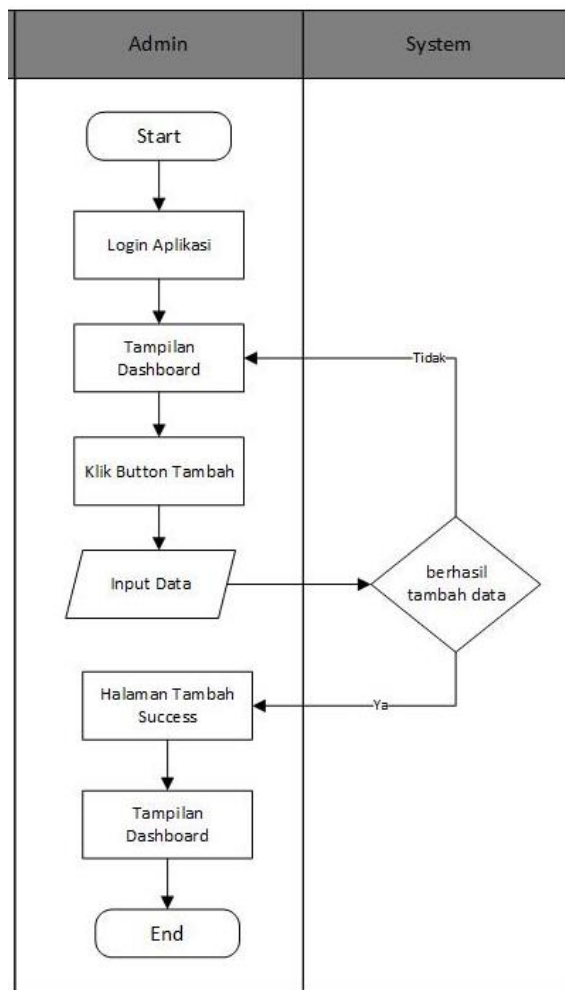
Halaman read data pada aplikasi ini selalu terlihat sederhana. Setelah login akan tampil sebuah halaman baru yang berisikan data-data penduduk yang hanya ditampilkan menurut nama.



Gambar 4. 11 Read Admin

4.1.2.2.2.8 **Activity Diagram Admin Tambah**

Admin tambah mempunyai alur 2 halaman. Setelah login sebagai admin maka akan diarahkan ke menu utama kemudian di sisi bawah terdapat sebuah button tambah yang akan berganti halaman apabila di klik. Selanjutnya setelah menginputkan data penduduk apabila berhasil akan muncul halaman baru yaitu data berhasil ditambahkan, jika tidak berhasil maka terdapat notifikasi berupa alasan tidak tersimpannya data tersebut dan diminta untuk mengisi ulang.

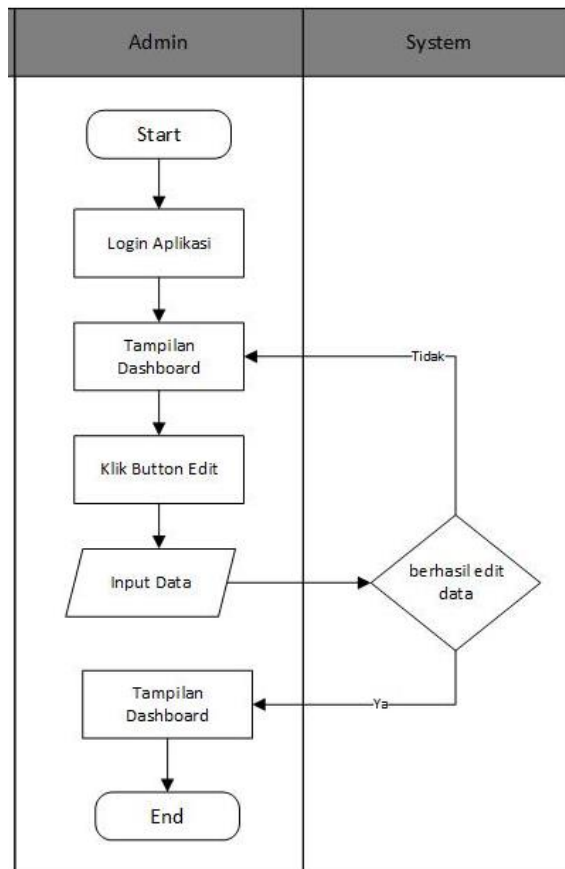


Gambar 4. 12 Tambah Admin

4.1.2.2.2.9 *Activity Diagram Admin Update*

Pada alur admin update, setelah tampilan utama admin pada bagian bawah terdapat button edit data apabila di klik

akan muncul dialog untuk memasukkan data yang akan di edit, apabila tidak berhasil maka pada tampilan utama data tidak akan terupdate.

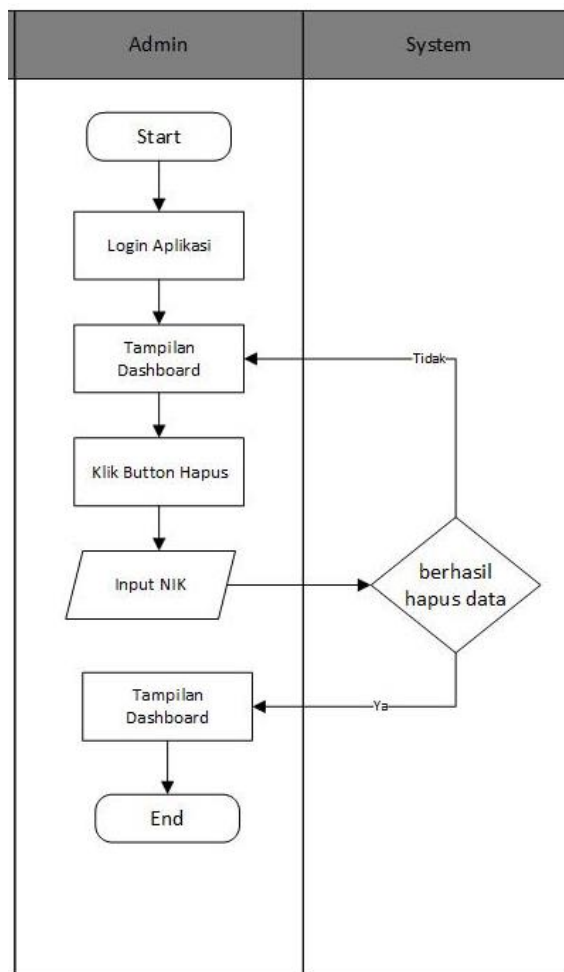


Gambar 4. 13 Update Admin

4.1.2.2.2.11 *Activity Diagram* Admin Hapus

Pada admin hapus hampir sama dengan update, jika admin

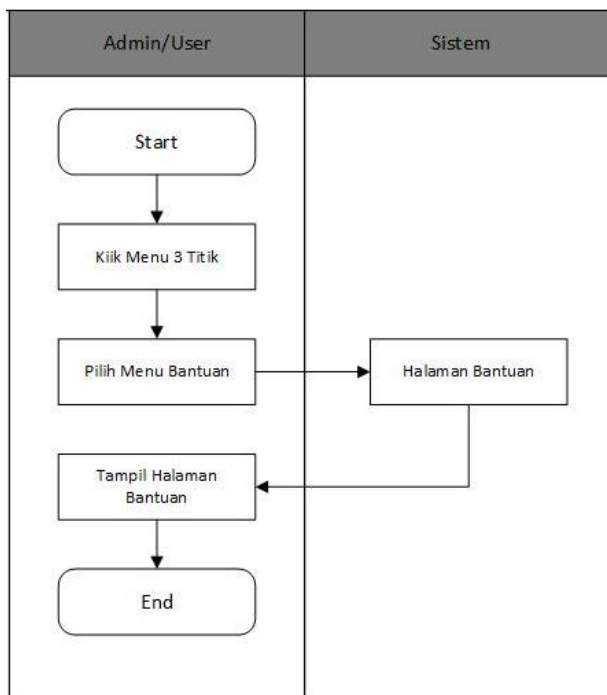
hapus hanya memasukkan NIK yang terdaftar, jika berhasil pada halaman utama akan terhapus data tersebut begitu pula sebaliknya.



Gambar 4. 14 Admin Hapus

4.1.2.2.2.11 *Activity Diagram Bantuan*

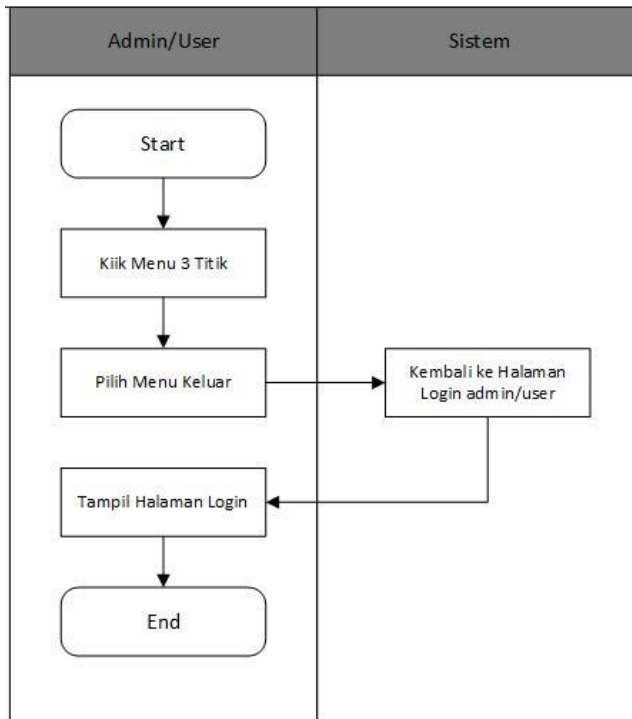
Pada Activity diagram ini menjelaskan bagaimana alur atau prosedur dari menu bantuan. Dimulai dari user membuka atau menjalankan aplikasi. Setelah user berada di menu utama, user dapat memilih menu tentang. Selanjutnya sistem akan mengarahkan ke menu tentang, dan user dapat melihat informasi dari tentang aplikasi.



Gambar 4. 15 Bantuan

4.1.2.2.2.12 *Activity Diagram Keluar*

Pada activity diagram ini, terdapat pilihan apakah user tetap ingin menjalankan aplikasi. Jika user ingin menjalankan aplikasi, Kembali ke menu utama. Jika user tidak ada perihai yang diperlukan lagi, user dapat melakukan exit dari aplikasi.



Gambar 4. 16 Keluar

Gambar 4.13 Use Case

4.2 Lingkungan Implementasi

Lingkungan implementasi adalah lingkungan wadah sistem

yang akan dibangun. Lingkungan implementasi terbagi menjadi dua, yaitu lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

4.2.1 Implementasi Perangkat Keras

Proses untuk membangun aplikasi ini menggunakan spesifikasi perangkat keras seperti berikut ini:

1. *Processor*: Intel Core I7
2. *Memory*: 16 GB RAM
3. *SSD*: 240 GB

4.2.2 Implementasi Perangkat Lunak

Proses untuk membangun aplikasi ini menggunakan spesifikasi perangkat lunak sebagai berikut ini:

1. Sistem Operasi: Windows 11
2. Bahasa Pemrograman: Java
3. Rest API: php framework CI
4. Software Pendukung: Android Studio
5. *Text Editor*: Visual Studio Code

4.3 Langkah Pembuatan

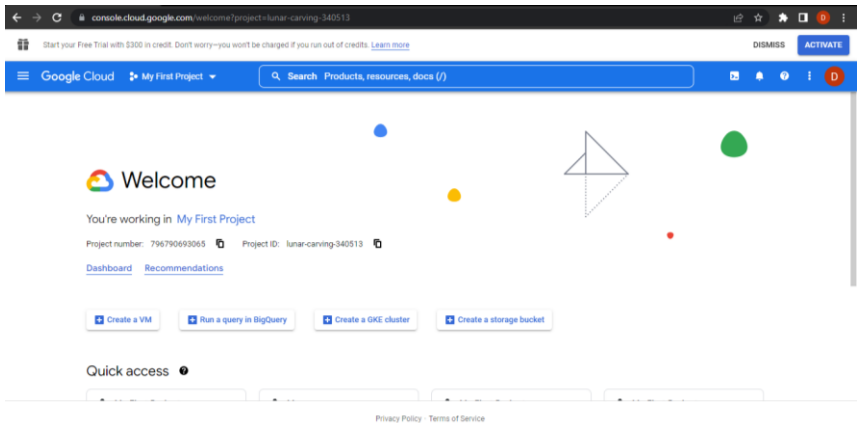
4.3.1 Persiapan Key Maps

Pada persiapan key maps disini ditujukan untuk

mengambil API Maps dari google maps yang dapat dilakukan dengan Langkah berikut:

1) Masuk ke website cloud google

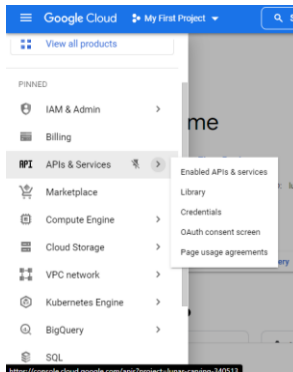
<https://console.cloud.google.com/welcome?project=lunar-carving-340513>



Gambar 4. 17 Cloud google

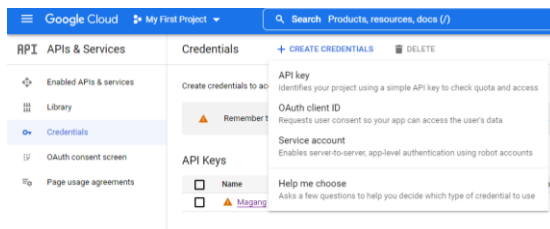
Buat sebuah proyek baru, saya membuat proyek, kali ini saya membuat nama dengan My First Project.

2) Selanjutnya buka navbar dan masuk ke dalam credentials.



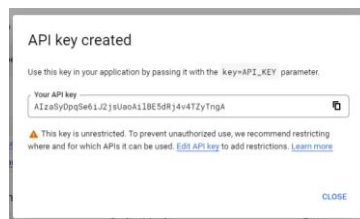
Gambar 4. 18 Navbar Cloud

- 3) Buat credentials pilih create credentials kemudian APIKey



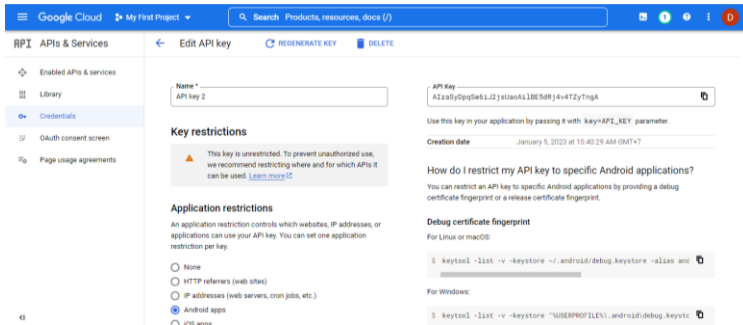
Gambar 4. 19 Create Credential

- 4) Jika berhasil akan tampil seperti ini



Gambar 4. 20 APIKey Create

- 5) Selanjutnya klik APIKey ubah Application restrictions menjadi Android apps save dan copy APIKey yang telah dibuat.



Gambar 4. 21 Edit APIKey

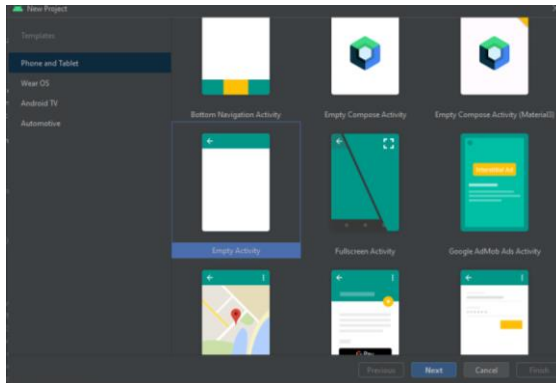
4.3.2 Persiapan Database

Buat sebuah database baru Bernama internship1, kemudian import sql dibawah ini.

https://drive.google.com/drive/u/0/folders/1ZGvkPSiTg66htRbfjM_iAAhu6QEY-rQI

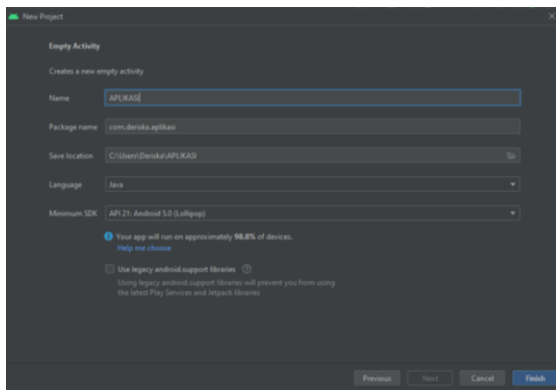
4.3.3 Memulai membuat Aplikasi

- 1) Selesai mendownload dan menginstall Android Studio <https://developer.android.com/studio> buat project baru, pilih aplikasi android untuk *Phone and Tablet* kemudian klik empty activity.



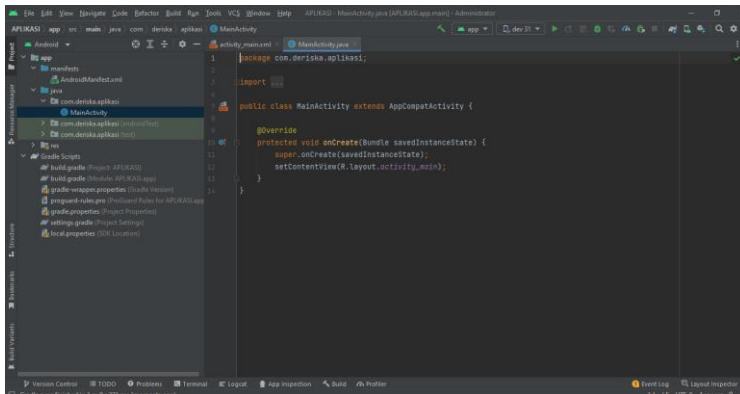
Gambar 4. 22 Permulaan 1

- 2) Kemudian berikan nama projectnya sesuai yang diinginkan, disini project diberikan nama “aplikasi” dengan bahas java dan SDK minimal 21 lalu finish.



Gambar 4. 23 Permulaan 2

- 3) Jika sudah tampil seperti ini maka berhasil



Gambar 4. 24 Permulaan 3

4.3.4 Mempersiapkan Gradle dan manifest

- 1) Buka menu gradle pada aplikasi yang sudah dibuat, kemudian tambahkan beberapa sourcode pada dependency. Gradle berfungsi untuk menginstall sebuah library pada java.

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.5.1'
    //noinspection GradleCompatible
    implementation 'com.android.support.design:28.0.0'
    implementation
    'com.google.android.material:material:1.7.0'
    implementation
    'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.android.gms:play-services-
maps:18.1.0'
    implementation 'androidx.navigation:navigation-
fragment:2.5.3'
    implementation 'androidx.navigation:navigation-ui:2.5.3'
    implementation 'com.google.firebase:firebase-
database:20.1.0'
    implementation 'com.squareup.picasso:picasso:2.71828'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation
    'androidx.test.espresso:espresso-core:3.5.0'
```

```

        implementation 'com.google.android.gms:play-services-
location:21.0.1'
        //noinspection GradleCompatible
        implementation 'com.android.support:cardview-v7:28.0.0'
        //noinspection GradleCompatible
        implementation 'com.android.support:recyclerview-
v7:28.0.0'
        implementation 'androidx.recyclerview:recyclerview:1.2.1'
        implementation
'com.google.android.material:material:1.7.0'
        implementation 'com.android.volley:volley:1.2.1'
    }

```

2) Kemudian lanjut sync now untuk menginstall library.

3) Selanjutnya mengkonfigurasi manifest

Masukkan source code dibawah setelah tag manifest ditutup untuk menyambungkan ke dalam internet sebagai volley.

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />

```

Source code dibawah merupakan konfigurasi untuk mengambil API dari maps yang telah dibuat sebelumnya, letakkan sebelum intent filter.

```

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="masukkan APIKEY"/>

```

4.3.5 Membuat Activity

Dalam membuat sebuah aplikasi memerlukan layout activity untuk tampilan UI, java activity sebagai fungsi serta

manifest sebagai penghubung.

4.3.4.1 *Splash Screen Activity*

Splash screen umumnya sederhana biasanya hanya menampilkan satu logo dengan background simple untuk mengidentifikasi bahwa itu adalah aplikasi tersebut.

Activity_splash_screen.xml

Pada splashscreen xml hanya menambahkan sebuah logo khusus untuk aplikasi itu sendiri dengan background polos.

```
<ImageView
    android:id="@+id/logo_utama"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:src="@drawable/logo_apk"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

SplashScreenActivity.java

Pada splashscreen java merupakan aksi dari splashscreen xml yang mempunyai timer tertentu untuk loading aplikasi kemudian akan mengarah ke sebuah halaman baru yaitu login user.

```
package com.deriska.apkmagang;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;

public class SplashScreenActivity extends AppCompatActivity {
    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_splash_screen);

            View decorView = getWindow().getDecorView();

            decorView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_FULLSCREEN);

            if (getSupportActionBar() != null) {
                getSupportActionBar().hide();
            }
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    Intent intent = new
                    Intent(SplashScreenActivity.this, LoginActivity.class);
                    startActivity(intent);
                    finish();
                }
            }, 3000);
        }
    }
}

```

4.3.4.2 *VolleySingleton*

Pada class volleysingleton disini berfungsi untuk request volley agar aplikasi dapat terhubung dengan internet dan dengan database local.

VolleySingleton.java

```

package com.deriska.apkmagang;
import android.content.Context;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class VolleySingleton {
    private static VolleySingleton vInstance;
    private RequestQueue requestQueue;
    private static Context vCtx;

    private VolleySingleton(Context context) {
        vCtx = context;
        requestQueue = getRequestQueue();
    }
}

```

```

        private RequestQueue getRequestQueue() {
            if (requestQueue == null) {
                requestQueue =
                Volley.newRequestQueue(vCtx.getApplicationContext());
            }
            return requestQueue;
        }

        public static synchronized VolleySingleton
        getInstance(Context context) {
            if (vInstance == null) {
                vInstance = new VolleySingleton(context);
            }

            return vInstance;
        }

        public<T> void addToRequestQue (Request<T> request) {
            getRequestQueue().add(request);
        }
    }

```

4.3.4.3 *Login Activity*

Login activity ditujukan untuk user atau pengguna, login ini dapat diakses setelah splashscreen selesai terbuka. Untuk source code bisa dilihat pada link dibawah ini.

<https://drive.google.com/drive/u/0/folders/19VZFpl1kjo5wAeKuuBOnUxMHMkog1M1J>

Pada login activity menggunakan API yang dibuat dalam bahasa php untuk koneksi ke dalam database.

Api.php(controller)

```

public function login()
{
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (isset($_POST['user']) && isset($_POST['pass']))
    {

```

```

        $user_login          =          $this->M_api-
>proses_login($_POST['user'], $_POST['pass']);
        $result['nik']      = null;

        if ($user_login->num rows() == 1) {
            $result['value'] = "1";
            $result['pesan'] = "sukses login!";
            $result['nik']   = $user_login->row()->nik;
        } else {
            $result['value'] = "0";
            $result['pesan'] = "username / password
salah!";
        }
    } else {
        $result['value'] = "0";
        $result['pesan'] = "beberapa inputan masih
kosong!";
    }
} else {
    $result['value'] = "0";
    $result['pesan'] = "invalid request method!";
}

echo json_encode($result);
}

```

M_Api.php

```

public function proses_login($user, $pass)
{
    return $this->db->query("SELECT nik FROM penduduk
WHERE username = '$user' AND password = MD5('$pass')");
}

```

4.3.4.4 *Main Activity*

Main Activity disini merupakan sebuah tampilan utama dari user. Source code pada activity ini terlalu Panjang maka silahkan dicek ke halaman

<https://drive.google.com/drive/folders/1y5NHxtepYQjtMf7e10Sba9C4xkw63doX?usp=sharing>

Untuk API pada main activity sebagai berikut

M_Api.php

```
public function cek_nik_register($nik)
{
    return $this->db->query("SELECT nik FROM penduduk
WHERE nik = '$nik'");
}

public function get_profile($nik)
{
    return $this->db->query("SELECT * FROM penduduk,
agama WHERE penduduk.id_agama = agama.id_agama AND nik =
'$nik'")->row();
}
```

Api.php

```
public function profile()
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        $result['hasil'] = null;

        if ($this->M_api->cek_nik_register($_POST['nik'])-
>num_rows() != 0) {

            $result['value'] = "1";
            $result['pesan'] = "response ok!";
            $result['hasil'] = [

                'nik' => $this->M_api-
>get_profile($_POST['nik'])->nik,
                'nama' => $this->M_api-
>get_profile($_POST['nik'])->nama,
                'ttl' => $this->M_api-
>get_profile($_POST['nik'])->tempat_lahir . ", " . $this->M_api-
>get_profile($_POST['nik'])->tanggal_lahir,
                'jk' => $this->M_api-
>get_profile($_POST['nik'])->jk == "L" ? "Laki-laki" :
"Perempuan",
                'agama' => $this->M_api-
>get_profile($_POST['nik'])->nama_agama
            ];
        }
    }
```

```

    } else {
        $result['value'] = "0";
        $result['pesan'] = "invalid request method!";
    }

    echo json_encode($result);
}

```

4.3.4.5 *ShowMenu Activity*

Show menu disini berada pada halaman utama admin dan user yang terdapat pada 3 titik di bagian atas yang akan menuju kearah bantuan dan keluar. Pada show menu hanya menambahkan xml saja untuk nantinya dipanggil ke dalam MainActivity.java dan AdminPenduduk.java

Menu_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:title="Bantuan"
        android:id="@+id/menu_item_bantuan"
        app:showAsAction="never"/>
    <item
        android:title="Logout"
        android:id="@+id/menu_item_logout"
        app:showAsAction="never"/>
</menu>

```

4.3.4.6 *Bantuan Activity*

Activity bantuan dapat diakses dan akan muncul pada halaman utama admin/user dan halaman login.

activity_bantuan.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout

```



```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@drawable/btn_circle_gradient"
tools:context=".BantuanActivity">

```

```

<LinearLayout
    android:layout_marginTop="80dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:src="@drawable/logo_apk"
        android:layout_width="match_parent"
        android:layout_height="150dp"/>
    <TextView
        android:layout_marginTop="20dp"
        android:textSize="14sp"
        android:fontFamily="@font/ss_regular"
        android:textColor="@color/colorWhite"
        android:paddingHorizontal="75dp"
        android:gravity="center"
        android:text="Jika Terdapat Kendala dalam
Penggunaan Aplikasi Pemetaan Lokasi Penduduk ini, dapat
Dilihat Video Berikut, Apabila Kurang Jelas Mohon Hubungi
Admin Dibawah ini"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <TextView
        android:layout_marginTop="20dp"
        android:textSize="14sp"
        android:fontFamily="@font/ss_semibold"
        android:textColor="@color/colorWhite"
        android:gravity="center"
        android:text="ADMIN 1 : "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <TextView
        android:textSize="14sp"
        android:fontFamily="@font/ss_regular"
        android:textColor="@color/colorWhite"
        android:gravity="center"
        android:text="deriskafadillam18@gmail.com"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <TextView
        android:layout_marginTop="20dp"
        android:textSize="14sp"
        android:fontFamily="@font/ss_semibold"
        android:textColor="@color/colorWhite"

```

```

        android:gravity="center"
        android:text="ADMIN 2 : "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
<TextView
    android:textSize="14sp"
    android:fontFamily="@font/ss_regular"
    android:textColor="@color/colorWhite"
    android:gravity="center"
    android:text="085723393512 (TLP / WA)"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</LinearLayout>
<TextView
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="16dp"
    android:textSize="14sp"
    android:fontFamily="@font/ss_semibold"
    android:textColor="@color/colorWhite"
    android:gravity="center"
    android:text="Terima Kasih"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</RelativeLayout>

```

BantuanActivity.java

```

package com.deriska.apkmagang;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class BantuanActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bantuan);
        getSupportActionBar().setElevation(0);
    }
}

```

4.3.4.6 *Login Admin Activity*

Login admin activity disini merupakan halaman untuk admin login agar dapat mengelola data.

<https://drive.google.com/drive/u/0/folders/1uH7Cqysit4utJ43sCt271Tu1PotfTj8u>

untuk login admin juga memerlukan API sebagai berikut

M_Api.php

```
public function proses_login_admin($username, $password)
{
    return $this->db->query("SELECT id_admin FROM admin
WHERE username = '$username' AND password =
MD5('$password')");
}
```

Api.php

```
public function login_admin()
{
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (isset($_POST['user']) && isset($_POST['pass']))
        {
            $user_login          =          $this->M_api-
>proses_login_admin($_POST['user'], $_POST['pass']);
            $result['id_admin']   = null;

            if ($user_login->num_rows() == 1) {
                $result['value'] = "1";
                $result['pesan'] = "sukses login!";
                $result['id_admin'] = $user_login->row()-
>id_admin;
            } else {
                $result['value'] = "0";
                $result['pesan'] = "username / password
salah!";
            }
        } else {
            $result['value'] = "0";
            $result['pesan'] = "beberapa inputan masih
kosong!";
        }
    } else {
        $result['value'] = "0";
        $result['pesan'] = "invalid request method!";
    }

    echo json_encode($result);
}
```

4.3.4.7 *Admin Penduduk Activity*

Halaman ini merupakan halaman utama admin, pada halaman ini admin bisa mengakses data penduduk mulai dari

create, read, update dan delete.

<https://drive.google.com/drive/u/0/folders/1WYWg-xCaxsZdhqfT1RQZwr0Zlj8PxK4B>

untuk API nya pada activity ini adalah

M_Api.php

```
public function get_users()
{
    return $this->db->query("SELECT    nik,    nama    FROM
penduduk");
}
```

Api.php

```
public function get_users()
{
    if ($_SERVER['REQUEST_METHOD'] == 'GET') {
        $result['value'] = "1";
        $result['pesan'] = "response ok!";
        $result['hasil'] = $this->M_api->get_users()-
>result_array();

    } else {
        $result['value'] = "0";
        $result['pesan'] = "invalid request method!";
    }

    echo json_encode($result);
}
```

4.3.4.8 *RecyclerView*

Pada recyclerview disini berfungsi untuk menampilkan semua data penduduk pada halaman admin penduduk yang diperoleh dari database dalam bentuk list data dan akan merefresh data apabila terjadi perubahan. Pada recyclerview

membutuhkan beberapa java class untuk menyambungkan pada aplikasi dan tidak memerlukan layout karena bukan tampilan.

Data.java

Pertama membuat class data terlebih dahulu untuk membuat fungsi yang akan dipanggil pada recyclerview.

```
package com.deriska.apkmagang;

public class Data {
    private String nik, nama;
    Data(String nik, String nama) {
        this.setNik(nik);
        this.setNama(nama);
    }

    public String getNik() {
        return nik;
    }

    public void setNik(String id) {
        this.nik = nik;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }
}
```

Recyclerview.java

```
package com.deriska.apkmagang;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
```

```

public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerView.MyViewHolder> {

    private ArrayList<Data> arrayList = new ArrayList<>();

    public RecyclerViewAdapter(ArrayList<Data> arrayList) {
        this.arrayList = arrayList;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent,
int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.list
_data, parent, false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, int
position) {
        holder.nik.setText(arrayList.get(position).getNik());
holder.nama.setText(arrayList.get(position).getNama());
    }

    @Override
    public int getItemCount() {
        return arrayList.size();
    }

    public static class MyViewHolder extends
RecyclerView.ViewHolder {

        TextView nik, nama;

        public MyViewHolder(View itemView) {
            super(itemView);
            nik = (TextView)
itemView.findViewById(R.id.txt_id);
            nama = (TextView)
itemView.findViewById(R.id.txt_nama);
        }
    }
}

```

4.3.4.9 *Tambah Data*

Pada create admin disini untuk membuat data baru

yaitu data penduduk yang hanya dapat dilakukan oleh admin. Dalam tambah data akan menampilkan sebuah halaman baru jika berhasil akan berganti halaman tambah sukses.

https://drive.google.com/drive/u/0/folders/1G2h_jTl3bGndiOp7qbGxg-8BX98Kde1L

M_Api.php

```
public function addPenduduk($nik, $nama, $ttl, $tanggal)
{
    // $tanggal = date('d-m-Y');
    $this->db->query("INSERT INTO penduduk (`nik`, `nama`,
`tempat_lahir`, `tanggal_lahir`)
                                VALUES ('$nik', '$nama', '$ttl',
$tanggal)");
}
```

Api.php

```
public function add_penduduk()
{
    if (isset($_POST['nik'])) {
        $this->M_api->addPenduduk($_POST['nik'],
$_POST['nama'], $_POST['tempat_lahir'],
$_POST['tanggal_lahir']);
        $result['value'] = "1";
        $result['pesan'] = "data penduduk berhasil
disimpan!";
    } else {
        $result['value'] = "0";
        $result['pesan'] = "beberapa inputan masih
kosong!";
    }

    echo json_encode($result);
}
```

4.3.4.10 *Ubah Data*

Pada ubah data atau update data pada admin disini

merupakan fungsi untuk mengubah data penduduk pada halaman admin penduduk dan hanya dapat diakses oleh admin. Jika data berhasil diupdate maka akan otomatis refresh pada halaman admin penduduk.

Update_data_dialog.xml

Pada layout xml ubah data hanya memakai dialog, berbeda dengan tambah data yang berganti halaman.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <EditText
        android:id="@+id/edt_updateId"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="0dp"
        android:hint="Id" />

    <EditText
        android:id="@+id/edt_updateNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edt_updateId"
        android:layout_marginTop="0dp"
        android:hint="Nama" />

    <EditText
        android:id="@+id/edt_updateAlamat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edt_updateNama"
        android:layout_marginTop="0dp"
        android:hint="Alamat" />

</RelativeLayout>
```

UpdateDataDialog.java

```
package com.deriska.apkmagang;
import android.app.AlertDialog;
```



```

import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class UpdateDataDialog extends AppCompatActivity {

    private EditText updateNik, updateNama, updateAlamat;
    private UpdateDataDialogListener listener;

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {

        AlertDialog.Builder builder = new
        AlertDialog.Builder(getActivity());

        LayoutInflater inflater =
        getActivity().getLayoutInflater();
        View view =
        inflater.inflate(R.layout.update_data_dialog, null);

        builder.setView(view)
            .setTitle("Update Data")
            .setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
        dialog, int which) {

                }

            })
            .setPositiveButton("Submit", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
        dialog, int which) {

                String nik =
        updateNik.getText().toString();
                String nama =
        updateNama.getText().toString();
                String alamat =
        updateAlamat.getText().toString();
                listener.update(nik, nama, alamat);

            }

        });

        updateNik = view.findViewById(R.id.edt_updateId);
        updateNama = view.findViewById(R.id.edt_updateNama);
    }
}

```

```

        updateAlamat =
view.findViewById(R.id.edt_updateAlamat);

        return builder.create();
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);

        try {
            listener = (UpdateDataDialogListener) context;
        } catch (ClassCastException e) {

        }
    }

    public interface UpdateDataDialogListener {
        void update(String nik, String nama, String alamat);
    }
}

```

M_Api.php

```

public function ubahPenduduk($nik, $nama, $alamat)
{
    $this->db->query("UPDATE penduduk SET nama = '$nama',
alamat = '$alamat'
                    WHERE nik = '$nik'");
}

```

Api.php

```

public function ubah_penduduk()
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        if ($this->M_api->cek_nik_register($_POST['nik'])-
>num_rows() != 0) {
            if (isset($_POST['nama'])) {
                $this->M_api->ubahPenduduk($_POST['nik'],
$ POST['nama'], $ POST['alamat']);
                $result['value'] = "1";
                $result['pesan'] = "Data Penduduk berhasil
diubah!";
            } else {
                $result['value'] = "0";
                $result['pesan'] = "beberapa inputan masih
kosong!";
            }
        }
    }
}

```

```

        } else {
            $result['value'] = "0";
            $result['pesan'] = "NIK Penduduk tidak
tersedia!";
        }
    } else {
        $result['value'] = "0";
        $result['pesan'] = "invalid request method!";
    }

    echo json_encode($result);
}

```

4.3.4.6 *Hapus Data*

Pada bagian delete data disini merupakan bagian tersimple daripada dialog yang lainnya. Delete data dialog disini berfungsi untuk menghapus data penduduk pada database yang tersambung dan akan direfresh di halaman admin penduduk.

Delete_data_dialog.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <EditText
        android:id="@+id/edt_deleteNik"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="1dp"
        android:hint="nik" />

</RelativeLayout>

```

DeleteDataDialog.java

```

package com.deriska.apkmagang;
import android.app.AlertDialog;
import android.app.Dialog;

```

```

import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class DeleteDataDialog extends AppCompatActivity
{
    private EditText deleteNik;
    private DeleteDataDialogListener listener;

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new
        AlertDialog.Builder(getActivity());
        LayoutInflater inflater =
        getActivity().getLayoutInflater();
        View view =
        inflater.inflate(R.layout.delete_data_dialog, null);

        builder.setView(view)
            .setTitle("Delete Data")
            .setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
        dialog, int which) {
            }
        })
            .setPositiveButton("Submit", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
        dialog, int which) {
                String id =
        deleteNik.getText().toString();
                listener.delete(id);
            }
        });

        deleteNik = view.findViewById(R.id.edt_deleteNik);
        return builder.create();
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        try {
            listener = (DeleteDataDialogListener) context;
        } catch (ClassCastException e) {

```

```

    }
}
public interface DeleteDataDialogListener {
    void delete(String Nik);
}
}

```

M_Api.php

```

public function hapusPenduduk($nik)
{
    $this->db->query("DELETE FROM penduduk WHERE nik = '$nik'");
}

```

Api.php

```

public function hapus_penduduk()
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        if ($this->M_api->cek_nik_register($_POST['nik'])->num_rows() != 0) {
            $this->M_api->hapusPenduduk($_POST['nik']);
            $result['value'] = "1";
            $result['pesan'] = "Data Penduduk berhasil dihapus!";
        } else {
            $result['value'] = "0";
            $result['pesan'] = "NIK Penduduk tidak tersedia!";
        }
    } else {
        $result['value'] = "0";
        $result['pesan'] = "invalid request method!";
    }

    echo json_encode($result);
}

```

4.3.6 Hasil Uji Coba

Berikut adalah hasil uji coba dari scenario

Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria Evaluasi	Hasil yang didapatkan	Kesimpulan
Splashscreen	System memuat splashscreen Activity	Membuka aplikasi	Diarahkan ke halaman splashscreen	Halaman splashscreen	Menampilkan splashscreen	Sesuai
Login User	System memuat halaman login activity	-	Diarahkan ke halaman login	Halaman login	Menampilkan halaman login	Sesuai
Register	System memuat halaman register activity	Memilih menu register	Diarahkan ke halaman register	Halaman register	Menampilkan halaman register	Sesuai
	System memuat halaman register success activity	Simpan registrasi	Diarahkan ke halaman register success	Halaman register success	Menampilkan halaman register success	Sesuai
Halaman Utama user	System memuat halaman main	Login user	Diarahkan ke halaman main	Halaman main	Menampilkan halaman main	Sesuai

	activity					
	System menyimpan perubahan data	Simpan data	Data tersimpan pada database	Halaman main	Data tersimpan pada database	Sesuai
Halaman bantuan	System memuat halaman bantuan	Memilih menu bantuan	Diarahkan ke halaman bantuan	Halaman bantuan	Menampilkan halaman bantuan	Sesuai
Halaman keluar	System memuat halaman login	Memilih menu keluar	Diarahkan ke halaman login	Keluar	Menampilkan halaman login	Sesuai
Login admin	System memuat halaman login admin activity	Memilih menu login admin	Diarahkan ke halaman admin	Halaman login admin	Menampilkan halaman login admin	Sesuai
Halaman utama admin	System memuat admin penduduk activity	Login admin	Diarahkan ke halaman admin penduduk	Halaman admin penduduk	Menampilkan halaman penduduk	Sesuai
	System merefresh data	Mengelola CRUD	Diarahkan ke halaman admin penduduk	Tambah, Update, Delete	Berhasil mengelola data	Sesuai

			k			
--	--	--	---	--	--	--

Tabel 4 10 hasil uji coba

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Permasalahan yang dihadapi oleh petugas survei lapangan dalam pencatatan data yang sulit untuk menemukan titik koordinat penduduk untuk survei sehingga perlunya system untuk melacak lokasi penduduk berbasis android. Sistem aplikasi dirancang sedemikian rupa sehingga pengguna dapat mengikuti alur secara runtut dan efisien. Dengan adanya aplikasi ini diharapkan memudahkan petugas lapangan survei untuk mencatat data penduduk.

5.2 Saran

Saran yang dapat diberikan oleh penulis adalah agar kedepannya sistem aplikasi pemetaan ini dapat berevolusi mengikuti kemajuan teknologi yang ada.

DAFTAR PUSTAKA

- A, A. (2018). Sistem Pakar Diagnosa Penyakit Ginjal Berbasis Android. In *Kajian Teori dan Hasil Penelitian Pendidikan* (pp. 43-67). IJIEEM.
- Alma, B. (2003). *Manajemen Pemasaran dan Pemasaran Jasa Edisi 2*. Bandung: Alfabeta.
- Awangga, R. (2019). *Pengantar Sistem Informasi Geografis*. Bandung: Kreatif Industri Nusantara.
- Elian, S., & H, S. (2012). *Layanan Informasi Kereta Api Menggunakan GPS, Google Maps dan Android*. Surabaya: Fakultas Teknologi Informasi Institut Teknologi Sepuluh November.
- Firmansyah, M. A. (Desember 2018). *Perilaku konsumen: (sikap dan pemasaran)*. Yogyakarta: Deepublish.
- Irwansyah, E. (2013). *Sistem Informasi Geografis*. Yogyakarta: Digibooks.
- Kadir, A. (2008). *Tuntunan Praktis Belajar Database Menggunakan MySQL*. 2008: C.V. Andi Offset.
- Kasmir. (2009). *Pengantar Manajemen Keuangan*. Jakarta: Kencana.
- Kustiyaningsih, Y. (2011). *Pemrograman Basis Data berbasis Web Menggunakan PHP dan MySQL*. Yogyakarta: Graha Ilmu.
- Mantra. (2009). *Demografi Umum*. Yogyakarta: Pusat Pelajar Offset.
- Mardison. (Agustus 2017). Perancangan Sistem Penunjang Keputusan untuk mengoptimalkan pemberian kredit pada bank BPR Kubang dengan Bahasa Pemrograman Java dan Didukung dengan Database MySQL. *Jurnal Processor*.
- Murya, Y. (2014). *Pemrograman Android Black Box*. Jakarta: Jasakom.
- N, S. H. (2014). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika.

- Pressman, R. (2010). *Rekayasa Perangkat Lunak . Pendekatan Praktisi*.
- Raharjo, B. (2011). *Belajar Otodidak Membuat Database Menggunakan MySQL*. Bandung: Informatika.
- Shalahuddin, M., & R. A. (2010). *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Penerbit Modula.
- Sibero, A. F. (2013). *Web Programing Power Pack*. Yogyakarta: mediaKom.
- Suwarman, U. (2004). *Perilaku Konsumen*. Jakarta: Ghalia Indonesia.
- Syam'ani. (2016). *Membangun Basisdata Spasial Menggunakan ArcGIS 10.3*.
- Yakub. (2011). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.

-oo00oo-

Pada data kependudukan terdapat data perseorangan yang meliputi NIK, nama lengkap, jenis kelamin, alamat dan sebagainya sebagai hasil kegiatan pendaftaran penduduk dan pencatatan sipil. Dalam menjalankan proses survei di lapangan perlu adanya proses pencatatan data, proses pencatatan data tersebut memerlukan survei lokasi penduduk. Pada data alamat penduduk petugas lapangan survei sulit memvalidasi kesesuaian lokasi hal tersebut yang menjadi masalah bagi petugas lapangan.

Dibutuhkan sistem yang mampu menampilkan lokasi penduduk dalam Web Geographic Information System (WebGIS). Penelitian ini menjelaskan tentang bagaimana cara sistem melacak lokasi penduduk menggunakan GPS Android, kemudian merepresentasikannya ke dalam WebGIS. Metode pelacakan dimulai Ketika penduduk mengisi form input data melalui aplikasi Android. Koordinat lokasi berupa latitude dan longitude akan terdeteksi secara otomatis. Kemudian seluruh data akan disimpan pada database MySQL

Tujuan dari penelitian ini yaitu membuat sistem pelacakan lokasi penduduk untuk memudahkan petugas survei melakukan pencarian penduduk. Metode yang digunakan dalam pengembangan sistem aplikasi adalah metode Agile Development.

