

***TRACKING POSISI OBJEK DENGAN MENGGUNAKAN
METODE REVERSE GEOCODING***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat Kelulusan Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh:

M. Irfan Bayu P

NIM: 1051915

Fifi Andini

NIM: 1051920

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2023**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

*TRACKING POSISI OBJEK DENGAN MENGGUNAKAN
METODE REVERSE GEOCODING*

Oleh:

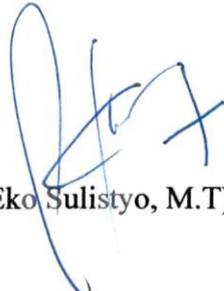
M. IRFAN BAYU P/NIM 1051915

FIFI ANDINI/NIM 1051920

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



(Eko Sulistyono, M.T)

Pembimbing 2



(Indra Dwisaputra, M.T.)

Penguji 1



(Irwan, M.Sc, Ph.D)

Penguji 2



(Yudhi, M.T)

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : M. Irfan Bayu P NIM: 1051915

Nama Mahasiswa 2 : Fifi Andini NIM: 1051920

Dengan Judul : *Tracking* Posisi Objek dengan menggunakan Metode
Reverse Geocoding

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 6 Februari 2023

Nama Mahasiswa

Tanda Tangan

1. M. Irfan Bayu P

.....

2. Fifi Andini

.....

ABSTRAK

Sistem tracking position atau pelacakan posisi adalah teknologi untuk menentukan posisi suatu objek, namun saat ini hanya digunakan untuk menentukan lokasi yang sudah diketahui dan belum dikembangkan untuk melacak objek. Oleh karena itu, diperlukan penelitian untuk pengembangan pencarian posisi objek yang memiliki informasi tentang lokasi dan alamat objek tersebut. Metodologi pada proyek akhir ini adalah dengan menerima sinyal dari modul GPS pada alat tracking yang dipancarkan satelit kemudian diproses oleh mikrokontroler menjadi koordinat dan dikirimkan ke database Google Firebase. Google Firebase mengirimkan koordinat ke aplikasi tracking objek. Penggunaan metode reverse geocoding adalah untuk mengubah koordinat tersebut menjadi sebuah lokasi dan alamat. Pengujian dilakukan dengan meletakkan alat tracking di beberapa lokasi yang berbeda dengan membandingkan aplikasi yang menggunakan metode reverse geocoding dan aplikasi tanpa metode reverse geocoding. Yang kedua, pengujian jarak terhadap alat tracking dengan membandingkan pengukurannya antara aplikasi dan google maps. Yang ketiga, pengiriman lokasi melalui pesan teks. Hasil pengujian aplikasi yang menggunakan metode reverse geocoding dapat menampilkan alamat dari alat tracking dengan persentase keberhasilan 100%. Sedangkan aplikasi tanpa metode reverse geocoding hanya menampilkan lokasi alat tracking saja tanpa menampilkan alamatnya. Alat tracking lebih cepat mendapatkan sinyal apabila dilakukan di luar ruangan dibandingkan dengan dalam gedung dimana alat tracking tidak mendapatkan koordinat dari satelit. Hasil pengukuran jarak yang dilakukan alat tracking posisi memiliki keakurasian 99% dengan validasi pengukuran jarak oleh google maps. Link google maps berhasil dikirim melalui sms dengan persentase keberhasilan 100%.

Kata Kunci : Tracking Posisi, GPS, Latitude, Longitude, Reverse Geocoding

ABSTRACT

An object's position may be determined using a position tracking system, however at the moment it is only used to locate both built and undeveloped sites. The development of the search for the position of items that contain information about their location and address, therefore, requires investigation. The tracking device's GPS module receives satellite-emitted signals for this final project, and the microcontroller converts those signals into coordinates before sending them to the Google Firebase database. The object tracking app receives coordinates from Google Firebase. The reverse geocoding technique is used to turn the coordinates into a place and an address. Testing was carried out by setting up tracking devices in various areas and contrasting applications that make use of reverse geocoding with those that don't. The second method is to compare tracking tool measurements between the app and Google Maps. The final step is to send the location by text message. App test results employing reverse geocoding methods can display the tracking tool's address with a 100% success rate. Applications lacking the reverse geocoding technique, however, just show the tracking device's location without showing the address. When tracking is done outside as opposed to inside a building, where the tracking tool does not get coordinates from the satellite, the tracking device receives the signal more quickly. With Google Maps' validation of distance measurements, the location tracking tool's findings for distance measures have an accuracy of 99%. 100% of SMS attempts to send the google maps link were successful.

Key Words : Tracking Position, GPS, Latitude, Longitude, Reverse Geocoding

KATA PENGANTAR

Puji syukur senantiasa kita ucapkan puji syukur kehadiran Allah SWT yang hingga saat ini masih memberikan kita nikmat iman dan kesehatan, sehingga penulis diberi untuk menyelesaikan tugas akhir yang berjudul “**TRACKING POSISI OBJEK DENGAN MENGGUNAKAN METODE REVERSE GEOCODING**”. Tugas akhir ini ditulis untuk memenuhi syarat mendapatkan gelar Sarjana Terapan Teknik Elektronika.

Tak lupa penulis juga mengucapkan terimakasih yang sebanyak-banyaknya kepada setiap pihak yang telah mendukung serta membantu penulis selama proses penyelesaian tugas akhir ini hingga selesainya tugas akhir ini. Ucapan terima kasih penulis sampaikan pada :

1. Allah SWT. Karena atas berkat dan karunia-Nya lah kami dapat menyelesaikan tugas akhir ini.
2. Bapak Eko Sulisty, M.T, selaku dosen pembimbing 1 tugas akhir atas bimbingan yang diberikan selama ini.
3. Bapak Indra Dwisaputra, M.T, selaku Kaprodi D4 Teknik Elektronika serta dosen pembimbing 2 tugas akhir.
4. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku direktur di Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak M. Iqbal Nugraha, M.Eng selaku kepala jurusan Teknik Elektro dan Informatika.
6. Seluruh staff pengajar dan karyawan di Politeknik Manufaktur Negeri Bangka Belitung.
7. Orang tua beserta keluarga yang selalu memberikan semangat dan dukungan kepada penulis.
8. Pihak-pihak lain yang banyak membantu penulis dalam penyusunan tugas akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan karya tulis ini masih jauh dari kata sempurna serta kesalahan yang penulis yakini diluar batas kemampuan penulis. Maka dari itu penulis dengan senang hati menerima kritik dan saran yang

membangun dari para pembaca. Penulis berharap karya tulis ini dapat bermanfaat bagi semua pihak.

Sungailiat, 6 Februari 2023

Hormat Kami,

Penulis



DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI.....	ii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	viii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Proyek Akhir	2
BAB II.....	3
DASAR TEORI	3
2.1 Sistem <i>Tracking Position</i>	3
2.2 Global Positioning System (GPS).....	4
2.3 Google Maps API.....	5
2.4 Metode Reverse Geocoding	6
2.5 Android Studio	8
2.6 Java.....	8
2.6.1 JDK.....	8
2.6.2 SDK.....	8
2.6.3 AVD	9
2.7 Database	9
2.8 Firebase	9
BAB III	10
METODE PENELITIAN.....	10
3.1 Pengumpulan dan Pengolahan Data	11
3.2 Perancangan <i>Hardware</i> dan <i>Software</i> Sistem <i>Tracking</i> Posisi	11

3.2.1	Perancangan <i>Hardware</i>	11
3.2.2	Perancangan <i>Software</i>	12
3.3	Pembuatan <i>Hardware</i> dan <i>Software</i> Sistem <i>Tracking</i> Posisi.....	12
3.3.1	Pembuatan <i>Hardware</i>	12
3.3.2	Pembuatan <i>Software</i>	13
3.4	Pengujian <i>Hardware</i> dan <i>Software</i> Sistem <i>Tracking</i> Posisi.....	13
3.4.1	Pengujian <i>Hardware</i>	14
3.4.2	Pengujian <i>Software</i>	14
3.4.3	Pengujian Keseluruhan	14
3.5	Evaluasi dan Perbaikan	15
3.6	Pembuatan Laporan Proyek Akhir.	15
BAB IV	16
PEMBAHASAN	16
4.1	Deskripsi Alat.....	16
4.2	Perancangan dan Pembuatan <i>Hardware</i> Sistem <i>Tracking</i> Posisi	16
4.2.1	Perancangan <i>Hardware</i> Secara Mekanik	17
4.2.2	Pembuatan <i>Hardware</i> Secara Mekanik	17
4.2.3	Perancangan <i>Hardware</i> Secara Elektrik.....	18
4.2.4	Pembuatan <i>Hardware</i> Secara Elektrik	19
4.3	Pengujian <i>Hardware</i>	20
4.3.1	Pengujian Modul GPS	20
4.3.2	Pengujian Modul SIM800L.....	22
4.3.3	Pengujian LCD ST7789	24
4.4	Perancangan dan Pembuatan <i>Software</i>	26
4.4.1	Perancangan Aplikasi pada Smartphone	26
4.4.2	Pembuatan Aplikasi pada Smartphone.....	27
4.4.3	Komunikasi NodeMCU ESP8266 dengan <i>Realtime Database</i> Firebase.....	31
4.4.4	Komunikasi <i>software</i> Android Studio dengan <i>Realtime Database</i> Firebase.....	31
4.5	Pengujian Aplikasi <i>Tracking</i> Posisi Objek.....	33
4.6	Pengujian Keseluruhan <i>Tracking</i> Posisi Objek dengan menggunakan Metode <i>Reverse Geocoding</i>	34

4.6.1	Pengujian Pelacakan Aplikasi <i>Tracking</i> Posisi Objek dari Beberapa Lokasi yang Berbeda	35
4.6.2	Pengujian Jarak pada Aplikasi <i>Tracking</i> Posisi Objek dengan Google Maps	40
4.6.3	Pengujian Pengiriman Lokasi Alat <i>Tracking</i> melalui Pesan Teks	42
BAB V	44
KESIMPULAN DAN SARAN	44
DAFTAR PUSTAKA	45
LAMPIRAN 1	47
DAFTAR RIWAYAT HIDUP	47
LAMPIRAN 2	49
PROGRAM KESELURUHAN	49



DAFTAR TABEL

Tabel 4.1 Pengujian Modul GPS.....	22
Tabel 4.2 Pengujian Modul SIM800L	24
Tabel 4.3 Pengujian LCD ST7789	25
Tabel 4.4 Pengujian Aplikasi	34
Tabel 4.5 Pengujian Pelacakan pada Alat <i>Tracking</i> 1.....	37
Tabel 4.6 Pengujian Pelacakan pada Alat <i>Tracking</i> 2.....	38
Tabel 4.7 Pengujian Pelacakan pada Alat <i>Tracking</i> 3.....	38
Tabel 4.8 Pengujian Jarak pada Alat <i>Tracking</i> 1	41
Tabel 4.9 Pengujian Jarak pada Alat <i>Tracking</i> 2	41
Tabel 4.10 Pengujian Jarak pada Alat <i>Tracking</i> 3	41
Tabel 4.11 Pengujian Pengiriman Lokasi pada Alat <i>Tracking</i> 1	42
Tabel 4.12 Pengujian Pengiriman Lokasi pada Alat <i>Tracking</i> 2	43
Tabel 4.13 Pengujian Pengiriman Lokasi pada Alat <i>Tracking</i> 3	43

DAFTAR GAMBAR

Gambar 2.1 Tampilan Aplikasi <i>Google Maps</i>	6
Gambar 3.1 Diagram Metode Pelaksanaan	10
Gambar 4.1 Desain alat <i>tracking</i> posisi secara keseluruhan	17
Gambar 4.2 Peletakan komponen pada box alat	18
Gambar 4.3 Skema pengkabelan alat <i>tracking</i> posisi	19
Gambar 4.4 Pengkabelan alat <i>tracking</i> posisi	20
Gambar 4.5 Blok Pengujian Modul GPS	21
Gambar 4.6 (a) Tampilan Titik Koordinat pada <i>Google Maps</i> , (b) Tampilan Titik Koordinat Modul GPS pada Serial Monitor, (c) <i>Delay Waktu Aktif Modul GPS</i>	21
Gambar 4.7 Blok Pengujian Modul SIM800L	23
Gambar 4.8 (a) Tampilan Pengiriman <i>Link Google Maps</i> pada SMS, (b) Tampilan Pengiriman <i>Link Google Maps</i> pada Serial Monitor, (c) <i>Delay Waktu Pengiriman Link Google Maps</i>	23
Gambar 4.9 Pengujian LCD ST7789	24
Gambar 4.10 (a) Tampilan Titik Kooordinat pada LCD ST7789, (b) Tampilan Titik Koordinat pada Serial Monitor	25
Gambar 4.11 Perancangan pembuatan aplikasi pada android studio	26
Gambar 4.12 Perancangan <i>Interface</i> Aplikasi	27
Gambar 4.13 Tampilan Menu Utama Aplikasi <i>Tracking</i> Posisi Objek pada <i>Software</i> Android Studio	28
Gambar 4.14 Tampilan Menu Utama Aplikasi <i>Tracking</i> Posisi Objek pada <i>Smartphone</i>	30
Gambar 4.15 Tampilan Menu <i>Maps</i> pada <i>software</i> Android Studio dan <i>Smartphone</i>	30
Gambar 4.16 Tampilan Koneksi <i>software</i> Android Studio dengan <i>Realtime database</i> firebase	32
Gambar 4.17 Tampilan <i>realtime database</i> firebase	32

Gambar 4.18 (a) Tampilan Menu Utama Aplikasi, (b) Tampilan Menu <i>Maps</i> Aplikasi	33
Gambar 4.19 Blok Diagram Pengujian Pelacakan Alat <i>Tracking</i>	35
Gambar 4.20 (a) Pengujian menggunakan Metode <i>Reverse Geocoding</i> ,	36
(b) Pengujian Tanpa menggunakan Metode <i>Reverse Geocoding</i>	36
Gambar 4.21 (a) Tampilan Jarak pada Aplikasi, (b) Pengukuran Jarak pada Google Maps	40



DAFTAR LAMPIRAN

LAMPIRAN 1 DAFTAR RIWAYAT HIDUP	47
LAMPIRAN 2 PROGRAM KESELURUHAN	49



BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Ranah teknologi telah maju dengan sangat cepat di masa sekarang. Salah satunya di bidang komunikasi elektronik. Belakangan ini banyak teknologi yang sudah dikenal masyarakat, salah satunya adalah GPS (Global Positioning System). Lokasi suatu objek di permukaan bumi dapat ditentukan dengan teknik ini. Lintang, bujur, dan ketinggian di atas permukaan laut adalah koordinat tiga dimensi yang digunakan untuk menemukan informasi[1].

Sistem *tracking position* atau pelacakan posisi adalah teknologi yang menggunakan koordinat lintang dan bujur untuk menentukan di mana GPS berada. Pengguna juga dapat melihat kondisi lingkungan di mana objek bergerak menggunakan GPS. Namun, tidak semua objek dapat digunakan sebagai media pembawa karena masing-masing objek ini memiliki banyak fungsi. Oleh karena itu, *GPS Tracker* harus memiliki sudut pandang yang lebar dan minimal untuk melakukan pemilihan objek bergerak sebagai media pembawa[1]. Pada penelitian ini, penulis menggunakan metode *Reverse Geocoding* untuk menentukan lokasi keberadaan objek. *Reverse Geocoding* adalah proses yang mengubah koordinat lokasi menjadi alamat atau nama tempat[2]. Hasil dari penelitian ini adalah alat *tracking* akan memberikan informasi lokasi objek melalui aplikasi dan SMS.

Alat ini akan dibuat menjadi bentuk kotak yang akan diletakkan pada objek. Sistem pelacakan ini menggunakan modul SIM800L juga modul NodeMCU. Sistem akan mengirimkan koordinat melalui NodeMCU dan diterima oleh aplikasi *smartphone*, dan modul SIM800L akan mengirim pesan singkat melalui SMS ke *smartphone* pengguna yang berisi tautan Google Maps. Alat ini menjadi salah satu dari beberapa opsi untuk mengurangi pengawasan dan mengoptimalkan sistem keamanan. Hal ini tentunya memudahkan pengguna untuk memantau objek dan mengurangi risiko pencurian terhadap benda berharga pengguna.

Terdapat beberapa penelitian serupa yang telah meneliti alat pendeteksi objek ini, sehingga pada proyek akhir ini akan dikembangkan lagi alat pendeteksi objek yang telah diteliti sebelumnya. Alat ini dilengkapi dengan aplikasi yang akan menampilkan alamat dari keberadaan objek dan akan menunjukkan jarak objek dengan pengguna.

1.2 Perumusan Masalah

Pada pembuatan proyek akhir “*Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*” muncul beberapa rumusan sebagai berikut :

- a. Bagaimana cara mendesain dan membuat “*Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*”?
- b. Bagaimana cara GPS memberikan koordinat ke aplikasi untuk mendeteksi keberadaan objek?
- c. Bagaimana cara aplikasi bisa menampilkan jarak antara objek dan pengguna?

1.3 Tujuan Proyek Akhir

Tujuan dari pembuatan proyek akhir “*Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*” yaitu :

- a. Merancang dan membangun alat *Tracking* posisi objek yang menggunakan teknologi GPS dengan keluaran berupa titik koordinat.
- b. Merancang dan membangun aplikasi yang bisa menampilkan posisi objek dan jarak objek dengan pengguna.

BAB II

DASAR TEORI

2.1 Sistem *Tracking Position*

Sistem pelacakan posisi adalah teknologi yang menggunakan koordinat lintang dan bujur untuk menentukan di mana GPS berada. Pengguna dapat melihat kondisi lingkungan di mana objek bergerak menggunakan *GPS Tracker*[1]. Salah satu contohnya adalah kendaraan, hewan, manusia, dll. Sebelumnya telah dilakukan beberapa penelitian yang berkaitan dengan sistem *tracking position* yang dapat dijadikan sebagai referensi yaitu penelitian yang dilakukan oleh Dicka Ariptian Rahayu dan Rifki Kosasih (2018) yang meneliti Aplikasi pelacak Kendaraan Berbasis Smartphone Menggunakan Arduino Uno dan Modul SIM808. Penelitian ini mengembangkan sistem pelacakan kendaraan menggunakan arduino uno dan modul SIM808, serta aplikasi berbasis *smartphone* menggunakan *framework*. Prototipe sistem pelacakan kendaraan ini terdiri dari tiga komponen yang saling berhubungan: perangkat pelacak, aplikasi seluler, dan server web[3].

Selain itu penelitian yang dilakukan oleh Budianto, Dkk (2019) yang berjudul Sistem Penjejak Posisi Kendaraan Bermotor Berbasis GPS Melalui Media SMS. Penelitian ini membahas alat pencari posisi kendaraan yang dapat menampilkan posisi kendaraan di suatu tempat menggunakan aplikasi Google Maps dengan bantuan Mikrokontroler AT-Mega 328 dan perantara modul SIM808 sehingga lokasi koordinat dapat dilihat pada Aplikasi Google Maps, yang kemudian dikonversi menjadi *link* Google Maps[4].

Penelitian serupa juga dilakukan oleh Anton Wahyudi, dkk (2013) yang meneliti tentang Aplikasi yang dapat melacak serta memberi tahu posisi device kepada pemilik apabila terjadi kehilangan. Selain itu pada aplikasi tersebut juga terdapat fitur untuk menampilkan pesan ke layar device tersebut dengan bantuan dari website yang dapat diakses melalui koneksi internet terdekat[5].

Selanjutnya penelitian oleh Imam Firdaus Kharisma Juda, dkk (2022) yang meneliti tentang alat pelacak yang dapat memantau dan melacak transportasi darat secara real time yang dapat memberikan titik koordinat lokasi dengan memantau pada aplikasi Blynk secara real time[6].

Dengan menyimpulkan dari hasil penelitian di atas, terciptalah ide untuk membuat proyek akhir yang berjudul “*Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*”. Proyek akhir ini merupakan ide dari penelitian di atas dengan tambahan menggunakan metode *reverse geocoding*.

2.2 Global Positioning System (GPS)

Departemen Pertahanan Amerika Serikat mengembangkan dan mengoperasikan *Global Positioning System* (GPS), satelit sistem navigasi radio. GPS dihitung dengan mengukur waktu yang dibutuhkan untuk mentransmisikan sinyal dari satelit dan mengalikannya dengan kecepatan cahaya (3×10^8 meter/detik) untuk menentukan seberapa jauh pesawat penerima GPS dari setiap satelit. Penerima GPS dapat menghitung posisi tetap suatu titik, yaitu lintang dan bujur menggunakan sinyal yang dikirim oleh satelit dari setidaknya tiga satelit yang berbeda[3]. Keberadaan objek (*tracking*) dapat ditentukan dengan menggunakan GPS ini berdasarkan posisi lintang dan bujur, dan posisi lintang dan bujur dapat divisualisasikan dalam bentuk peta. Ada dua jenis alat pelacak, yaitu:

1. Pemantauan Pasif / *Passive Tracking*

Alat untuk menyimpan data seperti lokasi, kecepatan, dan arah. Ketika kendaraan kembali, data dari perangkat yang diinstal dapat diunduh ke komputer dan dianalisis.

2. Pemantauan Aktif / *Active Tracking*

Alat yang digunakan untuk mengumpulkan data seperti lokasi, kecepatan, dan arah. Data kemudian ditransmisikan secara real time ke komputer server melalui jaringan seluler atau satelit.

2.3 Google Maps API

Google Maps adalah layanan peta digital yang paling efektif dan banyak digunakan di dunia saat ini. Google Maps menawarkan gambar pemetaan seluruh permukaan muka bumi. Layanan ini juga dapat memberikan petunjuk arah ke lokasi mana pun dan menghitung jarak terpendek dan rute perjalanan paling hemat biaya. Google Maps digunakan dalam aplikasi Android melalui layanan Google Maps Android API v2[2].

Ada banyak penawaran API yang tersedia, termasuk Google Maps Javascript API, Google Maps API for Flash, dan Google Static Maps API. Untuk menggunakan Google Maps API, pengguna harus terlebih dahulu mendaftar untuk mendapatkan kunci API, yang nantinya akan digunakan untuk mengakses Google Maps API[4].

Layanan ini dapat menampilkan penanda lokasi, objek pada peta, dan melampirkan satu atau beberapa fragmen peta. Kita dapat menghemat waktu dan uang untuk membuat aplikasi peta digital dengan menggunakan Google Maps Android API v2, memungkinkan kita untuk fokus hanya pada data yang akan ditampilkan.

Google Maps Android API juga menyertakan fitur penanda peta. Pengguna dapat menambahkan penanda ke peta untuk menandai lokasi tertentu. Selain penanda, Google Maps memungkinkan aplikasi untuk menggambar polyline dan poligon pada peta untuk menandai rute dan area. Pengguna juga bisa mendapatkan beberapa perspektif di peta menggunakan Google Maps API, yang mencakup kemampuan rotasi dan pembesaran.



Gambar 2.1 Tampilan Aplikasi *Google Maps*

2.4 Metode Reverse Geocoding

Reverse Geocoding adalah proses yang mengubah titik koordinat lokasi menjadi alamat atau nama tempat[2]. Teknologi ini dapat mengidentifikasi nama jalan, tempat, dan atau pembagian geografis seperti desa, kecamatan, provinsi, atau negara. Ketika dikombinasikan dengan geocoding dan rute layanan reverse geocoding adalah komponen penting dari aplikasi seluler berbasis lokasi untuk mengubah titik koordinat GPS menjadi alamat jalan yang lebih mudah dibaca dan dipahami. Layanan ini juga menjadi lebih banyak tersedia melalui API dan banyak digunakan dalam layanan situs web serta aplikasi perangkat seluler[5]. Layanan yang akan dipakai pada alat ini adalah menggunakan layanan *Google Maps Geocoding API* yang dimana titik koordinat akan dikirim dan diterjemahkan oleh geocoder menjadi alamat sesuai dengan *latitude* dan *longitude*. Secara umum, ada dua jenis *reverse geocoding conventional* dan *online*. Konvensional reverse geocoding sering dilakukan di bawah tanggung jawab profesional sistem informasi geografis (GIS) menggunakan alat geocoding utama perangkat lunak GIS. Pada

tahap prapemrosesan reverse geocoding, pengguna menyediakan dataset alamat referensi, menyiapkan pencari alamat, dan menetapkan aturan untuk pencocokan yang tidak pasti. Geocoding terbalik online sebagian besar diterbitkan oleh perusahaan komersial sebagai layanan Web. *Reverse geocoding online* biasanya beroperasi pada sistem yang dapat diakses jaringan dan biasanya dapat diakses melalui Antarmuka Pemrograman Aplikasi Web (API) melalui internet. Koordinat yang ditentukan pengguna diubah menjadi data tekstual melalui API *reverse geocoding*. Setiap komponen reverse geocoding dihitung secara otomatis oleh skrip Python yang menggunakan API dan paket Geocoder[10].

Sebelumnya telah dilakukan beberapa penelitian yang berkaitan dengan metode reverse geocoding yang dapat dijadikan sebagai referensi yaitu penelitian oleh Luthfi Rizaldi (2022) yang meneliti aplikasi *Sales Marketing* Berbasis Android Menggunakan Metode Reverse Geocoding di PT. Sri Indah Labetama. Penerapan metode *reverse geocoding* pada penelitian ini adalah dapat mengubah nilai atau titik koordinat yang dikirim oleh *sales marketing* ketika sedang melakukan kunjungan ke *customer* sehingga kepala bagian dapat mengetahui lokasi *sales marketing* tersebut berupa alamat[11].

Selanjutnya penelitian oleh Andi Juansyah (2015) yang meneliti Pembangunan Aplikasi *Child Tracker* Berbasis Assisted-Global Positioning System (A-GPS) Dengan Platform Android. Penelitian ini membahas tentang aplikasi yang digunakan untuk mengetahui keberadaan anak berupa alamat. Aplikasi ini dapat mengetahui keberadaan anak saat berpergian dengan menggunakan *smartphone* android. Selain itu juga dapat membantu anak memberikan lokasi dan menghubungi orangtua dengan cepat dan pada saat situasi darurat[12].

Penelitian serupa juga dilakukan oleh Fandy Ong, dkk (2021) yang meneliti Aplikasi Pengoptimalan Rute Pengiriman Barang pada PT. XYZ. Penelitian ini membahas tentang aplikasi yang dapat memberikan informasi tentang detail rute pengiriman barang beserta alamat pengiriman, dan barang yang dipesan dalam *delivery order* yang diantar dalam suatu rute[13].

2.5 Android Studio

Android Studio adalah lingkungan pengembangan baru dan terintegrasi penuh, yang dimiliki oleh Google untuk pengembangan aplikasi Android. Tujuan Android Studio adalah menggantikan Eclipse, IDE paling populer saat ini, sebagai alat baru untuk pengembangan aplikasi. Struktur proyek dan sebagian besar file di direktori SDK akan muncul saat Anda membuat proyek baru di Android Studio, perpindahan ke sistem manajemen berbasis Gradle ini meningkatkan fleksibilitas proses pengembangan. Anda dapat melihat setiap perubahan estetika yang Anda buat pada aplikasi secara *realtime* menggunakan Android Studio. Saat dimuat di banyak *smartphone* Android yang berbeda, Anda juga dapat melihat perbedaan dalam pengaturan dan resolusi sekaligus[12].

2.6 Java

Java adalah sebuah bahasa pemrograman yang dapat digunakan untuk membuat berbagai jenis aplikasi, seperti aplikasi desktop, web, dan lainnya. Seperti halnya bahasa pemrograman konvensional lainnya, Java juga memiliki kemampuan untuk membuat aplikasi yang sama. Java merupakan bahasa pemrograman yang berorientasi objek dan dapat berjalan di berbagai platform sistem operasi. Java tidak hanya terbatas pada satu sistem operasi saja, tetapi juga dapat berjalan pada berbagai sistem operasi open source[14].

2.6.1 JDK

JDK (Java Development Kit) adalah alat bantu yang mempermudah proses pembuatan aplikasi Android. Anda dapat mengunduh installer JDK dari situs resmi. Pastikan untuk memilih installer JDK yang sesuai dengan sistem operasi yang digunakan pada komputer Anda. Versi JDK yang dapat digunakan untuk membuat program Android adalah JDK 1.6 dan 1.7, atau versi terbaru dari JDK[15] .

2.6.2 SDK

Para developer aplikasi Android membutuhkan sebuah perangkat lunak seperti SDK (Software Development Kit) yang dibuat untuk membangun aplikasi Android

sesuai dengan versi Android yang digunakan. SDK ini dapat diunduh dari situs resmi developer.android.com dan pilihan package yang tersedia sesuai dengan versi Android yang dibutuhkan. Sangat penting bagi developer untuk memastikan bahwa package yang diunduh sesuai dengan kebutuhan mereka dalam membangun aplikasi Android[16].

2.6.3 AVD

AVD singkatan dari Android Virtual Device adalah sebuah perkembangan aplikasi untuk membantu dalam melakukan pengujian sementara dan debugging aplikasi Android. AVD adalah sebuah emulator aplikasi yang memungkinkan untuk menjalankan Android virtual tanpa menggunakan perangkat smartphone Android. Aplikasi yang sudah selesai diprogram tidak selalu dapat langsung diuji pada AVD, karena memerlukan kapasitas memori RAM yang besar[15].

2.7 Database

Database adalah kumpulan data yang terorganisir dan tersimpan secara sistematis di dalam komputer sehingga dapat dicari dan diambil informasinya melalui suatu program komputer. Perangkat lunak yang digunakan untuk mengatur dan memanggil data dalam database disebut Sistem Manajemen Basis Data. Ilmu informasi mempelajari tentang bagaimana mengelola dan mengakses data dalam sebuah database melalui sistem manajemen basis data[14].

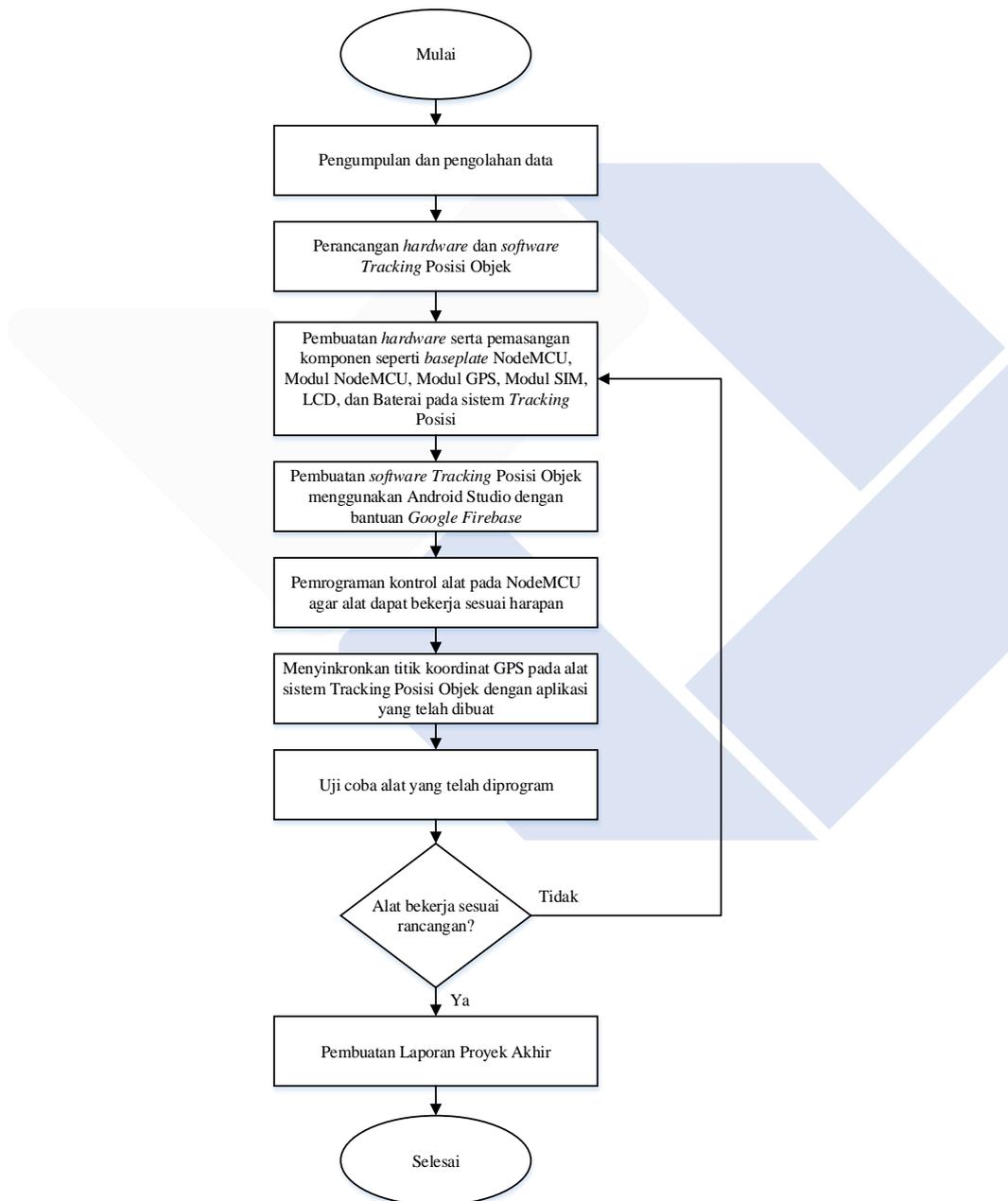
2.8 Firebase

Database adalah sekumpulan data yang disimpan dengan aturan dan struktur tertentu dalam komputer, sehingga informasi dapat diperoleh dan diolah dengan bantuan program komputer. Perangkat lunak yang digunakan untuk mengatur dan mengakses data dalam database disebut Sistem Manajemen Basis Data. Ilmu informasi mempelajari bagaimana mengelola dan mengambil data dari sebuah database melalui sistem manajemen basis data[14].

BAB III

METODE PENELITIAN

Dalam pelaksanaan proyek akhir ini, dilakukan beberapa tahapan penelitian yang bertujuan untuk memudahkan penulis dalam proses pembuatan proyek akhir. Berikut adalah tahapan-tahapan penelitian yang telah dibuat.



Gambar 3.1 Diagram Metode Pelaksanaan

Adapun tahap penelitian yang dilakukan sebagai berikut :

3.1 Pengumpulan dan Pengolahan Data

Pada tahap ini, dilakukan pengumpulan data melalui studi literatur yang sesuai dengan proyek akhir yang akan dibuat. Dalam melakukan tahap ini, akan diperoleh informasi yang berkaitan dengan proyek akhir yang kemudian akan dibuat dan sebagai referensi dalam proses pembuatan dan penulisan. Gambaran alat yang akan dibuat juga data diperoleh berdasarkan penelitian yang telah dilakukan sebelumnya yang diketahui yang kemudian akan dikembangkan. Penulis menggunakan dua metode pengumpulan data. Yang pertama yaitu pengumpulan jurnal-jurnal yang berkaitan dengan judul proyek akhir dan melakukan konsultasi dengan dosen pembimbing. Setelah mengumpulkan informasi, langkah selanjutnya adalah merancang perangkat keras dan perangkat lunak yang akan dibuat.

3.2 Perancangan *Hardware* dan *Software* Sistem *Tracking* Posisi

Tahapan kedua yang dilakukan dalam pembuatan proyek akhir ini adalah perancangan dan pembangunan *hardware* dan *software*. Tahapan ini merupakan tahapan awal yang akan menentukan gambaran proyek akhir yang akan dibuat, perancangan serta pembangunan *hardware* memiliki fungsi sebagai penentu bentuk fisik suatu *hardware* yang akan dirancang dari alat *tracking* posisi dan penempatan masing-masing komponen pada alat. Sementara itu pada tahapan perancangan *software* memiliki fungsi untuk melacak dan mengetahui keberadaan alat *tracking* tersebut.

3.2.1 Perancangan *Hardware*

Ada 2 jenis desain *hardware* yang dibuat pada alat *tracking* posisi yaitu perangkat mekanik dan perangkat elektrik. Pada tahap ini dilakukan rancang bangun konstruksi dari alat *tracking* menggunakan aplikasi *sketch up*. Tahap ini menentukan ukuran, alat, dan bahan yang digunakan untuk pembuatan alat *tracking*. Desain perangkat mekanik yang akan dibuat seperti berikut.

1. Merancang box dengan bahan plastik dengan ukuran panjang 12,5 cm, lebar 8,5 cm, dan tinggi 5 cm.
2. Pembuatan box kontrol sebagai tempat untuk meletakkan NodeMCU, Modul GPS, Modul SIM, LCD, dan Baterai.

Kemudian, pada tahap perancangan *hardware* ini merancang *wiring* kontrol menggunakan NodeMCU sebagai mikrokontroler untuk mengontrol dan mengirim data koordinat ke firebase.

3.2.2 Perancangan Software

Pada tahap perancangan *software* ini dibuat rancangan berupa:

1. Pemrograman Arduino yang digunakan untuk menampilkan titik koordinat menggunakan modul GPS NEO 6-M yang kemudian titik koordinat tersebut ditampilkan di modul LCD TFT 7789 dan mengirimkan SMS menggunakan modul SIM 800L.
2. Google Firebase digunakan untuk menyimpan data titik koordinat dan diteruskan ke dalam aplikasi.
3. Merancang tampilan aplikasi menggunakan *software* Android Studio untuk menampilkan *waypoint* beserta alamat keberadaan alat *tracking* dan menampilkan jarak antara alat *tracking* dengan pengguna.

3.3 Pembuatan Hardware dan Software Sistem Tracking Posisi

Setelah tahap perencanaan, selanjutnya tahapan pembuatan, yang di mana pada tahap ini direalisasikannya *hardware* dan *software* yang sudah dirancang sebelumnya.

3.3.1 Pembuatan Hardware

Tahap pembuatan *hardware* secara mekanis dibuat dengan membuat bentuk fisik dari alat *tracking* yang mengacu pada desain yang telah dibuat sebelumnya. Tahapan membuat perangkat keras secara mekanis seperti di bawah ini .

1. Membuat box kontrol dengan memotong box sesuai kebutuhan.
2. Meletakkan komponen seperti NodeMCU, Modul GPS, Modul SIM, LCD, dan Baterai.

3.3.2 Pembuatan *Software*

Tahap pembuatan *software* secara mekanis dibuat dengan membuat desain *software* yang telah dibuat sebelumnya. Pembuatan *software* pada proyek akhir ini terdiri dari dua jenis yaitu *software* mikrokontroler dan aplikasi untuk *tracking* posisi objek. Tahap-tahap membuat *software* untuk mikrokontroler sebagai berikut.

1. Membuat program NodeMCU ESP8266 untuk memprogram modul GPS untuk mendapatkan data titik koordinat yang kemudian akan dikirimkan ke Google Firebase.
2. Membuat program NodeMCU ESP8266 untuk memprogram modul SIM untuk mendapat pesan teks berupa *link* Google Maps.
3. Membuat program NodeMCU ESP8266 untuk memprogram modul LCD untuk menampilkan titik koordinat dari keberadaan alat *tracking*.

Tahap-tahap membuat *software* untuk aplikasi *tracking* posisi objek sebagai berikut.

1. Membuat aplikasi menggunakan *platform* Android Studio yang dapat di *install* pada *smartphone* pengguna.
2. Membuat program untuk menampilkan alamat dari alat *tracking* beserta jarak antara alat *tracking* dengan pengguna menggunakan *platform* Android Studio

3.4 Pengujian *Hardware* dan *Software* Sistem *Tracking* Posisi

Pengujian dilakukan apabila alat dan aplikasi telah selesai dikerjakan. Pengujian yang dilakukan yaitu dengan melihat apakah alat dan aplikasi sudah sesuai dengan yang dirancang dan apakah alat dapat diaplikasikan sesuai dengan target yang akan

dilacak keberadaannya. Pengujian dalam penelitian ini juga dibagi menjadi 2 bagian yaitu pengujian *hardware* dan *software*.

3.4.1 Pengujian Hardware

1. Pengujian NodeMCU dengan Modul GPS untuk mengetahui pengiriman data keberadaan alat *tracking* yang kemudian akan dikirimkan ke aplikasi.
2. Pengujian NodeMCU dengan Modul SIM800L yang untuk mengetahui pengiriman data berupa link *Google Maps*.
3. Pengujian NodeMCU dengan LCD untuk mengetahui keberhasilan LCD dalam menampilkan titik koordinat dari keberadaan alat *tracking*.

3.4.2 Pengujian Software

1. Pengujian Google Firebase bertujuan untuk mengetahui apakah Google Firebase dapat menyimpan data titik koordinat ke dalam *realtime* database.
2. Pengujian aplikasi untuk mengetahui apakah aplikasi yang diciptakan dapat mengakses titik koordinat yang dikirimkan oleh Google Firebase.
3. Pengujian aplikasi untuk mengkonversi titik koordinat menjadi sebuah alamat dengan meminta *request* pada Google API (*Reverse Geocoding*).
4. Pengujian aplikasi yang sudah diciptakan dalam menampilkan alamat keberadaan objek beserta jarak objek dengan pengguna.

3.4.3 Pengujian Keseluruhan

Pada tahap ini, dilakukan pengujian *hardware* dan *software* secara keseluruhan. Pengujian ini dilakukan untuk mengetahui apakah sistem sudah berjalan seperti yang diinginkan meliputi modul GPS sebagai penerima titik koordinat yang kemudian datanya akan diproses oleh aplikasi, modul SIM 800L, dan LCD. Modul SIM 800L sebagai pengirim *link* titik koordinat melalui pesan teks, dan LCD sebagai penampil data titik koordinat pada alat tracking posisi. Data koordinat akan dikirimkan ke *firebase* sebagai penyimpanan data, kemudian aplikasi akan mengolah data dari *firebase* menjadi sebuah *tag* lokasi atau alamat sesuai dengan metode

reverse geocoding yang digunakan sebagai penunjuk keberadaan alat *tracking* serta menampilkan jarak antara alat *tracking* dengan pengguna.

3.5 Evaluasi dan Perbaikan

Hasil dari pengujian pelacakan posisi dilakukan evaluasi dan jika masih terdapat kekurangan maka selanjutnya dilakukan perbaikan untuk menyempurnakan alat yang dibuat.

3.6 Pembuatan Laporan Proyek Akhir.

Pembuatan laporan akhir merupakan tahap akhir dalam pembuatan tugas akhir. Bertujuan untuk meringkas semua pembahasan tentang tugas akhir, meliputi latar belakang, perumusan masalah, tujuan, landasan teori, metode penelitian, diskusi, hasil, kesimpulan, dan saran. Laporan Proyek Akhir juga sebagai penyaluran informasi dari penulis kepada pembaca, dengan harapan akan dijadikan referensi oleh pembaca.

BAB IV

PEMBAHASAN

Pada bab ini menguraikan rancangan dari proses dan hasil pengerjaan proyek akhir berdasarkan metode yang telah dipaparkan pada bab sebelumnya. Sebelum dilakukan perancangan dan pembuatan Sistem *Tracking* Posisi, secara umum bab ini menguraikan tentang perancangan *hardware* dan *software*, pembuatan *hardware* dan *software*, dan tahap terakhir yaitu melakukan pengujian alat secara keseluruhan serta melakukan analisis data dan pembuatan laporan proyek akhir.

4.1 Deskripsi Alat

“*Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*” adalah alat yang digunakan untuk mendeteksi keberadaan suatu objek. Sistem *tracking* posisi objek yang digunakan ialah GPS, GPS merupakan teknologi yang digunakan untuk mendapatkan titik koordinat objek melalui satelit. Ketika modul GPS sudah mendapatkan titik koordinat, data tersebut akan dikirimkan ke dua *output* yaitu *database* dan pesan teks.

NodeMCU akan mengirimkan data titik koordinat ke *realtime database* (Firebase) dengan catatan harus terkoneksi dengan Wi-Fi sehingga aplikasi dapat mengakses data titik koordinat tersebut dan mengonversinya menjadi sebuah *tag* lokasi dan alamat serta menampilkan jarak antara objek dengan pengguna.

Sedangkan pesan teks, data koordinat yang telah didapatkan dari GPS akan diubah ke dalam sebuah link *Google Maps* dengan mengirim pesan teks ke nomor alat dengan sebuah kode berupa “FIND” yang sudah diprogram sebelumnya. Pengguna akan mendapatkan informasi berupa *link* pada pesan teks untuk mengetahui keberadaan objek tersebut.

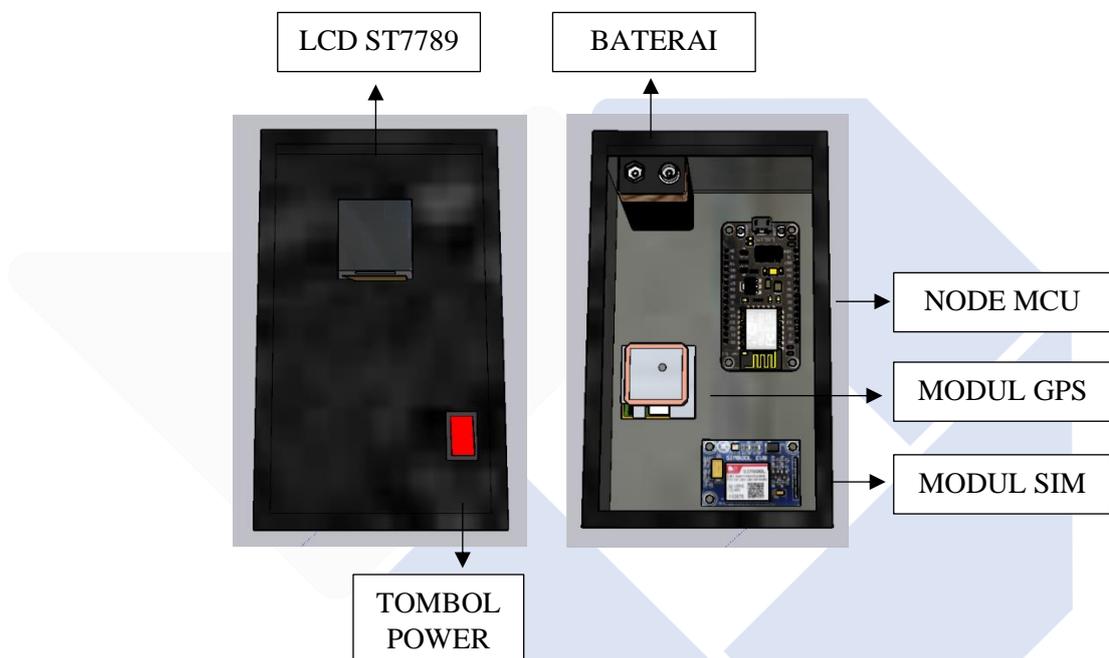
4.2 Perancangan dan Pembuatan Hardware Sistem *Tracking* Posisi

Perancangan dan pembuatan *hardware* dilakukan melalui 2 tahap yaitu, perancangan *hardware* secara mekanik dan perancangan *hardware* secara elektrik.

Berikut ini merupakan tahap yang dilakukan dalam perancangan dan pembuatan *hardware* proyek akhir ini.

4.2.1 Perancangan *Hardware* Secara Mekanik

Perancangan ini dilakukan dengan membuat desain keseluruhan dari alat *tracking* yang dibuat menggunakan aplikasi *Sketch Up*. Berikut adalah gambar rancangan yang telah dibuat.

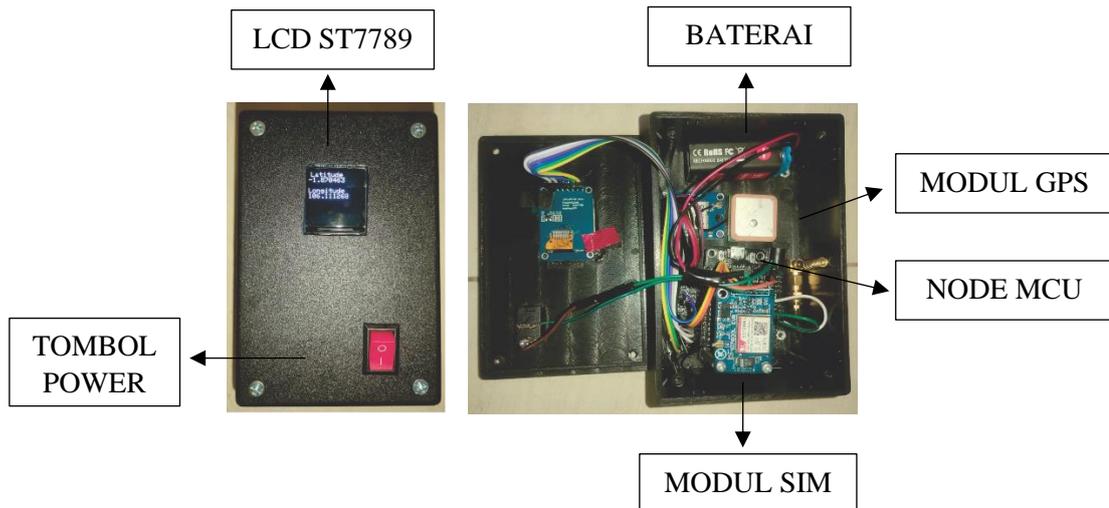


Gambar 4.1 Desain alat *tracking* posisi secara keseluruhan

4.2.2 Pembuatan *Hardware* Secara Mekanik

Pada tahap ini, dilakukan dengan membuat alat *tracking* yang telah dirancang sebelumnya secara nyata. Alat dan bahan yang digunakan juga disesuaikan dengan yang sudah direncanakan sebelumnya.

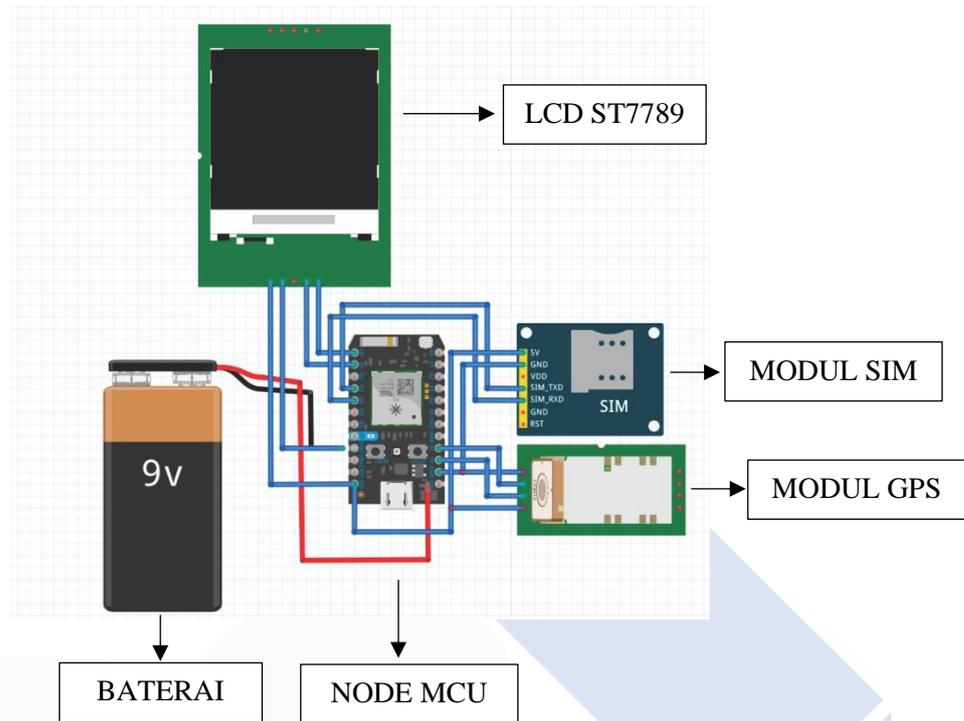
Tahap pembuatan dilakukan dengan meletakkan setiap komponen seperti NodeMCU, Modul GPS, Modul SIM, dan juga LCD pada box. Berikut adalah gambar peletakan komponen pada box.



Gambar 4.2 Peletakan komponen pada box alat

4.2.3 Perancangan *Hardware* Secara Elektrik

Hardware elektrik merupakan rangkaian yang dibuat untuk memenuhi tujuan dari proyek akhir. Adapun yang diinginkan dari rangkaian ini adalah dapat mendeteksi keberadaan objek, dapat menampilkan koordinat objek yang diterima, dapat mengirim lokasi objek melalui pesan teks, baterai yang digunakan dapat memberikan sumber daya terhadap alat, dan dapat melakukan penyimpanan data di *database*. Dari tujuan tersebut, maka digunakan modul GPS Neo-6M untuk melakukan pendeteksian keberadaan objek, LCD ST7789 untuk menampilkan koordinat objek yang diterima, modul SIM800L untuk mengirim lokasi objek melalui pesan teks, Baterai 9V agar alat yang digunakan dapat bertahan lama, dan NodeMCU ESP8266 sebagai perantara untuk melakukan penyimpanan data di *database*. Adapun perancangan yang dibuat adalah sebagai berikut.

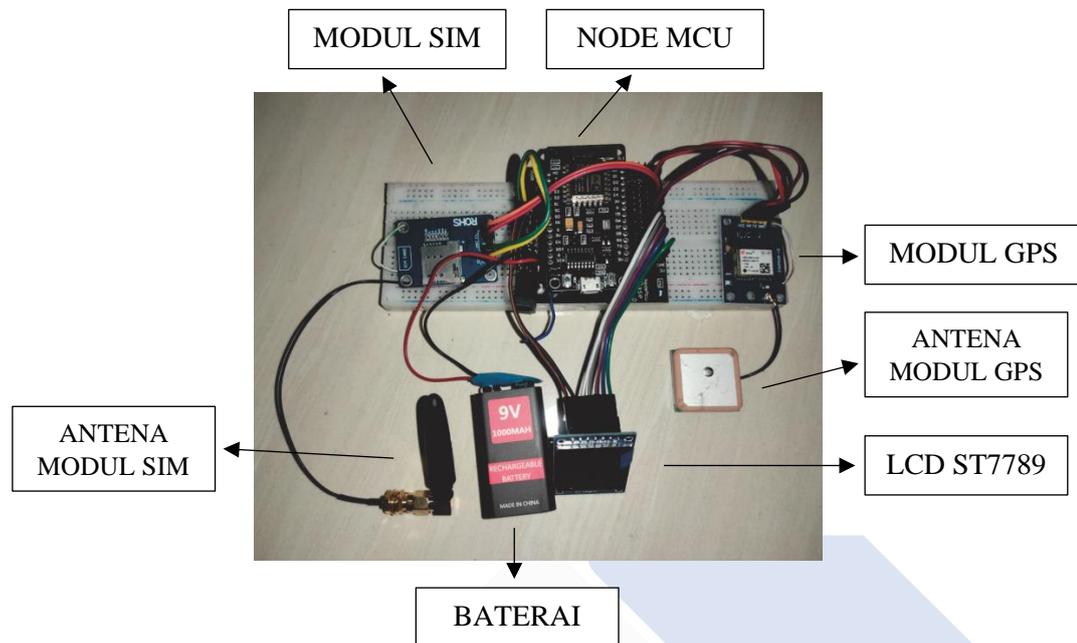


Gambar 4.3 Skema pengkabelan alat *tracking* posisi

4.2.4 Pembuatan *Hardware* Secara Elektrik

Pada tahap ini, dilakukan dengan merangkai atau menyambungkan setiap komponen yang telah dirancang sebelumnya secara nyata. Alat dan bahan yang digunakan juga disesuaikan dengan yang sudah direncanakan sebelumnya.

Tahap pembuatan dilakukan dengan menyambungkan setiap komponen seperti NodeMCU, Modul GPS, Modul SIM, Baterai, dan juga LCD pada box. Berikut adalah gambar pembuatan *hardware* secara elektrik.



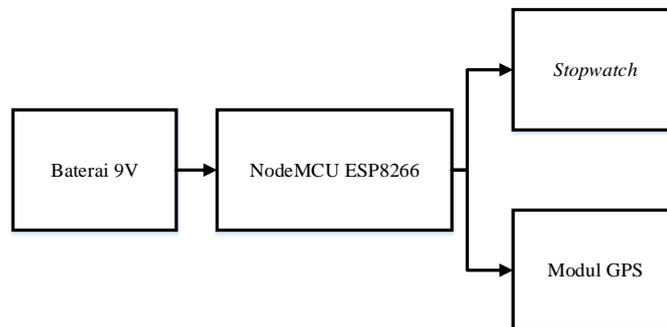
Gambar 4.4 Pengkabelan alat *tracking* posisi

4.3 Pengujian *Hardware*

Pengujian *hardware* bertujuan untuk melakukan pengecekan dalam setiap komponen yang digunakan untuk mengetahui apakah seluruh komponen yang dipakai sudah berfungsi. Pada tahap ini, dilakukan dengan menguji setiap komponen seperti Modul GPS, Modul SIM 800L, dan LCD ST7789.

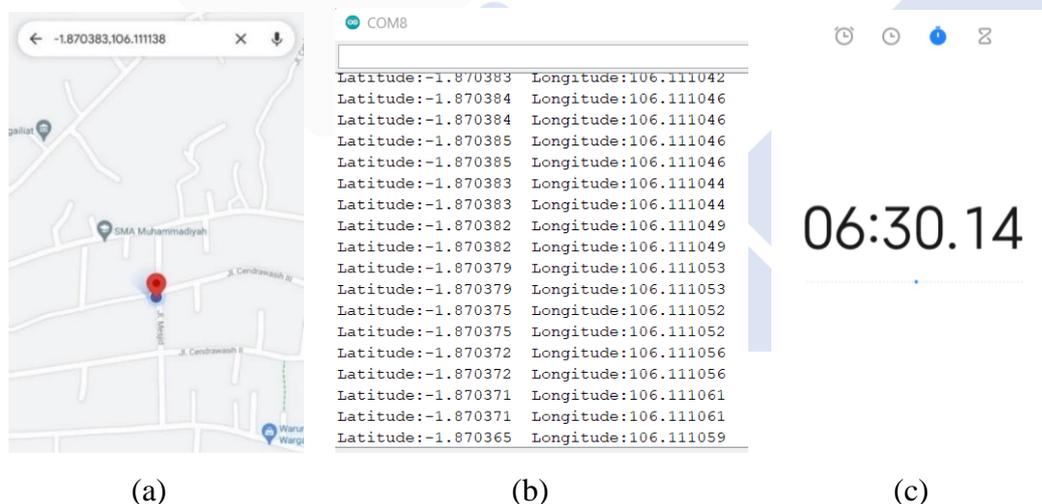
4.3.1 Pengujian Modul GPS

Pengujian modul GPS pada penelitian ini bertujuan untuk mendapatkan data titik koordinat objek yang dicari. Pengujian dilakukan dengan meletakkan alat pada lokasi yang telah diketahui titik koordinatnya. Kemudian data tersebut dibandingkan dari hasil pembacaan yang telah di dapat dari modul GPS dengan google *maps* sebagai pembacaan koordinat manual. Jika data koordinatnya telah sesuai dengan data koordinat manual, maka dapat dikatakan modul GPS ini telah berhasil. Jika tidak, akan dilakukan konfigurasi ulang terhadap alat yang dibuat. Dalam pengujian ini ada beberapa macam pengujian yaitu menguji modul GPS di luar ruangan, di dalam ruangan, dan di dalam gedung. Gambar blok rangkaian pengujian modul GPS dapat dilihat pada gambar berikut.



Gambar 4.5 Blok Pengujian Modul GPS

Pengujian modul GPS ini juga dilakukan untuk melihat kecepatan modul GPS dalam menangkap sinyal dari satelit menggunakan *stopwatch* dan apakah data koordinat dari modul GPS sudah muncul pada serial monitor yang akan berubah secara berkala. Ketika melakukan pengujian, GPS sebaiknya dilakukan di tempat yang terbuka atau di luar ruangan karena modul ini harus mendapatkan atau terhubung dengan minimal 3 satelit untuk mendapatkan data koordinatnya. Pengujian pada modul GPS dapat dilihat pada gambar berikut.



Gambar 4.6 (a) Tampilan Titik Koordinat pada Google Maps, (b) Tampilan Titik Koordinat Modul GPS pada Serial Monitor, (c) Delay Waktu Aktif Modul GPS

Hasil pengujian modul GPS dapat dilihat pada tabel berikut.

Tabel 4.1 Pengujian Modul GPS

Keterangan lokasi alat <i>tracking</i>	Delay GPS (<i>stopwatch</i>)	Koordinat alat <i>tracking</i>	Koordinat alat <i>tracking</i> (Google Maps)	Persentase Error
Luar Ruangan	6 menit 30 detik	-1.870360, 106.111206	-1.870366, 106.111151	0,1%
Dalam Ruangan	8 menit 15 detik	-1.870329, 106.111128	-1.870326, 106.111121	0,1%
Dalam Gedung	∞	-	-	100%

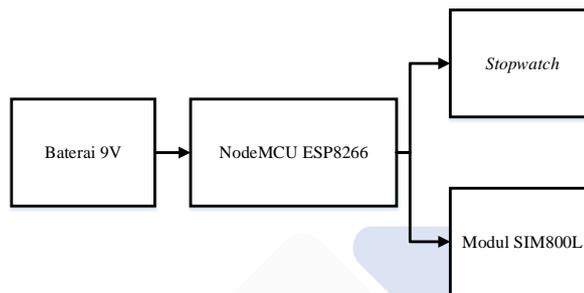
Kesimpulan :

Berdasarkan data hasil pengujian modul GPS yang telah dilakukan, dapat dilihat bahwa modul GPS berhasil mendapatkan sinyal dari satelit dan titik koordinat yang ditangkap oleh GPS memiliki nilai yang tidak jauh berbeda dengan titik koordinat yang dilakukan secara manual di *google maps*. Waktu yang diperlukan modul GPS untuk menangkap sinyal dari satelit yaitu sekitar 6 sampai 8 menit, hal ini dikarenakan adanya pengaruh dari lokasi keberadaan modul yang dapat memengaruhi kecepatan penangkapan sinyal satelit. Modul GPS akan lebih cepat menangkap sinyal satelit apabila dilakukan di luar ruangan dari pada dalam ruangan karena akan memakan waktu yang cukup lama untuk mendapatkan sinyal dikarenakan modul GPS ini sulit menerima sinyal dari satelit. Sedangkan di dalam gedung modul GPS sama sekali tidak bisa mendapat sinyal dari satelit dikarenakan sinyal satelit tidak dapat menembus benda yang tebal seperti gedung.

4.3.2 Pengujian Modul SIM800L

Pengujian modul SIM800L pada penelitian ini bertujuan untuk mendapatkan data titik koordinat objek yang dicari berupa *link google maps* melalui pesan teks. Untuk mengetahui keberadaan alat *tracking* melalui pesan teks, pengguna harus mengetik sebuah kode yaitu "FIND" yang sudah diprogram sebelumnya. Pengujian dilakukan dengan meletakkan alat pada lokasi yang telah diketahui titik koordinatnya. Kemudian data tersebut dibandingkan dari hasil *link google maps* yang diterima dengan *google*

maps sebagai pembacaan koordinat manual. Jika data koordinatnya telah sesuai dengan data koordinat manual, maka dapat dikatakan pengujian ini telah berhasil. Jika tidak, akan dilakukan pengiriman kode ulang terhadap alat yang dibuat. Gambar blok rangkaian pengujian modul SIM800L dapat dilihat pada gambar berikut.



Gambar 4.7 Blok Pengujian Modul SIM800L

Pengujian pada modul SIM 800L dapat dilihat pada gambar berikut.



Gambar 4.8 (a) Tampilan Pengiriman *Link Google Maps* pada SMS, (b) Tampilan Pengiriman *Link Google Maps* pada Serial Monitor, (c) *Delay Waktu* Pengiriman *Link Google Maps*

Pengujian modul SIM800L dilakukan sebanyak tiga kali untuk melihat apakah modul SIM berhasil mengirim pesan teks berupa *link* koordinat yang sama dengan koordinat yang ada di serial monitor dan menghitung kecepatan modul SIM800L dalam mengirim pesan teks menggunakan *stopwatch*. Hasil pengujian modul SIM800L dapat dilihat

pada tabel berikut.

Tabel 4.2 Pengujian Modul SIM800L

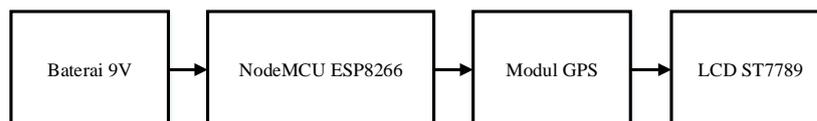
Pengujian SMS	Waktu Pengiriman (<i>stopwatch</i>)	Link Pesan Teks	Output <i>link</i> (Google Maps)	Keterangan
SMS Link 1	24 detik	http://www.google.com/maps/place/-1.870338,106.111126	-1.870338, 106.111126	Terkirim
SMS Link 2	25 detik	http://www.google.com/maps/place/-1.870316,106.111105	-1.870316, 106.111105	Terkirim
SMS Link 3	24 detik	http://www.google.com/maps/place/-1.870310,106.111091	-1.870310, 106.111091	Terkirim

Kesimpulan :

Berdasarkan data hasil pengujian modul SIM800L yang telah dilakukan, dapat dilihat bahwa modul SIM800L berhasil mengirim *link* yang dikirim ke pesan teks sama dengan koordinat yang muncul di serial monitor. Modul ini membutuhkan waktu sekitar 24 sampai 25 detik untuk mengirimkan pesan ke *handphone* pengguna, hal ini dikarenakan adanya pengaruh dari sinyal yang dapat memengaruhi kecepatan pengiriman pesan kepada pengguna. Semakin bagus sinyal, maka semakin cepat pesan tersebut terkirim kepada pengguna.

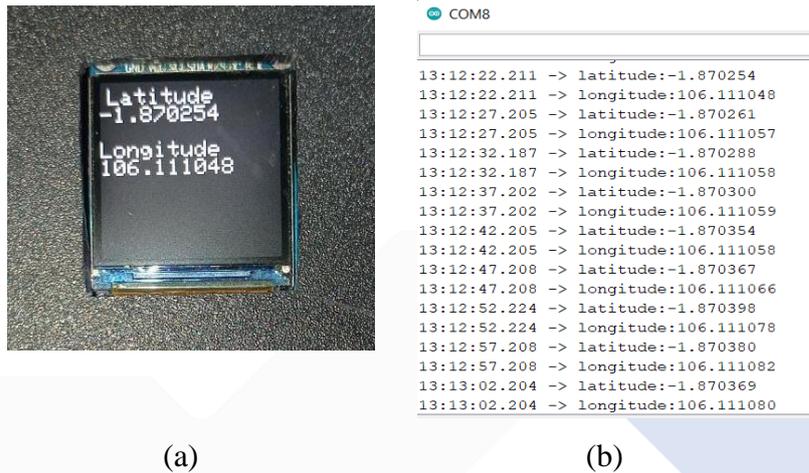
4.3.3 Pengujian LCD ST7789

Pengujian LCD ST7789 pada penelitian ini bertujuan untuk menampilkan data titik koordinat objek. Gambar blok rangkaian pengujian LCD ST7789 dapat dilihat pada gambar berikut.



Gambar 4.9 Pengujian LCD ST7789

Pengujian dari LCD ST7789 pada penelitian ini adalah untuk memperoleh informasi koordinat yang dikirim dari modul GPS yang kemudian data titik koordinatnya akan ditampilkan pada LCD. LCD hanya akan menampilkan data koordinat apabila modul GPS telah menerima data koordinat dari satelit. Pengujian pada LCD ST7789 dapat dilihat pada gambar berikut.



Gambar 4.10 (a) Tampilan Titik Kooordinat pada LCD ST7789, (b) Tampilan Titik Koordinat pada Serial Monitor

Hasil pengujian LCD ST7789 dapat dilihat pada tabel berikut.

Tabel 4.3 Pengujian LCD ST7789

Pengujian	Tampilan pada LCD	Tampilan Serial Monitor	Keterangan	Persentase Error
Pengujian ke-1	-1.870254, 106.111048	-1.870254, 106.111048	Sesuai	0 %
Pengujian ke-2	-1.870261, 106.111057	-1.870261, 106.111057	Sesuai	0 %
Pengujian ke-3	-1.870288, 106.111058	-1.870288, 106.111058	Sesuai	0 %

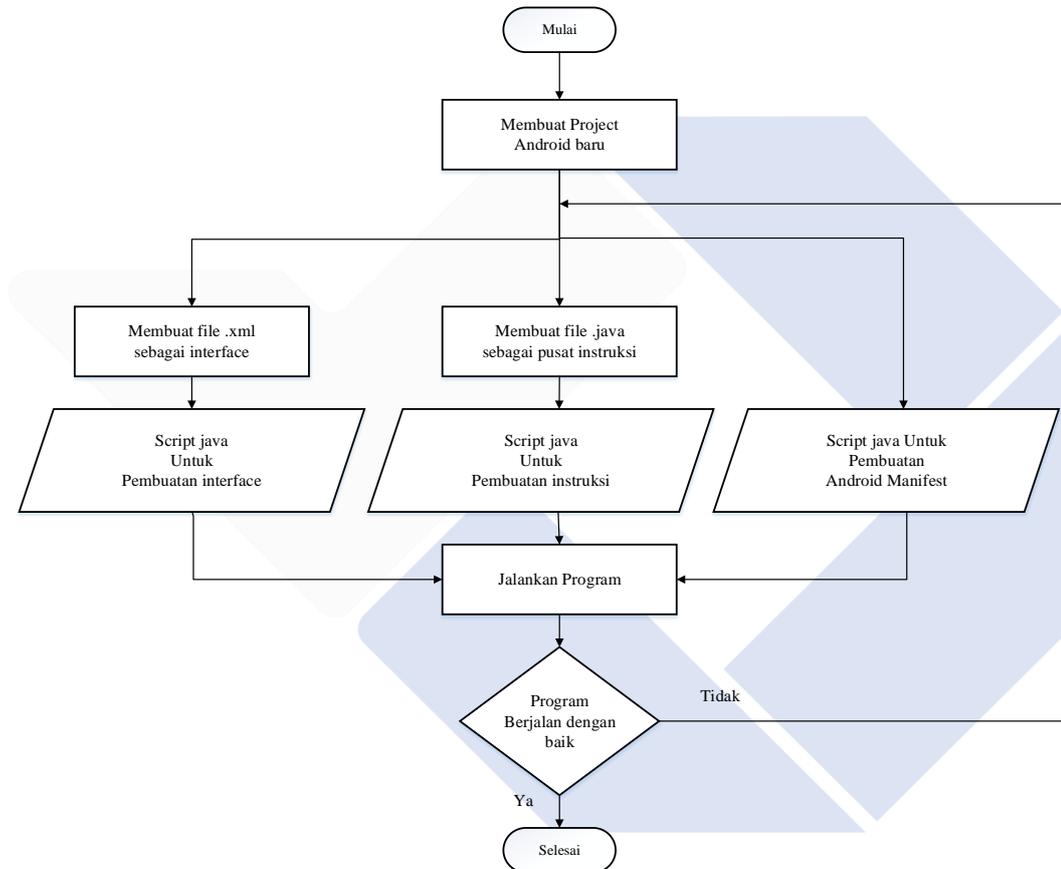
Kesimpulan :

Berdasarkan data hasil pengujian LCD ST7789 yang telah dilakukan, dapat dilihat bahwa LCD berhasil menampilkan titik koordinat yang sama dengan titik koordinat

yang muncul di serial monitor.

4.4 Perancangan dan Pembuatan *Software*

Tahap perancangan dan pembuatan *software* ini akan dilakukan dengan membuat rancangan aplikasi pada *smartphone*, rancangan dan pembuatan aplikasi ini dilakukan dengan menggunakan *software* Android Studio. Berikut adalah *flowchart* perancangan dan pembuatan aplikasi *Tracking* Posisi Objek.



Gambar 4.11 Perancangan pembuatan aplikasi *tracking* posisi objek pada android studio

4.4.1 Perancangan Aplikasi pada Smartphone

Perancangan ini dilakukan dengan membuat desain *interface* dari aplikasi *Tracking* Posisi Objek. Berikut adalah gambar rancangan yang telah dibuat.



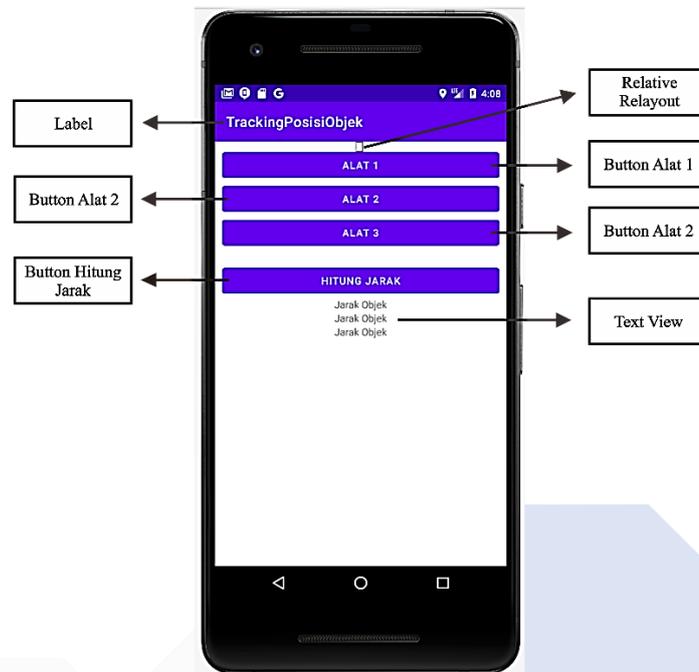
Gambar 4.12 Perancangan *Interface* Aplikasi *tracking* posisi objek

4.4.2 Pembuatan Aplikasi pada Smartphone

Pada tahap ini, dilakukan dengan membuat aplikasi yang telah dirancang sebelumnya secara nyata. Pembuatan aplikasi *Tracking* Posisi Objek ini bertujuan untuk memonitoring alat *tracking* yang ingin diketahui keberadaannya. Tampilan aplikasi menggunakan Android Studio ini dibuat dalam 2 tampilan yaitu, tampilan menu utama dan tampilan *maps*. Berikut tampilan yang telah dibuat menggunakan *software* Android Studio.

1. Menu Utama

Menu utama adalah menu awal saat aplikasi dibuka. Menu ini berisi sistem kontrol untuk melacak atau memonitoring keberadaan alat *tracking* dan mengetahui jarak alat *tracking* tersebut dengan pengguna. Pada menu ini pengguna dapat memilih untuk mengetahui alat *tracking* yang ingin diketahui keberadaannya serta mengetahui jaraknya.



Gambar 4.13 Tampilan Menu Utama Aplikasi *Tracking* Posisi Objek pada *Software* Android Studio

Berikut keterangan dan interface menu utama pada Aplikasi Tracking Objek.

- Label*, berfungsi sebagai penampil teks dalam aplikasi. Label merupakan tampilan dari judul program.
- Relative Layout*, merupakan program yang berfungsi mengatur tata letak tombol pada aplikasi di dalam program *relative layout* juga diatur jarak antar tombol satu dengan tombol yang lain sehingga tampilan pada aplikasi menjadi rapi dan teratur.
- Button Alat 1*, berfungsi sebagai pengganti layer dari menu utama ke menu maps yang digunakan untuk menampilkan *waypoint* beserta alamat dari objek1. Adapun program pembuatan "*Button Alat 1*" adalah sebagai berikut.

<**Button**

```

android:id="@+id/btn_second"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="ALAT 1"/>
```

- d. *Button Alat 2*, berfungsi sebagai pengganti layer dari menu utama ke menu maps yang digunakan untuk menampilkan *waypoint* beserta alamat dari objek2. Adapun program pembuatan "*Button Alat 2*" adalah sebagai berikut.

```
<Button
    android:id="@+id/btn_third"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ALAT 2"/>
```

- e. *Button Alat 3*, berfungsi sebagai pengganti layer dari menu utama ke menu maps yang digunakan untuk menampilkan *waypoint* beserta alamat dari objek3. Adapun program pembuatan "*Button Alat 3*" adalah sebagai berikut.

```
<Button
    android:id="@+id/btn_fourth"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ALAT 3"/>
```

- f. *Button Hitung Jarak*, berfungsi sebagai penghitung jarak alat 1 dengan pengguna, alat 2 dengan pengguna, dan alat 3 dengan pengguna. Adapun program pembuatan "*Button Hitung Jarak*" adalah sebagai berikut.

```
<Button
    android:id="@+id/btn_jarak"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Hitung jarak" />
```

- g. *Text View*, berfungsi untuk menampilkan hasil dari perhitungan jarak antara objek dan pengguna. Adapun program pembuatan "*Text View*" adalah sebagai berikut.

```
<TextView
    android:id="@+id/jarak_view1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Jarak Objek"
    android:textAlignment="center"/>
```

Tampilan pada menu utama pada *smartphone* dapat dilihat dari gambar berikut.



Gambar 4.14 Tampilan Menu Utama Aplikasi *Tracking* Posisi Objek pada *Smartphone*

2. Menu *Maps*

Menu *maps* adalah menu yang berisi tampilan peta yang di mana pada peta tersebut menampilkan *waypoint* keberadaan alat *tracking* dan juga menampilkan alamat dari alat *tracking* yang ingin diketahui keberadaannya. Berikut adalah tampilan menu *maps* pada *software* Android Studio dan *smartphone*.



Gambar 4.15 Tampilan Menu *Maps* pada *software* Android Studio dan *Smartphone*

Berikut keterangan dari tampilan kedua pada *software* Android Studio dan aplikasi.

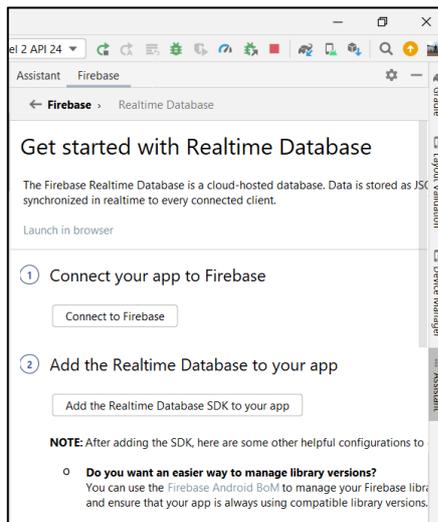
- a. *Addresses*, merupakan tampilan dari alamat yang sudah diterima dari google geocoding API.
- b. *Waypoint*, merupakan tampilan dengan gambar pin yang menunjukkan keberadaan objek yang sudah berhasil diketahui keberadaannya.
- c. *Drawable.redcar*, merupakan tampilan dengan gambar mobil yang menunjukkan posisi pengguna.

4.4.3 Komunikasi NodeMCU ESP8266 dengan *Realtime Database* Firebase

Interface dapat diartikan sebagai tampilan pada aplikasi dengan tujuan pengguna dapat terhubung dengan suatu system. *Interface* ini di desain dengan *software* Android studio yang berfungsi memudahkan pengguna dalam melacak keberadaan objek. Agar *interface* berjalan dengan baik maka dibuatlah komunikasi antara NodeMCU dengan *realtime database* firebase yaitu dengan membuat program terlebih dahulu agar NodeMCU dapat mentransfer data koordinat ke *realtime database* firebase. Firebase berfungsi sebagai penyimpan data yang akan dikirimkan NodeMCU. Pemrograman ini dapat dilihat pada lembar lampiran.

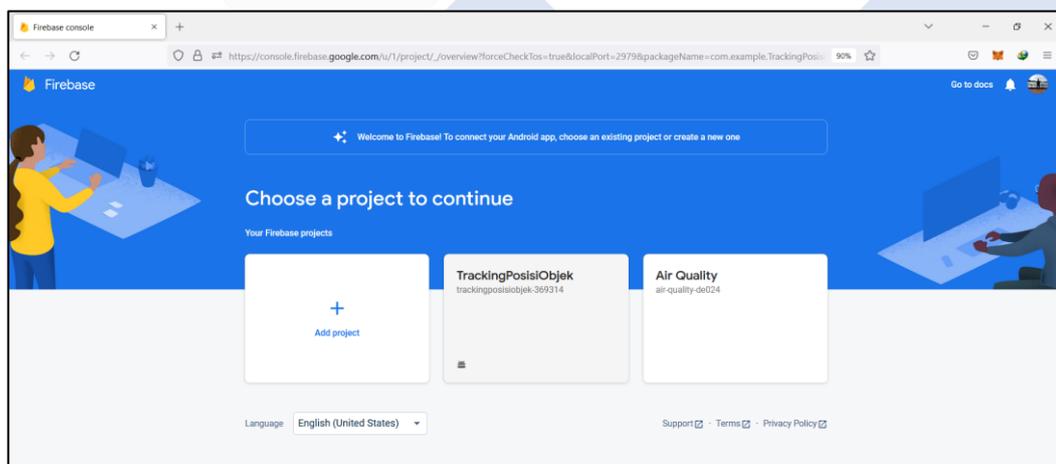
4.4.4 Komunikasi *software* Android Studio dengan *Realtime Database* Firebase

Realtime database firebase memiliki fungsi penyimpanan data yang akan digunakan menyimpan data koordinat dari nodeMCU. Titik koordinat yang diterima aplikasi akan diubah kedalam bentuk tampilan *waypoint* keberadaan dari alat *tracking* beserta alamatnya. Tampilan awal untuk menghubungkan aplikasi dengan firebase adalah dengan meng-klik *Connect to Firebase*.



Gambar 4.16 Tampilan Koneksi *software* Android Studio dengan *Realtime database* firebase

Setelah meng-klik *Connect to Firebase* maka akan diarahkan ke *website* google firebase dan selanjutnya pilih *database* yang sudah terhubung dengan NodeMCU untuk menerima data titik koordinat objek.



Gambar 4.17 Tampilan *realtime database* firebase

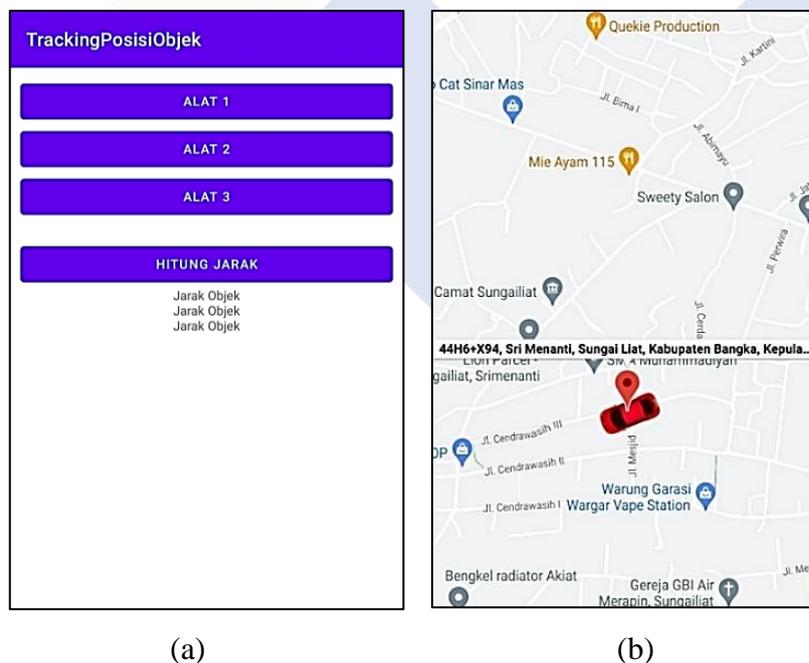
Setelah aplikasi Tracking Posisi Objek terhubung dengan *realtime database* firebase maka aplikasi dapat meminta request ke *realtime database* firebase untuk menggunakan data titik koordinat yang tersimpan pada *realtime database* untuk menampilkan *waypoint* dan alamat.

4.5 Pengujian Aplikasi *Tracking* Posisi Objek

Pengujian aplikasi “Tracking Posisi Objek” dilakukan dengan *install* aplikasi yang telah dibuat. Pengujian dilakukan pada setiap *button* yang dibuat, yaitu menampilkan *waypoint* dari masing-masing alat *tracking* serta menampilkan alamat dan jaraknya. Adapun untuk melakukan pengujian aplikasi dilakukan langkah-langkah berikut.

1. Aktifkan akses lokasi pada perangkat *smartphone*
2. Buka aplikasi *Tracking* Posisi Objek
3. Klik tombol Alat 1 untuk mengetahui keberadaan alat *tracking* 1
4. Klik tombol Alat 2 untuk mengetahui keberadaan alat *tracking* 2
5. Klik tombol Alat 3 untuk mengetahui keberadaan alat *tracking* 3
6. Klik *waypoint* yang terdapat pada maps aplikasi untuk mengetahui alamat dari alat *tracking*
7. Klik tombol Hitung Jarak untuk mengetahui jarak antara pengguna dengan alat *tracking*

Tampilan pengujian aplikasi dapat dilihat pada gambar berikut.



Gambar 4.18 (a) Tampilan Menu Utama Aplikasi, (b) Tampilan Menu *Maps* Aplikasi

Hasil pengujian aplikasi dapat dilihat pada tabel berikut.

Tabel 4.4 Pengujian Aplikasi

Pengujian Tombol Aplikasi	Tampilan <i>Waypoint</i> dan Alamat pada <i>Smartphone</i>	Keterangan (Berhasil / Tidak Berhasil)
Tombol Alat 1	Menampilkan <i>waypoint</i> dan alamat alat <i>tracking</i> 1	Berhasil
Tombol Alat 2	Menampilkan <i>waypoint</i> dan alamat alat <i>tracking</i> 2	Berhasil
Tombol Alat 3	Menampilkan <i>waypoint</i> dan alamat alat <i>tracking</i> 3	Berhasil
Tombol Hitung Jarak	Menampilkan jarak antara alat <i>tracking</i> dengan pengguna	Berhasil

Kesimpulan :

Berdasarkan data hasil pengujian aplikasi yang telah dilakukan, dapat disimpulkan bahwa aplikasi berhasil merapkan metode *reverse geocoding* dan tombol pada aplikasi berhasil menampilkan *waypoint* keberadaan alat *tracking* beserta alamat dan jaraknya.

4.6 Pengujian Keseluruhan *Tracking* Posisi Objek dengan menggunakan Metode *Reverse Geocoding*

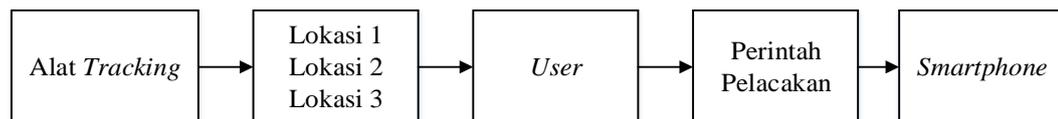
Pengujian secara keseluruhan ini bertujuan untuk mengetahui dan menganalisa kinerja alat *tracking* dan aplikasi apakah dapat bekerja dengan baik. Alat *tracking* ini terdiri dari nodeMCU, modul GPS, modul SIM, dan LCD. Pengujian dilakukan dengan menekan masing-masing tombol yang terdapat pada aplikasi yang kemudian akan menampilkan masing-masing keberadaan alat *tracking* dan jaraknya.

4.6.1 Pengujian Pelacakan Aplikasi *Tracking* Posisi Objek dari Beberapa Lokasi yang Berbeda

Pengujian ini bertujuan untuk mengetahui apakah aplikasi dapat menerima lokasi dari alat *tracking* yang telah dibuat. Pengujian dilakukan di 3 lokasi yang berbeda pada masing-masing alat *tracking* kemudian membandingkan alamatnya dengan alamat yang tertera google maps, pengujian ini bertujuan untuk menganalisa ketepatan alamat yang dihasilkan oleh aplikasi terhadap alamat sebenarnya dari google maps. Pada saat pengujian, alat *tracking* sebaiknya diaktifkan di luar ruangan. Berikut adalah langkah-langkah dalam melakukan pengujian.

1. Hidupkan tombol on/off pada alat *tracking* 1, 2, dan 3
2. Amati koordinat yang terdapat pada alat *tracking* 1, 2, dan 3
3. Aktifkan akses lokasi pada perangkat *smartphone*
4. Buka aplikasi *Tracking* Posisi Objek menggunakan Metode *Reverse Geocoding*
5. Klik salah satu tombol alat 1, alat 2, alat 3 untuk mengetahui keberadaan dari masing-masing alat *tracking*
6. Klik *waypoint* yang terdapat pada *maps* aplikasi
7. Bandingkan alamat dan titik koordinat yang didapat dari alat *tracking* dengan alamat dan titik koordinat yang tertera di *Google Maps*
8. Bandingkan juga alamat yang terdapat pada aplikasi yang menggunakan metode *reverse geocoding* dengan aplikasi yang tidak menggunakan metode *reverse geocoding*

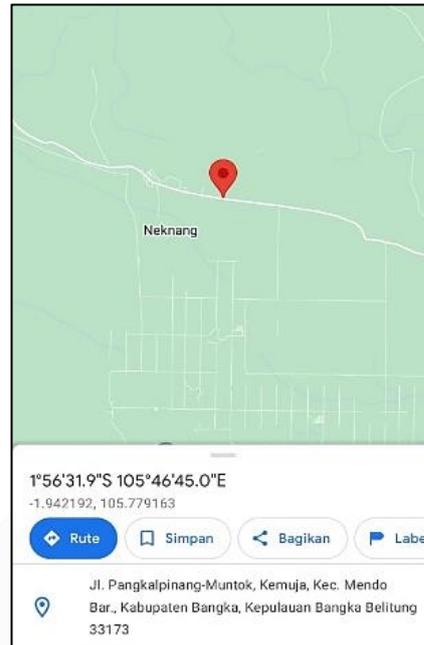
Berikut ini merupakan blok diagram pengujian yang ditunjukkan pada gambar 4.19 dan juga pengujian yang ditunjukkan pada gambar 4.20 dan 4.21



Gambar 4.19 Blok Diagram Pengujian Pelacakan Alat *Tracking*



(a)



(b)

Gambar 4.20 (a) Koordinat pada alat *tracking*, (b) Koordinat pada *Google Maps*



(a)



(b)

Gambar 4.20 (a) Pengujian menggunakan Metode *Reverse Geocoding*, (b) Pengujian Tanpa menggunakan Metode *Reverse Geocoding*

Tabel 4.5 Pengujian Pelacakan pada Alat *Tracking* 1

Delay GPS (<i>stopwatch</i>)	Koordinat alat <i>tracking</i>	Koordinat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi)	Alamat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi tanpa metode reverse geocoding)
20 menit 4 detik	-1.942192, 105.779163	-1.942190, 105.779165	3Q5H+5R X Jl. Pangkalpinang-Muntok, Kemuja, kec. Mendo Bar, Kab. Bangka	Jl. Pangkalpinang-Muntok, Kemuja, kec. Mendo Bar, Kab. Bangka	Objek1
19 menit 22 detik	-1.881419, 106.104343	-1.881435, 106.104341	4493+7Q C, Jl. Jend. A. Yani, Parit Padang, Sungailiat, Kab. Bangka	4493 7QC, Jl. Jend. A. Yani, Parit Padang, Sungailiat, Kab. Bangka	Objek1
10 menit 6 detik	-1.884610, 106.113465	-1.884600, 106.113464	4487+599, Jl. Jendral Sudirman, Parit Padang, Sungailiat, Kab. Bangka	4487 599, Jl. Jendral Sudirman, Parit Padang, Sungailiat, Kab. Bangka	Objek1

Tabel 4.6 Pengujian Pelacakan pada Alat *Tracking 2*

Delay GPS (<i>stopwatch</i>)	Koordinat alat <i>tracking</i>	Koordinat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi)	Alamat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi tanpa metode)
5 menit 20 detik	-1.883268, 106.113311	-1.883270, 106.113315	Jl. Jenderal Sudirman No. 6, Parit Padang, Sungailiat, Kabupaten Bangka	Jl. Jenderal Sudirman No. 6, Parit Padang, Sungailiat, Kabupaten Bangka	Objek2
6 menit 42 detik	-1.624648, 105.773001	-1.624652, 105.773010	Jl. Kp. Sunda No.16, Bukit Ketok, Belinyu, Kabupaten Bangka	Jl. Kp. Sunda No.16, Bukit Ketok, Belinyu, Kabupaten Bangka	Objek2
4 menit 22 detik	-1.872416, 106.11286	-1.872405, 106.11292	44G7+X4V, Jl. Cendrawasih I, Sri Menanti, Sungailiat, Kabupaten Bangka	44G7+X4V, Jl. Cendrawasih I, Sri Menanti, Sungailiat, Kabupaten Bangka	Objek2

Tabel 4.7 Pengujian Pelacakan pada Alat *Tracking 3*

Delay GPS (<i>stopwatch</i>)	Koordinat alat <i>tracking</i>	Koordinat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi)	Alamat alat <i>tracking</i> (Google Maps)	Alamat alat <i>tracking</i> (Aplikasi tanpa metode)
15 menit 2 detik	-1.853345, 106.124073	-1.853340, 106.124065	Jl. Nangnung Utara	Jl. Nangnung Utara	Objek3

			Sungailiat Bangka No. 372, Sungailiat, Kabupaten Bangka	Sungailiat Bangka No. 372, Sungailiat, Kabupaten Bangka	
9 menit 36 detik	-1.852340, 106.131928	-1.852336, 106.131936	Kawasan Industri Air Kantung, Sungailiat, Kabupaten Bangka	Kawasan Industri Air Kantung, Sungailiat, Kabupaten Bangka	Objek3
6 menit 2 detik	-1.851349, 106.118142	-1.851353, 106.118140	44X9+F7R Tugu Dulang, Sungailiat, Kabupaten Bangka	44X9 F7R Tugu Dulang, Sungailiat, Kabupaten Bangka	Objek3

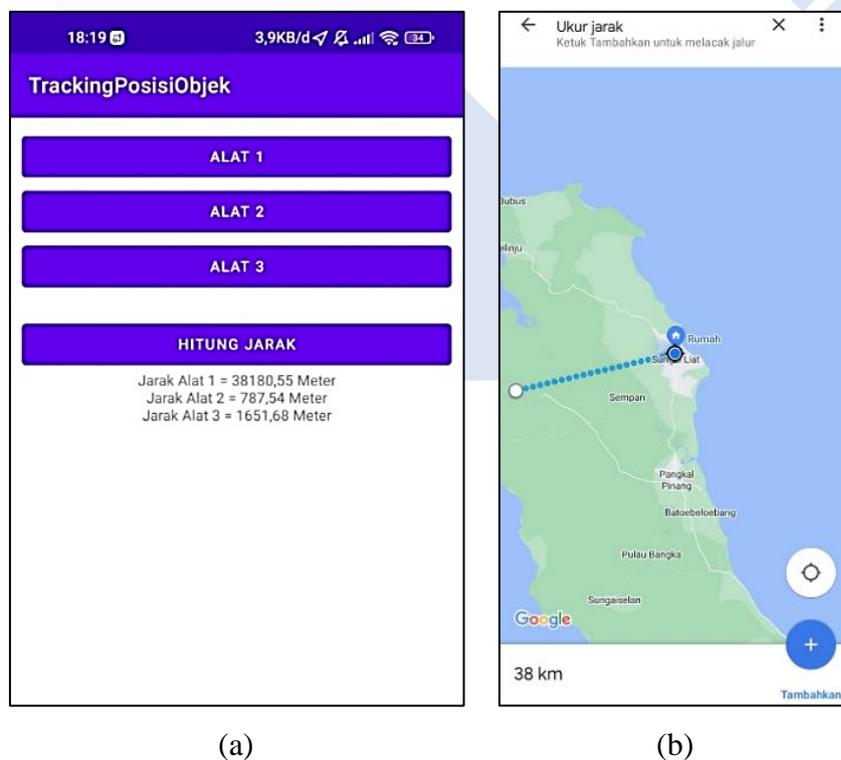
Kesimpulan :

Dari hasil pengujian di atas dapat dilihat bahwa alat *tracking* dapat mengirim koordinat ke *database* dan aplikasi dapat menerima koordinat dari alat *tracking* dalam bentuk *waypoint*, alamat, dan jarak. Titik koordinat dan alamat yang dihasilkan dari alat *tracking* hampir sama dengan titik koordinat dan alamat yang diuji secara manual di *google maps*. Perbedaan aplikasi yang menggunakan metode *reverse geocoding* dan tanpa menggunakan metode *reverse geocoding* yaitu aplikasi yang menggunakan metode *reverse geocoding* dapat menampilkan alamat keberadaan alat *tracking*, sedangkan tanpa menggunakan metode *reverse geocoding* hanya menampilkan keterangan Objek 1, Objek 2, atau Objek 3 saja. Metode *reverse geocoding* ini dapat mempermudah pengguna memperoleh informasi.

4.6.2 Pengujian Jarak pada Aplikasi *Tracking* Posisi Objek dengan Google Maps

Pengujian ini dilakukan sama dengan pelacakan alat *tracking* yaitu dengan melihat jarak dari aplikasi dan membandingkannya dengan jarak yang tertera google maps, pengujian ini bertujuan untuk menganalisa ketepatan jarak yang dihasilkan aplikasi terhadap jarak sebenarnya dari google maps. Berikut adalah langkah-langkah dalam melakukan pengujian.

1. Hidupkan tombol on/off pada alat *tracking* 1, 2, dan 3
2. Aktifkan akses lokasi pada perangkat *smartphone*
3. Buka aplikasi *Tracking* Posisi Objek
4. Klik tombol hitung jarak, untuk mengetahui jarak antara pengguna dan alat *tracking* posisi
5. Amati jarak yang diperoleh
6. Bandingkan jarak yang didapat dari alat *tracking* dengan jarak yang tertera di *Google Maps*



Gambar 4.21 (a) Tampilan Jarak pada Aplikasi, (b) Pengukuran Jarak pada Google Maps

Tabel 4.8 Pengujian Jarak pada Alat *Tracking* 1

Jarak (Aplikasi)	Jarak (Google Maps)	Persentase <i>Error</i>
38177,64 m	38 km	0.1%
1481,88 m	1,4 km	5.7%
1631,47 m	1,6 km	1.9%

Tabel 4.9 Pengujian Jarak pada Alat *Tracking* 2

Jarak (Aplikasi)	Jarak (Google Maps)	Persentase <i>Error</i>
3528,81 m	3.5 m	0.8%
46555,67 m	47 km	1%
305,09 m	305 m	0.02%

Tabel 4.10 Pengujian Jarak pada Alat *Tracking* 3

Jarak (Aplikasi)	Jarak (Google Maps)	Persentase <i>Error</i>
2339,93 m	2,4 km	2.5%
7,68 m	7 m	9%
1545,38 m	1,5 km	3%

Rumus perhitungan persentase error:

$$\left| \frac{\text{Nilai eksperimen} - \text{Nilai teoritis}}{\text{Nilai teoritis}} \right| \times 100\%$$

Ket:

Nilai eksperimen = Jarak pada aplikasi

Nilai teoritis = Jarak pada google maps

- Perhitungan persentase *error* pada tabel

$$\text{Persentase Error} = \left| \frac{38-38}{38} \right| \times 100\% = 0\%$$

Kesimpulan :

Dari hasil pengujian yang telah dilakukan untuk jarak yang jauh, dapat dilihat bahwa jarak yang ditampilkan oleh aplikasi *tracking* posisi objek memiliki perbedaan jarak pada *google maps* dengan persentase *error* tertinggi sebesar 9% dan persentase *error* terendah sebesar 0.02%. Persentase *error* tertinggi di dapat ketika melakukan pengujian dengan jarak yang terlalu dekat antara alat *tracking* dengan pengguna. Ketika melakukan pengujian pada jarak 1 meter antara pengguna dengan alat *tracking*, aplikasi menampilkan hasil sejauh 7 meter begitu juga dengan *google maps*. Dapat disimpulkan bahwa toleransi jarak pada alat *tracking* ini adalah sebesar 7 meter.

4.6.3 Pengujian Pengiriman Lokasi Alat *Tracking* melalui Pesan Teks

Pengujian ini dilakukan secara bertahap yaitu di mulai dari alat 1, alat 2, dan alat 3. Pengujian ini bertujuan untuk menganalisa respon modul sms ketika mendapatkan perintah untuk mengirimkan lokasi objek. Pada pengujian ini pengguna memberikan perintah berupa pesan teks, kemudian pesan teks tersebut akan diterima alat sehingga membuat alat mengirimkan pesan teks kembali berupa *link* *google maps* lokasi keberadaan objek. Berikut adalah langkah-langkah dalam melakukan pengujian.

1. Hidupkan tombol on/off pada alat *tracking* 1, 2, dan 3
2. Kirim pesan teks berupa kata kunci “*FIND*”
3. Tunggu hingga alat mengirimkan *link google maps*
4. Klik *link google maps*
5. Amati posisi objek yang muncul pada *google maps*

Tabel 4.11 Pengujian Pengiriman Lokasi pada Alat *Tracking* 1

Pengujian SMS	Waktu Pengiriman (<i>stopwatch</i>)	<i>Link</i> Pesan Teks	Keterangan
SMS Link 1	24 detik	http://www.google.com/maps/place/-1.942195,105.779160	Terkirim
SMS Link 2	24 detik	http://www.google.com/maps	Terkirim

SMS Link 3	25 detik	/place/-1.881430, 106.104339 http://www.google.com/maps /place/-1.884611, 106.113463	Terkirim
------------	----------	---	----------

Tabel 4.12 Pengujian Pengiriman Lokasi pada Alat *Tracking* 2

Pengujian SMS	Waktu Pengiriman (<i>stopwatch</i>)	Link Pesan Teks	Keterangan
SMS Link 1	18 detik	http://www.google.com/maps /place/-1.883265, 106.113309	Terkirim
SMS Link 2	25 detik	http://www.google.com/maps /place/-1.624648, 105.773001	Terkirim
SMS Link 3	25 detik	http://www.google.com/maps /place/-1.872390, 106.112781	Terkirim

Tabel 4.13 Pengujian Pengiriman Lokasi pada Alat *Tracking* 3

Pengujian SMS	Waktu Pengiriman (<i>stopwatch</i>)	Link Pesan Teks	Keterangan
SMS Link 1	24 detik	http://www.google.com/maps /place/-1.853343, 106.124069	Terkirim
SMS Link 2	19 detik	http://www.google.com/maps /place/-1.852345, 106.131930	Terkirim
SMS Link 3	25 detik	http://www.google.com/maps /place/-1.851349, 106.118142	Terkirim

Kesimpulan :

Dari hasil pengujian yang telah dilakukan, dapat dilihat bahwa waktu yang dibutuhkan untuk pengiriman *link* keberadaan alat *tracking* adalah sekitar 18-25 detik dan *link* yang dikirim oleh alat *tracking* telah sesuai dengan keberadaan alat *tracking* yang sebenarnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan dengan melihat posisi masing-masing dari 3 alat *tracking* posisi melalui aplikasi *tracking* posisi objek dengan menggunakan metode *reverse geocoding* diperoleh kesimpulan sebagai berikut.

1. Aplikasi yang menggunakan metode *reverse geocoding* dapat menampilkan alamat dari alat *tracking* dengan persentase keberhasilan 100%. Sedangkan aplikasi tanpa metode *reverse geocoding* hanya menampilkan lokasi alat *tracking* saja tanpa menampilkan alamatnya.
2. Hasil pengukuran jarak yang dilakukan alat *tracking* posisi memiliki keakurasian 99% dengan validasi pengukuran jarak oleh google maps.
3. Dari hasil pengujian, di dapatkan bahwa alat *tracking* ini hanya bisa digunakan apabila terkoneksi dengan internet dan terhubung dengan sinyal GPS dari satelit. Jika salah satu tidak terkoneksi, maka alat ini tidak bisa digunakan.

5.2 Saran

Proyek akhir yang telah dibuat masih terdapat beberapa kekurangan yang bisa dijadikan sebagai saran untuk pengembangan proyek akhir selanjutnya yaitu:

1. Alat *tracking* dibuat dengan ukuran yang lebih kecil agar dapat ditempatkan pada objek yang kecil.
2. Melakukan penelitian lanjutan untuk memperkuat penangkapan sinyal GPS pada ruangan yang tertutup.

DAFTAR PUSTAKA

- [1] M. Ridha Fahlivi, "Sistem Tracking Position Berdasarkan Titik Koordinat GPS Menggunakan Smartphone," *Jurnal Infomedia*, vol. 2, no. 1, 2017.
- [2] Kurniawan Iqbal Dwiki, Sunaryono Dwi, dan Ahmadyah Ahatu Solchah, "Aplikasi Monitoring Keberadaan Objek Melalui Perangkat Bergerak Berbasis Android," *Jurnal Teknik Pomits*, vol. 2, hlm. 1–6, 2014.
- [3] D. A. Rahayu dan R. Kosasih, "Aplikasi Sistem Lacak Kendaraan Berbasis Android Menggunakan Arduino Uno Dan Modul SIM808," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 23, no. 1, hlm. 55–64, 2018, doi: 10.35760/tr.2018.v23i1.2451.
- [4] Budianto, M. T. Taman, dan L. Hayat, "Sistem Penjejak Posisi Kendaraan Bermotor Berbasis GPS Melalui Media SMS," *Jurnal Riset Rekayasa Elektronika*, vol. 1, 2019.
- [5] A. Wahyudi, J. Andjarwirawan, dan A. Handojo, "Perancangan Dan Pembuatan Aplikasi Tracker Position Dan Alert Pada Mobile Device Berbasis Android Yang Dapat Dipantau Via Website," 2013, [Daring]. Available: <http://maps.google.com>.
- [6] I. F. Kharisma Juda, R. R. Yacoub, F. Imansyah, J. Marpaung, N. Tjahja, dan Mooniarsih, "Rancang Bangun Tracking System Transportasi Darat Menggunakan Komunikasi GSM Dengan Interface Aplikasi Blynk," *Jurnal Teknik Elektro Universitas Tanjungpura*, vol. 2, 2022.
- [7] Y. Yuniati, M. Ulvan, M. Azzahra, J. Teknik Elektro Fakultas Teknik Universitas Lampung Jalan Soemantri Brojonegoro No, dan B. Lampung, "Implementasi Modul Global Positioning System (Gps) pada Sistem Tracking Bus Rapid Transit (BRT) Lampung Menuju Smart Transportation," *Jurnal Sains, Teknologi dan Industri*, vol. 14, no. 2, hlm. 150–156, 2017.
- [8] I. Fitriati dan N. Afriani, "Pelacakan Koordinat dengan SMS-Tracking menggunakan App My GPS Coordinate berbasis Android," 2016.
- [9] K. Kishoreanthuvan, S. Balaji, S. Thamizharasan, dan K. Kuzhandaivelu, "An Intermediate Service Composer Architecture for Dynamic Discovery and Invocation of Web Services from Mobile Devices." [Daring]. Available: www.iosrjournals.org
- [10] B. Kilic dan F. Gülgen, "Investigating the Quality of Reverse Geocoding Services Using Text Similarity Techniques and Logistic Regression

Analysis,” *Cartogr Geogr Inf Sci*, vol. 47, no. 4, hlm. 336–349, Jul 2020, doi: 10.1080/15230406.2020.1746198.

- [11] L. Rizaldi, “Aplikasi Tracking Sales Marketing Berbasis Android Menggunakan Metode Reverse-Geocoding Di PT. Sri Indah Labetama,” 2022.
- [12] A. Juansyah, “Pembangunan Aplikasi Child Tracker Berbasis Assisted-Global Positioning System (A-GPS) Dengan Platform Android,” *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 1, 2015.
- [13] F. Ong, A. Setiawan, dan N. Sepadyati, “Aplikasi Pengoptimalan Rute Pengiriman Barang pada PT.XYZ,” 2021.
- [14] A. Sukamerta, I. Gede Putu Krisna Juliharta, K. Queena Fredlina, J. Teknik Informatika, S. Primakara, dan J. Tukad Badung, “Model Sistem Informasi Pura Kawitan Berbasis Android (Studi Kasus di Provinsi Bali),” *Jutisi*, vol. 9, 2020.
- [15] Mushlihun, Ilhamsyah, dan Y. Brianorman, “Prototipe Aplikasi Pengukur Luas Lahan Wilayah Berbasis Android,” *Jurnal Coding, Sistem Komputer Untan*, vol. 4, 2016.
- [16] I. Fitriati dan N. Afriani, “Pelacakan Koordinat dengan SMS-Tracking menggunakan App My GPS Coordinate berbasis Android,” *Jurnal Tambora*, vol. 1, no. 2, hlm. 1, Jun 2016.

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : M. Irfan Bayu P
Tempat & tanggal lahir : Simpang Jelita, 29 Juli 2001
Alamat : Paket 4 PT.SMI Desa Neknang,
Bakam, Bangka
Telp : 082179625649
HP : 082179625649
Email : irfanbayu890@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 121 Jeramba	2006-2013	-
Bolong Jambi		
SMP Sultan Iskandar Muda	2013-2016	-
Medan		
SMK Negeri 1 Bakam	2016-2019	TKJ

3. Pendidikan Non Formal

Orbit AI for Gen-Z	2022	Kemendikbud - MSIB
--------------------	------	--------------------

Sungailiat, 6 Februari 2023

(M. Irfan Bayu P)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Fifi Andini
Tempat & tanggal lahir : Sungailiat, 06 Oktober 2001
Alamat : Jl. Kartini 1 No. 184 Sungailiat,
Bangka
Telp : 082175232554
HP : 082175232554
Email : fifiandini154@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

TK Negeri Pembina Sungailiat	2006-2007	-
SD Negeri 4 Sungailiat	2007-2013	-
SMP Setia Budi Sungailiat	2013-2016	-
SMK Negeri 1 Sungailiat	2016-2019	Multimedia

3. Pendidikan Non Formal

Orbit AI for Gen-Z	2022	Kemendikbud - MSIB
--------------------	------	--------------------

Sungailiat, 6 Februari 2023

(Fifi Andini)

LAMPIRAN 2

PROGRAM KESELURUHAN

- PROGRAM *NODEMCU* ESP8266

```
#include <TinyGPS++.h> //library untuk modul gps
#include <SoftwareSerial.h> //library untuk komunikasi serial
#include <Adafruit_GFX.h> // library untuk lcd tft
#include <Adafruit_ST7789.h> // library for this display
#include <SPI.h> //library untuk menangani komunikasi serial
#define TFT_CS D8 // if your display has CS pin
#define TFT_RST D2 // reset pin
#define TFT_DC D1 // data pin

Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);
SoftwareSerial SIM800L(D3,D4);
SoftwareSerial Serialgps(D5,D6);
TinyGPSPlus gps;
double latitude, longitude;
String response;
String link,linkmap;
String link1;
String kirim;
double Lat;
long Long;
double link2;
double link3;
int lastStringLength = response.length();
bool kirimsms = 1;

#include <Arduino.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Berikan info proses pembuatan token.
#include "addons/TokenHelper.h"
//Berikan info proses RTDB dan fungsi pembantu lainnya.
#include "addons/RTDBHelper.h"
```

```

// masukkan wifi yang akan digunakan
#define WIFI_SSID "hehe"
#define WIFI_PASSWORD "12345678"

// masukan API Key firebase
#define API_KEY "AIzaSyD_sw6PnvCvxs-hr2NpOitxxnhbSwvmGDE "

// masukkan RTDB URL Firebase */
#define DATABASE_URL "https://trackingposisiobjek-369314-default-
rtdb.asia-southeast1.firebaseio.com"

//Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
bool signupOK = false;

long millisBaca1;

void setup() {
  Serial.begin(9600);
  Serialgps.begin(9600);
  SIM800L.begin(9600);
  delay(1000);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Tetapkan api KEY (wajib) */
  config.api_key = API_KEY;

```

```

/* Tetapkan URL RTDB (wajib) */
config.database_url = DATABASE_URL;

/* menghubungkan ke firebase */
if (Firebase.signUp(&config, &auth, "", "")){
  Serial.println("ok");
  signupOK = true;
}
else{
  Serial.printf("%s\n", config.signer.signupError.message.c_str());
}

/* Tetapkan fungsi panggilan balik untuk proses token yang berjalan lama*/
config.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

}

void loop() {

GPS();

if(millisBaca1< millis()) {
  millisBaca1 = millis() + 3000;
  sms();
  Serial.println("jalan");
  if (SIM800L.available(>0){
    response = SIM800L.readStringUntil('\n');
    tft.init(240, 240, SPI_MODE2);
    tft.setRotation(2);
    tft.fillScreen(ST77XX_BLACK);
    tft.setCursor(10, 10);
    tft.setTextColor(ST77XX_WHITE);
    tft.setTextSize(3);
    tft.setTextWrap(true);
    tft.println(response);
    Serial.println("modul sms");
    Serial.println(response);
  }
}

```

```

if(lastStringLength != response.length()){
    //Perintah ON
    //GPS();
    if(response.indexOf("FIND")!=-1){ //keyword sms

        SIM800L.println("AT+CMGF=1"); //tetapkan modul kedalam mode sms
        delay(1000);
        SIM800L.println("AT+CMGS=\"082175232554\"\r"); //no hp yang
dituju
        delay(1000);
        SIM800L.println(linkmap);// sms yang ingin dikirim
        delay(100);
        SIM800L.println((char)26);// kode CTRL+Z untuk keluar proses sms
        delay(1000);
        Serial.print("SMSterkirim");
        tft.setTextColor(ST77XX_WHITE);
        tft.setTextSize(3);
        tft.setTextWrap(true);
        tft.println("SMS Terkirim");
    }
}
}
}
void GPS(){
    if(Serial.available()) {
        gps.encode(Serial.read());
    }
    if(gps.location.isUpdated()) {
        latitude = gps.location.lat();
        longitude = gps.location.lng();
        link = String(latitude, 6) ;
        link1 = String(longitude, 6) ;
        linkmap = "www.google.com/maps/place/" + String(latitude, 6) + "," +
String(longitude, 6) ;
        link2 = link.toDouble();
        link3 = link1.toDouble();
        Serial.println(link);
        Serial.println(link2);
        tft.init(240, 240, SPI_MODE2);
        tft.setRotation(2);
        tft.fillScreen(ST77XX_BLACK);
        tft.setCursor(10, 10);
        tft.setTextColor(ST77XX_WHITE);

```

```

tft.setTextSize(3);
tft.setTextWrap(true);
tft.println("Latitude");
tft.setTextColor(ST77XX_WHITE);
tft.setTextSize(3);
tft.setTextWrap(true);
tft.println(link);
tft.setTextColor(ST77XX_WHITE);
tft.setTextSize(3);
tft.setTextWrap(true);
tft.println("");
tft.setTextColor(ST77XX_WHITE);
tft.setTextSize(3);
tft.setTextWrap(true);
tft.println("Longitude");
tft.setTextColor(ST77XX_WHITE);
tft.setTextSize(3);
tft.setTextWrap(true);
tft.print(link1);
delay(500);

```

```

    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis >
15000 || sendDataPrevMillis == 0)){
    sendDataPrevMillis = millis();

```

```

if (Firebase.RTDB.setString(&fbdo, "latitude", link2)){
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
}
else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

```

```

if (Firebase.RTDB.setString(&fbdo, "longitude", link3)){
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
}
else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

```

```

    }
    }
}

void updateSerial()
{
    delay(500);
    if(SIM800L.available())
    {
        Serial.write(SIM800L.read()); //meneruskan apa yang diterima software
        serial ke Port Serial
    }
}

void sms(){
    if(kirimsms==1){
        SIM800L.println("AT+CMGF=1");
        delay(1000);
        SIM800L.println("AT+CNMI=2,2,0,0,0");
        delay(1000);
        kirimsms=0;
    }
}
}

```

- PROGRAM ANDROID STUDIO (Main Activity)

```
package com.example.TrackingPosisiObjek;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.annotation.NonNull;
```

```
import androidx.core.app.ActivityCompat;
```

```
import androidx.core.content.ContextCompat;
```

```
import android.Manifest;
```

```
import android.content.IntentSender;
```

```
import android.content.pm.PackageManager;
```

```
import android.location.Geocoder;
```

```
import android.location.Location;
```

```
import android.os.Bundle;
```

```
import android.os.Looper;
```

```
import android.util.Log;
```

```
import com.google.android.gms.common.api.ResolvableApiException;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsRequest;
import com.google.android.gms.location.LocationSettingsResponse;
import com.google.android.gms.location.SettingsClient;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.Circle;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.maps.android.SphericalUtil;
```

```
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
```

```
import java.text.DecimalFormat;
import java.util.Map;
```

```
public class MainActivity extends AppCompatActivity {
    private Button secondButton, thirdButton, fourthButton;
    Button jarakbutton;
    TextView tampiljarak, tampiljarak2, tampiljarak3;
    private static final String TAG = "MainActivity";
    int LOCATION_REQUEST_CODE = 10001;
    private GoogleMap mMap;
    private Geocoder geocoder;
    private DatabaseReference rootDatabaseref;
    private FusedLocationProviderClient fusedLocationProviderClient;
    LocationRequest locationRequest;

    Marker userLocationMarker;
    Circle userLocationAccuracyCircle;
```

```

LocationCallback locationCallback = new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
        if (locationResult == null) {
            return;
        }
        for (Location location : locationResult.getLocations()) {
            Log.d(TAG, "onLocationResult: " + location.toString());
        }
    }
};

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ActivityCompat.requestPermissions(MainActivity.this, new
String[] { Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION }, 0);
    secondButton = findViewById(R.id.btn_second);
    thirdButton = findViewById(R.id.btn_third);
    fourthButton = findViewById(R.id.btn_fourth);
    jarakbutton = findViewById(R.id.btn_jarak);
    tampiljarak = findViewById(R.id.jarak_view1);
    tampiljarak2 = findViewById(R.id.jarak_view2);
    tampiljarak3 = findViewById(R.id.jarak_view3);
    fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
    locationRequest = LocationRequest.create();
    locationRequest.setInterval(4000);
    locationRequest.setFastestInterval(2000);

```

```

locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURAC
Y);

```

```

secondButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, SecondActivity.class);
        startActivity(intent);
    }
});
thirdButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

        Intent intent = new Intent(MainActivity.this, ThirdActivity.class);
        startActivity(intent);
    }
});
fourthButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, FourthActivity.class);
        startActivity(intent);
    }
});
if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

    return;
}
jarakbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        rootDatabaseref = FirebaseDatabase.getInstance().getReference();
        rootDatabaseref.addValueEventListener(new ValueEventListener()
{

            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Map map = (Map) snapshot.getValue();

                    String lat = map.get("latitude").toString();
                    Double data1 = Double.valueOf(lat);
                    String lon = map.get("longitude").toString();
                    Double data2 = Double.valueOf(lon);
                    LatLng latLng = new LatLng(data1, data2);
                    String lat2 = map.get("latitude2").toString();
                    Double data3 = Double.valueOf(lat2);
                    String lon2 = map.get("longitude2").toString();
                    Double data4 = Double.valueOf(lon2);
                    LatLng latLng2 = new LatLng(data3, data4);
                    String lat3 = map.get("latitude3").toString();
                    Double data5 = Double.valueOf(lat3);
                    String lon3 = map.get("longitude3").toString();

```

```

        Double data6 = Double.valueOf(lon3);
        LatLng latLng3 = new LatLng(data5, data6);

        GetLocation getLocation = new
        GetLocation(getApplicationContext());
        Location location = getLocation.getLocation();
        if (location != null) {

            DecimalFormat x = new DecimalFormat("#.##");
            double latitudeSaya = location.getLatitude();
            double longitudeSaya = location.getLongitude();
            LatLng latLngSaya = new
            LatLng(latitudeSaya, longitudeSaya);
            double latitudeTujuan = data1;
            double longitudeTujuan = data2;
            double distance =
            SphericalUtil.computeDistanceBetween(latLng, latLngSaya);
            tampiljarak.setText("Jarak Alat 1 =
            "+x.format(distance)+" Meter");

            double latitudeTujuan2 = data3;
            double longitudeTujuan2 = data4;
            double distance2 =
            SphericalUtil.computeDistanceBetween(latLng2, latLngSaya);
            tampiljarak2.setText("Jarak Alat 2 =
            "+x.format(distance2)+" Meter");

            double latitudeTujuan3 = data5;
            double longitudeTujuan3 = data6;
            double distance3 =
            SphericalUtil.computeDistanceBetween(latLng3, latLngSaya);
            tampiljarak3.setText("Jarak Alat 3 =
            "+x.format(distance3)+" Meter");

        }
    }

}

@Override
public void onCancelled(@NonNull DatabaseError error) {

}

});

```

```

    }
});

}

@Override
protected void onStart() {
    super.onStart();
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
//    getLastLocation();
    checkSettingsAndStartLocationUpdates();
    } else {
    askLocationPermission();
    }
}

@Override
protected void onStop() {
    super.onStop();
    stopLocationUpdates();
}

private void checkSettingsAndStartLocationUpdates() {
    LocationSettingsRequest request = new
LocationSettingsRequest.Builder()
    .addLocationRequest(locationRequest).build();
    SettingsClient client = LocationServices.getSettingsClient(this);

    Task<LocationSettingsResponse> locationSettingsResponseTask =
client.checkLocationSettings(request);
    locationSettingsResponseTask.addOnSuccessListener(new
OnSuccessListener<LocationSettingsResponse>() {
        @Override
        public void onSuccess(LocationSettingsResponse
locationSettingsResponse) {
            //Settings of device are satisfied and we can start location updates
            startLocationUpdates();
        }
    });
    locationSettingsResponseTask.addOnFailureListener(new
OnFailureListener() {
        @Override

```

```

    public void onFailure(@NonNull Exception e) {
        if (e instanceof ResolvableApiException) {
            ResolvableApiException apiException =
(ResolvableApiException) e;
            try {
                apiException.startResolutionForResult(MainActivity.this,
1001);
            } catch (IntentSender.SendIntentException ex) {
                ex.printStackTrace();
            }
        }
    }
});
}

```

```

    private void startLocationUpdates() {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

            return;
        }
        fusedLocationProviderClient.requestLocationUpdates(locationRequest,
locationCallback, Looper.getMainLooper());
    }

```

```

    private void stopLocationUpdates() {
        fusedLocationProviderClient.removeLocationUpdates(locationCallback);
    }

```

```

    private void askLocationPermission() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_FINE_LOCATION)) {
                Log.d(TAG, "askLocationPermission: you should show an alert
dialog...");
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},

```

