

## BAB III

### METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah D&D (*Design and Development*) atau Desain dan Pengembangan. Model D&D (*Design and Development*) merupakan metode yang berfokus pada desain produk atau sistem dengan pendekatan yang sistematis terhadap proses desain, pengembangan, dan evaluasi dengan tujuan untuk menciptakan pembuatan atau menyempurnakan produk dan alat, baik bersifat instruksional maupun non - instruksional (Yunanda & Efrizon, 2022). Metode ini digunakan dalam rancang bangun sebuah alat yang terdapat usulan didalamnya seperti menggunakan *Geofencing* dan *GPS Tracking*, serta memanfaatkan implementasi *Internet of Things* yang dihubungkan dengan aplikasi Telegram. Pada Gambar 3.1 merupakan alur tahapan dalam penelitian ini, adapun tahapan penelitian ini dilakukan dengan menggunakan metode D&D (*Design and Development*).



Gambar 3.1 Alur Tahapan Penelitian

#### 3.1 Analisis Kebutuhan

Pada tahapan analisis kebutuhan, diawali dengan menganalisis kebutuhan baik secara fungsionalitas dari komponen yang ada didalamnya maupun interaksi pengguna terhadap fitur yang diusulkan. Hal ini melibatkan pemahaman mengenai fitur dan komponen dari alat yang akan digunakan. Sebelum mengetahui kebutuhan yang diperlukan, terlebih dahulu mengetahui analisis permasalahan dari penelitian sebelumnya yang terkait pada penelitian ini. Berdasarkan analisis beberapa penelitian sebelumnya, pada sistem keamanan pencurian sepeda motor, pemilik tidak mengetahui kondisi sepeda motor dalam keadaan aman atau tidak, dikarenakan belum terdapat fitur untuk mengatur mode jaga melalui *smartphone*. Sehingga masih kesulitan mengetahui kondisi ketika sepeda motornya berhasil dicuri saat berada jauh dari kendaraan, karena tidak ada alarm berupa pengiriman pesan atau notifikasi pencurian melalui *smartphone* pemilik sepeda motor. Selain itu, kesulitan dalam melacak atau memantau posisi lokasi sepeda motor pada saat

dicuri, oleh karena itu pemilik sepeda motor belum terdapat sistem monitoring lokasi keberadaan sepeda motor melalui *smartphone*, serta belum mengimplementasikan sistem mematikan mesin secara otomatis ketika sepeda motor berhasil dicuri dalam keadaan menyala, sehingga pencuri akan tetap masih bisa menjalankan sepeda motor dalam jarak yang jauh.

Dari hasil analisis permasalahan tersebut, dibutuhkan sebuah alat yang mendukung untuk menjadi solusi yang inovatif dari kekurangan tersebut. Sehingga membutuhkan sebuah alat yang dilengkapi beberapa komponen utama yang mendukung seperti modul GPS, modul GSM, ESP32, dan Relay. Sistem ini dibutuhkan sebuah aplikasi yang dapat menghubungkan alat tersebut melalui *Internet of Things* ke *smartphone*, pada penelitian ini menggunakan aplikasi Telegram yang dapat memberikan kemudahan bagi pengguna, selain itu sebagai sistem komunikasi dengan alat yang akan dirancang sehingga dapat digunakan untuk sistem informasi serta sistem kontrol seperti memberikan perintah. Dibutuhkan juga sebuah tempat untuk menyimpan data yang dikirim oleh alat, penelitian ini menggunakan *Firestore Database* sebagai tempat penyimpanan data tersebut.

Pada penggunaan aplikasi Telegram memanfaatkan fitur pada telegram yaitu bot telegram, bot tersebut dijalankan beberapa perintah seperti mengirim lokasi pengguna, *setting* radius aman, memilih opsi mode jaga dan mode normal, lacak dan pantau sepeda motor selain itu mengirim informasi lokasi keberadaan sepeda motor, serta jarak antara alat dengan pengguna. Komponen yang terpasang pada alat yaitu modul GPS digunakan untuk memantau atau melacak lokasi alat berada, modul GSM digunakan untuk pengiriman data ke Internet yang dihubungkan dengan mikrokontroler, ESP32 digunakan sebagai mikrokontroler tersebut dalam implementasi *Internet of Things*, Relay digunakan untuk memutuskan dan menghubungkan kembali arus tegangan dari *power supply*.

### **3.2 Perancangan Sistem**

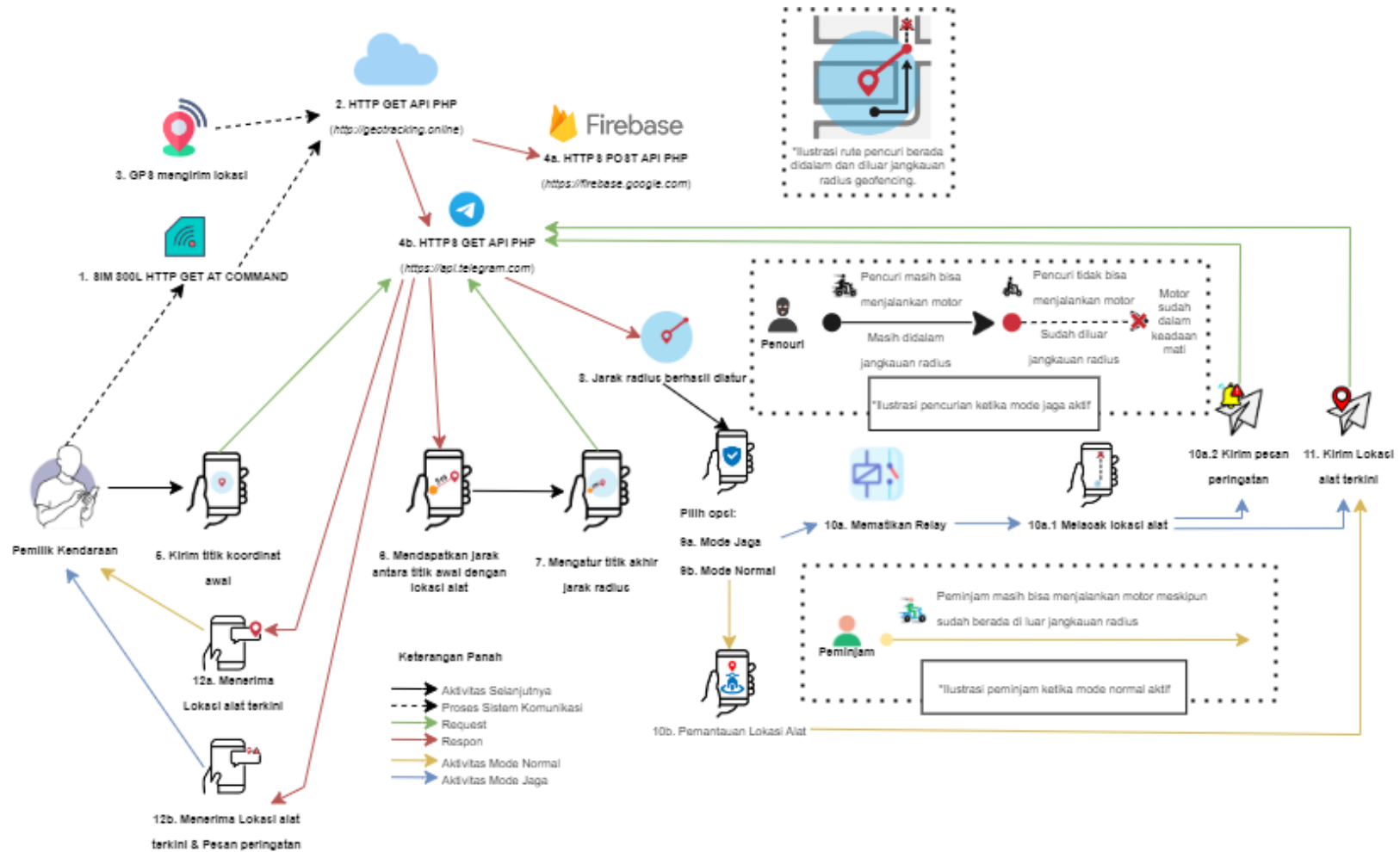
Berdasarkan analisis kebutuhan yang sudah dijelaskan, penelitian ini akan membuat perancangan sebuah alat dengan mengusulkan sistem keamanan pencurian sepeda motor tersebut menggunakan teknologi GPS. Tentunya alat ini membutuhkan komponen perangkat keras lainnya dan *software* yang mendukung

supaya dapat berjalan dan digunakan dengan baik, berikut komponen tersebut diantaranya:

1. Laptop
2. Aplikasi Telegram
3. Firebase *Database*
4. *Software* Visual Studio Code (*PlatformIO*)
5. Mikrokontroler ESP32 DEV KIT
6. Modul GPS Ublox Neo 6M V.2
7. Modul GSM SIM800L V.2
8. Relay
9. Step Down LM2596
10. LED merah & kuning
11. PCB
12. Power Supply 12 Volt

Pada komponen yang sudah disebutkan diatas, selanjutnya merancang sistem secara keseluruhan dari beberapa komponen tersebut. Perancangan ini memberikan gambaran alur cara kerja sistem berjalan. Pada gambar 3.2 dapat dilihat arsitektur sistem yang menunjukkan perancangan sistem yang terdiri dari komponen baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang terhubung membentuk alur sistem secara keseluruhan. Dalam perancangan sistem keseluruhan ini terdapat beberapa alur yang bekerja didalamnya dan kemudian menjadi satu alur yang bekerja.

Proses perancangan melibatkan identifikasi kebutuhan, pemilihan komponen yang tepat, serta integrasi antar komponen sehingga semua bagian dapat berfungsi dengan baik. Pada tahap ini, setiap komponen dihubungkan melalui penghubung yang sesuai, seperti untuk beberapa modul yang digunakan, serta protokol jaringan untuk komponen yang terhubung. Hal ini penting untuk memastikan bahwa sistem dapat beroperasi dengan stabil dan benar sesuai dengan spesifikasi atau hasil yang diharapkan.



Gambar 3.2 Arsitektur Sistem

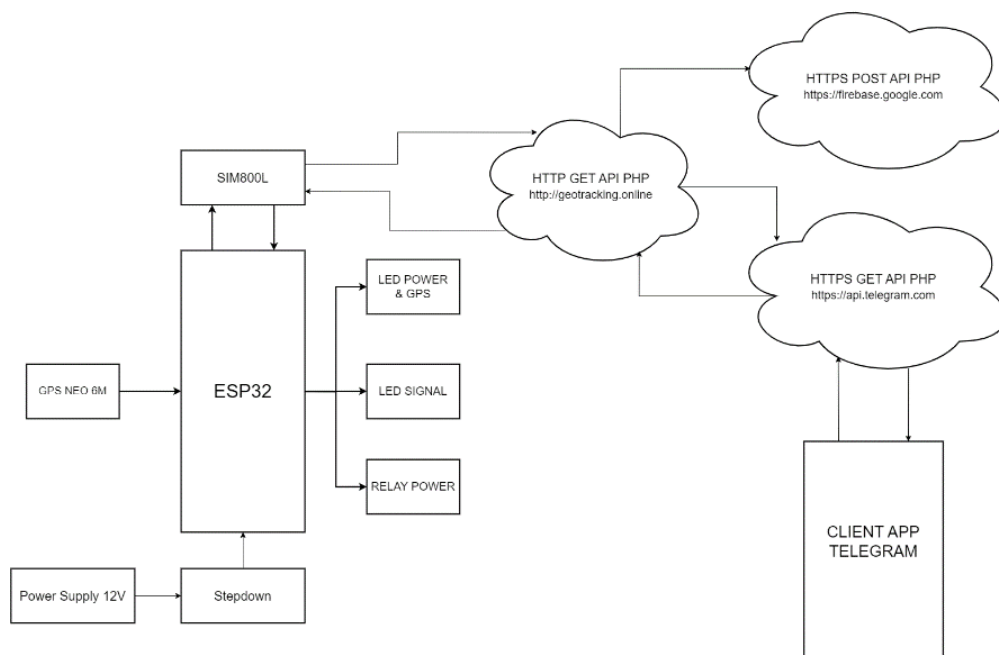
Terlihat pada gambar 3.2 arsitektur tersebut, proses dimulai yaitu pengguna mengirim perintah HTTP GET ke server menggunakan modul SIM800L. Permintaan HTTP GET diteruskan ke API PHP yang berada di server *http://geotracking.online*, pengguna mendapatkan data lokasi dan mengirimkannya ke server menggunakan modul GPS. Pada HTTP *Requests* terdapat dua alur yang berbeda yaitu lokasi yang diperoleh dari GPS dikirim ke Firebase menggunakan HTTPS POST, dan data yang dikirim ke API Telegram menggunakan HTTPS GET untuk notifikasi.

Setelah proses HTTP Request berhasil, pengguna mengirim lokasinya yang digunakan sebagai titik koordinat awal ke server. Kemudian sistem akan proses menghitung jarak antara titik awal dengan lokasi alat yang dikirim melalui modul GPS. Setelah berhasil mendapatkan jarak, pengguna mengatur titik akhir jarak radius yang digunakan sebagai radius *Geofencing*. Jarak radius berhasil diatur dan membentuk batas area virtual (*Geofencing*). Kemudian, pengguna memilih opsi mode jaga atau mode normal. Jika pengguna memilih mode jaga, maka sistem keamanan aktif, sehingga sistem dapat mematikan motor yaitu *Relay off*, melacak lokasi alat dan mengirim lokasi tersebut, serta mengirim pesan peringatan.

Pada sistem mode jaga diperuntukan jika pengguna ingin menjaga kendaraannya dari pencuri. Terlihat seperti ilustrasi pencuri ketika mode jaga aktif, pencuri masih bisa menjalankan motor ketika masih didalam jangkauan radius, lalu pencuri tidak bisa menjalankan motor ketika sudah berada diluar jangkauan radius dikarenakan sistem mematikan Relay yang dari semula *on* menjadi *off*, kemudian motor yang sudah dalam keadaan mati tersebut mengirim lokasi motor tersebut berada. Terlihat pada gambar rute pencuri berada didalam dan diluar jangkauan radius *Geofencing*. Jika pengguna memilih mode normal, sistem keamanan hanya berupa sistem pemantauan lokasi alat, dan mengirimkan lokasi tersebut. Sistem mode normal di peruntukan jika pengguna ingin memakai atau meminjamkan motor tersebut ke orang lain yang diberi akses untuk memakainya tanpa harus mengaktifkan sistem keamanan ketika motor sedang dipakai, sistem ini hanya berupa pemantauan lokasi motor, tidak terdapat sistem mematikan Relay, dan tidak terdeteksi pencurian pada saat motor melewati radius yang sudah berhasil diinput

sebelumnya. Setelah proses pengiriman dari beberapa mode yang dipilih, maka jika mode jaga diaktifkan maka pengguna menerima lokasi alat terkini dan pesan peringatan, namun jika mode normal diaktifkan pengguna hanya menerima lokasi alat terkini.

Dari pemaparan alur dari arsitektur sistem tersebut, terdapat tiga alur proses sistem yang berjalan, yaitu alur komunikasi SIM800L, sistem keamanan seperti monitoring dan pendeteksi pencurian, serta integrasi aplikasi telegram. Pada gambar 3.3 blok diagram sistem, menunjukkan perancangan sistem keamanan sepeda motor yang terdiri dari beberapa komponen penting yang saling berhubungan dengan fungsi yang berbeda - beda.



Gambar 3.3 Blok Diagram Sistem

Pada gambar 3.3 Blok Diagram Sistem menunjukkan alur sistem dimulai dengan *power supply* 12V yang masuk ke modul step - down untuk mengurangi tegangan menjadi 5V yang dibutuhkan oleh ESP32 dan komponen lainnya. ESP32 menerima daya dari step - down dan berfungsi untuk mengontrol beberapa komponen yang terhubung. Modul GPS Neo 6M mengirimkan data lokasi ke ESP32, selanjutnya ESP32 mengirimkan perintah HTTP GET melalui modul SIM800L ke endpoint <http://geotracking.online> . Jika proses respons HTTP dari *Endpoint* Geotracking berhasil, ESP32 akan mengirimkan data ke API HTTPS

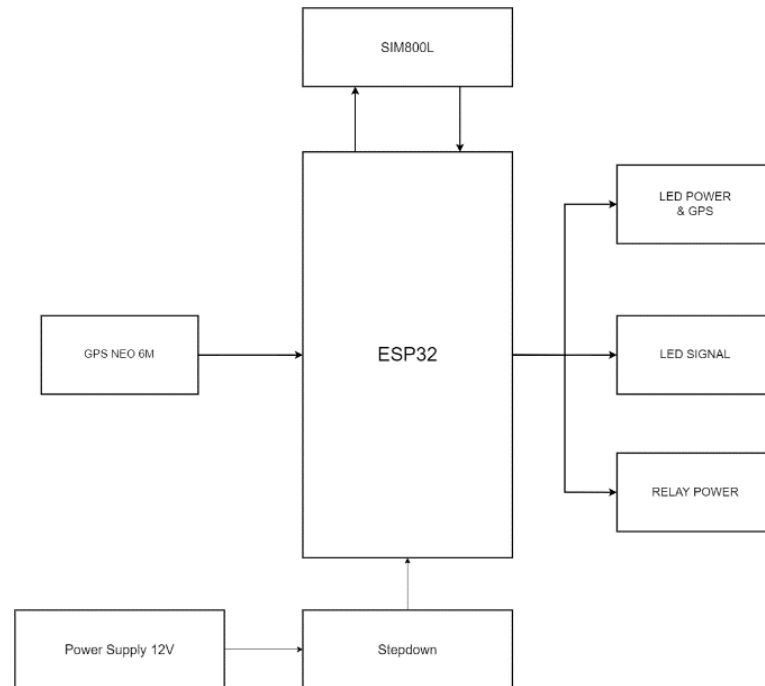
Firestore (<https://firebase.google.com>) melalui SIM800L menggunakan metode POST, kemudian data akan tersimpan di *database* Firestore. Selain itu jika proses respons HTTP berhasil, ESP32 mengirim atau menerima pesan melalui API HTTPS Telegram (<https://api.telegram.com>) menggunakan metode GET dan respons JSON dikirim kembali ke SIM800L.

Terdapat komponen lainnya seperti LED, ESP32 mengontrol LED *Power* dan GPS serta LED *Signal* untuk memberikan indikasi visual mengenai status daya, serta sinyal GPS dan GSM. Selain itu juga terdapat Relay *power*, ESP32 mengontrol Relay *power* untuk mengendalikan daya ke perangkat seperti memutuskan dan menghubungkan kembali. Blok diagram sistem ini memastikan data dari modul GPS diproses oleh ESP32, dikomunikasikan oleh SIM800L ke berbagai layanan web, dan memungkinkan kontrol serta monitoring melalui aplikasi Telegram. Dari pemaparan tersebut, pada perancangan sistem terdapat perancangan alat dan proses komunikasi sistem.

### 3.2.1 Perancangan Alat

Tahap berikutnya adalah merancang alat berdasarkan desain sistem yang telah dibuat sebelumnya, perangkat tersebut dirancang sesuai dengan blok diagram yang sudah ada. Pada perancangan ini alat yang dirancang terdiri dari beberapa komponen dengan fungsi yang berbeda - beda. Modul GPS NEO 6M mengirimkan data lokasi GPS ke ESP32 yang berfungsi sebagai pengontrol utama sistem. ESP32 menerima data lokasi dari GPS NEO 6M dan mengontrol beberapa komponen output, yaitu LED *Power & GPS*, LED *Signal*, dan Relay *Power*. Serta *power supply* sebesar 12V dan Step Down untuk menurunkan tegangan menjadi 5V.

*Power supply* terhubung untuk memberikan sumber daya ke setiap komponen yang terhubung pada ESP32. Kemudian ESP32 terhubung ke modul SIM800L untuk komunikasi GSM/GPRS, berkomunikasi dengan ESP32 untuk mengirim data ke server atau indikasi visual terkait status daya dan GPS, sedangkan LED *Signal* memberikan indikasi visual terkait status sinyal. Relay *Power* dikendalikan oleh ESP32 untuk mengontrol daya ke perangkat seperti memutuskan dan menghubungkan kembali. Berikut adalah gambar 3.4 menunjukkan blok diagram alat, pada gambar tersebut memberikan gambaran pada perancangan setiap komponen yang saling terhubung.

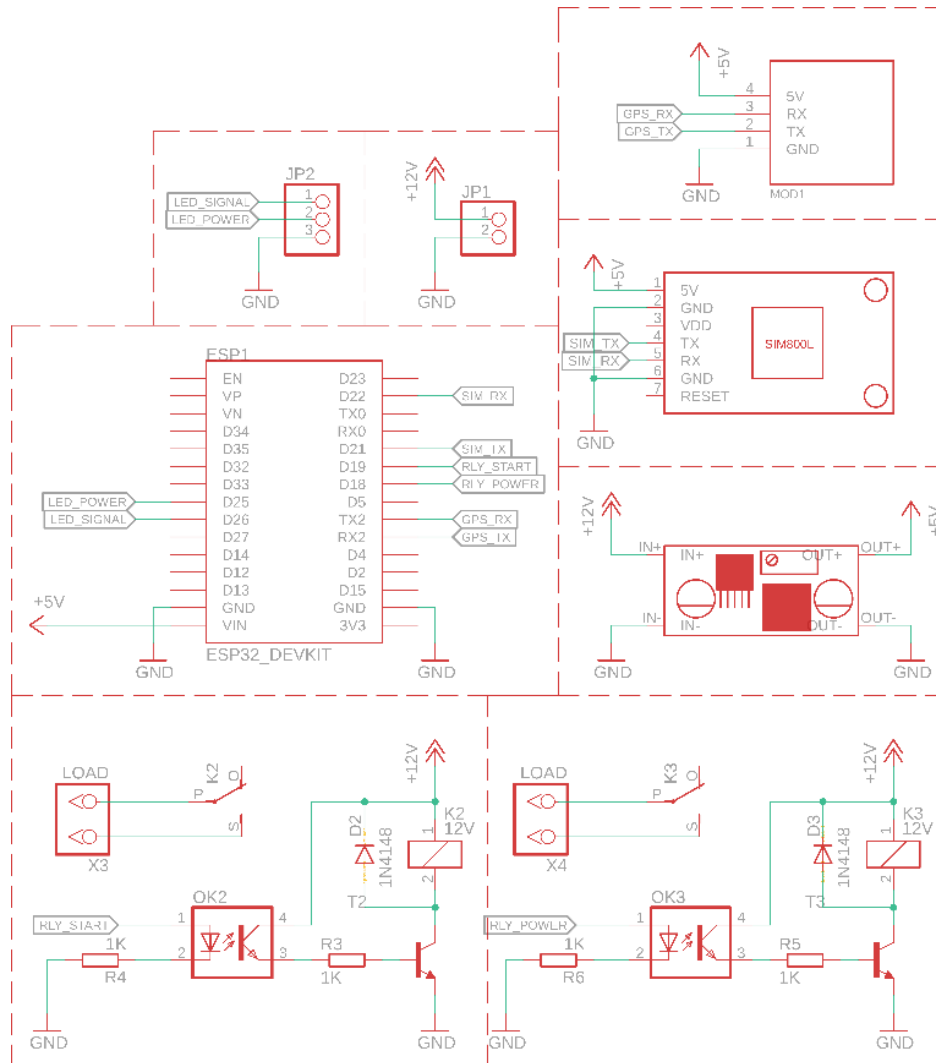


Gambar 3.4 Blok Diagram Alat

Pada Gambar 3.4 Blok Diagram Alat tersebut dalam alur prosesnya, *Power Supply* serta step down untuk mengirimkan sumber daya ke ESP32 yang terhubung keseluruhan komponen. Modul GPS NEO 6M mengirimkan data lokasi ke ESP32 yang kemudian memprosesnya dan mengontrol LED *Power & GPS*, LED *Signal* GSM maupun GPS, dan Relay *Power* sesuai dengan perintah yang diprogram. ESP32 juga mengirimkan data atau status ke modul SIM800L, yang kemudian mengirimkan data melalui jaringan GSM/GPRS ke server atau menerima perintah dari server untuk diproses lebih lanjut oleh ESP32. Dengan alur ini, sistem dapat mengumpulkan data lokasi, mengontrol beberapa indikator LED dan relay, serta berkomunikasi dengan server eksternal untuk pertukaran data.

Selanjutnya, merancang skematik alat bertujuan memberikan gambaran yang jelas mengenai rangkaian hubungan antara berbagai komponen pada setiap pin yang tersambung ke ESP32, sebagai pengontrol utama, akan diperinci dalam skematik ini. Skematik ini diharapkan memberikan kemudahan dalam memahami setiap komponen terhubung dengan benar. Perancangan ini dapat dilihat pada gambar 3.5, yang menunjukkan desain skematik alat.



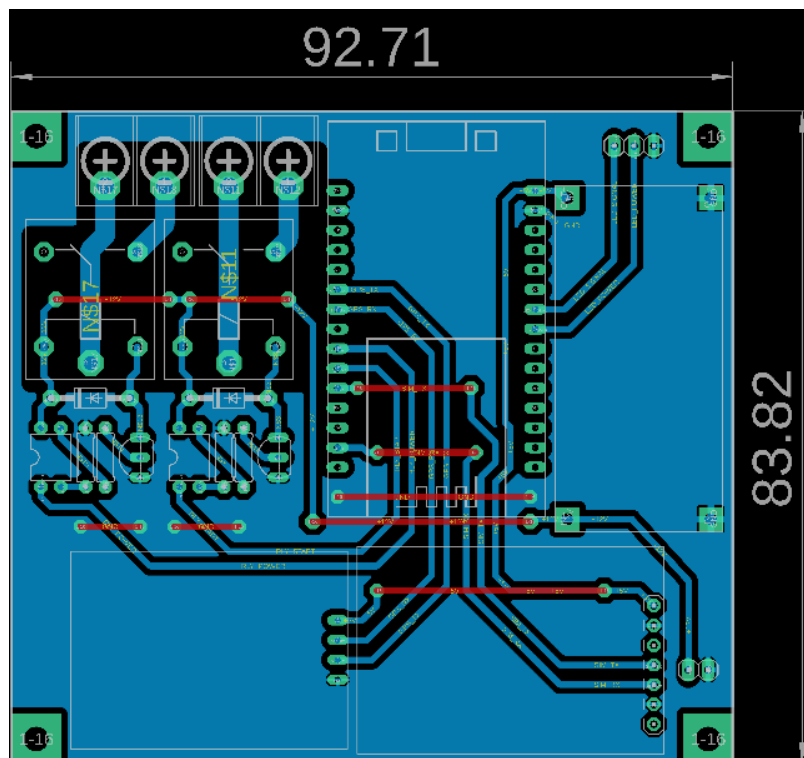


Gambar 3.5 Desain Skematik Alat

Rangkaian ini dirancang dengan beberapa komponen utama yang saling terhubung. *Power supply* +12V digunakan sebagai input untuk rangkaian, dengan GND sebagai *ground*. Stepdown kemudian mengonversi tegangan +12V menjadi +5V yang diperlukan oleh ESP32, GND dari stepdown terhubung ke GND ESP32. Modul GPS menerima daya +5V dan GND dari ESP32, dengan RX terhubung ke TX2 ESP32 dan TX terhubung ke RX2 ESP32. Modul GSM juga menerima daya +5V dan GND dari ESP32, dengan TX terhubung ke pin D21 dan RX terhubung ke pin D22 ESP32. Relay menggunakan +12V dari power supply untuk *input* positif dan GND dari *power supply* sebagai *ground*. Pengontrol Relay RLY\_START dihubungkan ke pin D19 ESP32. JP1 menerima +12V dari power supply untuk input positif, dengan GND JP1 terhubung ke GND dari power supply. JP2

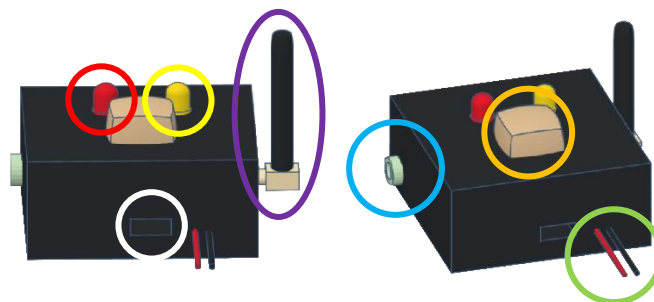
terhubung ke ESP32, dengan GND JP2, LED\_POWER (D25), dan LED\_SIGNAL (D26) yang digunakan untuk indikator daya dan sinyal.

Pada tahap selanjutnya menggabungkan beberapa komponen tersebut menjadi satu rangkaian yang disusun dan dipasang dalam papan rangkaian cetak atau yang disebut dengan PCB (*Printed Circuit Board*). Desain PCB tersebut dibuat dengan panjang 92.71mm dan lebar 83.82mm dan menggunakan software eagle, berikut ini Gambar 3.6 Desain PCB.



Gambar 3.6 Desain PCB

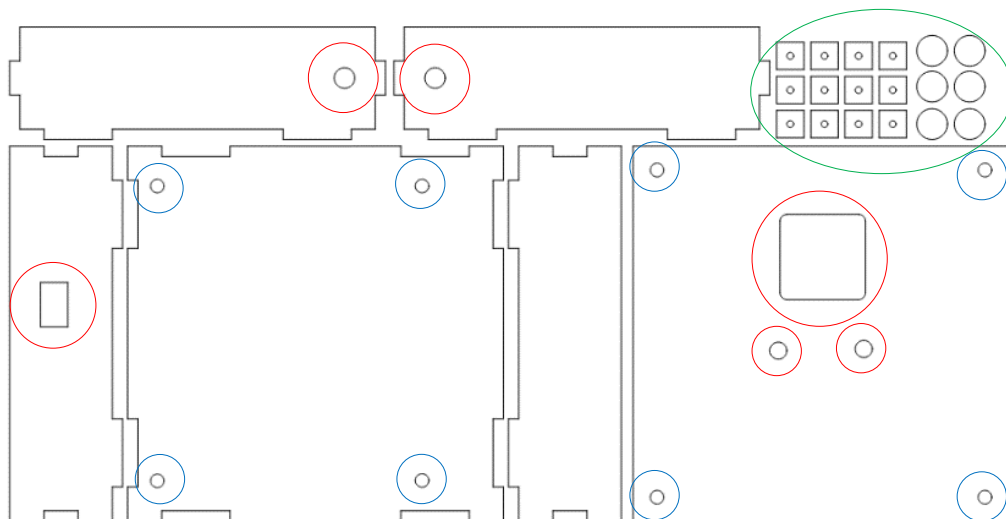
Tahap berikutnya yaitu pembuatan sketsa alat dalam bentuk 3D modelling agar penulis mendapatkan gambaran bentuk dan tampilan jika dilihat dari luar box. Berikut gambar 3.7 Sketsa Alat.



Gambar 3.7 Sketsa Alat

Pada gambar terlihat berbagai komponen penting ditandai dengan berbagai warna untuk memudahkan identifikasi. Antena GPS, yang ditandai dengan lingkaran berwarna *orange*, dan antena GSM yang ditandai dengan oval berwarna ungu, diposisikan diluar box untuk memastikan sinyal yang diterima optimal. Selain itu, terdapat LED merah, yang ditandai dengan lingkaran berwarna merah, berfungsi sebagai indikator visual untuk sinyal GPS dan daya. Sementara itu, LED kuning, yang ditandai dengan lingkaran berwarna kuning, digunakan sebagai indikator sinyal GSM.

Dapat dilihat juga pada gambar tersebut, terlihat konektor *input power supply*, yang ditandai dengan lingkaran berwarna biru, dan *input ESP3*, yang ditandai dengan lingkaran berwarna putih. Selain itu, kabel Relay yang keluar dari box ditandai dengan lingkaran berwarna hijau. Semua komponen ini dirancang untuk memastikan sistem bekerja dengan baik dan memberikan indikasi visual yang jelas mengenai status operasi yang sedang berjalan. Pada tahapan selanjutnya, membuat desain box dan dipersiapkan untuk di cetak. Rangkaian desain box tersebut dibuat menggunakan *Software Adobe Illustrator*, seperti yang terlihat pada gambar 3.8.



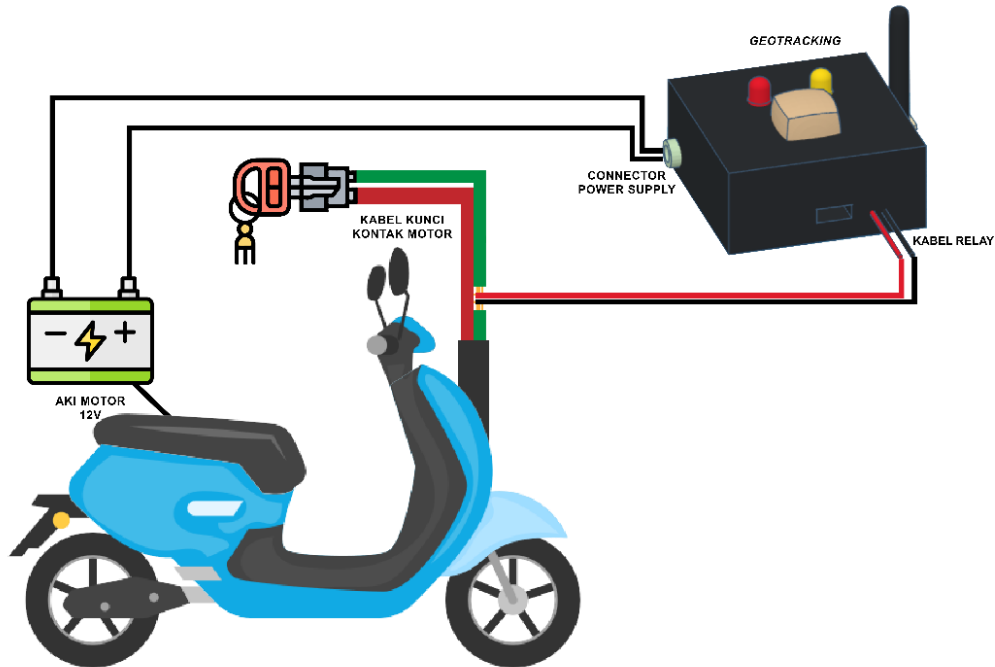
Gambar 3.8 Sketsa Box

Pada gambar sketsa box penyimpanan alat, setiap ukuran lubang diperhatikan agar pas saat pemasangan. Terdapat enam lubang (lingkaran merah) untuk komponen yang keluar dari box, 18 lubang mur (lingkaran hijau) untuk menguatkan PCB di dalam box, dan delapan lubang baut (lingkaran biru) untuk

menguatkan box dari luar.

### 3.2.2 Perancangan Alat Pada Sepeda Motor

Selanjutnya tahap perancangan alat yang dihubungkan ke sepeda motor, dapat dilihat pada gambar 3.9 berikut ini.

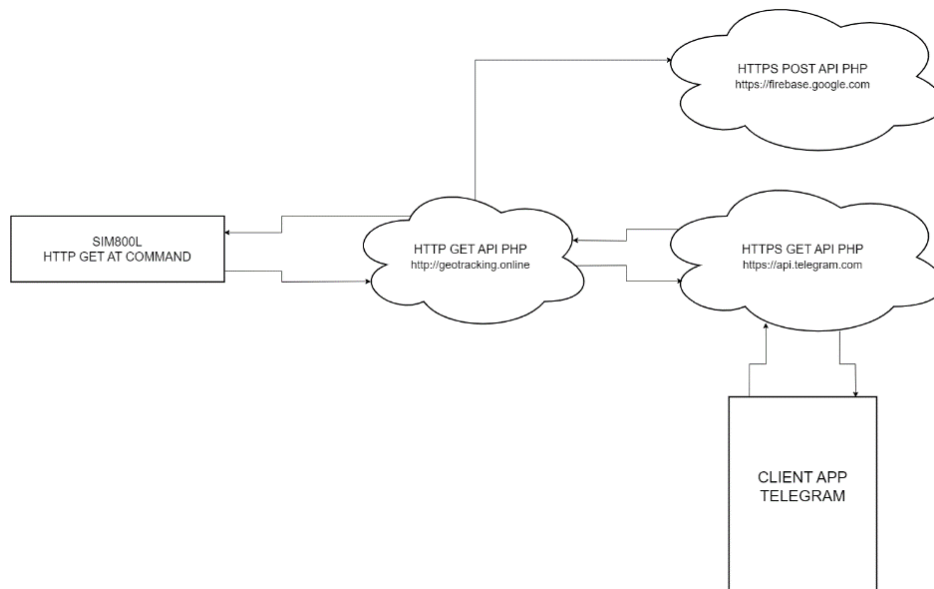


Gambar 3.9 Rancangan Alat Pada Sepeda Motor

Terlihat pada gambar perancangan alat pada sepeda motor memberikan sebuah gambaran pemasangan alat yang terhubung ke sepeda motor. Proses tersebut dengan menghubungkan alat pada sepeda motor dengan cara menghubungkan salah satu dari dua kabel kontak motor dan memotongnya menjadi dua potongan kabel, yang kemudian diberi nama kabel A dan kabel B. Kabel A dihubungkan ke kabel Relay NO, sedangkan Kabel B dihubungkan ke Relay COM. Selanjutnya, positif aki disambungkan ke konektor positif *power supply* pada alat, dan negatif aki dihubungkan ke konektor negatif *power supply* pada alat. Gambar tersebut membantu memastikan pemasangan alat dilakukan dengan benar dan sesuai dengan prosedur yang ditetapkan.

### 3.2.3 Perancangan Komunikasi SIM800L

Selain perancangan alat, terdapat juga proses komunikasi sistem, yaitu komunikasi antara perangkat SIM800L dan berbagai layanan melalui HTTP dan HTTPS API seperti Firebase dan API Telegram. Gambar 3.10 menunjukkan diagram alir proses komunikasi sistem, menggambarkan langkah-langkah yang diambil oleh sistem dari awal hingga akhir.

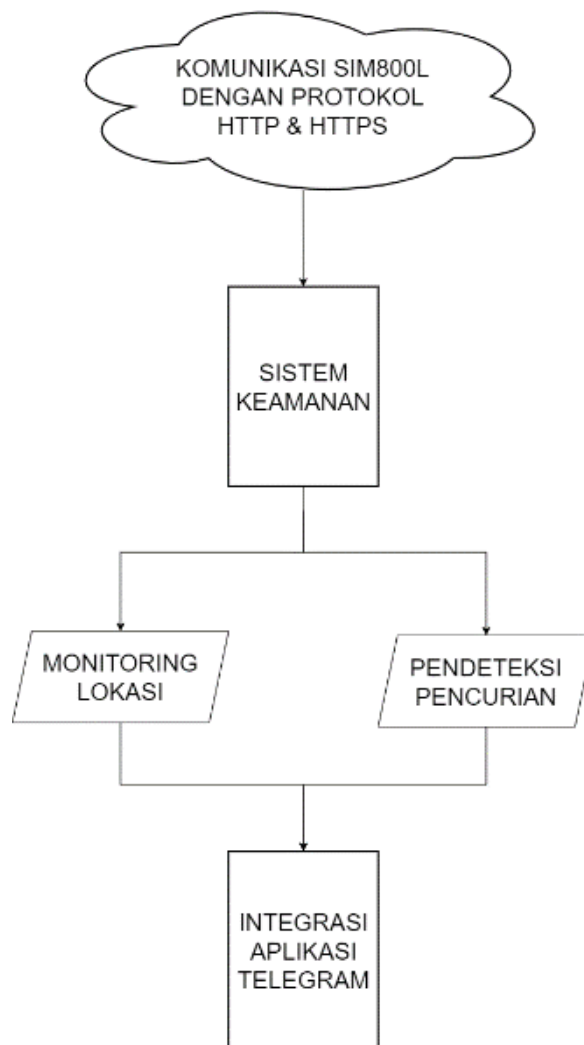


Gambar 3.10 Diagram Alir Proses Komunikasi Sistem

Pada gambar diagram alir proses komunikasi sistem, terlihat modul SIM800L mengirimkan perintah AT untuk melakukan perintah HTTP GET, dengan menginisiasi pengambilan data dari server melalui koneksi GPRS. Permintaan HTTP GET dikirim ke API PHP yang di *hosting* di <http://geotracking.online>, mengambil data berupa lokasi, jarak dan radius dari server. Data yang diperoleh dari langkah sebelumnya dikirim ke <https://firebase.google.com> menggunakan permintaan HTTPS POST, menyimpan data di Firebase. Mengirim permintaan HTTPS GET ke API <https://api.telegram.com> untuk mengirim pesan atau notifikasi, mengirim pesan melalui Telegram. Pengguna menerima pesan atau notifikasi di aplikasi Telegram, menyediakan *interface* bagi pengguna untuk menerima pesan dan berinteraksi dengan sistem.

### 3.3 Pengembangan

Tahap pengembangan adalah tahapan yang diusulkan oleh penulis berdasarkan perancangan sistem sebelumnya. Pada tahap perancangan tersebut, penulis mengusulkan tiga alur proses utama berdasarkan arsitektur sistem. Pertama, alur komunikasi antara SIM800L dan berbagai layanan menggunakan protokol HTTP dan HTTPS. Kedua, sistem keamanan yang mencakup monitoring dan pendeteksi pencurian. Ketiga, integrasi dengan aplikasi Telegram untuk menyediakan notifikasi dan kontrol jarak jauh. Gambar 3.11 memperlihatkan secara jelas alur proses yang diusulkan oleh penulis, yang mencakup semua aspek ini.



Gambar 3.11 Alur Proses yang Diusulkan

Pada gambar tersebut, alur proses yang diusulkan oleh penulis dimulai dengan komunikasi antara SIM800L dan berbagai layanan menggunakan protokol

HTTP & HTTPS. Proses ini mencakup pengiriman dan penerimaan data yang diperlukan untuk mengoperasikan sistem. Setelah itu, sistem keamanan diaktifkan, berupa monitoring lokasi dan pendeteksi pencurian. Langkah ini memastikan bahwa lokasi kendaraan selalu terpantau dan setiap aktivitas yang mencurigakan dapat segera terdeteksi. Selanjutnya, sistem keamanan ini diintegrasikan dengan aplikasi Telegram, memungkinkan pengoperasian dan pemantauan sistem dari jarak jauh melalui *smartphone*.

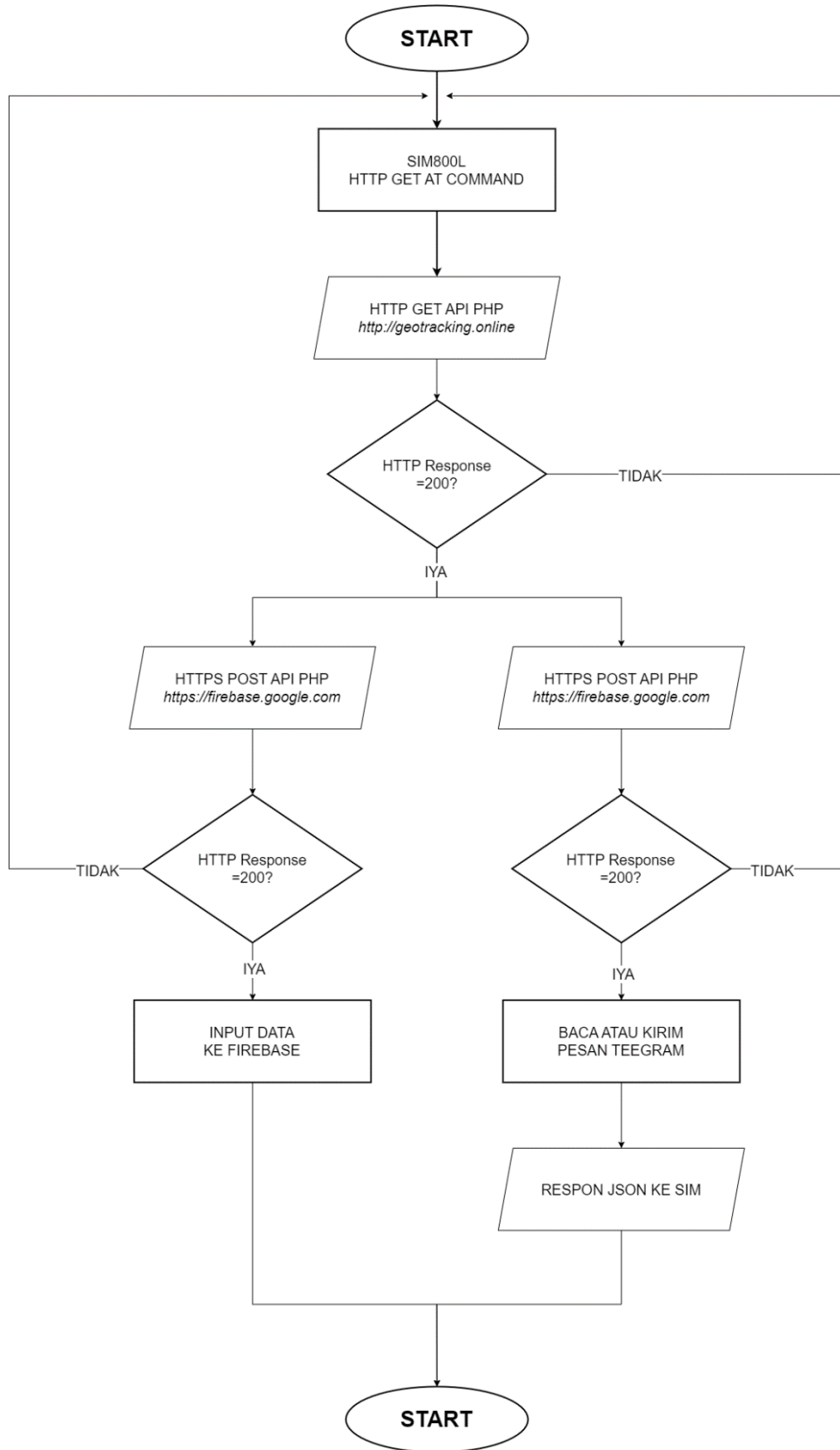
### 3.3.1 Komunikasi SIM800L Dengan Protokol HTTP dan HTTPS

Dalam penelitian ini, SIM800L digunakan untuk pengiriman data ke Internet melalui jaringan GPRS, dengan memanfaatkan protokol komunikasi standart modem pada Modul GSM SIM800L yaitu *AT Command*. Terlihat pada gambar 3.12 alur komunikasi SIM800L dengan protokol HTTP & HTTPS.

Proses dimulai dengan Inisialisasi SIM800L melalui HTTP GET *AT COMMAND*. Setelah itu, proses dilanjutkan dengan mengirimkan perintah HTTP GET menggunakan perangkat SIM800L ke *endpoint* <http://geotracking.online>. Jika respons HTTP yang diterima tidak sama dengan 200, maka sistem akan mengulang kembali pengiriman perintah HTTP GET dengan SIM 800L. Namun, jika respons HTTP adalah 200, maka proses dilanjutkan ke tahap berikutnya. Sistem kemudian bercabang menjadi dua alur yaitu pengiriman data ke firebase dan pengiriman data ke Telegram.

Untuk pengiriman data ke Firebase, sistem menggunakan HTTPS PHP POST API <https://firebase.google.com>. Setelah mengirimkan data, jika respons HTTP tidak sama dengan 200, proses kembali mengirim perintah HTTP GET; jika respons HTTP adalah 200, data akan ditulis ke database Firebase.

Sementara itu, untuk pengiriman data ke Telegram, sistem menggunakan PHP HTTPS GET API <https://api.telegram.com>. Setelah mengirim atau membaca pesan Telegram, jika respons HTTP tidak sama dengan 200, proses kembali mengirim perintah HTTP GET; jika respons HTTP adalah 200, pesan diproses dan respons JSON dikirim kembali ke SIM800L. Setelah data ditulis ke Firebase atau pesan Telegram diproses, proses berakhir dan mengulang lagi dari awal.



Gambar 3.12 Alur Komunikasi SIM800L Dengan Protokol HTTP & HTTPS



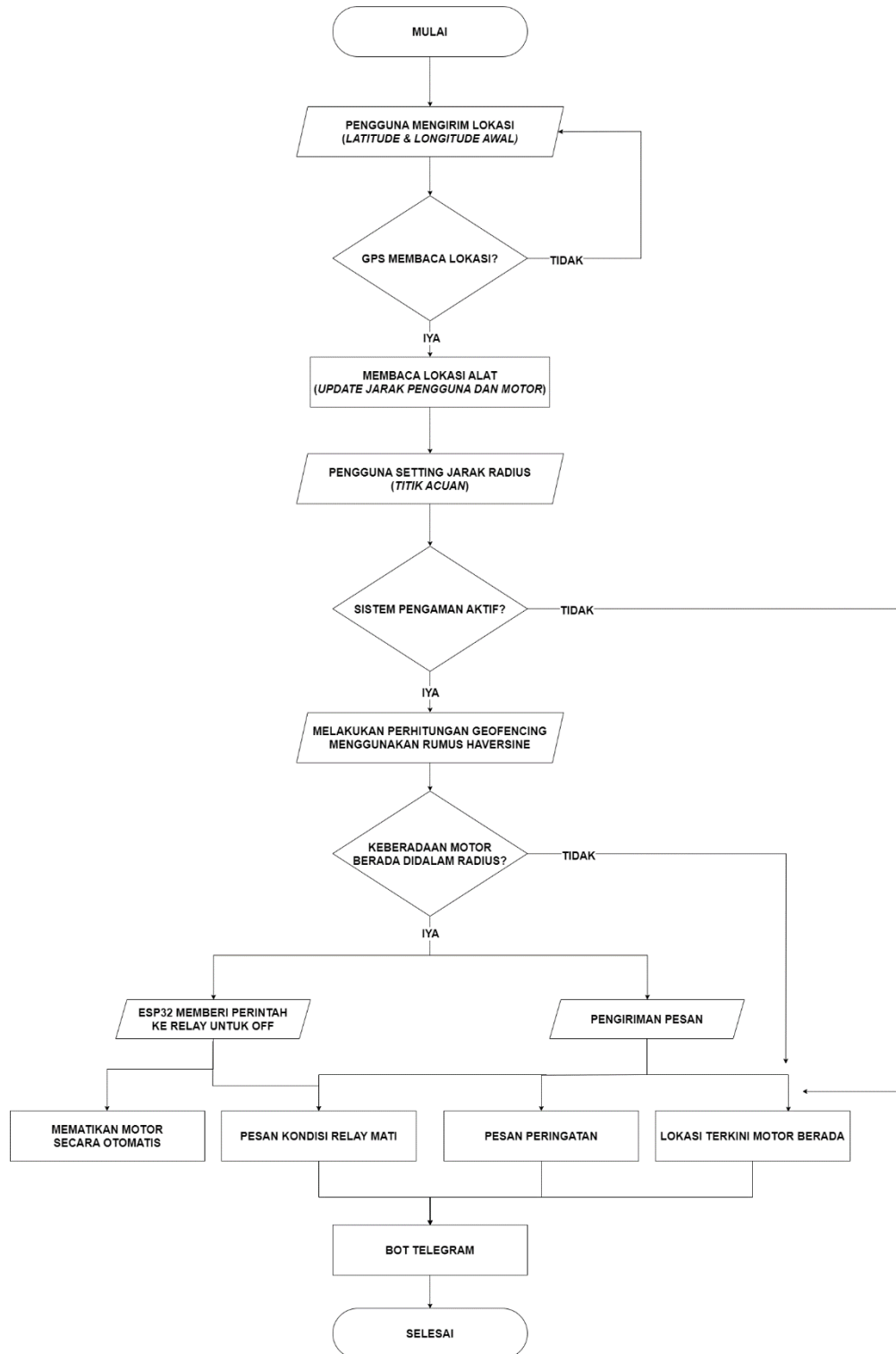
### 3.3.2 Keseluruhan Sistem Keamanan

Dalam penelitian ini, penulis mengusulkan sistem keamanan sepeda motor menggunakan *Geofencing* dan GPS terintegrasi dengan IoT dan aplikasi Telegram. Sistem ini mendeteksi pergerakan keluar dari area tertentu dan memungkinkan pemantauan lokasi sepeda motor dari jarak jauh melalui *smartphone*. Dapat dilihat pada gambar 3.13 menunjukkan keseluruhan sistem yang diusulkan.

Pada gambar diagram alir proses sistem keamanan secara keseluruhan yang diusulkan penulis dalam penelitian ini, dapat dilihat proses dimulai dengan pengguna mengirimkan lokasinya. Lokasi tersebut dijadikan sebagai titik koordinat awal. Kemudian, modul GPS membaca lokasi, jika modul GPS belum berhasil mendapatkan sinyal, maka update jarak pengguna dan motor akan gagal, dan pengguna perlu mengirimkan lokasinya kembali sampai modul GPS mendapatkan sinyal. Jika modul GPS berhasil mendapatkan sinyal, maka update jarak pengguna dan motor berhasil dilakukan.

Tahap selanjutnya adalah pengguna mengatur jarak radius sebagai acuan. Ketika sistem pengaman diaktifkan, *Geofencing* dihitung dengan rumus Haversine. Jika modul GPS mendeteksi motor berada di luar radius yang ditentukan, sistem akan mematikan motor secara otomatis dengan mengirim perintah ke ESP32 untuk mematikan relay. Selain itu, sistem akan mengirim pesan peringatan, lokasi motor terkini, dan status relay melalui bot Telegram. Jika motor masih berada dalam radius, hanya lokasi motor terkini yang dikirim melalui bot Telegram.

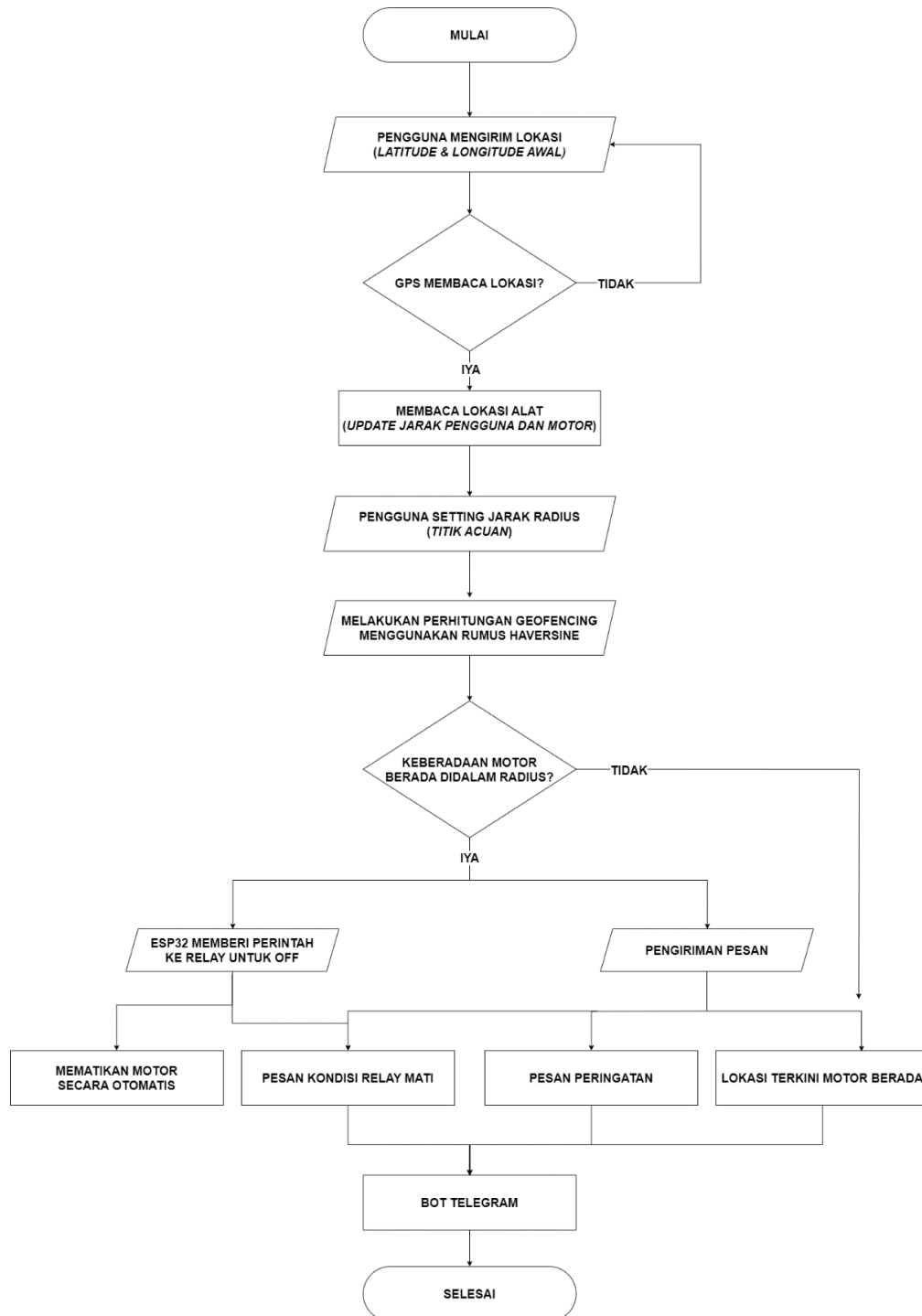
Sistem pengaman akan aktif ketika pengguna mengaktifkannya, tetapi hanya menjalankan modul GPS yang digunakan untuk mengirim lokasi terkini motor melalui bot Telegram. Setelah proses selesai dijalankan, baik dengan sistem pengaman diaktifkan maupun tidak, program akan berhenti atau kembali ke proses awal tergantung pada keputusan pengguna. Pada sistem keamanan ini secara keseluruhan yang sudah dipaparkan, terdapat dua mekanisme operasional yang berjalan seperti pendeteksian pencurian menggunakan *Geofencing* dan monitoring lokasi menggunakan GPS.



Gambar 3.13 Diagram Alir Sistem Keamanan Keseluruhan

Pada fitur pendeteksian pencurian dalam sistem keamanan sepeda motor menggunakan *Geofencing* dengan rumus Haversine untuk menghitung jarak antara dua titik koordinat. Sistem ini membentuk pembatas virtual untuk memantau

pergerakan motor dan mendeteksi pencurian ketika motor berada di luar radius yang ditentukan. Pada gambar 3.14 menunjukkan diagram alir cara kerja sistem pendeteksian pencurian menggunakan *Geofencing*.



Gambar 3.14 Diagram Alir Sistem Pendeteksi Pencurian

Pada gambar tersebut ditunjukkan cara kerja dari aktivasi sistem pendeteksi pencurian menggunakan *Geofencing*. Alur proses awal pada sistem ini sama seperti yang telah dipaparkan pada diagram alir sistem keamanan secara keseluruhan. Hanya saja yang membedakan pada saat proses sistem pengaman diaktifkan, maka sistem akan mengaktifkan fitur pendeteksian pencurian dengan menjalankan perhitungan *Geofencing* menggunakan rumus *Haversine* yang sudah dijelaskan pada subbab 2.3. Rumus *Haversine* di implementasikan kedalam program dapat dilihat pada gambar 3.15 sebagai berikut:

```

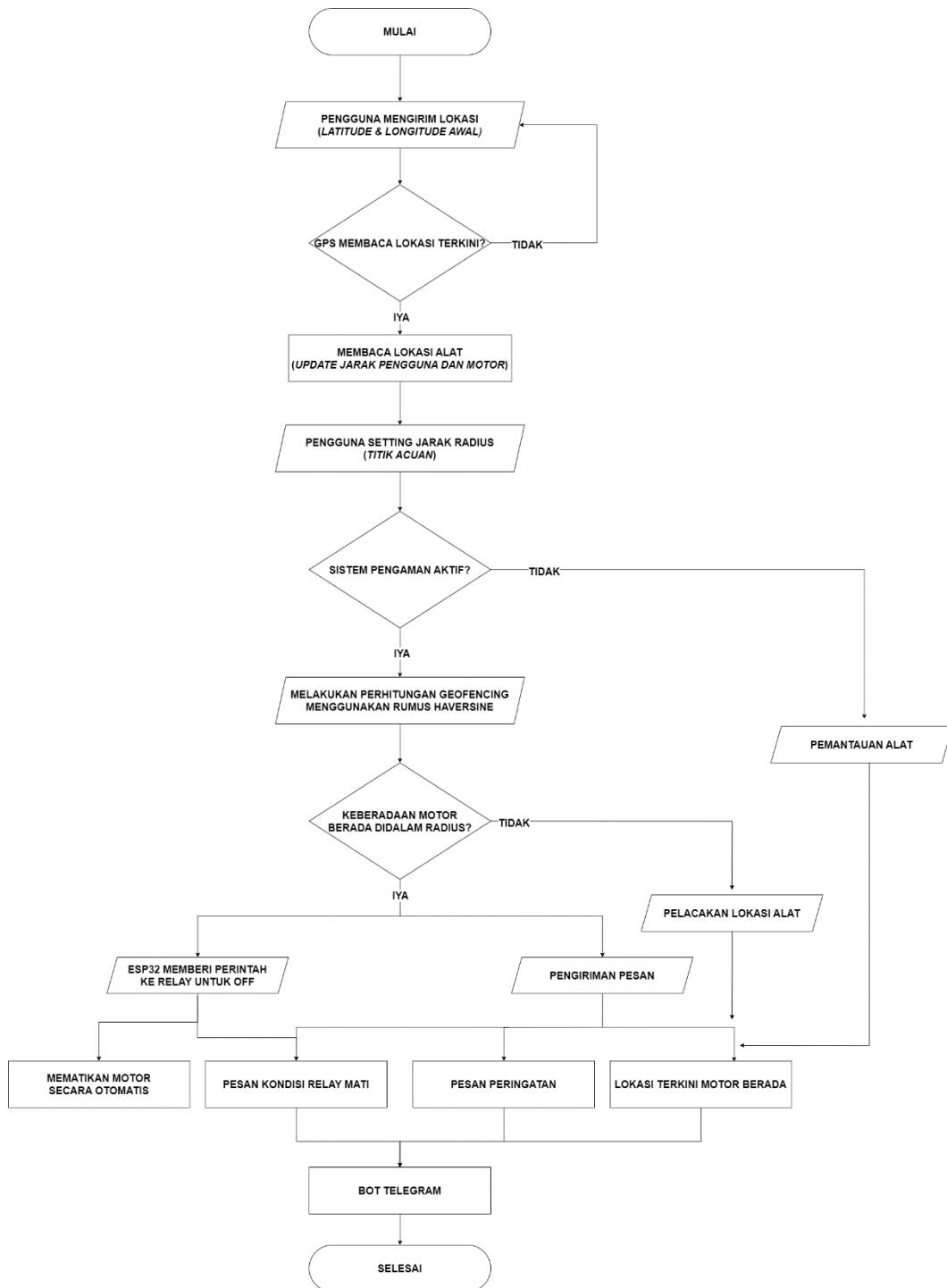
float getDistance(double lat1, double lon1, double lat2, double lon2)
{
    if (lat1 == 0.0 or lon1 == 0.0 or lat2 == 0.0 or lon2 == 0.0)
        return 0.0;
    double dLat = radians(lat2 - lat1);
    double dLon = radians(lon2 - lon1);
    double a = sin(dLat / 2) * sin(dLat / 2) + cos(radians(lat1)) * cos(radians(lat2)) * sin(dLon / 2) * sin(dLon / 2);
    double c = 2 * atan2(sqrt(a), sqrt(1 - a));
    double d = 6371000 * c;
    return d;
}

```

Gambar 3.15 Script Program Implementasi Rumus Haversine

Pada fitur tersebut, jika motor terdeteksi pencurian atau berada di luar radius yang telah ditentukan atau melebihi titik acuan, sistem pendeteksi pencurian akan memerintahkan ESP32 untuk mengontrol Relay menjadi Off. Hal ini akan mematikan motor secara otomatis melalui perintah yang telah diprogram. Selain itu, sistem juga akan mengirimkan pesan berupa lokasi terkini keberadaan motor, pesan peringatan, dan kondisi Relay Off melalui bot Telegram.

Ketika posisi sepeda motor masih berada di dalam radius yang ditentukan, sistem akan menganggapnya aman dan hanya mengirimkan lokasi terkini motor melalui bot Telegram. Fitur ini bertujuan untuk memantau kondisi keamanan sepeda motor dan lokasi terkini sepeda motor dari jarak jauh melalui *smartphone*. Selanjutnya terdapat fitur lainnya yang diusulkan oleh penulis, yaitu fitur monitoring yang memungkinkan dapat melakukan baik pemantauan dan pelacakan. Dapat dilihat pada gambar 3.16 flowchart sistem monitoring lokasi, sebagai berikut.



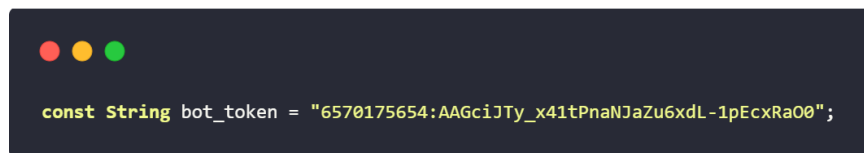
Gambar 3.16 Diagram Alir Sistem Monitoring

Pada alur proses awal fitur monitoring lokasi sama seperti pada tahapan sistem keamanan keseluruhan. Hanya saja pada fitur monitoring lokasi ini, dapat memantau dan melacak. Pada fitur pemantauan dijalankan baik ketika sistem pengaman tidak diaktifkan ataupun fitur pelacakan aktif ketika sistem pengaman

diaktifkan. Untuk membedakan ketika fitur pengaman diaktifkan maka sistem pelacakan yang dijalankan, dan ketika tidak mengaktifkan fitur pengaman maka sistem pemantauan yang diaktifkan.

### 3.3.3 Integrasi Aplikasi Telegram


Dengan memanfaatkan fitur Bot Telegram, sistem ini dapat diintegrasikan melalui *Internet of Things* sehingga pengguna dapat mengoperasikannya dari jarak jauh menggunakan *smartphone*. Sebelum integrasi, bot Telegram harus dirancang terlebih dahulu. Proses pembuatan bot dimulai dengan membuka aplikasi Telegram, mencari "*BotFather*", mengklik menu Start, dan mengikuti langkah-langkah untuk membuat bot baru. Setelah itu, pengguna akan menerima token API HTTP dan *link* akses bot yang dibuat. Token API ini diimplementasikan ke dalam program, seperti yang terlihat pada gambar 3.17.



```
const String bot_token = "6570175654:AAGciJT_yx41tPnaNJaZu6xdL-1pEcXRa00";
```

Gambar 3.17 Script Program Bot Token

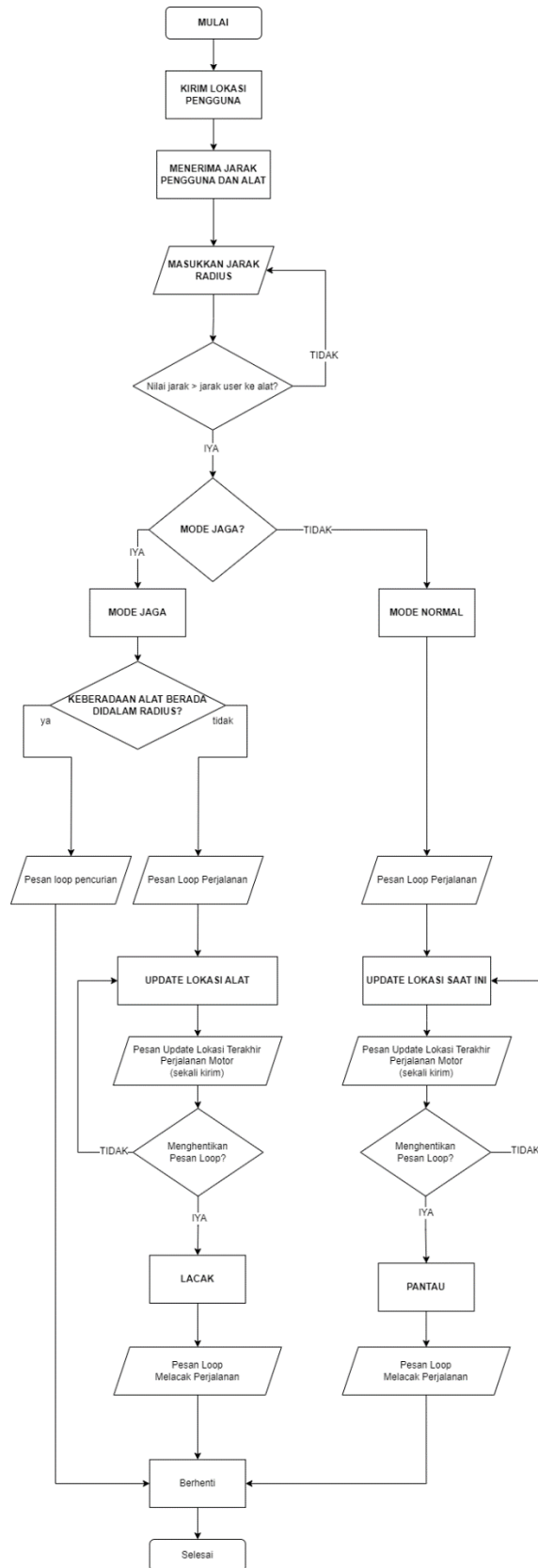
Tahap selanjutnya menambahkan ID Telegram agar pengguna dapat menjalankan bot yang telah dibuat, yaitu pertama pada bagian kolom pencarian masukkan kata "*IDBot*", selanjutnya klik perintah */getid* untuk mendapatkan ID Bot. Kemudian pengguna akan menerima ID. Pada ID yang didapat kemudian diimplementasikan ke dalam program dapat dilihat pada gambar 3.18 script program chat id sebagai berikut.



```
const String chat_id = "5260723107";
```

Gambar 3.18 Script Program Chat ID

Setelah bot Telegram dibuat dan ID chat diimplementasikan, tahap berikutnya adalah menjalankan bot dengan perintah yang telah ditentukan. Pada gambar 3.19 menunjukkan diagram alir integrasi bot, menjelaskan mekanisme dari menerima perintah hingga mengeksekusinya dalam program.



Gambar 3.19 Diagram Alir Sistem Integrasi Bot Telegram

Pada langkah awal pengguna masuk dan memulai kemudian menjalankan bot Telegram dengan memberikan *command* pesan /mulai. Bot Telegram merespon dengan mengirim pesan berupa "**Mulai / Start System**. Kirim Lokasi Anda untuk melanjutkan program. Gunakan Fitur **Send My Current Location** atau Kirim Lokasi Saya", pengguna diminta untuk mengirim lokasinya berada melalui fitur yang tersedia di Telegram "**Send My Current Location** atau Kirim Lokasi Saya", agar lokasi pengguna yang dikirim nantinya akan dijadikan titik awal. Jika modul GPS yang terpasang belum mendapatkan sinyal GPS maka bot Telegram akan merespon dengan mengirim pesan berupa "**Gagal Lokasi User Berhasil**, Motor tidak mendeteksi Sinyal Lokasi GPS" kemudian pengguna diminta untuk mengatur ulang sampai mendapatkan sinyal lokasi GPS.

Namun, ketika sudah berhasil mendapatkan sinyal GPS maka Bot Telegram merespon dengan mengirim pesan balasan berupa "**Update Lokasi Alat Berhasil**. Jarak User dan Motor: ... meter", kemudian pengguna diminta untuk memasukkan jarak radius dengan contoh *command* yang digunakan yaitu **Radius=...** meter, syarat nilai yang dimasukkan lebih dari nilai jarak lokasi alat dengan posisi pengguna berada. Radius yang dimasukkan pengguna kurang dari jarak pengguna dengan alat, maka pengguna diminta untuk memasukkan ulang. Namun, jika radius yang dimasukkan pengguna lebih dari jarak pengguna ke alat maka bot Telegram akan merespon radius yang sudah dimasukkan oleh pengguna berupa pesan "**Berhasil Set Radius GPS**. Radius: ... meter. Pilih Mode; /modenormal, /modejaga".

Pengguna terlebih dahulu memilih opsi mode normal atau mode jaga untuk melanjutkan tahap selanjutnya. Jika pengguna memilih mode jaga maka sistem akan mengaktifkan *geofencing* dan bot juga mengirim pesan berupa "**Berhasil Set Mode Jaga**. Mode Jaga Aktif", jika posisi keberadaan motor belum berpindah atau masih didalam radius *geofencing* maka bot akan merespon dengan berupa pesan "**Pelacakan Alat Aktif**. Jarak User dan Motor: ... meter. Status Relay On, /updatelokasialat" pesan tersebut akan terus terkirim (*loop*), dan untuk memberhentikan pesan *loop* tersebut dengan menggunakan *command* "/updatelokasialat" selanjutnya bot Telegram akan merespon dengan pesan berupa "**Update Lokasi Alat Saat Ini**. Jarak User dan Motor: ... meter. Looping



Dihentikan, /lacak” jika pengguna ingin menjalankan kembali pesan *loop* perjalanan yaitu dengan mengirim *command* “/lacak” maka bot akan mengirim pesan berupa “**Pelacakan Alat Aktif**. Jarak User dan Motor: ... meter. Looping Diaktifkan, /updatelokasialat”. Melanjutkan kasus sebelumnya jika posisi keberadaan motor sudah berada diluar radius *geofencing* maka bot akan merespon dengan berupa pesan peringatan “**Warning! Motor melebihi Radius**. Jarak User dan Motor: ... meter, Status Relay Off” , pesan peringatan tersebut akan terus terkirim dan hanya bisa diberhentikan menggunakan *command* “/berhenti”, selain pesan peringatan *loop* berhenti dan program juga berhenti.

Sistem akan menonaktifkan *Geofencing* ketika pengguna memilih mode normal, kemudian bot akan mengirim pesan berupa “**Berhasil Set Mode Normal**. Mode Normal Aktif”, dan kemudian bot akan mengirim secara terus menerus berupa pesan “**Pemantauan Alat Aktif**. Jarak User dan Motor: ... meter. Status Relay On, /updatelokasisaatini”. jika pengguna ingin memberhentikan pesan *loop* perjalanan tersebut yaitu dengan mengirim *command* “/updatelokasisaatini” maka bot akan mengirim pesan berupa “**Pemantauan Alat Aktif**. Jarak User dan Motor: ... meter. Looping Dihentikan, /pantau”, jika pengguna ingin menjalankan kembali pesan *loop* perjalanan yaitu dengan mengirim *command* “/pantau” maka bot akan mengirim pesan berupa “**Pemantauan Alat Aktif**. Jarak User dan Motor: ... meter. Looping Diaktifkan, /updatelokasisaatini”. Pada mode normal ini hanya berupa pengiriman lokasi saja, jika lokasi terbaru motor lebih dari jarak radius maka tidak terdeteksi pencurian dan bot tidak mengirim pesan peringatan ke pengguna.

Kemudian, tahapan terakhir yaitu jika pengguna ingin memberhentikan program yang dijalankan maka pengguna diminta untuk mengirim *command* “/berhenti” bot akan merespon dengan mengirim pesan berupa “**Berhenti / Stop System**. Untuk Start System kembali, gunakan perintah /mulai” ini menandakan program sudah berhenti dan pengguna mengulang kembali pada tahap mulai.

### 3.4 Pengujian

Proses pengujian dalam penelitian ini akan menggunakan metode black box yaitu pengujian yang dilakukan hanya mengambil hasil eksekutif melalui data uji dan memeriksa fungsionalitas baik dari perangkat lunak maupun perangkat keras (Yordani & Sudaryanto, 2021). Pada tahapan pengujian ini, akan dilakukan

beberapa pembuktian sistem yang berjalan secara keseluruhan. Hal ini dilakukan untuk mengetahui sistem keamanan sepeda motor dapat dijalankan sesuai harapan atau tidak. Pengujian ini meliputi pengujian pada perangkat keras maupun perangkat lunak dan pengujian sistem secara keseluruhan.

### 3.4.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak akan menggunakan *Blackbox* dengan cara mengamati hasil pengoperasian dan memverifikasi fungsionalitas perangkat lunak melalui data pengujian yang berfokus pada kinerja sistem, sehingga dapat mengetahui software berjalan sesuai dengan harapan atau tidak. Pada tabel 3.1 pengujian perangkat lunak, pada proses ini melibatkan berbagai skenario uji untuk memastikan bahwa output yang dihasilkan sesuai dengan yang diharapkan.

Tabel 3.1  
Pengujian Perangkat Lunak

| Aktivitas Pengujian   | Hasil Pengujian yang Diharapkan  |
|---|--|
| Pegguna mulai menjalankan program bot Telegram                  | Bot Telegram merespon dengan mengirim pesan berupa " <b>Mulai / Start System</b> ". Kirim Lokasi Anda untuk melanjutkan program. Gunakan Fitur <b>Send My Current Location</b> atau Kirim Lokasi Saya" |
| Pegguna mengirimkan lokasi nya berada pada bot Telegram         | Bot Telegram merespon dengan mengirim pesan balasan berupa " <b>Update Lokasi Alat Berhasil</b> ". Jarak User dan Motor: ... meter"  |
| Pegguna diminta untuk memasukkan jarak radius pada bot Telegram | Bot Telegram akan merespon radius yang sudah dimasukkan oleh pengguna berupa pesan " <b>Berhasil Set Radius GPS</b> ". Radius: ... meter. Pilih Mode; /modenormal, /modejaga".                         |

|   |  |
|---|--|
| Pengguna memilih /modejaga  | Sistem akan mengaktifkan <i>geofencing</i> dan bot juga mengirim pesan berupa “ <b>Berhasil Set Mode Jaga.</b> Mode Jaga Aktif”  |
| Posisi keberadaan motor belum berpindah atau masih didalam radius <i>geofencing</i> | Bot akan merespon dengan berupa pesan “ <b>Pelacakan Alat Aktif.</b> Jarak User dan Motor: ... meter. Status Relay On, /updatelokasialat”  |
| Posisi keberadaan motor sudah berada diluar radius <i>geofencing</i>                | Bot akan merespon dengan berupa pesan peringatan “ <b>Warning! Motor melebihi Radius.</b> Jarak User dan Motor: ... meter, Status Relay Off”   |
| Command “/updatelokasialat”   | Menghentikan pesan <i>loop</i> “ <b>Pelacakan Alat Aktif</b> ”, Bot Telegram akan merespon dengan pesan berupa “ <b>Update Lokasi Alat Saat Ini.</b> Jarak User dan Motor: ... meter. Looping Dihentikan, /lacak”            |
| Command “/lacak”  | Menjalankan kembali pesan <i>loop</i> “ <b>Pelacakan Alat Aktif</b> ”, Bot Telegram akan merespon dengan pesan berupa “ <b>Pelacakan Alat Aktif.</b> Jarak User dan Motor: ... meter. Looping Diaktifkan, /updatelokasialat” |
| Pengguna memilih /modenormal  | Sistem akan menonaktifkan <i>geofencing</i> dan bot juga mengirim pesan berupa “ <b>Berhasil Set Mode Normal.</b> Mode Normal Aktif”   |

|                                       |   |
|---------------------------------------|---|
|                                       | Bot akan mengirim secara terus menerus berupa pesan “ <b>Pemantauan Alat Aktif</b> . Jarak User dan Motor: ... meter. Status Relay On, /updatelokasisaatini”  |
| Command “/updatelokasisaatini”        | Menghentikan pesan <i>loop</i> “ <b>Pemantauan Alat Aktif</b> ” dan Bot akan mengirim pesan berupa “ <b>Pemantauan Alat Aktif</b> . Jarak User dan Motor: ... meter. Looping Dihentikan, /pantau”                     |
| Command “/pantau”                     | Menjalankan kembali pesan <i>loop</i> “ <b>Pemantauan Alat Aktif</b> ” dan Bot akan mengirim pesan berupa “ <b>Pemantauan Alat Aktif</b> . Jarak User dan Motor: ... meter. Looping Diaktifkan, /updatelokasisaatini” |
| Command “/berhenti”                   | Bot akan merespon dengan mengirim pesan berupa “ <b>Berhenti / Stop System</b> . Untuk Start System kembali, gunakan perintah /mulai”   |
| Penyimpanan Data ke Database Firebase | Menyimpan data ke Database Firebase seperti data lokasi (pengguna, maupun modul GPS), Jarak, dan Radius   |

### 3.4.2 Pengujian Perangkat Keras

Pengujian perangkat keras bertujuan untuk memastikan bahwa setiap komponen berfungsi sesuai dengan tujuan yang diharapkan. Proses ini mencakup pengujian yang dilakukan secara langsung oleh pengguna untuk mengevaluasi hasil yang didapatkan pada masing – masing komponen dalam berbagai kondisi

operasional. Tahapan pengujian perangkat keras dapat dilihat pada tabel 3.2, proses ini melibatkan berbagai skenario untuk menguji berbagai komponen yang terdapat pada alat sistem keamanan sepeda motor.

Tabel 3.2  
Pengujian Perangkat Keras

| No. | Komponen               | Hasil yang Diharapkan   |
|-----|------------------------|---|
| 1.  | Modul GPS Ublox Neo 6M | Modul GPS Ublox Neo 6M dapat membaca latitude dan longitude.                      |
| 2.  | Modul GSM SIM800L V2   | Modul GSM SIM800L V.2 dapat mengirim data melalui sinyal internet jaringan GPRS   |
| 3.  | Relay                  | Relay dapat memutuskan dan menghubungkan kembali arus tegangan dari power supply. |

### 3.4.3 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dengan menyusun skenario pengujian terlebih dahulu. Bertujuan untuk mendapatkan hasil yang diperoleh dan mengevaluasi hasil yang didapat dari beberapa usulan oleh penulis dalam perancangan sistem keamanan pencurian sepeda memanfaatkan *Geofencing* dan penggunaan GPS yang diintegrasikan *Internet of Things* (IoT) melalui Aplikasi Telegram. Berikut skenario pengujian yang dilakukan:

1. Pengujian terhadap hasil rancangan implementasi sistem monitoring lokasi sepeda motor dilakukan dengan menguji keakuratan lokasi jarak dari titik awal ke titik terkini yang terbaca oleh modul GPS jika dibandingkan dengan lokasi di Google Maps.
2. Pengujian terhadap penerapan teknik *Geofencing* pada sistem monitoring lokasi sepeda motor dilakukan dengan menganalisis tingkat keberhasilan respon notifikasi melalui aplikasi Telegram dan sistem mematikan mesin sepeda motor secara otomatis ketika sepeda motor berada diluar area yang telah ditentukan.

3. Pengujian terhadap kinerja sistem keamanan pencegahan pencurian sepeda motor menggunakan metode *Geofencing* dan *GPS Tracking* melalui Telegram dilakukan dengan menganalisis hasil evaluasi keseluruhan mulai dari tingkat keakuratan dalam monitoring lokasi, sampai tingkat keberhasilan respon notifikasi melalui Telegram dan sistem mematikan mesin sepeda motor secara otomatis ketika berada diluar area yang telah ditentukan.

Beberapa skenario tersebut dapat membantu dalam memvalidasi hasil kemampuan pada sistem dalam berbagai kondisi operasional, dan memastikan bahwa sistem keamanan sepeda motor yang diusulkan penulis diharapkan dapat mendeteksi pencurian sepeda motor serta dapat memonitoring lokasi dan diintegrasikan melalui Aplikasi Telegram.

### **3.5 Evaluasi**

Setelah pengujian dilakukan, tahapan selanjutnya yaitu analisis terhadap data yang diperoleh dari hasil pengujian dan mengevaluasi dari beberapa aspek pengujian yang telah dilakukan dengan membandingkan hasil atau *output* yang diharapkan berdasarkan *input* yang diberikan pada sistem. Evaluasi dilakukan untuk mendapatkan keakuratan, dan fungsionalitas setiap komponen yang berjalan dengan baik.