

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap ini merupakan proses realisasi rancangan *GPS Tracker* berbasis *Internet of Things (IoT)* yang telah dibahas pada bab sebelumnya. Implementasi dilakukan melalui integrasi perangkat keras dan perangkat lunak untuk memastikan sistem dapat bekerja secara optimal dalam mendeteksi dan mengirimkan lokasi kendaraan bermotor secara *real-time* melalui pesan SMS.

Sistem yang dikembangkan bertujuan memberikan solusi keamanan tambahan pada sepeda motor dengan memanfaatkan modul GPS Neo-6M untuk penentuan posisi, modul GSM SIM800L V2 sebagai media pengiriman data, serta mikrokontroler Arduino Uno sebagai pusat pengendali. Proses implementasi dilakukan melalui dua tahapan utama, yaitu penyusunan perangkat keras (*hardware*) dan pengembangan perangkat lunak (*software*).

4.1.1 Perangkat Keras (Hardware)

Perangkat keras berperan penting dalam melakukan akuisisi data lokasi serta mengirimkan informasi tersebut kepada pengguna. Setiap komponen memiliki fungsi khusus yang saling mendukung agar sistem dapat bekerja sesuai perancangan. Tabel berikut menunjukkan daftar perangkat keras yang digunakan dalam penelitian ini:

Tabel 4. 1 Perangkat Keras

No	Komponen	Fungsi
1	Arduino Uno	Mengolah data dari modul GPS dan mengirimkan perintah ke modul GSM untuk

		mengirim SMS lokasi
2	Modul GPS Neo-6M	Menerima sinyal satelit dan menentukan koordinat lokasi (latitude & longitude) kendaraan
4	Modul GSM SIM800L V2	Mengirimkan data lokasi ke ponsel pengguna dalam bentuk SMS
5	Baterai Li-Ion 3,7V (2 unit)	Sumber daya utama perangkat GPS Tracker
6	Modul TP4056	Mengatur proses pengisian baterai dan mencegah <i>overcharging</i>
7	Project Box	Melindungi rangkaian dari gangguan fisik dan lingkungan.
8	Kabel Jumper	Menghubungkan komponen dalam rangkaian elektronik
9	Modul Relay 1 Channel	Memutus atau menghubungkan arus listrik ke sistem pengapian motor berdasarkan perintah SMS pengguna

Perakitan dimulai dengan menghubungkan modul GPS Neo-6M ke Arduino Uno melalui komunikasi serial menggunakan pin digital yang ditentukan, diikuti pemasangan modul GSM SIM800L V2 pada port yang berbeda untuk menghindari konflik data. Sumber daya diatur melalui rangkaian power supply yang terhubung dengan modul TP4056 agar baterai dapat diisi ulang dengan aman. Semua komponen ditempatkan di dalam project box untuk melindungi perangkat dari benturan dan memudahkan pemasangan pada kendaraan.

Modul relay dihubungkan ke pin digital Arduino Uno dan dikonfigurasi agar aktif saat menerima perintah tertentu dari pengguna melalui SMS perintah "MATIKAN". Relay akan memutus arus ke koil pengapian motor sehingga mesin

berhenti. Sebaliknya, perintah "HIDUPKAN" akan mengaktifkan kembali arus dan memungkinkan motor dinyalakan.

4.1.2 Perangkat Lunak (*Software*)

Perangkat lunak berperan sebagai pengatur logika kerja sistem, mulai dari pembacaan koordinat *GPS*, pengolahan data, hingga pengiriman SMS berisi tautan Google Maps kepada pengguna. Pengembangan dilakukan menggunakan Arduino IDE dengan bahasa pemrograman *C/C++*.

1. Perangkat Lunak Utama

Arduino IDE digunakan untuk menulis, mengompilasi, dan mengunggah kode program ke mikrokontroler Arduino Uno. Lingkungan pengembangan ini dipilih karena kompatibilitasnya yang tinggi terhadap modul GPS dan GSM, serta dukungan *library* yang luas untuk mempermudah integrasi.

2. *Library* Pendukung

Agar sistem berfungsi dengan baik, digunakan beberapa *library* berikut:

- a. *SoftwareSerial.h* → Membuat port serial tambahan untuk komunikasi terpisah antara Arduino dan modul GPS/GSM.
- b. *TinyGPS++.h* → Mengolah data mentah dari GPS menjadi format koordinat yang mudah digunakan.

3. Perangkat Lunak Pendukung Dokumentasi Sistem

Selain pengembangan sistem inti, beberapa aplikasi tambahan digunakan untuk membuat dokumentasi visual guna menjelaskan struktur dan alur sistem secara lebih jelas, yaitu:

- a. *Fritzing* : Membuat skema rangkaian elektronik perangkat *GPS Tracker* dalam mode *breadboard* maupun *schematic*.

- b. *Microsoft Visio* : Menggambarkan *flowchart* proses kerja sistem mulai dari pencarian sinyal *GPS* hingga pengiriman SMS lokasi.

Penggunaan *Fritzing* dan *Microsoft Visio* membantu memberikan gambaran yang lebih *komprehensif* terhadap rancangan sistem, baik dari sisi fisik (rangkaian) maupun logika (alur kerja), sehingga mempermudah proses implementasi maupun penjelasan dalam laporan.

4.2 Rangkaian Sistem

Rangkaian sistem pada penelitian ini dirancang untuk menghubungkan seluruh komponen perangkat keras *GPS Tracker* sehingga dapat bekerja secara terintegrasi dalam memantau lokasi kendaraan dan mengendalikan mesin dari jarak jauh. Arsitektur yang digunakan membagi tugas setiap modul: Arduino Uno R3 berperan sebagai pusat pengendali, *modul GPS Neo-6M* berfungsi membaca koordinat lokasi, *modul GSM SIM800L V2* digunakan untuk mengirim/menerima perintah melalui SMS, dan modul *relay* digunakan untuk memutus atau menyambung arus kelistrikan pengapian motor.

Perancangan rangkaian dilakukan melalui dua tahap, yaitu:

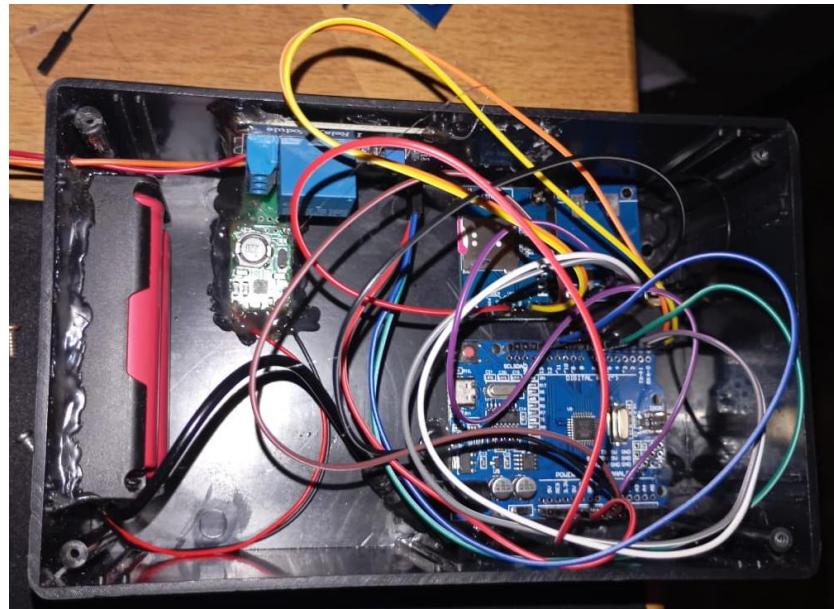
1. Pembuatan desain rangkaian virtual menggunakan *Fritzing* untuk memastikan koneksi antar komponen sesuai.
2. Perakitan fisik di dalam project box untuk melindungi perangkat saat dipasang pada kendaraan, sekaligus memudahkan proses instalasi dan perawatan.

Gambar 4.1 berikut menunjukkan tampilan perangkat *GPS Tracker* dari sisi depan setelah dipasang di dalam project box.



Gambar 4. 1 Tampilan Perangkat GPS Tracker dari Depan

Selanjutnya, Gambar 4.2 memperlihatkan tata letak rangkaian di dalam *project box* yang berisi Arduino *Uno R3*, GPS *Neo-6M*, GSM *SIM800L*, *relay*, modul catu daya, dan baterai. Dokumentasi ini menunjukkan bagaimana seluruh komponen diatur agar tetap rapi, aman, dan tidak mudah bergeser saat kendaraan digunakan.



Gambar 4. 2 Rangkaian Perangkat di Dalam *Project Box*

Penjelasan Fungsi Masing-Masing Modul:

1. Modul GPS Neo-6M

Digunakan untuk membaca koordinat lokasi (*latitude & longitude*) kendaraan dengan menerima sinyal dari minimal tiga satelit. Data yang diperoleh dalam format *NMEA* kemudian diproses oleh Arduino menggunakan *library* *TinyGPS++* untuk menghasilkan informasi lokasi yang akurat.

2. Arduino Uno

Berfungsi sebagai mikrokontroler utama yang:

- a. Membaca data koordinat dari *GPS Neo-6M*.
- b. Menerima perintah SMS dari pengguna melalui modul *GSM*.
- c. Mengirimkan balasan SMS berisi tautan Google Maps.
- d. Mengendalikan *modul relay* untuk memutus/menyambung arus kelistrikan pengapian motor sesuai perintah SMS.

3. Modul GSM SIM800L V2

Bertugas sebagai media komunikasi dengan pengguna melalui jaringan *GSM*.

Modul ini menerima perintah seperti “*FIND*” untuk melacak lokasi atau “*MATIKAN*”/“*HIDUPKAN*” untuk mematikan/menyalakan mesin, dan mengirimkan informasi kembali dalam bentuk SMS.

4. Modul Relay 1 Channel 5V

Mengendalikan aliran listrik ke sistem pengapian motor. Saat menerima perintah “*MATIKAN*” melalui SMS, relay memutus arus sehingga mesin berhenti. Perintah “*HIDUPKAN*” akan mengaktifkan kembali arus pengapian.

5. Modul TP4056

Digunakan sebagai pengatur pengisian *baterai Li-Ion*. Modul ini mencegah *overcharging* dan memastikan baterai terisi dengan aman melalui *port micro-USB*.

6. Baterai Li-Ion 3,7V (2 Unit Seri)

Menyediakan sumber daya sebesar 7,4V yang disalurkan ke Arduino dan seluruh modul pendukung melalui rangkaian *power supply*.

7. Breadboard & Kabel Jumper

Digunakan untuk perakitan *prototipe*; kabel *jumper* mempermudah koneksi antar komponen.

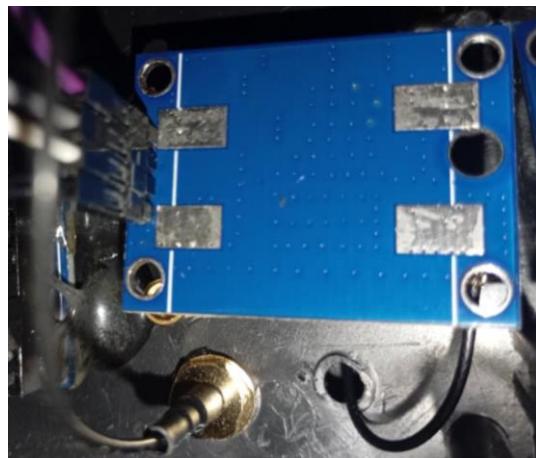
8. Project Box

Menjadi wadah pelindung bagi semua komponen, menjaga perangkat dari benturan, debu, dan cipratan air saat dipasang pada kendaraan.

4.2.1 Rangkaian Modul GPS Neo-6M

Modul GPS Neo-6M digunakan untuk menerima sinyal dari satelit dan menghasilkan data koordinat lokasi (*latitude* dan *longitude*) yang kemudian dikirim ke Arduino melalui komunikasi *serial*. Data yang diperoleh masih dalam format

NMEA dan diolah menggunakan *library TinyGPS++* agar dapat dibaca serta dikirimkan ke pengguna melalui SMS.



Gambar 4. 3 Rangkaian *Modul GPS Neo-6M*

Keterangan koneksi:

- a. VCC (Neo-6M) → 5V Arduino
- b. GND → GND Arduino
- c. TX (Neo-6M) → Pin 4 Arduino (RX pada SoftwareSerial)
- d. RX (Neo-6M) → Pin 5 Arduino (TX pada SoftwareSerial)

Komunikasi data antara *GPS Neo-6M* dan Arduino dilakukan menggunakan *protokol UART* pada *baud rate 9600 bps*. Pemilihan pin 4 dan 5 untuk komunikasi ini dilakukan dengan memanfaatkan *library SoftwareSerial*, sehingga *port serial* bawaan Arduino tetap bisa digunakan untuk proses *debugging* melalui *USB*.

Untuk mendapatkan hasil pembacaan lokasi yang akurat, *modul GPS* harus diletakkan pada posisi yang memungkinkan antena menangkap sinyal satelit secara optimal. Area terbuka seperti bagian atas kendaraan akan mempercepat proses *satellite lock*. Waktu yang dibutuhkan modul untuk mengunci posisi pertama kali (*cold start*) bisa memakan waktu 30–60 detik, sedangkan jika modul baru saja aktif kembali (*warm start*) biasanya hanya memerlukan waktu beberapa detik.

Dalam implementasi ini, data koordinat yang diperoleh dari *GPS Neo-6M* akan dikirim ke fungsi `kirimLokasi()` pada program Arduino. Fungsi ini bertugas memformat data menjadi tautan Google Maps sehingga pengguna dapat langsung melihat lokasi kendaraan hanya dengan membuka tautan tersebut di ponsel.

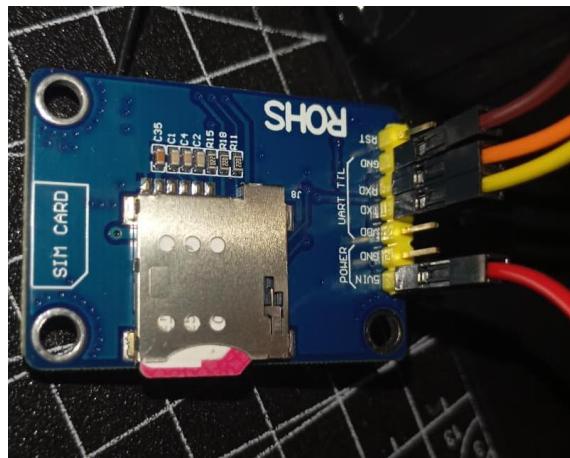
4.2.2 Rangkaian Modul GSM SIM800L V2

Modul GSM SIM800L V2 digunakan sebagai sarana komunikasi antara pengguna dan perangkat *GPS Tracker* melalui layanan *SMS (Short Message Service)*. Modul ini berfungsi menerima perintah teks dari pengguna, seperti CEK untuk mengetahui lokasi terkini, MATIKAN untuk memutus arus kelistrikan mesin, HIDUPKAN untuk mengaktifkan kembali mesin, serta STATUS untuk mengecek kondisi kendaraan. Selain menerima perintah, *SIM800L* juga mengirimkan balasan berisi informasi yang diminta, misalnya koordinat *GPS* dalam bentuk tautan Google Maps atau status kelistrikan mesin.

Komunikasi antara Arduino Uno dan *SIM800L* dilakukan melalui jalur serial menggunakan *library SoftwareSerial*. Hal ini dilakukan agar *port serial* bawaan Arduino tetap tersedia untuk keperluan *debugging* melalui koneksi USB. Pin 2 Arduino digunakan sebagai *RX* untuk menerima data dari *SIM800L*, sedangkan pin 3 digunakan sebagai *TX* untuk mengirim data ke modul.

Karena *SIM800L V2* bekerja optimal pada tegangan sekitar 3,7V hingga 4,2V, diperlukan catu daya khusus yang stabil untuk menghindari kerusakan modul akibat tegangan berlebih. Pada implementasi ini digunakan *step-down regulator* (*buck converter*) untuk menurunkan tegangan dari *baterai Li-Ion 7,4V* menjadi sekitar 4V. Selain itu, modul ini membutuhkan suplai arus yang cukup besar saat proses inisialisasi jaringan *GSM* (*burst current ±2A*), sehingga jalur daya harus

menggunakan kabel yang cukup tebal dan konektor yang kokoh untuk menghindari penurunan tegangan mendadak.



Gambar 4. 4 Rangkaian *GSM SIM800L V2*

Keterangan koneksi:

- a. VCC (SIM800L) → *Output 4,0V* dari *step-down regulator* (tidak langsung ke 5V Arduino)
 - b. GND → *GND* Arduino dan *GND* baterai (*common ground*)
 - c. TX (SIM800L) → Pin 2 Arduino (*RX SoftwareSerial*)
 - d. RX (SIM800L) → Pin 3 Arduino (*TX SoftwareSerial*)
 - e. ANTENA → Antena eksternal bawaan modul untuk memperkuat sinyal GSM

Pada saat perangkat dinyalakan, SIM800L akan secara otomatis mencari jaringan GSM terdekat. LED indikator bawaan modul akan berkedip cepat saat mencari jaringan, dan berkedip lambat jika sudah berhasil terhubung. Setelah koneksi GSM siap, perangkat dapat mengirim dan menerima SMS sesuai perintah pengguna.

Agar modul bekerja secara optimal di lapangan, antena harus diposisikan pada area yang bebas dari penghalang logam dan sedapat mungkin berada di lokasi

yang terbuka. Pada implementasi di kendaraan, antena biasanya dipasang di area tersembunyi tetapi tetap terbuka terhadap sinyal, seperti di bawah jok motor namun tidak terbungkus logam tebal.

4.2.3 Rangkaian Modul Relay 1 Channel

Modul relay 1 channel digunakan sebagai *aktuator* yang mengendalikan arus listrik menuju sistem pengapian kendaraan. *Relay* ini berfungsi memutus atau menyambungkan kelistrikan mesin berdasarkan perintah SMS yang dikirim oleh pengguna melalui modul *GSM SIM800L*. Dengan adanya *relay*, pengguna dapat mematikan mesin dari jarak jauh (*remote engine cut-off*) untuk mencegah pencurian atau menghentikan kendaraan yang digunakan tanpa izin.

Relay bekerja layaknya *saklar elektrik* yang dikendalikan oleh sinyal digital dari Arduino Uno. Pada sistem ini, *relay* dihubungkan ke pin digital 7 Arduino. Saat pin 7 diberi logika *HIGH*, *relay* akan aktif (kontak *NO* tertutup) dan arus listrik ke pengapian motor tersambung, sehingga mesin dapat hidup. Sebaliknya, saat pin 7 diberi logika *LOW*, *relay* akan nonaktif (kontak *NO* terbuka) dan arus pengapian terputus, membuat mesin tidak dapat dinyalakan.



Gambar 4. 5 Rangkaian *Relay 1 Channel*

Keterangan koneksi:

- a. VCC (Relay) → 5V Arduino

- b. GND → GND Arduino
- c. IN → Pin 7 Arduino (sebagai *trigger* kontrol *relay*)
- d. COM & NO/NC → Terhubung ke jalur pengapian motor:
 - 1. COM (*Common*) dihubungkan ke sumber arus dari aki menuju sistem pengapian.
 - 2. NO (*Normally Open*) digunakan agar arus hanya mengalir saat *relay* aktif (mesin ON).
 - 3. NC (*Normally Closed*) dapat digunakan jika ingin logika terbalik (jarang dipakai di sistem *cut-off*).

Modul relay memiliki *optocoupler internal* yang berfungsi sebagai *isolasi* antara rangkaian kontrol Arduino dengan rangkaian kelistrikan motor, sehingga gangguan tegangan tinggi dari motor tidak merusak *mikrokontroler*. Untuk mengurangi percikan listrik (*electrical arc*) pada saat memutus arus pengapian, disarankan menggunakan *relay* dengan kapasitas minimal $10A/250VAC$ atau yang sesuai dengan spesifikasi kelistrikan motor.

Dalam implementasi program, logika kontrol *relay* diatur melalui fungsi matikanMotor() dan hidupkanMotor(). Ketika menerima perintah SMS "MATIKAN", Arduino akan memberikan logika *LOW* ke pin 7 sehingga *relay* terbuka dan mesin mati. Sebaliknya, perintah "HIDUPKAN" akan memberi logika *HIGH* pada pin 7, menutup kontak *relay* dan menghubungkan kembali arus pengapian. Status motor (HIDUP atau MATI) juga dikirimkan kembali ke pengguna melalui *SMS* sebagai konfirmasi.

Penggunaan *relay* pada *GPS Tracker* ini menambah fungsi keamanan karena tidak hanya memberikan informasi lokasi kendaraan, tetapi juga mampu

mengendalikan kondisi mesin secara langsung dari jarak jauh. Hal ini menjadi nilai tambah yang signifikan dibandingkan sistem pelacakan biasa yang hanya bersifat pasif.

4.2.4 Modul Catu Daya & Baterai

Catu daya merupakan bagian penting dalam sistem *GPS Tracker* ini karena berfungsi menyediakan tegangan dan arus yang stabil untuk seluruh komponen. Sistem ini menggunakan *baterai Li-Ion 2 sel* ($2 \times 3,7V$) yang disusun seri sehingga menghasilkan tegangan nominal 7,4V. Baterai ini dipilih karena memiliki kepadatan energi tinggi, ukuran kompak, dan mampu menyuplai arus cukup besar untuk *modul GSM SIM800L* yang membutuhkan *burst current* hingga 2A.

Tegangan dari baterai kemudian diatur menggunakan *step-down (buck) converter* menjadi 5V sesuai kebutuhan arduino dan modul lainnya. Selain *step-down converter*, digunakan juga *modul TP4056* sebagai pengatur proses pengisian ulang baterai. Modul ini dilengkapi dengan fitur *overcharge protection* dan *overdischarge protection*, sehingga baterai terlindungi dari kerusakan akibat pengisian berlebihan atau pengosongan berlebih. Proses pengisian dilakukan melalui *port micro-USB* pada *modul TP4056*, yang dapat menggunakan *adaptor 5V* seperti charger ponsel.



Gambar 4. 6 Rangkaian Catu Daya & Baterai

Keterangan koneksi:

- a. Baterai Li-Ion 2S (7,4V) → Input ke modul TP4056 untuk pengisian.
- b. Output TP4056 → Input step-down converter.
- c. Output step-down 5V → Arduino Uno, GPS Neo-6M, Relay.
- d. Output step-down 4V → GSM SIM800L V2.
- e. GND semua modul → Disatukan (common ground) untuk menghindari perbedaan referensi tegangan.

Stabilitas catu daya sangat mempengaruhi kinerja sistem. Apabila tegangan *drop* saat *modul GSM* melakukan *transmisi SMS*, perangkat dapat *restart* atau gagal mengirim pesan. Oleh karena itu, pemilihan baterai dengan kapasitas cukup besar (misalnya 2200mAh atau lebih) sangat dianjurkan untuk memastikan perangkat dapat beroperasi dalam waktu lama tanpa pengisian ulang.

Penempatan baterai di dalam *project box* harus mempertimbangkan keamanan dan ventilasi. Baterai tidak boleh tertekan atau terkena panas berlebih dari komponen lain. Selain itu, seluruh jalur daya diberikan isolasi yang baik untuk mencegah hubungan arus pendek (*short circuit*).

4.3 Implementasi Program Sistem

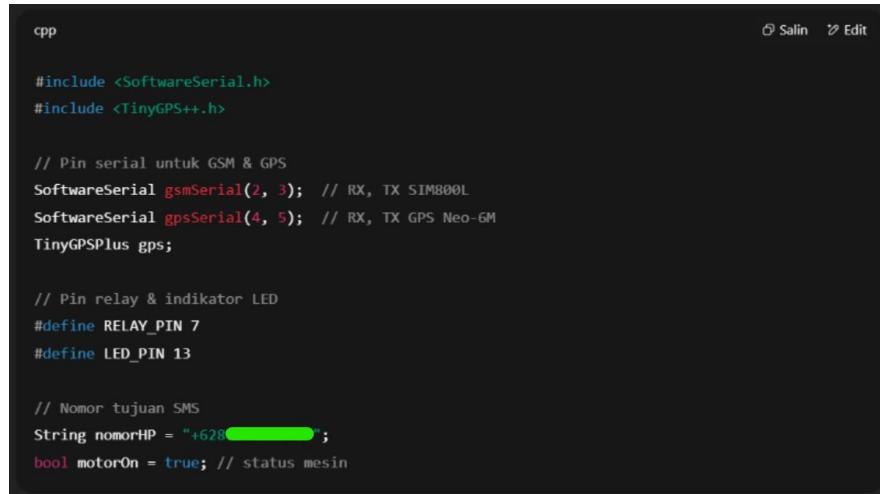
Tahap implementasi perangkat lunak pada sistem *GPS Tracker* ini dilakukan menggunakan bahasa pemrograman *C++* dengan bantuan Arduino IDE. Program yang dibuat berfungsi untuk mengatur interaksi antara seluruh komponen: *GPS Neo-6M*, *GSM SIM800L*, *relay*, dan catu daya.

Struktur program dibagi menjadi beberapa bagian utama:

- a. Inisialisasi *library* dan pin
- b. Fungsi *setup()* untuk inisialisasi awal
- c. *Loop* utama untuk membaca SMS dan memproses perintah
- d. Fungsi untuk mengirim lokasi, mematikan/menghidupkan mesin, serta mengirim status
- e. Fungsi utilitas seperti pengiriman SMS dan penghapusan SMS lama

4.3.1 Inisialisasi *Library* dan Pin

Bagian ini memuat pemanggilan *library* dan pendefinisian pin yang digunakan. *Library SoftwareSerial* digunakan untuk membuat komunikasi serial tambahan untuk *GSM* dan *GPS*, sementara *TinyGPS++* digunakan untuk mengolah data lokasi dari *GPS*.



```

cpp

#include <SoftwareSerial.h>
#include <TinyGPS++.h>

// Pin serial untuk GSM & GPS
SoftwareSerial gsmSerial(2, 3); // RX, TX SIM800L
SoftwareSerial gpsSerial(4, 5); // RX, TX GPS Neo-6M
TinyGPSPlus gps;

// Pin relay & indikator LED
#define RELAY_PIN 7
#define LED_PIN 13

// Nomor tujuan SMS
String nomorHP = "+628XXXXXX";
bool motorOn = true; // status mesin

```

Gambar 4. 7 Inisialisasi Library

Pada bagian ini juga dilakukan inisialisasi variabel global yang digunakan untuk menyimpan status mesin, nomor tujuan SMS, dan objek komunikasi serial untuk *GSM* serta *GPS*.

4.3.2 Fungsi *setup()*

Fungsi ini dijalankan sekali saat perangkat dinyalakan. Tujuannya adalah menginisialisasi komunikasi *serial*, mengatur mode SMS pada *SIM800L*, serta mengatur status awal *relay* (mesin aktif).

```

void setup() {
    Serial.begin(9600);

    pinMode(RELAY_PIN, OUTPUT);
    pinMode(LED_PIN, OUTPUT);

    // Status awal motor hidup
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);

    gsmSerial.begin(9600);
    gpsSerial.begin(9600);

    // Inisialisasi GSM untuk mode SMS
    gsmSerial.println("AT");
    gsmSerial.println("AT+CMGF=1");
    gsmSerial.println("AT+CNMI=1,2,0,0");
    gsmSerial.println("AT+CMGDA=\"DEL ALL\"");
}

```

Gambar 4. 8 Fungsi *Setup()*

4.3.3 Loop Utama

Loop utama (*loop()*) merupakan bagian inti dari program yang dieksekusi berulang-ulang selama perangkat GPS Tracker aktif. Pada bagian ini, sistem secara terus-menerus memantau apakah ada pesan SMS yang masuk melalui modul GSM SIM800L.

```

void loop() {
    gsmSerial.listen();
    if (gsmSerial.available()) {
        String sms = gsmSerial.readString();
        sms.toUpperCase();
        sms.trim();

        if (sms.indexOf("CEK") >= 0) kirimLokasi();
        else if (sms.indexOf("MATIKAN") >= 0) matikanMotor();
        else if (sms.indexOf("HIDUPKAN") >= 0) hidupkanMotor();
        else if (sms.indexOf("STATUS") >= 0) kirimStatus();

        hapusSMS();
    }
}

```

Gambar 4. 9 *Void Loop()*

Proses kerja dimulai dengan mengaktifkan listener pada `gsmSerial` untuk memastikan data yang diterima berasal dari jalur komunikasi GSM. Ketika data SMS terdeteksi, program akan membacanya dalam bentuk string, kemudian mengubah seluruh huruf menjadi kapital (uppercase) untuk memudahkan proses pencocokan perintah. Hal ini penting karena pengguna dapat mengetik perintah dengan variasi huruf besar/kecil, dan sistem tetap dapat mengenaliinya.

Setelah itu, program melakukan pencarian kata kunci perintah menggunakan metode `indexOf()`. Jika ditemukan kata kunci "CEK", sistem akan memanggil fungsi `kirimLokasi()` untuk mengirim koordinat GPS kepada pengguna. Perintah "MATIKAN" akan memanggil fungsi `matikanMotor()` untuk memutus arus pengapian, sedangkan "HIDUPKAN" memanggil fungsi `hidupkanMotor()` untuk mengaktifkan kembali arus pengapian. Perintah "STATUS" akan mengirimkan kondisi mesin saat ini melalui fungsi `kirimStatus()`.

Apabila perintah tidak dikenali, sistem tetap menghapus SMS tersebut menggunakan fungsi `hapusSMS()` untuk mencegah penumpukan pesan di memori SIM card, yang berpotensi mengganggu penerimaan pesan berikutnya. Dengan mekanisme ini, loop utama bertindak sebagai pusat pengambilan keputusan yang menentukan alur kerja sistem berdasarkan input yang diterima.

4.3.4 Fungsi `kirimLokasi()`

Fungsi `kirimLokasi()` memiliki peran vital dalam menyampaikan posisi kendaraan kepada pengguna. Saat fungsi ini dipanggil, proses dimulai dengan mengaktifkan *listener* pada `gpsSerial` agar Arduino dapat membaca data mentah (*raw data*) dari *modul GPS Neo-6M*.

Data *GPS* biasanya dikirim dalam format *NMEA* yang memuat berbagai

informasi, seperti koordinat, kecepatan, dan waktu. Namun, tidak semua data tersebut dibutuhkan dalam aplikasi ini. Oleh karena itu, digunakan library *TinyGPS++* untuk mengekstrak hanya informasi yang relevan, yaitu *latitude* dan *longitude*.

```

void kirimLokasi() {
    unsigned long startTime = millis();
    bool lokasiDidapat = false;
    float lat, lng;

    while (millis() - startTime < 15000) {
        gpsSerial.listen();
        while (gpsSerial.available()) {
            gps.encode(gpsSerial.read());
        }
        if (gps.location.isUpdated()) {
            lat = gps.location.lat();
            lng = gps.location.lng();
            lokasiDidapat = true;
            break;
        }
    }

    String pesan;
    if (lokasiDidapat) {
        String link = "https://maps.google.com/?q=" + String(lat, 6) + "," + String(lng, 6);
        String statusMotor = motorOn ? "HIDUP" : "MATI";
        pesan = "LOKASI MOTOR:\n" + link + "\nSTATUS: " + statusMotor;
    } else {
        pesan = "GPS tidak tersedia!\nSTATUS MOTOR: " + (motorOn ? "HIDUP" : "MATI");
    }
    kirimSMS(pesan);
}

```

Gambar 4. 10 *Void KirimLokasi()*

Fungsi ini juga dilengkapi mekanisme batas waktu (*timeout*) selama 15 detik. Apabila dalam waktu tersebut modul *GPS* tidak berhasil mendapatkan sinyal (*satellite lock*), maka sistem akan mengirimkan pesan peringatan kepada pengguna bahwa koordinat tidak tersedia, namun tetap menyertakan status mesin saat ini.

Jika koordinat berhasil diperoleh, fungsi akan memformatnya menjadi tautan Google Maps agar pengguna cukup mengklik tautan tersebut untuk melihat lokasi kendaraan secara visual. Langkah ini mempermudah pengguna yang mungkin tidak familiar dengan membaca angka koordinat mentah.

Pengiriman pesan lokasi dilakukan melalui fungsi *kirimSMS()* yang bertugas mengirimkan pesan ke nomor HP pengguna. Dengan demikian, *kirimLokasi()* berfungsi sebagai jembatan antara data *GPS* mentah dan informasi lokasi yang mudah dipahami oleh pengguna.

4.3.5 Fungsi Kontrol Mesin (Relay)

Bagian ini terdiri dari dua fungsi, yaitu *matikanMotor()* dan *hidupkanMotor()*, yang bertugas mengendalikan kondisi mesin kendaraan melalui *modul relay*.

1. Fungsi *matikanMotor()* akan memberikan logika *LOW* pada pin *relay*, sehingga kontak *relay* terbuka dan arus pengapian terputus. Kondisi ini membuat mesin kendaraan tidak dapat dinyalakan meskipun kunci kontak dalam posisi ON. Fungsi ini diakhiri dengan pengiriman SMS konfirmasi kepada pengguna untuk memastikan bahwa perintah telah dijalankan.
2. Fungsi *hidupkanMotor()* sebaliknya akan memberikan logika *HIGH* pada pin *relay*, menutup kontak *relay* sehingga arus pengapian kembali tersambung dan mesin dapat dinyalakan. SMS konfirmasi dikirim untuk memberikan kepastian kepada pengguna bahwa mesin telah diaktifkan.

```
void matikanMotor() {
    digitalWrite(RELAY_PIN, LOW);
    digitalWrite(LED_PIN, LOW);
    motorOn = false;
    kirimSMS("MOTOR DIMATIKAN! Tidak dapat dinyalakan.");
}

void hidupkanMotor() {
    digitalWrite(RELAY_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);
    motorOn = true;
    kirimSMS("MOTOR DIHIDUPKAN! Siap digunakan.");
}
```

Gambar 4. 11 Kontrol Mesin (Relay)

Penggunaan *relay* sebagai aktuator memiliki keunggulan karena dapat memutus arus listrik berdaya besar dengan kontrol sinyal berdaya kecil dari Arduino. Dengan demikian, kedua fungsi ini tidak hanya berperan dalam operasional sistem, tetapi juga menjadi elemen penting dalam fitur keamanan *GPS Tracker*.

4.3.6 Fungsi kirimStatus()

Fungsi kirimStatus() bertujuan untuk memberikan informasi terkini mengenai kondisi mesin kendaraan kepada pengguna. Status mesin disimpan dalam variabel *motorOn* yang bernilai *true* jika mesin dapat dinyalakan, dan *false* jika mesin dalam kondisi mati.

```
void kirimStatus() {
    String pesan = "STATUS MOTOR: " + String(motorOn ? "HIDUP" : "MATI");
    pesan += motorOn ? "\nMotor siap digunakan" : "\nMotor tidak dapat dinyalakan";
    kirimSMS(pesan);
}
```

Gambar 4. 12 Void kirimStatus()

Ketika fungsi ini dipanggil, program akan memeriksa nilai *motorOn* dan membentuk pesan sesuai kondisinya. Jika mesin aktif, pesan akan menyatakan "STATUS MOTOR: HIDUP" dan memberi keterangan bahwa motor siap digunakan. Jika mesin mati, pesan akan menyatakan "STATUS MOTOR: MATI" beserta keterangan bahwa motor tidak dapat dinyalakan.

Fungsi ini berguna dalam situasi ketika pengguna ingin memastikan kondisi mesin sebelum mengambil tindakan lain, seperti menghidupkan atau mematikan dari jarak jauh. Hal ini juga membantu menghindari perintah ganda yang tidak diperlukan

4.3.7 Fungsi kirimSMS() dan hapusSMS()

Fungsi kirimSMS() bertanggung jawab untuk mengirimkan pesan teks ke nomor tujuan yang telah ditentukan. Proses pengiriman diawali dengan perintah AT AT+CMGS yang diikuti dengan nomor HP penerima. Setelah jeda singkat untuk memberi waktu modul GSM memproses perintah, pesan teks dikirim, kemudian diakhiri dengan karakter CTRL+Z (ASCII 26) sebagai tanda bahwa pesan telah selesai diketik.

Fungsi hapusSMS() digunakan untuk membersihkan seluruh pesan masuk pada SIM card menggunakan perintah AT+CMGDA="DEL ALL". Hal ini penting karena memori penyimpanan SMS pada modul GSM umumnya sangat terbatas (sekitar 20 pesan). Dengan rutin menghapus SMS yang sudah diproses, sistem terhindar dari kegagalan menerima perintah baru akibat memori penuh.

Delay diberikan di beberapa titik untuk memastikan modul GSM memiliki cukup waktu memproses perintah, mengingat pengiriman SMS melalui jaringan seluler dapat memerlukan sinkronisasi sinyal. Respons dari modul dapat dibaca kembali untuk memverifikasi keberhasilan pengiriman.

```

void kirimSMS(String pesan) {
    gsmSerial.println("AT+CMGS=\"" + nomorHP + "\"");
    delay(1000);
    gsmSerial.println(pesan);
    gsmSerial.write(26); // CTRL+Z
    delay(3000);
}

void hapusSMS() {
    gsmSerial.println("AT+CMGDA=\"DEL ALL\"");
    delay(1000);
}

```

Gambar 4. 13 *Void kirimSMS() Dan hapusSMS()*

Kedua fungsi ini bekerja sama menjaga kelancaran komunikasi antara pengguna dan perangkat, memastikan bahwa setiap perintah dapat dikirim, diterima, dan diproses tanpa hambatan.

4.4 Pengujian Modul Perangkat Keras

Pengujian modul perangkat keras dilakukan sebelum proses integrasi penuh sistem *GPS Tracker*. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap komponen yang digunakan berfungsi sesuai dengan spesifikasinya. Dengan melakukan pengujian secara terpisah, kesalahan atau kegagalan pada modul dapat diidentifikasi dan diperbaiki lebih cepat sebelum sistem dioperasikan secara keseluruhan.

Pengujian dilakukan pada empat modul utama, yaitu *modul GPS Neo-6M*, *modul GSM SIM800L V2*, *modul relay*, dan *modul catu daya*.

4.4.1 Pengujian Modul GPS Neo-6M

1. Tujuan:

Memastikan modul GPS Neo-6M mampu membaca koordinat lokasi dengan akurasi tinggi dan waktu tangkap sinyal (time to first fix) yang wajar.

2. Metode:

- a. Modul *GPS* dihubungkan langsung ke *Arduino Uno* dan diuji menggunakan program sederhana dengan *library TinyGPS++*.
- b. Pengujian dilakukan di area terbuka dan semi-tertutup.
- c. Hasil koordinat ditampilkan melalui *Serial Monitor* dan diverifikasi menggunakan *Google Maps*.

3. Hasil Pengujian

Tabel 4. 2 Pengujian *Modul GPS Neo-6M*

Kondisi Lokasi	Waktu Lock Sinyal (detik)	Akurasi (\pm meter)	Hasil Uji
Area terbuka	10	5–7	Berhasil
Semi-tertutup	20	5–10	Berhasil

4. Kesimpulan

Modul GPS bekerja baik di area terbuka dengan akurasi <10 meter. Di area semi-tertutup, waktu lock sinyal sedikit lebih lama dan akurasi berkurang, namun tetap dapat digunakan.

4.4.2 Pengujian Modul GSM SIM800L V2

1. Tujuan

Memastikan modul *GSM* dapat mengirim dan menerima SMS dengan stabil.

2. Metode

- Modul GSM* dihubungkan ke Arduino Uno dengan catu daya 5V yang stabil.
- Mengirim perintah *AT* langsung melalui *Serial Monitor* untuk menguji koneksi SMS.
- Mengirim dan menerima pesan dari nomor pengguna.

3. Hasil Pengujian

Tabel 4. 3 Pengujian *Modul GSM SIM800L V2*

Fungsi Diuji	Waktu Respon (detik)	Keberhasilan (%)
Mengirim SMS	4–6	100%
Menerima SMS	3–5	100%

4. Kesimpulan

Modul GSM berfungsi dengan baik dan dapat berkomunikasi melalui SMS tanpa kendala, asalkan suplai daya mencukupi dan sinyal operator seluler stabil.

4.4.3 Pengujian Modul Relay

1. Tujuan

Memastikan *relay* dapat memutus dan menyambung arus kelistrikan mesin kendaraan sesuai perintah.

2. Metode

- Relay* dihubungkan ke Arduino dan diuji dengan program sederhana yang mengaktifkan dan menonaktifkan pin kontrol.
- Diuji dalam kondisi terhubung dengan sistem pengapian motor.

3. Hasil Pengujian

Tabel 4. 4 Pengujian *Modul Relay*

Perintah	Kondisi Relay	Respon Mesin	Hasil Uji
HIGH	ON (kontak tertutup)	Mesin hidup	Berhasil
LOW	OFF (kontak terbuka)	Mesin mati	Berhasil

4. Kesimpulan

Relay bekerja sesuai perintah, mampu memutus dan menyambung arus pengapian dengan respons cepat tanpa *delay* berarti.

4.5 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi pada perangkat *GPS Tracker* bekerja sesuai dengan rancangan. Pengujian ini mencakup pengiriman perintah melalui SMS, penerimaan respon dari sistem, serta eksekusi aksi pada perangkat keras.

Metode pengujian dilakukan dengan mengirimkan perintah SMS dari telepon seluler pengguna ke nomor SIM yang dipasang pada *modul GSM SIM800L V2*. Hasil pengujian diamati dari respon SMS yang dikirim balik oleh sistem dan perilaku perangkat keras seperti *relay*.

Tabel berikut menunjukkan hasil pengujian masing-masing perintah:

Tabel 4. 5 Hasil Pengujian Perintah SMS

No	Perintah SMS	Deskripsi Fungsi	Respon Sistem (SMS Balik)	Kondisi Relay / Mesin	Keterangan Hasil Uji
1	CEK	Mengirimkan lokasi GPS kendaraan dan status mesin saat ini	Tautan Google Maps berisi koordinat lokasi + status mesin (HIDUP/MATI)	Tidak berubah (hanya membaca data)	Berhasil, lokasi akurat dalam radius ±5-10 m
2	MATIKAN	Memutus arus kelistrikan pengapian mesin	"MOTOR DIMATIKAN! Tidak dapat dinyalakan."	Relay OFF, Mesin mati	Berhasil, mesin tidak bisa dinyalakan meskipun kunci kontak ON
3	HIDUPKAN	Menyambung kembali arus kelistrikan pengapian mesin	"MOTOR DIHIDUPKAN! Siap digunakan."	Relay ON, Mesin hidup	Berhasil, mesin kembali normal
4	STATUS	Menginformasikan kondisi mesin terkini	"STATUS MOTOR: HIDUP" atau "STATUS MOTOR: MATI"	Tidak berubah (hanya membaca status)	Berhasil, status sesuai kondisi sebenarnya

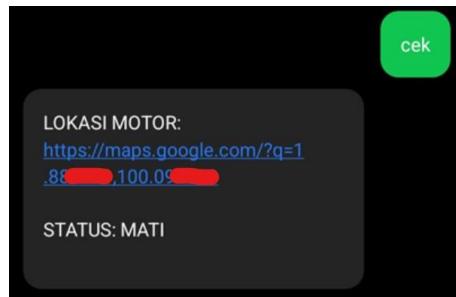
4.5.1 Analisis Hasil Pengujian

1. Perintah CEK berhasil memberikan tautan Google Maps yang dapat langsung dibuka pada browser atau aplikasi Google Maps. Tingkat akurasi lokasi dipengaruhi oleh kondisi sinyal GPS; pada area terbuka akurasi berada di kisaran 5–10 meter, sedangkan di area tertutup atau dekat gedung tinggi akurasi dapat menurun.
2. Perintah MATIKAN berfungsi memutus arus pengapian motor melalui modul relay. Hasil pengujian menunjukkan bahwa setelah perintah ini dijalankan, motor tidak dapat dinyalakan meskipun kunci kontak diputar ke posisi ON.
3. Perintah HIDUPKAN mengembalikan arus pengapian motor. Setelah perintah ini dijalankan, motor dapat dinyalakan kembali seperti semula.

4. Perintah STATUS memberikan informasi yang konsisten dengan kondisi mesin yang sebenarnya. Fitur ini membantu pengguna memastikan status kendaraan tanpa harus mengirimkan perintah MATIKAN atau HIDUPKAN terlebih dahulu.

4.5.2 Dokumentasi Balasan SMS Sistem

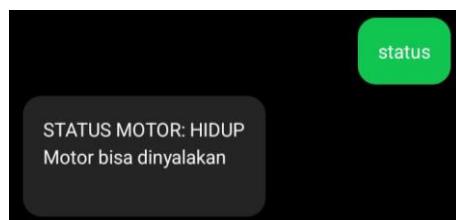
1. Balasan SMS Perintah CEK



Gambar 4. 14 Balasan SMS Perintah CEK

Gambar ini menampilkan balasan SMS dari sistem *GPS Tracker* ketika pengguna mengirimkan perintah CEK. Pada balasan tersebut, sistem memberikan tautan Google Maps yang memuat koordinat terkini lokasi kendaraan, lengkap dengan status mesin (HIDUP atau MATI). Tautan dapat dibuka langsung di aplikasi Google Maps untuk menampilkan posisi kendaraan secara *real-time*.

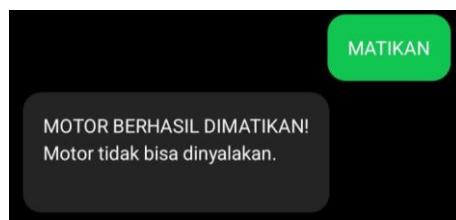
2. Balasan SMS Perintah STATUS



Gambar 4. 15 Balasan SMS Perintah STATUS

Gambar ini memperlihatkan balasan dari sistem saat pengguna mengirimkan perintah STATUS. Dalam balasan tersebut tercantum informasi kondisi mesin kendaraan, apakah dalam keadaan HIDUP atau MATI, beserta keterangan tambahan yang memudahkan pengguna memahami status kendaraan tanpa harus melihat langsung.

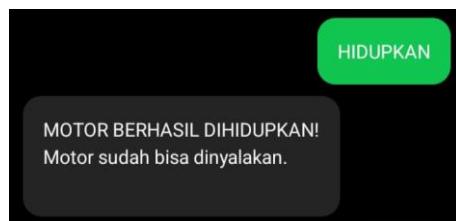
3. Balasan SMS Perintah MATIKAN



Gambar 4. 16 Balasan SMS Perintah MATIKAN

Gambar ini menunjukkan respon sistem ketika menerima perintah MATIKAN dari pengguna. Sistem memproses perintah tersebut, memutus arus kelistrikan pengapian mesin melalui *modul relay*, lalu mengirimkan konfirmasi bahwa “MOTOR DIMATIKAN! Tidak dapat dinyalakan.” Balasan ini menjadi bukti bahwa kendali jarak jauh berjalan sesuai rencana.

4. Balasan SMS Perintah HIDUPKAN



Gambar 4. 17 Balasan SMS Perintah HIDUPKAN

Gambar ini menampilkan balasan dari sistem setelah pengguna mengirim perintah HIDUPKAN. Sistem mengaktifkan kembali *relay* sehingga mesin dapat dinyalakan, dan mengirimkan konfirmasi “MOTOR DIHIDUPKAN! Siap digunakan.”