BAB IV
PERANCANGAN

Bab ini menguraikan semua hal yang berkaitan dengan penelitian mulai dari penyusunan kebutuhan penelitian lalu pembentukan prototipe hingga tahap pengujian prototipe.

4.1. Analisis Terdahulu

Peneliti melakukan analisis terhadap penelitian sebelumnya untuk mengevaluasi sistem yang meliputi area uji, spesifikasi alat, cara kerja, hasil yang dicapai dari penelitian terdahulu. Hasil analisis ini berupa penjelasan sederhana mengenai alat yang digunakan, cara kerja sistem dan area uji dari penelitian sebelumnya, kemudian memberikan penjelasan mengenai penelitian yang dilakukan peneliti untuk memberikan gambaran perbedaan dan kebaharuan penelitian ini dari penelitian terdahulu. Analisis ini dimanfaatkan untuk mengembangkan sistem pada penelitian ini berdasarkan sistem pada penelitian sebelumnya.

Penelitian yang telah dilakukan oleh Adarsh, Arjun C, Koushik S, Midhun Krishnan, Sourav Purushothaman V, Aswathy K Nair, dan Gayathri Narayanan (2020) berjudul "Integrated Real-time Vehicle Speed Control System using RFID and GPS". Area uji penelitian ini adalah sebuah prototipe kendaraan yang dikendalikan secara elektronik yang diuji di area terpilih dan menyertakan simulasi GPS pada perangkat lunak Proteus. Spesifikasi alat utamanya meliputi Mikrokontroler Arduino Uno (ATmega328P), Modul RFID RC522, Motor Driver L298N, dan Modul GPS U-Blox NEO 6M. Cara kerja sistemnya dimulai dengan prioritas pada data geolokasi dari Modul GPS U-Blox NEO 6M yang diterima oleh Arduino Uno. Data lintang dan bujur ini kemudian dibandingkan dengan nilai koordinat yang ada dalam memori Arduino, di mana setiap koordinat tersebut sudah dipetakan ke nilai batas kecepatan tertentu untuk zona spesifik. Jika sinyal GPS tidak tersedia (dalam penelitian ini menggunakan sebuah terowongan sebagai area uji coba), sistem akan beralih menggunakan RFID. Pada kasus ini, Tag RFID pasif

ditempatkan di zona batas kecepatan, dan pembaca RFID RC522 di kendaraan akan menangkap sinyal unik terenkripsi (UID) dari Tag RFID tersebut. UID ini sudah dipetakan ke nilai batas kecepatan spesifik. Untuk keamanan, UID Tag RFID tersebut dienkripsi menggunakan algoritma DES (Data Encryption Standard), dan pada sisi penerima, kode terenkripsi ini akan didekripsi untuk mendapatkan nilai batas kecepatan. Nilai batas kecepatan yang diperoleh baik dari GPS maupun RFID kemudian dikirimkan ke Motor Driver L298N melalui pin PWM (*Pulse Width Modulation*) Arduino. PWM ini berfungsi untuk mengontrol kecepatan motor DC dengan mengatur durasi pulsa digital, yang secara efektif memvariasikan tegangan rata-rata yang disuplai ke motor. Jika kecepatan kendaraan saat ini melebihi batas kecepatan yang ditentukan, sistem akan memutus akselerasi dan menerapkan pengereman secara berkala. Proses ini berulang hingga kecepatan kendaraan sama dengan atau di bawah batas yang diizinkan. Hasil yang dicapai dari penelitian ini menunjukkan bahwa kecepatan prototipe berhasil dikurangi hingga batas yang ditentukan di area uji, dan model gabungan RFID-GPS terbukti meningkatkan efisiensi serta akurasi sistem, dengan implementasi algoritma DES yang berhasil.

Berbeda dengan penelitian terdahulu, penelitian ini yang berjudul "Sistem Pengontrol Kecepatan Otomatis pada Kendaraan Listrik di Zona Selamat Sekolah dengan Fitur Geofence" berfokus pada kendaraan listrik dengan prototipe menggunakan motor DC. Area uji penelitian ini adalah prototipe mobil RC yang digerakkan oleh motor DC, diterapkan secara spesifik di Zona Selamat Sekolah (ZoSS). Spesifikasi alat yang digunakan meliputi Mikrokontroler ESP32S3 WROOM1 N16R8, dua unit Motor Driver L298N Dual H-Bridge, empat unit Motor DC, empat unit HC-020K *Speed Encoder*, modul GPS U-BLOX NEO M8N, dan OLED Display 0,96 Inch 128x64 Pixel, serta Modul MP1584 *Fixed Output* Step Down DC untuk regulasi daya. Cara kerja sistemnya dimulai dengan ESP32S3 yang secara *real-time* membaca geolokasi dari Modul GPS U-Blox NEO M8N. Kecepatan prototipe diukur secara presisi oleh empat unit HC-020K yang terpasang pada setiap motor DC, dan data kecepatan ini diolah serta ditampilkan pada OLED. ESP32S3 WROOM1 N16R8 akan terus membandingkan lokasi prototipe dengan koordinat ZoSS yang telah ditentukan. Jika prototipe terdeteksi memasuki area ZoSS, sistem akan membatasi catu daya yang dialirkan ke motor DC melalui driver

L298N , sehingga kecepatan prototipe secara otomatis berkurang dan stabil di bawah batas maksimal yang diizinkan di ZoSS. Pentingnya, sistem ini juga dirancang untuk mengabaikan upaya penambahan kecepatan melalui mekanisme akselerasi saat prototipe berada di dalam ZoSS, memastikan kepatuhan terhadap batas kecepatan. Ketika data GPS menunjukkan bahwa prototipe telah keluar dari area ZoSS, pembatasan kecepatan akan secara otomatis dilepaskan, dan motor DC dapat beroperasi pada kecepatan normalnya.

4.2. Perancangan Prototipe

Bagian ini akan menjelaskan proses perancangan prototipe sistem pengontrol kecepatan otomatis pada kendaraan listrik. Pembahasan mencakup identifikasi komponen perangkat keras yang digunakan, tata letak pin, prinsip kerja, logika sistem, diagram skema, serta rancangan papan sirkuit cetak. Selain itu, bagian ini juga akan memaparkan desain produk akhir dari prototipe yang dikembangkan.

4.2.1. Analisis Kebutuhan Perangkat Keras

Bagian ini akan membahas komponen-komponen perangkat keras utama yang dipilih dan digunakan dalam pembangunan sistem, termasuk spesifikasi teknis dan peran masing-masing komponen. Berikut penjabaran dari komponen-komponen yang akan digunakan dalam penelitian ini.

Tabel 4.1 Daftar Komponen

Komponen	Spesifikasi	Peran
ESP32S3 WROOM1 N16R8	• Prosesor: Dual-core Tensilica Xtensa LX7 (hingga 240 MHz). • Memori: 512 KB SRAM, 16 MB Flash, 8 MB PSRAM. • Konektivitas: Wi-Fi (802.11 b/g/n), Bluetooth LE 5.0.	Berfungsi sebagai unit pemrosesan sentral. Bertanggung jawab untuk memproses data dari modul GPS dan sensor kecepatan, mengimplementasikan logika *geofence* dan mode operasi, mengontrol *Motor Driver* melalui sinyal PWM,

Komponen	Spesifikasi	Peran
	• Pin GPIO: Lebih dari 40 pin GPIO multifungsi, mendukung ADC, PWM, I2C, SPI, UART, dan *interrupt*. • Pin Khusus: Pin internal untuk RGB LED (GPIO 48).	serta mengelola tampilan pada layar OLED dan indikator LED RGB.
Dua unit Motor Driver L298N Dual H-Bridge	• Tegangan Operasi Motor: 5V hingga 35V. • Arus Saluran: Hingga 2A masing-masing saluran (total 4A maksimal). • Saluran Kontrol: Dua saluran independen per modul. • Input Logika: 5V (IN1, IN2, IN3, IN4), Input Enable (ENA, ENB) untuk PWM.	Berfungsi sebagai antarmuka daya antara mikrokontroler dan motor DC. Modul ini menerima sinyal logika rendah dari ESP32S3 dan mengubahnya menjadi arus dan tegangan yang lebih tinggi yang diperlukan untuk menggerakkan Motor DC.
Empat unit Motor DC	Tegangan Operasi: 3V hingga 6V	Komponen aktuator yang mengubah energi listrik menjadi gerakan mekanis untuk menggerakkan kendaraan. Masing-masing dua motor dihubungkan secara paralel ke satu

Komponen	Spesifikasi	Peran
		saluran output Driver L298N.
Empat unit HC-020K Speed Encoder	• Sensor Slot Optik (*photocoupler)* • Tegangan Operasi: 4,5V hingga 5,5V.	Dipasang pada poros atau roda masing-masing dari empat motor DC untuk mengukur kecepatan putaran secara individual. Pulsa digital (HIGH/LOW) yang dihasilkan saat *disk encoder* melewati celah sensor dihitung oleh ESP32S3. Data ini esensial untuk menentukan kecepatan aktual kendaraan dan fungsi kontrol kecepatan.
Satu unit MP1584 Fixed Output Step Down DC	• Tegangan Input: 7V hingga 28V. • Tegangan Output: Tetap 5V. • Arus Output: Hingga 3A.	Berfungsi sebagai regulator daya utama untuk komponen sensitif. Menurunkan tegangan tinggi dari baterai 3S (Maksimal 12.6V) menjadi 5V yang stabil untuk memberikan daya pada ESP32S3, Modul GPS, OLED, dan *speed encoder* memastikan operasi yang aman dan stabil.
Satu unit Modul GPS U-Blox NEO M8N	• Chipset: U-Blox NEO-M8N.	Menyediakan data lokasi (lintang, bujur, ketinggian) dengan menerima sinyal

Komponen	Spesifikasi	Peran
	• Komunikasi: Antarmuka serial UART (RXD, TXD). • Fitur: Mendukung berbagai sistem satelit (GPS, GLONASS, Galileo, BeiDou)	dari satelit. Data ini digunakan untuk implementasi fitur *geofence*, yang secara otomatis membatasi kecepatan saat kendaraan memasuki Zona Selamat Sekolah (ZoSS).
Satu unit OLED Display 0,96 Inch 128×64 Pixel	• Resolusi: 128×64 piksel. • Antarmuka: I2C	Menyediakan antarmuka visual bagi pengguna untuk menampilkan informasi penting secara real-time, seperti lokasi GPS, kecepatan saat ini, dan mode operasi sistem.
Satu unit Tempat Baterai 3S 18650 dengan Tiga buah Baterai 18650	• Tegangan Nominal: 3,7V×3 (11,1V). • Tegangan Penuh: 4,6V×3 (12,6V). • Arus; 40A	Menyediakan daya listrik utama untuk seluruh sistem.
Satu unit Saklar Daya KCD11 10×15MM	• SPST (*Single Pole Single Throw*) ON/OFF. • Peringkat Arus/Tegangan: 3A pada 250VAC, 6A pada 125VAC.	Berfungsi sebagai saklar fisik utama untuk menghidupkan dan mematikan seluruh sistem.
Empat buah Kapasitor Elektrolit Elco 1000uF 16V	• Kapasitansi: 1000uF • Nilai Tegangan Kerja: 16V	Berfungsi sebagai kapasitor *bulk decoupling*. Menyimpan cadangan

Komponen	Spesifikasi	Peran
		energi untuk mengatasi lonjakan arus yang mendadak dan menstabilkan tegangan suplai daya pada jalur utama 5V dan 12V.
Tiga Belas buah Kapasitor Keramik 1uF 50V	• Kapasitansi: 1uF • Nilai Tegangan Kerja: 50V	Berfungsi sebagai kapasitor decoupling dan penyaring noise frekuensi tinggi. Dipasang di dekat pin daya setiap IC atau modul untuk menyerap noise lokal dan menyediakan arus instan yang dibutuhkan, memastikan operasi yang stabil.

4.2.2. Rancangan Pin Prototype

Bagian ini akan menguraikan konfigurasi dan penugasan pin untuk setiap komponen perangkat keras yang terhubung pada mikrokontroler. Berikut merupakan tabel yang menjabarkan penugasan pin untuk setiap komponen yang terhubung ke mikrokontroler.

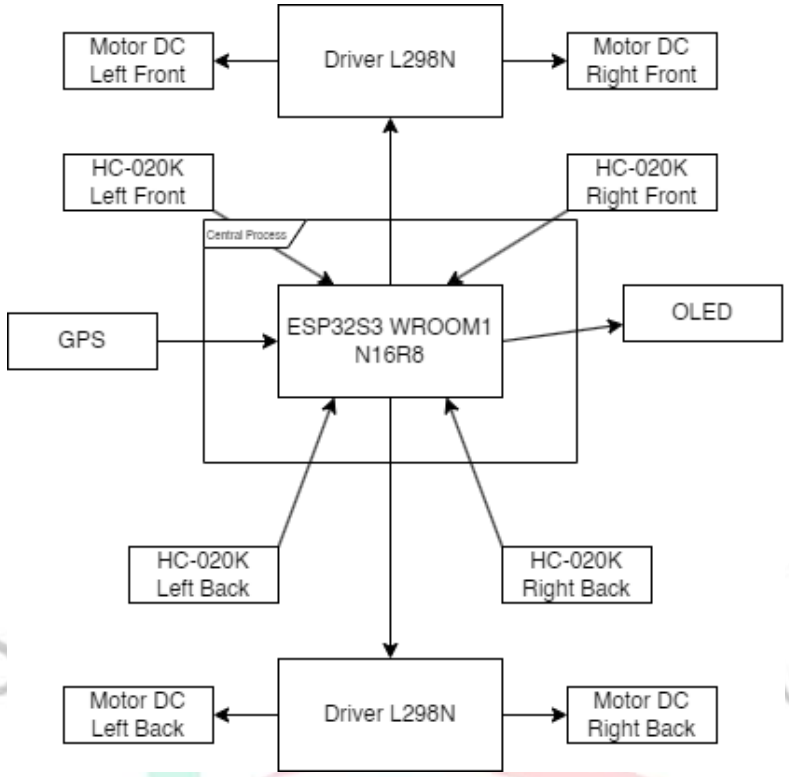
Tabel 4.2 Tabel Rancangan Pin Komponen

Komponen	Pin ESP32S3 → Komponen	Fungsi
Motor Driver L298N #1	GPIO 1 → IN1	Kontrol Motor 1 (Kiri depan)
	GPIO 2 → IN2	
	GPIO 3 → ENA	
	GPIO 4 → IN3	Kontrol Motor 2 (Kanan depan)
	GPIO 5 → IN4	
	GPIO 7 → ENB	

Komponen	Pin ESP32S3 → Komponen	Fungsi
Motor Driver L298N #2	GPIO 10 → IN1	Kontrol Motor 3 (Kanan belakang)
	GPIO 11 → IN2	
	GPIO 12 → ENA	
	GPIO 13 → IN3	Kontrol Motor 4 (Kiri belakang)
	GPIO 14 → IN4	
	GPIO 15 → ENB	
Modul GPS U-Blox NEO M8N	GPIO 18 → RX	Mengirim data untuk mengkonfigurasi pengaturan modul.
	GPIO 17 → TX	Menerima data.
OLED Display 0.96 inci 128x64 Piksel I2C	GPIO 41 → SDA	Mengirim dan menerima data.
	GPIO 42 → SCL	Sinkronisasi transfer data.
Empat unit HC-020K Speed Encoder	GPIO 19 → OUT	Menerima data dari *speed encoder* untuk Motor DC kanan depan.
	GPIO 20 → OUT	Menerima data untuk Motor DC kiri depan.
	GPIO 21 → OUT	Menerima data untuk Motor DC kanan belakang.
	GPIO 45 → OUT	Menerima data untuk Motor DC kiri belakang.

4.2.3. Rancangan Prinsip Kerja

Bagian ini akan menjelaskan secara konseptual bagaimana sistem secara keseluruhan beroperasi, mulai dari input sensor hingga output kendali motor. Berikut merupakan blok diagram dari sistem yang akan dibuat.



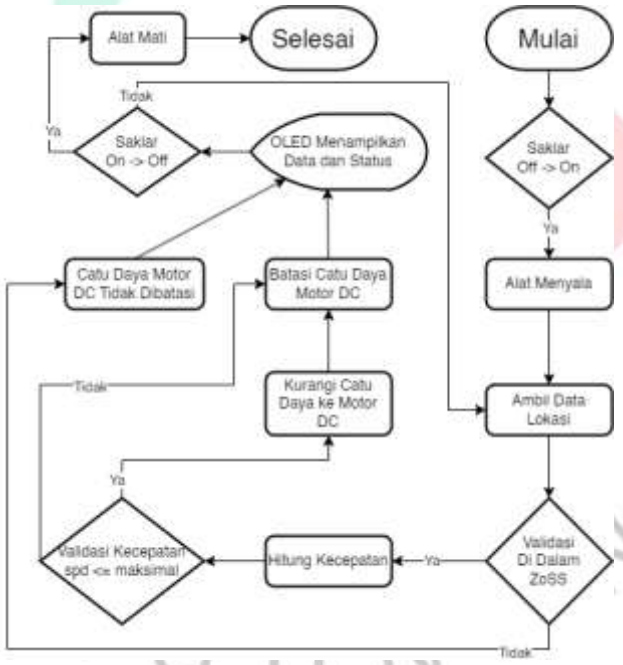
Gambar 4.1 Blok Diagram

Berdasarkan blok diagram yang ditunjukkan pada gambar 4.1 sistem mulai bekerja setelah mendapatkan daya. Pertama kali sistem akan mengaktifkan GPS dengan mengirimkan data koordinat ini ke ESP32 untuk dapat menentukan geolokasi secara *real-time* lokasi keberadaan prototipe. Informasi hasil pembacaan geolokasi yang telah dikirimkan ke ESP32 kemudian dikirimkan ke OLED untuk ditampilkan. Proses pengolahan selanjutnya ESP32 akan menentukan apakah lokasi keberadaan prototipe berada di dalam atau di luar *geofence* yang telah ditentukan, kemudian akan mengirimkan instruksi ke Driver L298N agar dapat mengendalikan arah dan kecepatan dari Motor DC untuk menjalankan prototipe sesuai instruksi yang telah ditentukan. Setelah prototipe bergerak, HC-020K akan membaca putaran dari Motor DC yang kemudian hasil pembacaan tersebut dikirimkan ke ESP32. Data tersebut diubah menjadi satuan kecepatan yang lebih mudah dipahami dan dikirimkan ke OLED untuk ditampilkan.

4.2.4. Rancangan Logika

Bagian ini akan menjabarkan logika dari sistem pada prototipe dan divisualisasikan pada gambar 4.2 menggunakan Flowchart. Proses diawali dengan

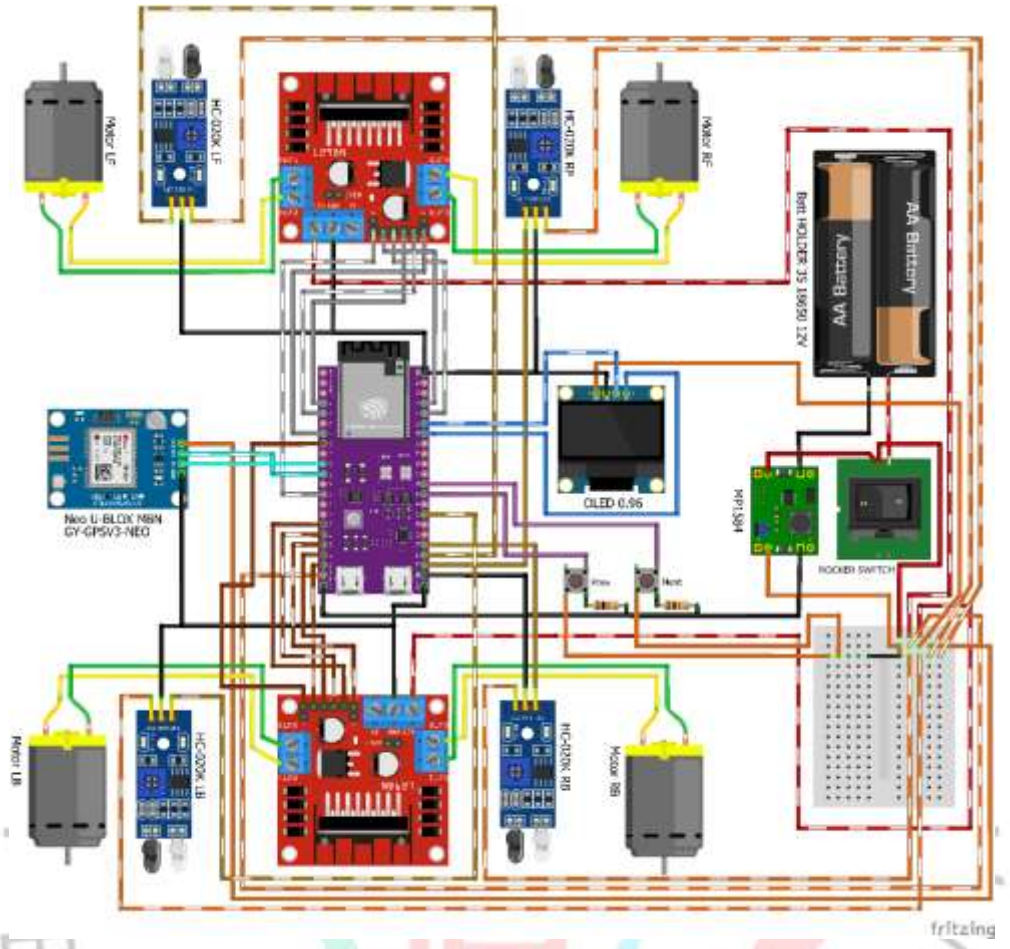
kondisi di mana sistem memverifikasi status saklar daya. Apabila "Tombol OFF menjadi ON", maka "Alat Menyala" dan inisiasi sistem berhasil. Sebaliknya, jika saklar tidak berada dalam posisi "ON", sistem akan tetap dalam keadaan tidak aktif atau kembali ke kondisi awal hingga saklar diaktifkan. Setelah alat menyala, sistem akan membaca koordinat posisi *real-time* dari prototipe secara. Prototipe kemudian akan secara otomatis mengaktifkan pembatasan kecepatan saat keberadaan prototipe terdeteksi memasuki area ZoSS, kemudian pembatasan kecepatan akan dinonaktifkan ketika sistem mendeteksi keberadaan prototipe keluar area ZoSS. Perubahan koordinat, kecepatan, status"OLED Menampilkan Data dan Status" terkini. Proses ini terus berulang hingga "Tombol ON menjadi OFF". Ketika kondisi ini terpenuhi, "Alat Mati" dan *flowchart* berakhir pada kondisi "Selesai".



Gambar 4.2 Diagram Flowchart

4.2.5. Rancangan Diagram Skema Perangkat Keras

Tahapan ini mencakup diagram skematik yang menunjukkan interkoneksi antar komponen perangkat keras. Berikut merupakan Diagram Skema Perangkat Keras dari prototipe yang dikembangkan pada penelitian ini dengan menggunakan alat bantu dari Software Fritzing.



Gambar 4.3 Diagram Skema Perangkat Keras

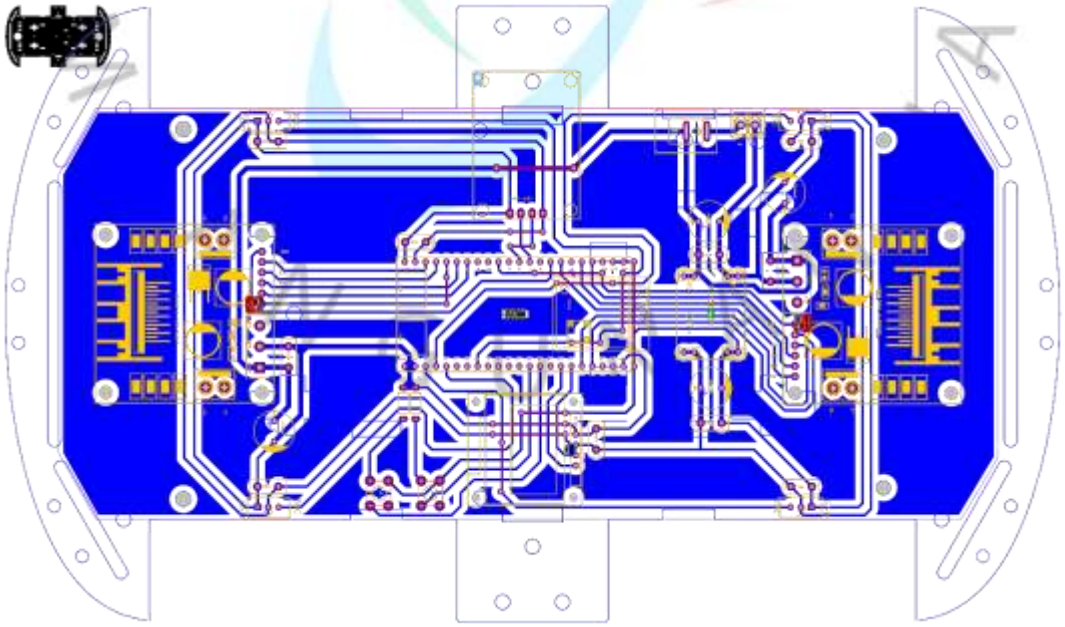
Berikut merupakan penjabaran secara sederhana mengenai Gambar 4.3 yang menjelaskan tentang Diagram Skema Perangkat Keras.

1. Mikrokontroler ESP32S3 WROOM1 N16R8 (Bagian Tengah, Ungu): Otak sistem, memproses data dan mengirim perintah.
2. Modul GPS U-Blox NEO M8N (Kiri Bagian Tengah, Biru): Menyediakan data lokasi ke mikrokontroler untuk fitur *geofence*.
3. Driver L298N (Atas dan Bawah Bagian Tengah, Merah): Mengontrol arah dan kecepatan empat Motor DC berdasarkan sinyal dari mikrokontroler dengan mengatur daya.
4. Motor DC (Empat Sudut, Abu-abu): Mengubah energi listrik menjadi gerakan putar, berfungsi sebagai penggerak kendaraan.

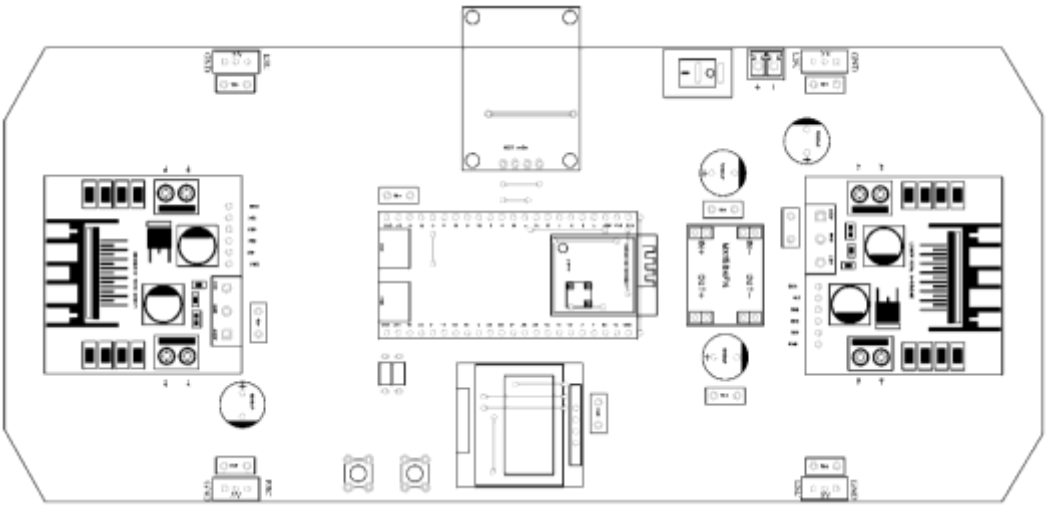
5. Sensor Kecepatan HC-020K (Dekat Setiap Motor, Biru): Memberikan umpan balik kecepatan putaran motor ke mikrokontroler untuk perhitungan kecepatan kendaraan.
6. Layar OLED (Kanan Tengah, Hitam): Menampilkan informasi seperti kecepatan dan status sistem kepada pengguna.
7. Modul MP1584 (Kanan Bagian Bawah, Hijau): Penurun tegangan untuk menyediakan daya yang sesuai bagi komponen.
8. Dudukan Baterai (Kanan Bagian Atas, Coklat): Sumber daya listrik utama untuk seluruh sistem.
9. Rocker Switch (Kanan Bagian Tengah, Hitam): Untuk menghidupkan dan mematikan sistem.
10. Breadboard (Kanan Bagian Bawah, Merah): Papan untuk menghubungkan jalur catu daya komponen sementara.

4.2.6. Rancangan Papan Sirkuit Cetak

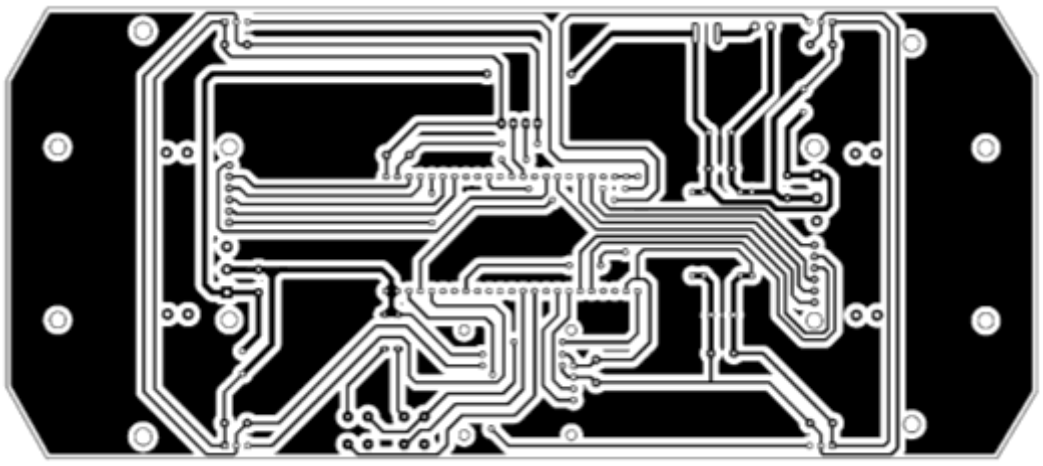
Bagian ini akan menjelaskan desain tata letak papan sirkuit cetak (PCB) untuk integrasi komponen secara fisik. Berikut merupakan desain tata letak papan sirkuit cetak.



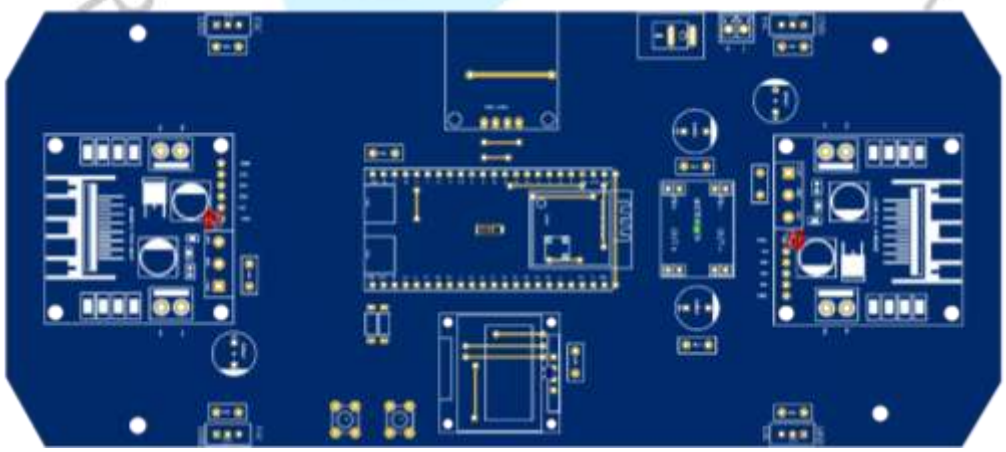
Gambar 4.4 Desain Tata Letak PCB pada Prototipe



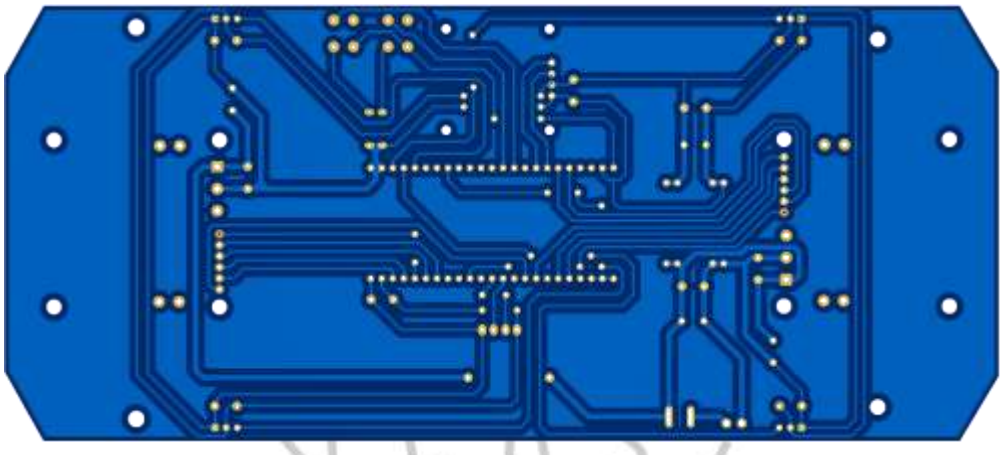
Gambar 4.5 Desain *Top Layer* PCB



Gambar 4.6 Desain *Bottom Layer* PCB



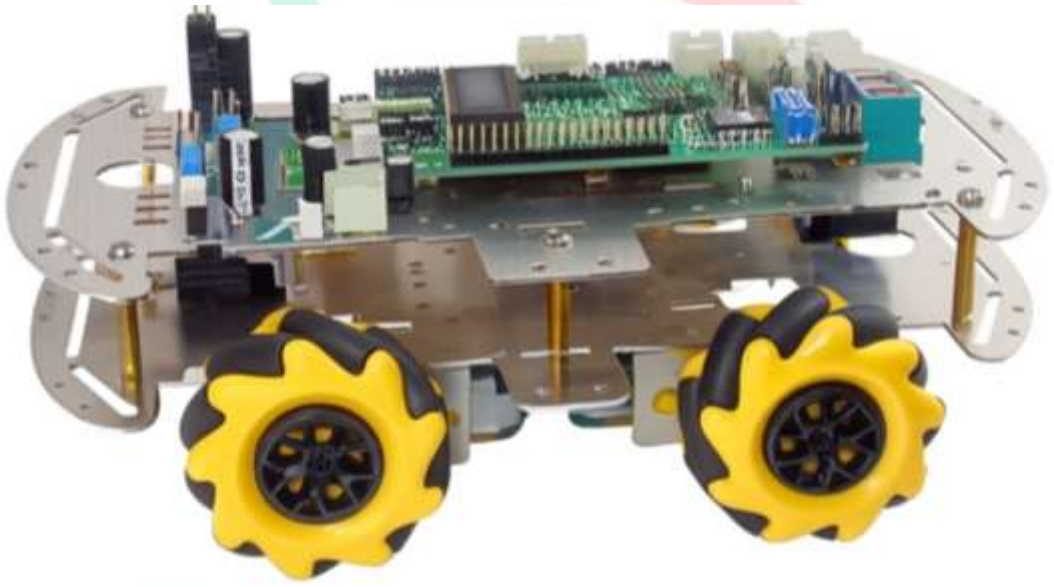
Gambar 4.7 Desain 3D Top Layer PCB



Gambar 4.8 Desain 3D *Bottom Layer* PCB

4.2.7. Rancangan Produk Akhir

Tahap ini akan memberikan gambaran tampilan akhir dari prototipe berdasarkan bentuk prototipe dalam masa perancangan. Berikut merupakan gambaran awal dari bentuk fisik prototipe.



Gambar 4.9 Rancangan Desain Bentuk Fisik Prototipe

Gambar 4.9 menampilkan rancangan awal prototipe yang akan dijadikan gambaran dalam produk akhir prototipe yang digunakan dalam penelitian ini. Prototipe akan menggunakan *base plate* berbahan aluminium alloy sebagai fondasi utama kendaraan. Di atas *base plate* ini, terpasang sebuah PCB yang dirancang khusus untuk sistem pengontrol kecepatan. PCB ini menampung berbagai

komponen elektronik, termasuk mikrokontroler, Driver L298N, OLED dan Modul GPS.

4.3.Perancangan Pengujian

Pada tahapan ini akan menjabarkan mengenai ini skenario pengujian menggunakan Metode Pengujian *BlackBox* yang akan digunakan untuk memverifikasi fungsionalitas sistem yang telah dirancang. Berikut tabel pengujian dengan menggunakan Metode Pengujian *BlackBox*.

Tabel 4.3 Tabel Rancangan Pengujian Prototipe

Pengujian	**Ke-1**
Skenario Pengujian	Pengujian Fungsionalitas Saklar ON/OFF Prototipe.
Deskripsi	Mengaktifkan saklar utama pada prototipe dari posisi "OFF" ke "ON".
Hasil yang Diharapkan	Sistem menyala, ditandai dengan layar OLED menampilkan status awal, serta LED pada setiap komponen dan WS2812B pada mikrokontroler menyala.
Pengujian	**Ke-2**
Skenario Pengujian	Pengujian Kontrol Gerak "UP" via Aplikasi Dabble.
Deskripsi	Menahan (menekan dan menahan) Button UP pada antarmuka aplikasi Dabble.
Hasil yang Diharapkan	Prototipe merespon dengan bergerak maju, dan motor DC berputar sesuai instruksi. Layar OLED memperbarui informasi kecepatan.
Pengujian	**Ke-3**
Skenario Pengujian	Pengujian Kontrol Gerak "DOWN" via Aplikasi Dabble.
Deskripsi	Menahan (menekan dan menahan) Button Down pada antarmuka aplikasi Dabble.

Hasil yang Diharapkan	Prototipe merespon dengan bergerak mundur, dan motor DC berputar sesuai instruksi. Layar OLED memperbarui informasi kecepatan.
Pengujian	**Ke-4**
Skenario Pengujian	Pengujian Kontrol Gerak "Lingkaran" via Aplikasi Dabble
Deskripsi	Menahan (menekan dan menahan) Button Right pada antarmuka aplikasi Dabble.
Hasil yang Diharapkan	Prototipe merespon dengan bergerak searah jarum jam, dan motor DC berputar sesuai instruksi. Layar OLED memperbarui informasi kecepatan.
Pengujian	**Ke-5**
Skenario Pengujian	Pengujian Kontrol Gerak "Kotak" via Aplikasi Dabble
Deskripsi	Menahan (menekan dan menahan) Button Left pada antarmuka aplikasi Dabble.
Hasil yang Diharapkan	Prototipe merespon dengan bergerak berlawanan arah dengan jarum jam, dan motor DC berputar sesuai instruksi. Layar OLED memperbarui informasi kecepatan.
Pengujian	**Ke-6**
Skenario Pengujian	Pengujian Deteksi dan Pembatasan Kecepatan di ZoSS
Deskripsi	Mengoperasikan prototipe dengan menggerakkannya hingga memasuki area Zona Selamat Sekolah (ZoSS) yang telah ditentukan.
Hasil yang Diharapkan	Sistem berhasil mendeteksi prototipe masuk ke ZoSS, kecepatan motor DC secara otomatis berkurang dan stabil di bawah batas tertentu. Layar OLED menampilkan indikasi masuk ZoSS dan kecepatan prototipe.
Pengujian	**Ke-7**
Skenario Pengujian	Pengujian Pelepasan Pembatasan Kecepatan di Luar ZoSS

Deskripsi	Mengoperasikan prototipe dengan menggerakkannya keluar dari area Zona Selamat Sekolah (ZoSS).
Hasil yang Diharapkan	Sistem berhasil mendeteksi keluar dari ZoSS, pembatasan kecepatan dihilangkan secara otomatis, dan motor DC dapat beroperasi pada kecepatan normal. Layar OLED memperbarui
Pengujian	**Ke-8**
Skenario Pengujian	Pengujian Perilaku di Luar ZoSS
Deskripsi	Mengoperasikan prototipe tetap berada di luar area ZoSS.
Hasil yang Diharapkan	Sistem tidak melakukan pembatasan kecepatan pada motor DC, memungkinkan prototipe untuk berakselerasi penuh.
Pengujian	**Ke-9**
Skenario Pengujian	Pengujian Fungsionalitas Saklar OFF Prototipe
Deskripsi	Mengubah saklar utama pada prototipe dari posisi "ON" ke "OFF".
Hasil yang Diharapkan	Sistem mati sepenuhnya, ditandai dengan layar OLED, LED komponen lain dan WS2812B pada mikrokontroler tidak menyala lagi.

4.4 Analisis Risiko Sistem

Dalam perancangan sistem pengontrol kecepatan otomatis ini, identifikasi potensi risiko dan dampaknya menjadi krusial untuk memastikan keandalan dan keamanan sistem, meskipun penelitian ini berfokus pada tahap prototipe. Analisis risiko ini bertujuan untuk mengantisipasi kemungkinan kegagalan pada komponen perangkat keras utama dan kesalahan pada sisi perangkat lunak, serta mengidentifikasi langkah mitigasi yang dapat diterapkan. Berikut merupakan penjabaran dari potensi kegagalan dan mitigasi pada penelitian ini:

1. Kegagalan Modul GPS U-Blox NEO M8N:
    - Mode Kegagalan: Modul GPS tidak mendapatkan data koordinat atau memberikan data koordinat yang tidak akurat.

- Dampak: Sistem tidak dapat menentukan lokasi keberadaan prototipe secara tepat, sehingga fitur *geofence* tidak berfungsi. Ini dapat mengakibatkan prototipe tidak membatasi kecepatan ketika memasuki area *geofence* (ZoSS).
- Mitigasi:
    - Pada Tingkat Perancangan Prototipe: Sistem dirancang untuk terus mencoba mendapatkan sinyal dan menampilkan status "tidak ada sinyal" atau "menunggu fix" pada layar OLED. Dalam kasus sinyal hilang sementara, sistem prototipe kembali ke skenario ketika prototipe berada di luar ZoSS.
    - Saran untuk Pengembangan Lanjut: Penggunaan antena GPS eksternal yang lebih sensitif atau modul GPS dengan kemampuan *dead reckoning* dapat dipertimbangkan untuk menjaga akurasi lokasi di area *blank spot* sinyal. Implementasi *logic fallback* yang lebih kompleks, seperti batas kecepatan *default* terendah, juga dapat ditambahkan jika sinyal hilang dalam periode waktu yang lama.

2. Kegagalan HC-020K Speed Encoder:
    - Mode Kegagalan: Sensor tidak membaca putaran motor dengan benar atau memberikan data kecepatan yang tidak stabil atau salah.
    - Dampak: Sistem tidak mendapatkan informasi kecepatan aktual prototipe, sehingga pengendalian kecepatan menjadi tidak akurat atau tidak berfungsi sama sekali. Hal ini dapat menyebabkan kecepatan tidak sesuai dengan batas yang ditentukan.
    - Mitigasi:
        - Pada Tingkat Perancangan Prototipe: Perancangan sistem melibatkan empat unit sensor HC-020K pada setiap motor, yang secara inheren menyediakan sedikit redundansi untuk pembacaan kecepatan rata-rata. Data kecepatan juga ditampilkan pada OLED yang memungkinkan pemantauan visual.
        - Saran untuk Pengembangan Lanjut: Implementasi algoritma filter data untuk mereduksi *noise* atau *outlier* dari pembacaan sensor.

Pertimbangan penggunaan sensor dengan resolusi lebih tinggi atau penambahan sistem kalibrasi sensor otomatis.

3. Kegagalan Motor Driver L298N:
   - Mode Kegagalan: Driver mengalami *overload*, *overheat*, atau kerusakan yang menyebabkan gagal mengontrol daya ke motor DC.
   - Dampak: Kontrol kecepatan tidak dapat diimplementasikan, sehingga tujuan utama sistem untuk membatasi kecepatan di ZoSS tidak tercapai.
   - Mitigasi:
     - Pada Tingkat Perancangan Prototipe: Pemilihan Driver L298N dilakukan berdasarkan spesifikasi motor DC dan kebutuhan daya. Desain PCB juga mempertimbangkan tata letak komponen untuk aliran daya yang optimal.
     - Saran untuk Pengembangan Lanjut: Penggunaan driver motor dengan kapasitas arus yang lebih tinggi atau fitur proteksi internal yang lebih canggih untuk meningkatkan ketahanan. Pertimbangan penambahan *heatsink* eksternal jika operasi jangka panjang menghasilkan panas berlebih.
4. Kesalahan Pemrograman:
   - Mode Kegagalan: *Bug* pada kode program dapat menyebabkan sistem *hang*, melakukan perhitungan yang salah atau *crash* secara acak.
   - Dampak: Perilaku sistem menjadi tidak dapat diprediksi, dapat membahayakan atau gagal berfungsi sama sekali dalam mengontrol kecepatan.
   - Mitigasi:
     - Pada Tingkat Perancangan Prototipe: Pengembangan perangkat lunak dilakukan melalui Metode *Prototyping* yang melibatkan siklus berulang analisis kebutuhan, desain, pengujian, dan revisi. Proses Pengujian *BlackBox* dilakukan untuk memverifikasi fungsionalitas sistem. Penggunaan *watchdog timer* pada ESP32 juga dapat diimplementasikan untuk melakukan *reset* otomatis jika sistem mengalami *hang* atau *stuck*.

- Saran untuk Pengembangan Lanjut: Menerapkan praktik pengembangan perangkat lunak yang lebih formal, seperti *code review* oleh pihak ketiga, pengujian unit otomatis yang lebih mendalam, atau simulasi perangkat lunak untuk mengidentifikasi *bug* sebelum implementasi pada perangkat keras.

Analisis risiko ini menjadi landasan untuk perbaikan dan pengembangan sistem di masa mendatang, memastikan bahwa sistem tidak hanya berfungsi sesuai yang diharapkan tetapi juga *resilient* terhadap potensi kegagalan yang mungkin terjadi di lingkungan nyata.