

LAPORAN TUGAS AKHIR 2

Connected Vehicle: Speedometer dan Odometer Berbasis Bluetooth



Penyusun:

Daffis Hesa Pratama (20524015)

Ichiro Rizky Abdul Aziz (20524017)

**Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia Yogyakarta**

2024

HALAMAN PENGESAHAN

***Connected Vehicle: Speedometer dan Odometer Berbasis
Bluetooth***

Penyusun:

Daffis Hesa Pratama (20524015)

Ichiro Rizky Abdul Aziz (20524017)

Yogyakarta, 09 Juli 2024

Dosen Pembimbing 1

Dosen Pembimbing 2



Dr. Hasbi Nur Prasetyo Wisudawan, S.T., M.T.

Dr. Wahyudi Budi Pramono, S.T., M.Eng.

195240101

985240104

**Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia Yogyakarta**

2024

LEMBAR PENGESAHAN TUGAS AKHIR

Connected Vehicle: Speedometer dan Odometer Berbasis



Ketua Penguji

: Dr. Hasbi Nur Prasetyo Wisudawan, S.T., M.T.

Anggota Penguji 1

: Husein Mubarak, S.T., M.Eng.

Anggota Penguji 2

: I Gede Sudiasa, S.T

**Tugas akhir ini telah disahkan sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Teknik**

Tanggal: 5 Agustus 2024

Ketua Program Studi Teknik Elektro



Dwi Ana Ratna Wati, S.T., M.Eng.

035240102

PERNYATAAN

Dengan ini kami menyatakan bahwa:

1. Tugas Akhir ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi lainnya, dan sepanjang pengetahuan kami juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Tugas Akhir yang terkait hak milik, hak intelektual, dan potensi merupakan milik bersama antara tiga pihak, yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal ini, penggunaan informasi dan materi Tugas Akhir terkait paten maka akan didiskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 3 Agustus 2024




Daffis Hesa Pratama (20524015)




Ichiro Rizky Abdul Aziz (20524017)

DAFTAR ISI

HALAMAN PENGESAHAN	1
LEMBAR PENGESAHAN TUGAS AKHIR.....	2
PERNYATAAN	3
DAFTAR ISI.....	4
DAFTAR GAMBAR	7
DAFTAR TABEL.....	9
RINGKASAN.....	10
BAB 1 PENDAHULUAN	11
1.1 Latar belakang dan Identifikasi Masalah	11
1.2 Rumusan Masalah.....	14
1.3 Tujuan	15
1.4 Batasan Masalah	15
1.5 Batasan Realistis Aspek Keteknikan	15
BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM	16
2.1 Studi Literatur dan Observasi	16
2.2 Dasar Teori.....	18
2.2.1 Speedometer.....	18
2.2.2 Odometer.....	19
2.2.3 Controller Votol.....	20
2.2.4 CAN BUS Shield (MCP2515).....	21
2.2.5 Arduino Uno	22
2.2.6 Modul Bluetooth	23
2.2.7 Liquid Crystal Display (LCD)	24
2.3 Analisis Stakeholder	25
2.3.1 PT Fusi Global Teknologi.....	25
2.3.2 Perusahaan Motor Listrik.....	25
2.3.3 Pengguna Motor Listrik	25
2.4 Analisis Aspek yang Mempengaruhi Sistem	25
2.4.1 Aspek Ekonomi.....	25
2.4.2 Aspek Keamanan	25
2.4.3 Aspek Teknologi.....	26
2.4.4 Aspek Regulasi	26

2.5 Spesifikasi Sistem	26
BAB 3. USULAN SOLUSI	27
3.1 Usulan Solusi 1	28
3.1.1 Desain Sistem 1.....	28
3.1.2 Rencana Anggaran Desain 1	34
3.1.3 Analisis Risiko Desain.....	35
3.1.4 Pengukuran Performa.....	35
3.1.5 Aspek Keteknikan dan K3	35
3.2 Usulan Solusi 2	36
3.2.1 Desain Sistem 2.....	36
3.2.2 Rencana Anggaran Desain Sistem 2	42
3.2.3 Analisis Risiko Desain 2.....	43
3.2.4 Pengukuran Performa.....	43
3.2.5 Aspek Ketenikan dan K3	43
3.3 Analisis dan Penentuan Usulan Solusi/Desain Terbaik	44
3.4 Skenario Pengujian	45
3.5 Gantt Chart	46
3.6 Realisasi Pelaksanaan Tugas Akhir 1	47
BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN	52
4.1 Hasil Rancangan Sistem	52
4.1.1 Rangkaian Elektronik.....	52
4.1.2 Design 3D	58
4.1.3 Antarmuka Pengguna.....	59
4.1.4 Pembuatan Kode Program	62
4.1.5 Foto hasil akhir perancangan	62
4.2 Metode Pengukuran Kinerja Hasil Perancangan	63
4.2.1 Standar Keteknikan.....	64
4.2.2 Parameter yang diukur	65
BAB 5. HASIL PENGUKURAN DAN ANALISIS	69
5.1. Analisis Hasil.....	69
5.1.1 Hasil dan Analisis Pengujian Indikator.....	69
5.1.2 Perbandingan Performa dengan sistem Lain.....	82
5.1.3 Pemenuhan Spesifikasi Sistem	84
5.1.4 Pengalaman Pengguna	85
5.1.5 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya.....	87
5.2 Dampak Implementasi Sistem	92
5.3 Kelebihan dan Kekurangan Sistem.....	93
BAB 6. KESIMPULAN DAN SARAN	96

6.1 Kesimpulan	96
6.2 Saran	97
DAFTAR PUSTAKA	98
LAMPIRAN – LAMPIRAN.....	100

DAFTAR GAMBAR

Gambar 2. 1 Speedometer Analog	17
Gambar 2. 2 Speedometer Digital	17
Gambar 2. 3 Odometer Analog	18
Gambar 2. 4 Odometer Digital	18
Gambar 2. 5 CAN Bus Shield MCP2515	19
Gambar 2. 6 Arduino Uno	20
Gambar 2. 7 Modul <i>Bluetooth</i>	21
Gambar 2. 8 Liquid Crystal Display TFT	22
Gambar 3. 1 Diagram Blok Sistem	26
Gambar 3. 2 Proses cara kerja sistem	27
Gambar 3. 3 Diagram Alir Sistem	28
Gambar 3. 4 Desain Sistem Tampak Samping	30
Gambar 3. 5 Desain Sistem 1 Tampak Depan	30
Gambar 3. 6 Usulan rancangan aplikasi untuk pengguna	32
Gambar 3. 7 Diagram Blok Sistem	34
Gambar 3. 8 Proses Cara Kerja Sistem	35
Gambar 3. 9 Diagram Alir Sistem	36
Gambar 3. 10 Desain Sistem 2 Tampak Samping	38
Gambar 3. 11 Desain Sistem 2 Tampak Samping	38
Gambar 3. 12 Usulan rancangan aplikasi untuk pengguna	40
Gambar 4. 1 Rangkaian Elektronik SPEETER	52
Gambar 4. 2 Rangkaian Elektronik Transmitter	54
Gambar 4. 3 Rangkaian Elektronik Receiver	55
Gambar 4. 4 Rangkaian Elektronis Transmitter dan Receiver	56
Gambar 4. 5 Diagram Alir	57
Gambar 4. 6 Cover LCD SPEETER	58
Gambar 4. 7 Rangkaian Car Kit	59
Gambar 4. 8 Tampilan Aplikasi SPEETER	59
Gambar 4. 9 Tampilan LCD Pertama	60
Gambar 4. 10 Tampilan LCD Utama	61
Gambar 4. 11 Hasil Akhir perancangan SPEETER	53
Gambar 4. 12 Rangkaian Akhir Car Kit	54
Gambar 4. 13 Pengujian Tachometer	58
Gambar 5. 1 Pembacaan Tachometer (RPM)	60
Gambar 5. 2 Pembacaan Serial Monitor	61
Gambar 5. 3 Pembacaan LCD	62
Gambar 5. 4 Pembacaan LCD	64
Gambar 5. 5 Hasil Pembacaan Aplikasi Android	65

Gambar 5. 6 Perhitungan Teoritis Tachometer	67
Gambar 5. 7 Pembacaan Aplikasi	68
Gambar 5. 8 Rata-rata <i>Error</i> Kecepatan dan Jarak Tempuh	70
Gambar 5. 9 Rata-rata Akurasi Kecepatan dan Jarak Tempuh	71

DAFTAR TABEL

Tabel 1. 1 Hasil survei antara pengembang dan pengguna	11
Tabel 2. 1 Hasil Studi Literatur Solusi Sejenis.....	15
Tabel 3. 1 Inventarisasi kebutuhan usulan sistem perangkat keras.....	31
Tabel 3. 2 Rencana anggaran pengembangan sistem	32
Tabel 3. 3 Inventarisasi kebutuhan usulan sistem perangkat keras SPEETER	39
Tabel 3. 4 Rencana anggaran pengembangan sistem SPEETER	40
Tabel 3. 5 Perbandingan 2 Usulan Solusi	42
Tabel 3. 6 Skenario Pengujian	43
Tabel 3. 7 Gantt Chart Pelaksanaan Capstone Project Sistem	44
Tabel 3. 8 Realisasi aktivitas pelaksanaan tugas akhir 1	45
Tabel 5. 1 Pembacaan Tachometer dan Serial Monitor	60
Tabel 5. 2 Data Pembacaan LCD	63
Tabel 5. 3 Data Pengukuran Pembacaan Teoritis RPM dan Aplikasi	67
Tabel 5. 4 Data Error dan Akurasi Kecepatan dan Jarak	70
Tabel 5. 5 Perbandingan performa antara sistem yang dibuat dengan sistem lain	74
Tabel 5. 6 Perbandingan usulan dan hasil perancangan sistem	75
Tabel 5. 7 Pengalaman Pengguna	77
Tabel 5. 8 Kesesuaian antara usulan dan realisasi timeline pengerjaan Tugas Akhir	79
Tabel 5. 9 RAB SPEETER Motor Listrik	79
Tabel 5. 10 RAB SPEETER Car Chasis	80
Tabel 5. 11 Realisasi aktivitas pelaksanaan tugas akhir 2	81
Tabel 5. 12 Perbandingan keseluruhan Speedometer	85

RINGKASAN

Motor listrik menjadi solusi transportasi ramah lingkungan di era modern, mengurangi emisi gas rumah kaca dan polusi udara. Namun, kebanyakan motor listrik saat ini tidak memiliki speedometer dan odometer yang terhubung dengan smartphone, mengakibatkan pengguna tidak dapat memantau kecepatan dan jarak tempuh secara *real-time*. Pemantauan ini penting untuk keamanan berkendara dan perencanaan perjalanan, serta pemeliharaan kendaraan. Penelitian ini mengusulkan sistem speedometer dan odometer berbasis *bluetooth* yang terintegrasi dengan *smartphone*. Sistem ini dirancang untuk memungkinkan pengguna motor listrik memantau kecepatan dan jarak tempuh melalui aplikasi *mobile* secara *real-time*.

Sistem yang diusulkan akan diukur performanya berdasarkan beberapa parameter: keandalan, yaitu menghitung waktu rata-rata antara kegagalan atau perbaikan yang diperlukan; akurasi, dengan membandingkan data kecepatan yang ditampilkan oleh speedometer kendaraan dengan data yang diukur oleh aplikasi "GPS Speedometer - Odometer For App", dan Tachometer serta antarmuka pengguna, yang mengukur respon antarmuka pengguna seperti layar LCD dan aplikasi *mobile*, termasuk waktu respon, kejelasan informasi, dan tingkat kepuasan pengguna.

Hasil rancangan menunjukkan dapat memberikan akurasi yang tinggi dalam membaca kecepatan dan jarak tempuh. Rata-rata *error* keseluruhan yang dihasilkan dari setiap kecepatan (2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam) sebesar 1.15% untuk *error* jarak dan 1.17% untuk *error* kecepatan. Dari kedua *error* tersebut, rata-rata akurasi jarak sebesar 98.85% dan kecepatan sebesar 98.83%. Selain itu, bahwa sistem dapat mengukur dan mentransmisikan data kecepatan dan jarak dengan akurasi tinggi melalui koneksi *bluetooth*. Sistem ini juga memungkinkan pengguna untuk memantau data secara *real-time* melalui aplikasi *Mobile*, meningkatkan kenyamanan dan keamanan berkendara. Kelebihan sistem ini meliputi kemampuan untuk menyediakan data yang akurat dan *real-time*, desain antarmuka pengguna yang ramah, dan integrasi yang baik dengan perangkat *mobile*.

BAB 1 PENDAHULUAN

1.1 Latar belakang dan Identifikasi Masalah

Sepeda motor berbasis listrik merupakan inovasi besar dalam dunia transportasi di era modern seperti saat ini. Dengan perkembangan teknologi yang sangat pesat, menjadikannya pilihan yang sangat tepat untuk dipakai di era ini. Berbeda dengan mesin konvensional yang masih menggunakan bahan bakar fosil sebagai sumber utamanya, motor listrik ini memakai energi listrik sebagai sumber daya utama yang tersimpan dalam baterai. Energi listrik pada motor ini akan diubah menjadi energi mekanik untuk menggerakkan motor. Hingga saat ini peminat dari motor listrik semakin tinggi setiap tahunnya. Menurut hasil riset Deloitte dan Foundry, jumlah penggunaan motor listrik di Indonesia naik 13 kali lipat selama periode 2020 – 2022. Riset ini menunjukkan, penggunaan motor listrik pada 2020 sebanyak 1947 unit lalu pada tahun 2022 naik signifikan menjadi 25782 unit [1].

Penggunaan motor listrik memberikan dampak positif yaitu bersifat ramah lingkungan karena dapat mengurangi emisi gas rumah kaca dan polusi udara. Saat ini kita ketahui kualitas udara di beberapa kota besar sudah menandakan tidak sehat yang salah satu penyebabnya adalah asap dari kendaraan konvensional. Maka dari itu, penggunaan motor listrik ini merupakan salah satu solusi untuk mengatasi polusi udara tersebut. Selain ramah lingkungan, motor listrik memiliki tingkat efisiensi yang tinggi karena biaya operasional yang rendah. Biaya operasional motor listrik ini disebutkan lebih hemat dibandingkan dengan motor yang berbasis bahan bakar minyak (BBM), bahkan biaya ini disebutkan bisa hemat hingga 75% [2].

Salah satu keprihatinan dalam sektor transportasi adalah meningkatnya penggunaan kendaraan bermotor yang masih memakai bahan bakar fosil. Praktik ini dapat mengakibatkan peningkatan konsumsi bahan bakar fosil yang berdampak pada pengurangan cadangan bahan bakar secara berkelanjutan. Dengan memperhitungkan tren ini, penipisan cadangan bahan bakar fosil, berupa cadangan minyak, batu bara, dan gas diprediksi akan habis dalam jangka waktu 35, 107, dan 37 tahun masing-masing [2].

Dampak lingkungan yang timbul dari penggunaan bahan bakar fosil adalah kekhawatiran besar di seluruh dunia. Di satu sisi, sektor ini mengonsumsi sejumlah besar bahan bakar yang mengakibatkan penurunan cadangan bahan bakar yang signifikan. Di sisi lain, hal ini juga menciptakan masalah polusi. Situasi ini dilema karena sektor transportasi merupakan kebutuhan

masyarakat tetapi juga dapat memiliki dampak negatif. Tanpa campur tangan strategis, permasalahan konsumsi bahan bakar dan emisi CO_2 dapat terus berlanjut atau bahkan meningkat dengan cepat. Mencari solusi di mana sistem transportasi beroperasi dengan menggunakan sumber daya yang ramah lingkungan akan menjadi pilihan terbaik dalam situasi ini [2].

Penelitian bidang transportasi listrik merupakan topik yang mendapatkan banyak perhatian. Lingkup topik ini sangat luas, meliputi aspek kebijakan, simulasi, pengembangan perangkat keras dan perangkat lunak, analisis ekonomi, dan perencanaan masa depan. Penelitian transportasi listrik yaitu motor listrik mencakup berbagai komponen yang salah satunya adalah inovasi dalam memodifikasi speedometer dan odometer [3].

Terdapat dua fitur penting pada motor listrik yaitu speedometer dan odometer. Speedometer memegang peranan penting yaitu dapat menampilkan dan mengukur kecepatan kendaraan. Seorang pengemudi kendaraan akan selalu memantau dan menjaga kecepatan kendaraan terhadap kondisi dari jalan atau medan yang dilaluinya sehingga hal ini dapat meningkatkan keselamatan pengendara. Odometer juga memegang peranan penting yaitu dapat mengukur dan mencatat jarak tempuh kendaraan dan seorang pengemudi dapat mengevaluasi berapa jarak tempuh sebelum mengisi daya ulang baterai [4].

Saat ini speedometer dan odometer ada yang berjenis analog dan digital. Penggunaan speedometer analog menggunakan mekanisme roda bergigi yang terhubung ke poros roda dan dilengkapi dengan jarum sebagai penunjuk kecepatan pada *instrument cluster* di mobil Anda. Pada indikator tertulis angka dari nol hingga kecepatan maksimum. Jenis analog tersebut dioperasikan melalui kabel yang dilengkapi dengan kawat baca. Keunggulan dari jenis speedometer analog adalah harga suku cadangnya terjangkau dan mudah didapat pada setiap dealer resmi [13].

Perbedaannya dengan speedometer digital, menggunakan pulsa magnetik dan pulser pada roda (sistem gigi). Cara kerjanya putaran roda depan melalui sistem gir memutar kawat kopel yang terhubung ke *instrument display*. Di dalam *instrument display* tersebut, kawat speedo memutar roda berlubang. Pada setiap lubang melalui *optocoupler* akan menghasilkan sinyal *logic 1*. Sinyal tersebut akan bekerja secara terus menerus selama roda depan berputar. Sinyal *logic* ini dihitung melalui *microcontroller* untuk kemudian ditampilkan melalui LCD 7-segmen, yang juga dapat mengkalkulasi berapa digit di belakang koma [13].

Namun, saat ini kebanyakan motor listrik mempunyai speedometer dan odometer yang tidak bisa terhubung oleh *smartphone*. Hal ini membuat pengguna motor listrik tidak bisa memantau kecepatan dan jarak tempuh yang telah mereka lalui. Pemantauan kecepatan dapat membantu pengemudi memastikan bahwa kendaraan beroperasi pada kecepatan yang sesuai dan aman. Hal ini penting untuk menjaga keamanan berkendara, mematuhi peraturan lalu lintas, dan mencegah potensi bahaya saat di jalan. Menurut Sekretaris Direktorat Jenderal Perhubungan Darat, Amirulloh, secara umum motor itu menjadi salah satu penyebab kecelakaan terbesar [3]. Selain itu, dengan adanya pemantauan jarak tempuh membantu pengemudi untuk mengawasi sejauh mana kendaraan mereka telah berjalan. Data ini dapat digunakan untuk menjadwalkan perawatan secara berkala dan menghitung efisiensi bahan bakar atau konsumsi daya baterai pada motor listrik serta membantu pengendara untuk merencanakan perjalanan mereka dengan lebih baik agar tidak kehabisan daya saat di tengah perjalanan.

Proses wawancara dengan *stakeholder* diawali dengan dosen pembimbing menghubungi CEO PT Fusi Global Teknologi. Setelah mendapatkan waktu wawancara dengan narasumber, kami menyiapkan beberapa daftar pertanyaan untuk menyelesaikan permasalahan. Terdapat beberapa pertanyaan yang disiapkan dan jawaban dari CEO PT Fusi Global Teknologi yang ditunjukkan pada Tabel 1.1. Wawancara dilakukan pada Senin, 19 September 2023 jam 13.00 sampai 14.00 WIB.

Tabel 1. 1 Hasil survei antara pengembang dan pengguna

Pertanyaan	Jawaban/tanggapan
Apa jenis kendaraan yang digunakan oleh PT Fusi Global Teknologi?	Jenis kendaraan yang digunakan kendaraan bermotor roda dua dan roda empat
Apa jenis kendaraan yang digunakan untuk implementasi pada proyek kali ini?	Dalam implementasi pada proyek kali ini berfokus di kendaraan bermotor roda dua (motor listrik)
Pada proyek kali ini, sistem komunikasinya dengan menggunakan sistem apa?	Menggunakan sistem <i>Controller Area Network</i> (CAN) Bus dengan integritas modul <i>bluetooth</i>

Pertanyaan	Jawaban/tanggapan
Apakah anda melihat manfaat penggunaan Speedometer dan Odometer berbasis <i>bluetooth</i> pada kendaraan?	Banyak manfaatnya, dikarenakan pengguna dapat mengakses data secara <i>real time</i> melalui server
Apakah diperlukan pemantauan kecepatan dan jarak melalui perangkat secara <i>real-time</i> ?	Sangat diperlukan, dikarenakan dapat memantau dari jarak jauh dengan melihat dari server
Apakah dengan teknologi ini dapat meningkatkan kenyamanan dan keamanan saat berkendara?	Betul, dikarenakan nantinya pengguna dapat melihat <i>history</i> dan memantau secara langsung
Apakah dengan pemantauan kecepatan dapat mengurangi insiden pelanggaran batas kecepatan kendaraan?	Iya, dikarenakan kita mengetahui data pengguna di dalam riwayat pengguna.
Pada proyek ini selain memantau speedometer dan odometer, apakah ada parameter lainnya yang akan ditambahkan?	Ada, salah satunya menambahkan parameter pembacaan baterai, suhu, dan parameter lainnya.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, beberapa permasalahan dapat dirumuskan sebagai berikut:

1. Bagaimana merancang sistem speedometer dan odometer pada motor listrik yang terhubung dengan *bluetooth*?
2. Bagaimana cara pengguna motor listrik dapat memantau kecepatan dan jarak tempuh?
3. Bagaimana tingkat akurasi pengukuran kecepatan dan jarak dapat dicapai sesuai dengan standar regulasi internasional untuk speedometer dan odometer motor listrik?

1.3 Tujuan

Dalam Tugas Akhir ini, memiliki beberapa tujuan. Adapun tujuan Tugas Akhir 1 ini adalah sebagai berikut:

1. Merancang speedometer dan odometer pada motor listrik yang dapat terhubung ke aplikasi berbasis Android dengan menggunakan *bluetooth*.
2. Membuat sistem yang dapat terhubung dengan *smartphone* untuk mengakses data kecepatan dan jarak tempuh terkait riwayat selama berkendara.
3. Merancang sistem yang dapat mengukur dan mentransmisikan data kecepatan dan jarak dengan akurasi tinggi melalui koneksi *bluetooth*.

1.4 Batasan Masalah

Dalam Tugas Akhir ini, memiliki beberapa tujuan. Adapun batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

- 1) Sistem ini menggunakan *bluetooth* sebagai sarana komunikasi antara motor listrik dengan *smartphone*.
- 2) Sistem ini berfokus kepada speedometer dan odometer pada motor listrik.
- 3) Sistem ini hanya menggunakan *smartphone* berbasis Android untuk menerima data hasil pembacaan data kecepatan dan jarak tempuh.

1.5 Batasan Realistis Aspek Keteknikan

- 1) Sistem ini dirancang memakai daya yang rendah agar tidak membebani baterai pada motor listrik.
- 2) Sistem ini harus memastikan koneksi *bluetooth* dengan Android dalam batas aman (< 10 meter) agar koneksi tetap stabil.
- 3) Sistem ini didesain menggunakan layar *Liquid-crystal display* (LCD) berukuran 3,6 *inch* dan aplikasi Android untuk mempermudah pembacaan parameter oleh pengguna motor listrik.
- 4) Sistem ini dirancang untuk memantau kecepatan dan jarak tempuh di aplikasi Android dengan fitur-fitur seperti kecepatan maksimal, kecepatan tertinggi, kecepatan rata-rata dan jarak tempuh.

BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM

2.1 Studi Literatur dan Observasi

Studi literatur diperlukan guna mengumpulkan berbagai macam informasi berkaitan dengan solusi yang akan dirancang untuk menanggulangi permasalahan yang telah dibahas sebelumnya. Selain itu, studi literatur juga berfungsi untuk memberikan kerangka berpikir yang jelas sehingga memberikan gambaran mengenai pembahasan yang akan dikaji. Di sisi lain, bahan perbandingan dan acuan juga menjadi fokus dalam kajian ini. Berikut merupakan beberapa kumpulan sumber informasi terkait alternatif solusi untuk memecahkan permasalahan pada pembahasan sebelumnya.

Pada penelitian yang dibuat oleh Ridwan et al. dengan judul “Rancang Bangun Sistem Akuisisi Data untuk *Dashboard* Android pada Sepeda Motor Listrik”, peneliti ini memberikan sebuah solusi yang ditawarkan Solusi yang ditawarkan peneliti berupa sebuah rancang bangun akuisisi data untuk dashboard pada sepeda motor listrik dengan teknologi komunikasi data *Universal Asynchronous Receiver Transmitter* (UART) dengan sistem ini dapat mengumpulkan berbagai jenis data dan mengirimkannya dengan kebutuhan dan memiliki kelebihan dalam mengumpulkan data dengan akurasi tinggi dan memiliki persentase *error* yang rendah. Didapatkan hasil rata-rata persentase *error* sekitar 1.072% dari pengiriman data dengan parameter jeda waktu 0.1 detik. Parameter 0.1 detik dipilih berdasarkan waktu tunggu yang diperlukan untuk menunggu setiap data dari perangkat berbeda dapat terkumpul pada perangkat Raspberry PI. Rata-rata jeda waktu pengiriman data dari perangkat BLDC *controller* serta pengiriman data dari perangkat BMS masing-masing berada pada sekitar 0.05 detik [11].

Pada penelitian yang dibuat oleh Agus Ulinuha et al. dengan judul “*Design and Implementation of Data Acquisition Device and Instrumentation Based on Microcontroller for Electric Motorcycle*”, peneliti ini memberikan solusi yang ditawarkan yaitu sistem tertanam untuk pengukuran dan akuisisi data pada sepeda motor listrik yang meliputi penggerak mula, pengontrol, penyimpan daya, dan beberapa komponen fisik dengan solusi tersebut dapat pengembangan sistem instrumentasi termasuk akuisisi data yang menggunakan mikrokontroler arduino mega memiliki kelebihan kemampuannya dalam mengumpulkan data dengan akurasi tinggi dan memiliki persentase *error* yang rendah [12].

Tabel 2. 1 Hasil Studi Literatur Solusi Sejenis

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
Rancang Bangun Sistem Akuisisi Data untuk <i>Dashboard</i> Android pada Sepeda Motor Listrik	Solusi yang ditawarkan peneliti berupa sebuah rancang bangun akuisisi data untuk dashboard pada sepeda motor listrik dengan teknologi komunikasi data <i>Universal Asynchronous Receiver Transmitter</i> (UART) dengan menggunakan RS-232 sebagai protokol komunikasi serial.	Hasil: Sistem ini mampu mengumpulkan berbagai jenis data yang diperlukan dan mengirimkannya sesuai dengan kebutuhan sistem Android. Kelebihan: kemampuannya dalam mengumpulkan data dengan akurasi tinggi dan memiliki persentase <i>error</i> yang rendah. Kekurangan: Sistem ini tidak dapat melakukan produksi mikrokontroler secara mandiri.
<i>Design and Implementation of Data Acquisition Device and Instrumentation Based on Microcontroller for Electric Motorcycle</i>	Sistem tertanam untuk pengukuran dan akuisisi data pada sepeda motor listrik yang meliputi penggerak mula, pengontrol, penyimpanan daya, dan beberapa komponen fisik	Hasil: Pengembangan sistem instrumentasi termasuk akuisisi data yang menggunakan mikrokontroler Arduino Mega Kelebihan: Sistem instrumentasi menggunakan Arduino Mega dan sensor-sensor, dengan tampilan TFT Shield 3,5 inci, mengukur dan menyajikan data kondisi pengoperasian secara <i>real time</i> . Hasil pengukuran menunjukkan akurasi tinggi, memberikan informasi komprehensif untuk berkendara optimal dengan kinerja dan efisiensi maksimal.

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
		Kekurangan: Penelitian tersebut tidak menyebutkan kekurangan apa pun dari sistem instrumentasi atau sepeda motor listrik yang dikembangkan.

Hasil dari studi literatur yang sudah ada, peneliti membuat sebuah sistem speedometer dan odometer yang akan diterapkan di pengguna motor listrik dengan menggunakan speedometer dan odometer digital. Perbedaan dari penelitian sebelumnya adalah sistem ini akan terhubung dengan *smartphone*.

2.2 Dasar Teori

2.2.1 Speedometer

Speedometer adalah sebuah alat pengukur kelajuan kendaraan darat, yang merupakan perlengkapan standar setiap kendaraan yang beroperasi di jalan. Speedometer berfungsi agar pengemudi mengetahui kelajuan kendaraan yang dijalkannya dan dijadikan informasi utama untuk mengendalikan kelajuan di kawasan/jalan agar tidak terlalu lambat atau terlalu cepat, bisa mengatur waktu perjalanan dan mengendalikan kelajuan di jalan yang kelajuannya dibatasi. Speedometer turun bersamaan dengan kelajuan kendaraan. Speedometer memiliki dua jenis, yakni tipe analog dan digital [7].

a. Speedometer Analog

Speedometer analog memakai jarum untuk menunjukkan angka kecepatan kendaraan. Secara tampilan, speedometer ini desain unik dan retro. Itulah sebabnya meski sekarang penggunaan speedometer digital sudah marak tetap ada kendaraan yang menggunakan jenis speedometer analog seperti Gambar 2.1 [7].



Gambar 2. 1 Speedometer Analog

Speedometer digital sudah menggunakan penunjuk angka pada sebuah layar display. Speedo ini, memiliki tampilan yang futuristik dan akurat. Oleh sebab itu banyak dipakai pada kendaraan yang mengedepankan nilai futuristik seperti Gambar 2.2.



Gambar 2. 2 Speedometer Digital

2.2.2 Odometer

Odometer adalah sebuah alat ukur yang menunjukkan jarak yang telah ditempuh pada suatu kendaraan. Apabila speedometer di kendaraan tersebut menggunakan satuan Kilometer/Jam, maka angka odometer yang ditunjukkan berupa jarak dalam satuan kilometer. Biasanya Odometer

digunakan untuk mengetahui kapan waktunya mengganti oli dikarenakan penggantian oli pada kendaraan berdasarkan jarak yang telah ditempuh dan sering dikaitkan untuk menentukan kendaraan tersebut jarang digunakan atau sering digunakan pada saat kita akan membeli kendaraan bekas. Berikut Gambar 2.3 adalah odometer analog sedangkan Gambar 2.4 adalah odometer dengan penunjukkan angka digital [7].



Gambar 2. 3 Odometer Analog



Gambar 2. 4 Odometer Digital

2.2.3 Controller Votol

Votol merupakan perusahaan teknologi yang menghadirkan solusi pengendalian motor cerdas untuk kendaraan listrik dan *Internet of Things* (IoT). Mereka menyajikan beragam pengendali yang dapat diprogram, seperti pengendali motor *mid-drive* BLDC EM-100 dan EM-150. EM100 merupakan pengendali gelombang sinus dengan rentang tegangan 48V hingga 72V dengan arus baterai maksimum 100 A dan arus fasa 330 A. Berbagai fitur utama yang dimilikinya yaitu fitur mengendalikan kecepatan sehingga kecepatan penuh sudah disesuaikan, bisa mengontrol tiga kecepatan: 60%, 80%, dan 100%, kemampuan dalam bergerak mundur,

meningkatkan kontrol pengereman, dan fitur pembangkit ulang energi dan pelemahan fluks yang dapat meningkatkan efisiensi energi. *Controller* Votol digunakan dalam berbagai kendaraan listrik sepeda motor listrik, skuter listrik, dan mobil listrik. Selain itu, protokol komunikasi serial Votol juga telah diuji dan diinterpretasikan oleh beberapa pengguna [8].

Controller Votol bersifat *programmable* yaitu pengguna dapat mengkonfigurasi suatu pengaturan tertentu seperti penyesuaian dengan kebutuhan motor BLDC yang digunakan dengan perangkat lunak. Votol bisa dikatakan sebagai *controller* yang tugasnya mirip dengan CDI pada motor konvensional. Kebutuhan *controller* harus disesuaikan dengan motor yang ingin dimiliki dan harus sesuai dengan karakter motornya. Biasanya, untuk motor-motor harian, bisa memakai standar dengan output 20 A. Semakin besar jenis votol yang dipakai maka output yang dihasilkan semakin tinggi dan tenaga motor yang dihasilkan juga semakin besar [8].

2.2.4 CAN BUS Shield (MCP2515)

Jaringan *Controller Area Network* (CAN) Bus banyak dipakai dalam pengembangan teknologi kendaraan, peralatan pertanian dan di dunia industri. Jaringan CAN Bus digunakan untuk pengontrolan dan akuisisi data. CAN terdiri dari dua kabel *CAN High* dan *CAN Low*, dalam pengaplikasiannya kabel dirancang mampu menahan interferensi dari *noise* dari luar, kabel CAN biasanya menggunakan kabel *twisted pair*. CAN Bus membutuhkan resistor terminasi 120 Ohm di setiap ujung BUS. Pada *shield module* sudah terpasang resistor terminasi. Ketika dua *shield* CAN Bus dihubungkan maka CAN Bus sudah terhubung dengan terminal resistor 120 Ohm [9].



Gambar 2. 5 CAN Bus Shield MCP2515

Data yang dikirim dalam komunikasi CAN terdiri dari data dan CAN ID. Data terdiri dari 1 hingga 8 *byte* untuk setiap pesan. Setiap *byte* dapat memiliki nilai antara 0 hingga 255 atau dalam

format *hexadecimal* 0x00 hingga 0xFF. Ketika sebuah pesan dikirim melalui jaringan CAN Bus dengan menggunakan ID. Jumlah *byte* yang terkirim disebut sebagai *Data Length Code* (DLC) dan melibatkan data aktual. Penerima hanya membaca jumlah *byte* yang terdapat dalam pesan tersebut. Misalnya, jika pengirim mengirim pesan dengan DLC 4 tetapi pesan sebenarnya berisi 8 *byte*, penerima hanya akan membaca 4 *byte* pertama. Meskipun 8 *byte* per pesan memiliki keandalan bus menjadi tinggi. Terkadang, diperlukan kreativitas untuk menyelipkan data ke dalam *byte* tersebut. Jika nilai yang dikirim kurang dari 255, cukup menggunakan satu *byte* [9].

2.2.5 Arduino Uno

Arduino UNO adalah sebuah *board mikrokontroler* yang didasarkan berbaris ATmega328P. Arduino UNO mempunyai 14 pin *digital input/output* (6 di antaranya dapat digunakan sebagai *output PWM*), 6 input *analog*, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino UNO memuat semua yang dibutuhkan untuk menunjang *mikrokontroler*, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai [6].

Arduino Uno memiliki SRAM sebesar 2KB, EEPROM sebesar 1KB, dan dilengkapi *Flash Memory* sebesar 32KB. *Static Random Access Memory* (SRAM) digunakan sebagai memori kerja selama program dijalankan. Memori yang digunakan untuk menyimpan variabel. *Electrically Erasable Programmable Read-Only Memory* (EEPROM) adalah memori yang digunakan untuk menyimpan data secara permanen. *Flash Memory* digunakan untuk menyimpan program [6].



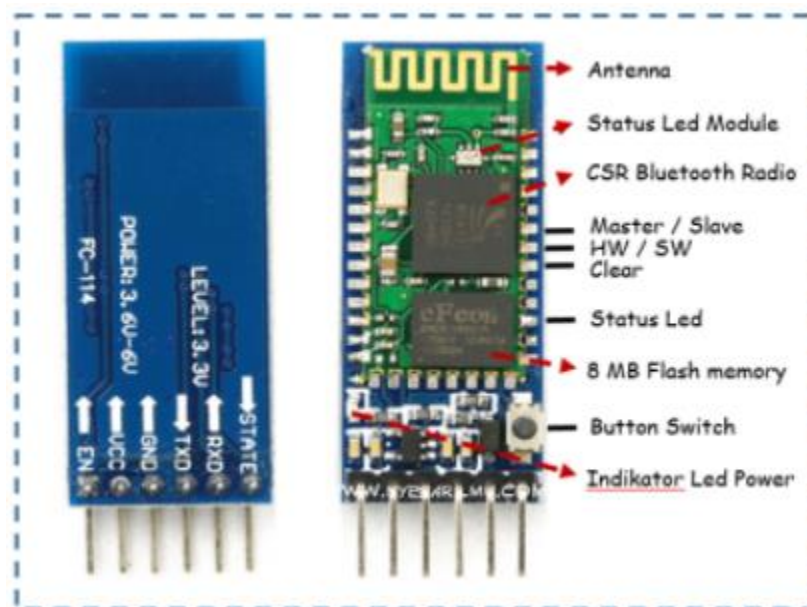
Gambar 2. 6 Arduino Uno

2.2.6 Modul Bluetooth

Bluetooth adalah spesifikasi industri untuk jaringan kawasan pribadi (*Personal Area Networks*) tanpa kabel. *Bluetooth* dapat menghubungkan dan atau dapat dipakai untuk melakukan tukar-menukar informasi atau data di antara peralatan-peralatan seperti PDA, laptop, HP, dan lain-lain. *Bluetooth* sendiri beroperasi dalam pita frekuensi 2,4 Ghz yang mampu menyediakan layanan komunikasi data dan suara secara *real time* antara *host Bluetooth* dengan jarak terbatas. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05 [6].

Modul *Bluetooth* HC-05 adalah sebuah modul *Bluetooth Serial Port Protocol* yang mudah digunakan untuk komunikasi nirkabel yang mengkonversi port serial ke *Bluetooth*. HC-05 menggunakan modulasi *bluetooth V2.0 + Enhanced Data Rate (EDR)* 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz. Modul ini dapat digunakan sebagai *slave* maupun master. *Bluetooth* HC-05 memiliki dua mode konfigurasi, yaitu AT mode dan *Communication mode*. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi *bluetooth* dengan piranti lain [6].

Modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor yang setiap pin konektor memiliki fungsi yang berbeda-beda seperti Gambar 2.7.



Gambar 2. 7 Modul *Bluetooth*

2.2.7 *Liquid Crystal Display (LCD)*

Liquid Crystal Display (LCD) adalah salah satu jenis display elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka maupun Gambar.

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan *sandwich* memiliki polarizer cahaya vertikal depan dan polarizer cahaya horisontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan [6].



Gambar 2. 8 *Liquid Crystal Display TFT*

2.3 *Analisis Stakeholder*

2.3.1 *PT Fusi Global Teknologi*

Perusahaan sebagai mitra yang memberikan informasi terkait potensi *dashboard* speedometer dan odometer yang akan digunakan oleh pengguna motor listrik.

2.3.2 *Perusahaan Motor Listrik*

Perusahaan motor listrik terlibat dalam pemasangan speedometer dan odometer pada kendaraan listrik, memastikan peningkatan efisiensi dan kualitas produksi.

2.3.3 Pengguna Motor Listrik

Pengguna motor listrik umumnya juga terlibat dalam penggunaan speedometer dan odometer karena dapat mengukur kecepatan dan jarak yang telah ditempuh serta membantu pengguna motor listrik memantau riwayat perjalanan mereka.

2.4 Analisis Aspek yang Mempengaruhi Sistem

2.4.1 Aspek Ekonomi

Perancangan harus mempertimbangkan biaya produksi yang efisien sehingga dapat memahami permintaan pasar, tren konsumen, dan kondisi persaingan untuk mengidentifikasi peluang dan tantangan.

2.4.2 Aspek Keamanan

Sistem speedometer dan odometer harus dirancang dengan mempertimbangkan keselamatan pengendara. Informasi yang ditampilkan harus jelas dan mudah dibaca untuk mengurangi gangguan saat mengemudi. Desain antarmuka harus ramah pengguna agar pengendara dapat dengan mudah memahami dan menggunakan sistem tersebut.

2.4.3 Aspek Teknologi

Sistem ini menggunakan teknologi nirkabel yaitu *bluetooth* sehingga tidak membutuhkan kabel transmisi dalam menghubungkan ke smartphone berbasis Android. Sistem harus dirancang agar mudah diperbaiki atau diperbarui jika diperlukan

2.4.4 Aspek Regulasi

Perancangan harus mematuhi regulasi yang berlaku terkait dengan speedometer dan odometer. Menurut peraturan internasional, kecepatan motor tidak boleh lebih dari $110\% + 4$ km/jam dari kecepatan asli kendaraan[14].

2.5 Spesifikasi Sistem

Spesifikasi sistem meliputi hal-hal yang dibutuhkan untuk melakukan rancang bangun speedometer dan odometer secara umum, adapun spesifikasi yang di gunakan sebagai berikut.

- Pembacaan kecepatan dan jarak tempuh menggunakan LCD digital dan aplikasi Android.
- Sistem harus memenuhi kisaran harga yang dapat diterima oleh pasar dan konsumen.
- Memiliki sistem pengaman kelistrikan.
- Sistem harus dirancang untuk memudahkan pemeliharaan dan perbaikan jika diperlukan.

- Sistem memiliki kemampuan untuk memonitor kecepatan dan jarak tempuh motor listrik melalui *Bluetooth*.
- Sistem dilengkapi dengan kemampuan pengujian dan pemecahan masalah untuk memastikan koneksi *bluetooth* berfungsi dengan baik.
- Sistem ini memakai protokol komunikasi CAN Bus.
- Sistem ini dirancang untuk memakai daya yang rendah agar tidak memberatkan baterai motor listrik.
- Sistem ini hanya menggunakan *smartphone* berbasis Android untuk menerima data hasil pembacaan kecepatan dan jarak tempuh motor listrik.

BAB 3. USULAN SOLUSI

Penyelesaian masalah dilakukan menggunakan pendekatan *design thinking*. *Design thinking* secara umum dapat diartikan sebagai konsep berpikir dalam mengobservasi ide. Lockwood mengungkapkan bahwa konsep *design thinking* merupakan teknik yang sangat dasar bagi individu yang dipakai sebagai pusat proses inovasi melalui pengamatan, kolaborasi, belajar cepat, visualisasi ide, *prototyping* konsep, serta analisis yang sangat dibutuhkan pada inovasi dan strategi. *Design thinking* secara umum dapat diartikan sebagai konsep berpikir dalam mengobservasi suatu ide. Metode ini terdiri dari enam tahap yakni *empathize*, *define*, *ideate*, *prototype*, *test* dan *implement*. Tahapan-tahapan tersebut seperti siklus yang didalamnya dapat terjadi perubahan, perbaikan, dan penambahan yang bertujuan untuk memenuhi spesifikasi dan batasan masalah sistem [10]. Deskripsi penggunaan metode tersebut untuk penyelesaian masalah sebagai berikut.

Langkah-langkah dalam membuat proyek sistem keamanan rumah berbasis IoT ini menggunakan konsep *engineering cycle*. Dimana konsep ini akan menghasilkan solusi terbaik dari suatu permasalahan yang ada [10]. Tahapan-tahapan *engineering cycle* terdiri dari *understand*, *explore*, dan *materialize*. Dari ketiga tahapan tersebut masih bisa diperjelas sebagai berikut:

1. *Empathize*

Pada tahap *empathize*, menemukan masalah terhadap pengguna motor listrik tidak bisa memantau kecepatan dan jarak tempuh yang mereka lalui.

2. *Define*

Pada tahap *define*, dikarenakan motor listrik tidak ada fitur yang dapat memantau kecepatan dan jarak secara *real time* sehingga pengguna motor listrik tidak memiliki informasi terkait riwayat perjalanan mereka.

3. *Ideate*

Pada tahap *Ideate*, tmendapatkan sebuah solusi yaitu suatu sistem yang dapat memantau kecepatan dan jarak secara *real time*.

4. *Prototype*

Pada tahap *prototype*, membuat desain 3D dan alat serta membuat kode program sistem elektronik.

5. *Test*

Pada tahap *test*, dapat melakukan uji coba dan evaluasi sistem yang telah dibuat. Dengan tujuan untuk mengetahui apakah sistem ini layak untuk dipakai atau masih ada kekurangan.

6. Implement

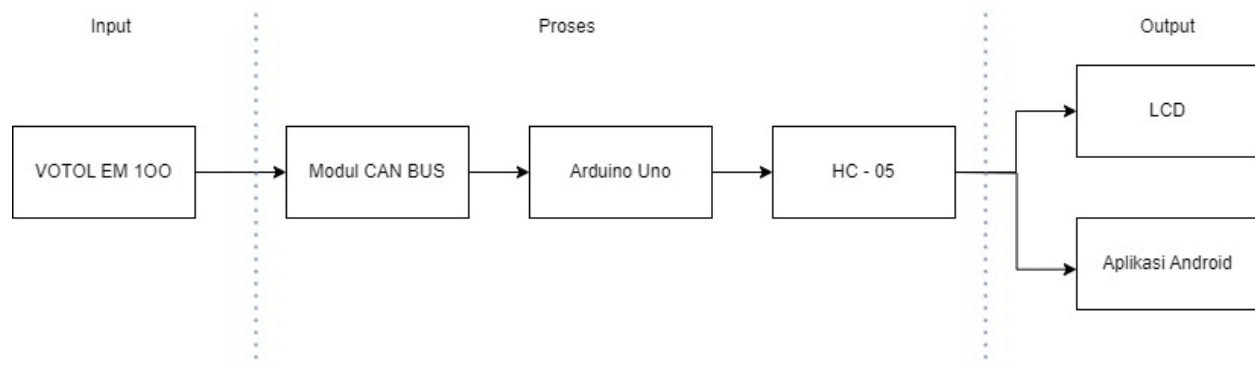
Setelah berhasil melewati tahap uji coba, alat ini akan diimplementasikan lebih lanjut.

Pada bab ini, sudah mencapai tahapan *prototype*, yaitu menganalisis komponen-komponen yang diperlukan untuk membuat sistem speedometer dan odometer. Dalam tahap *prototype* memerlukan perancangan desain *prototype* untuk menganalisis komponen yang digunakan tepat guna untuk menyelesaikan permasalahan. Berikut merupakan beberapa usulan sebagai bahan pertimbangan rancangan sistem.

3.1 Usulan Solusi 1

Usulan solusi kedua adalah sistem *connected vehicle* yang diusulkan dengan mendeteksi nilai keluaran speedometer dan odometer yang terintegritas sistem *Internet of Things* yaitu menggunakan *bluetooth* dengan menampilkan di layar LCD dan aplikasi SPEETER. "SPEETER" adalah nama alat speedometer dan odometer untuk motor listrik.

3.1.1 Desain Sistem 1

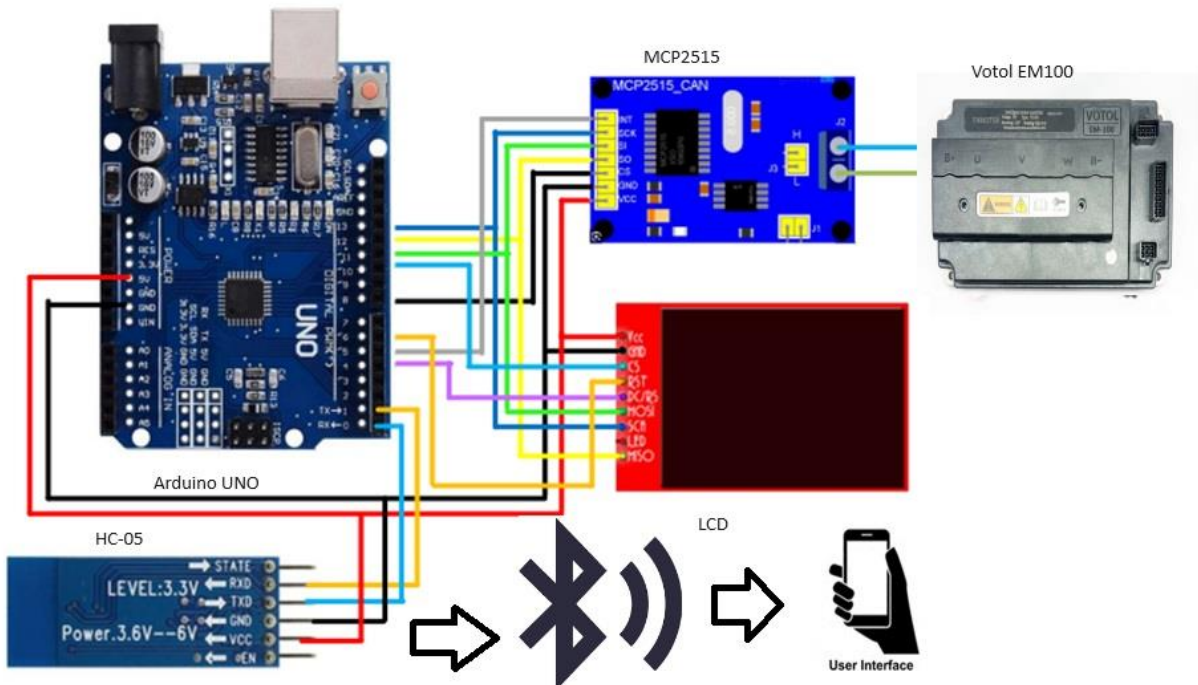


Gambar 3. 1 Diagram Blok Sistem

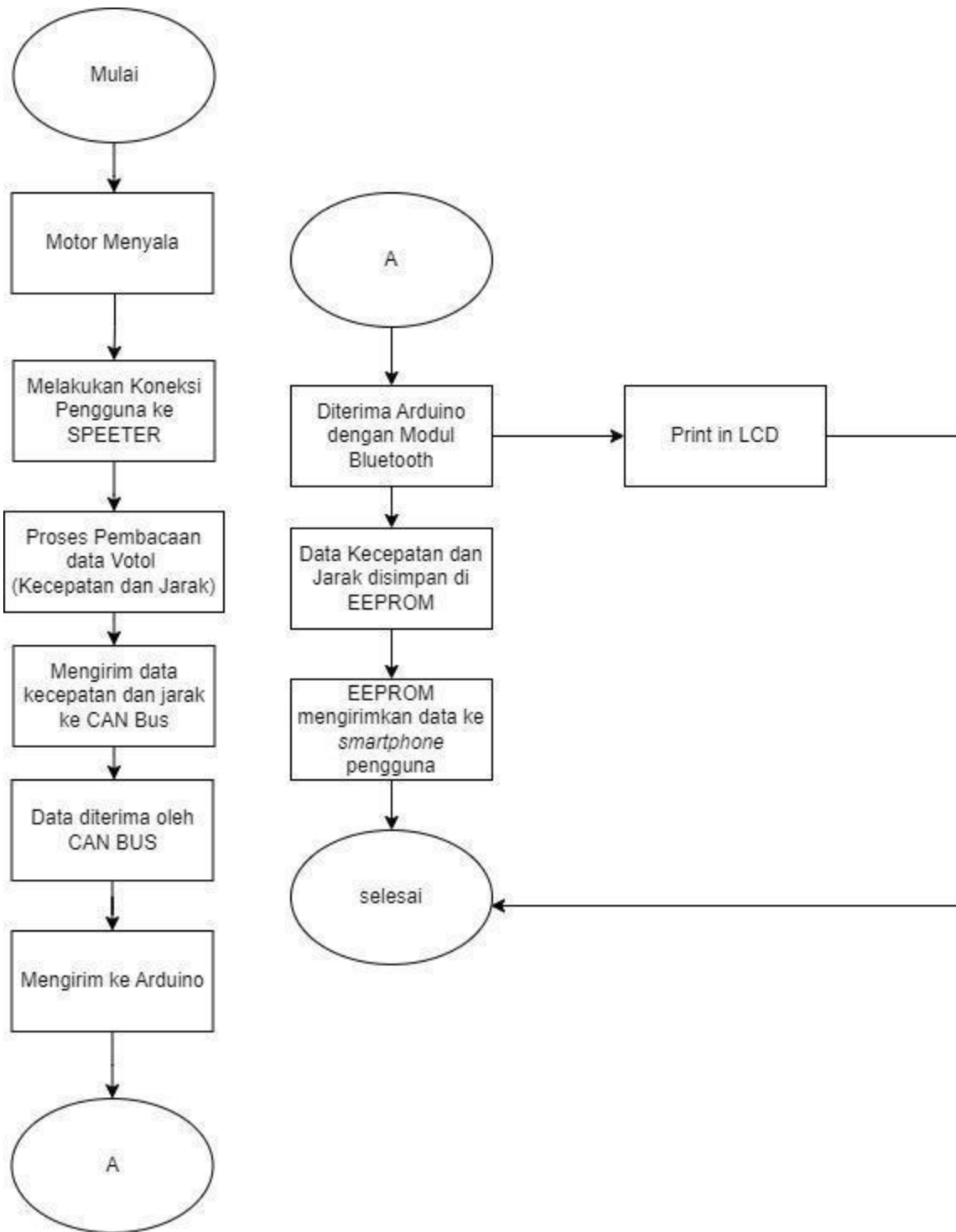
Cara kerja sistem ini melibatkan *controller* Votol EM100, modul CAN Bus, Arduino Uno, LCD, dan aplikasi pembaca data pada *smartphone*. *Controller* Votol EM100 berfungsi sebagai otak utama pengendalian motor listrik yang dapat mengatur kecepatan dan jarak kendaraan. Modul CAN Bus menjadi perantara antara Arduino Uno dan sistem kendaraan. Hal ini memungkinkan Arduino untuk berkomunikasi dengan CAN Bus.

Arduino UNO dengan modul *bluetooth* berfungsi untuk mengolah data yang menerima informasi dari sensor-sensor pada speedometer dan odometer melalui modul CAN Bus. Setelah menerima data, arduino Uno akan memproses dan menampilkannya pada LCD sebagai tampilan informasi. LCD ini memberikan pengguna kemampuan untuk secara *real-time* memantau kecepatan dan jarak tempuh kendaraan.

Selanjutnya menggunakan aplikasi *smartphone* agar pengguna dapat dengan mudah membaca dan memantau informasi speedometer dan odometer dari jarak jauh. Aplikasi ini berupa Aplikasi SPEETER yang dapat menampilkan data informasi pembacaan speedometer dan odometer. Berikut desain sistem elektronik dan diagram blok sistem yang ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Proses cara kerja sistem



Gambar 3. 3 Diagram Alir Sistem

Cara kerja sistem ini akan berjalan ketika motor sudah menyala. Selanjutnya, pengguna melakukan koneksi dengan speedometer dan odometer (SPEETER) pada motor listrik untuk mengakses informasi tentang kecepatan dan jarak. Setelah melakukan koneksi, Votol mulai membaca data berupa kecepatan dan jarak tempuh. Data ini kemudian dikirim ke *Controller Area*

Network (CAN Bus), yang mana CAN Bus berperan sebagai media komunikasi untuk mengirim data kecepatan dan jarak tempuh. Data kecepatan dan jarak tempuh akan disimpan di *Electrically Erasable Programmable Read-Only Memory* (EEPROM) untuk menyimpan data. EEPROM bertugas mengirimkan data yang telah tersimpan ke *smartphone* pengguna melalui koneksi *bluetooth*.

Modul *bluetooth* memiliki kelebihan yaitu beroperasi tanpa memerlukan driver khusus dan memastikan integrasi yang mudah dengan berbagai perangkat dan *platform*. Selain itu, memiliki konsumsi daya yang rendah menjadikannya usulan yang cocok untuk perangkat bertenaga baterai, seperti perangkat *wearable* dan sensor nirkabel. Keunggulan lainnya adalah kemampuannya untuk mendukung berbagai jenis aplikasi, mulai dari *transfer file* hingga audio nirkabel, serta konektivitas yang luas dengan berbagai perangkat seperti headset dan speaker. Dengan fleksibilitas dan efisiensi yang ditawarkan, modul *bluetooth* menjadi solusi untuk memenuhi kebutuhan konektivitas nirkabel dalam berbagai konteks, menyederhanakan proses integrasi, dan memberikan pengalaman pengguna yang optimal [5].

Desain Gambar 3.4 dan 3.5 ini mencakup integrasi teknologi terkini untuk mengukur dan menyajikan informasi kecepatan serta jarak tempuh dengan akurasi tinggi. Selain itu, antarmuka pengguna dirancang untuk memberikan informasi bagi pengemudi. Desain ini tidak hanya memastikan kinerja yang optimal dalam mengukur dan melaporkan data kendaraan, tetapi juga mempertimbangkan aspek estetika dan kenyamanan pengguna. Dengan demikian, desain ini dianggap sebagai solusi terbaik dan siap untuk diimplementasikan dalam produksi speedometer dan odometer yang memenuhi standar tinggi serta harapan pengguna.



Gambar 3. 4 Desain Sistem Tampak Samping



Gambar 3. 5 Desain Sistem 1 Tampak Depan

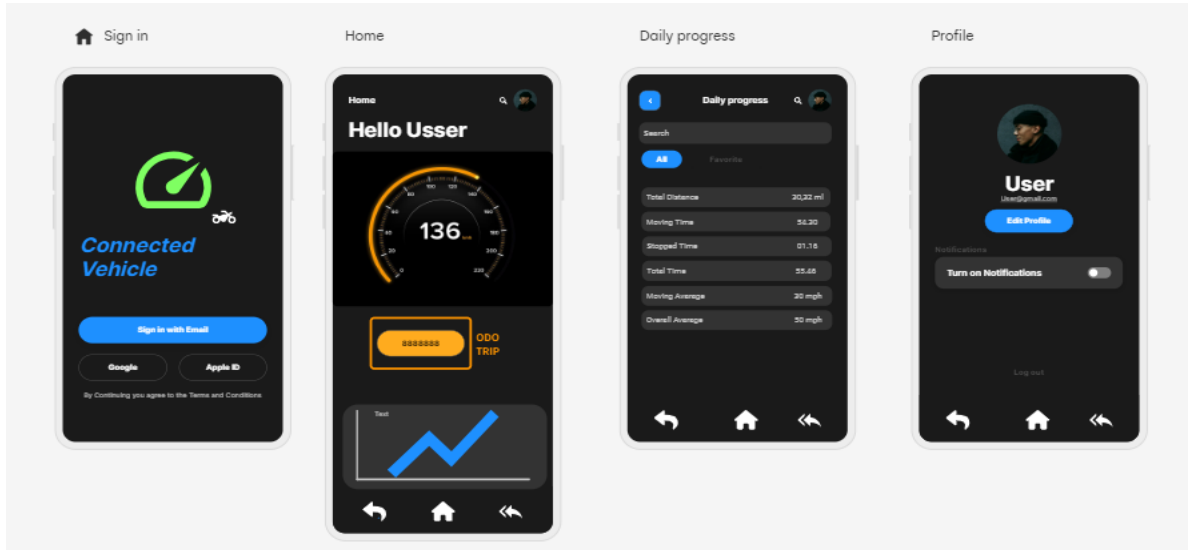
Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.1 menunjukkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3. 1 Inventarisasi kebutuhan usulan sistem perangkat keras

No	Nama Alat	Keterangan
1	Votol EM 100	Merupakan kontroler motor listrik yang digunakan untuk mengendalikan motor listrik di kendaraan elektrik. EM100 menerima sinyal dari Arduino Uno untuk mengatur kecepatan dan arah motor.
2	Modul CAN Bus	Modul CAN Bus digunakan untuk berkomunikasi dengan berbagai perangkat dalam kendaraan, seperti sensor suhu, sensor kecepatan, sensor baterai, dan lainnya. Modul ini menerima dan mengirim data melalui protokol <i>Controller Area Network</i> (CAN) untuk mengumpulkan informasi tentang kondisi kendaraan.
3	Arduino Uno	Arduino Uno sebagai otak utama dalam sistem yang menerima data dari Votol EM100 melalui Modul CAN Bus dan mengontrol operasi motor atau perangkat lainnya. Arduino Uno juga memiliki kemampuan untuk mengendalikan perangkat lain, seperti modul <i>bluetooth</i> dan LCD.
4	HC-05	HC-05 Adalah sebuah modul <i>bluetooth Serial Port Protocol</i> (SPP) yang mudah digunakan untuk komunikasi serial wireless (nirkabel) yang mengkonversi port serial ke <i>bluetooth</i> . HC-05 menggunakan modulasi <i>bluetooth V2.0 + Enhanced Data Rate</i> (EDR) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz.
5	LCD	Layar LCD digunakan untuk menampilkan informasi yang berguna kepada pengemudi atau pengguna kendaraan, seperti kecepatan, suhu, tingkat baterai, dan informasi lainnya yang diperoleh dari Modul CAN Bus. LCD ini terhubung ke Arduino Uno dan menerima data dari mikrokontroler untuk ditampilkan.
6	Aplikasi Android	Aplikasi untuk <i>Internet of Things</i> (IoT) adalah program perangkat lunak yang mengintegrasikan data dari berbagai perangkat IoT, seperti sensor, kamera, dan peralatan latihan, dengan tujuan menganalisis dan membuat keputusan yang matang berdasarkan data tersebut

Dikarenakan sistem ini tidak hanya menggunakan sistem perangkat keras tetapi juga menggunakan perangkat lunak maka dalam usulan perancangan ini, terdapat usulan sistem aplikasi yang digunakan. Aplikasi ini didesain untuk aplikasi Android dengan desain aplikasi dibuat

compatible dengan *smartphone* dengan spesifikasi yang rendah, dengan desain tampilan seperti Gambar 3.7.



Gambar 3. 6 Usulan rancangan aplikasi untuk pengguna

3.1.2 Rencana Anggaran Desain 1

Tabel 3. 2 Rencana anggaran pengembangan sistem

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	Arduino Uno	Pcs	Rp. 135.000, -	1	Rp. 135.000, -
2	Modul CAN Bus MCP2515	Pcs	Rp. 37.500, -	1	Rp. 37.500, -
3	Modul <i>bluetooth</i> HC-05	Pcs	Rp. 44.000, -	1	Rp. 44.000, -
4	LCD TFT	Pcs	Rp. 150.000, -	1	Rp. 150.000, -
5	Bahan Filament ABS dan PLA Printer 3D	Pcs	Rp. 172.000, -	1	Rp. 172.000, -
Total					Rp. 538.500, -

3.1.3 Analisis Risiko Desain

Sistem ini menggunakan berbagai komponen yaitu dari perangkat keras (Votol EM100, Arduino Uno, dan modul *bluetooth*) dan perangkat lunak (Aplikasi SPEETER). Usulan desain pertama ini memiliki kelemahan seperti modul *bluetooth* yang memiliki jarak terbatas untuk

komunikasi. Jika perangkat terlalu jauh dari *bluetooth*, sinyalnya bisa melemah, bahkan bisa terputus. Walaupun modul *bluetooth* memiliki kecepatan *transfer* data yang cukup untuk sebagian besar aplikasi tetapi jika dibandingkan dengan teknologi nirkabel lainnya seperti Wi-Fi, kecepatannya lebih rendah sehingga perangkat yang dapat terhubung secara bersamaan memungkinkan data yang didapatkan tidak akurat.

3.1.4 Pengukuran Performa

Untuk mengukur performa dari desain alat atau sistem, ada beberapa parameter dan metode yang dapat digunakan yaitu:

1. Keandalan: Performa dapat diukur dari sudut pandang keandalan. Hal ini bertujuan untuk mengetahui seberapa sering sistem mengalami kegagalan atau masalah teknis. Ini dapat diukur dengan menghitung waktu rata-rata antara kegagalan atau perbaikan yang diperlukan.
2. Akurasi: Performa dapat diukur menggunakan Aplikasi seperti "GPS Speedometer - Odometer For App" yang dapat membandingkan data kecepatan yang ditampilkan oleh speedometer kendaraan melalui *server* dengan data kecepatan yang diukur oleh aplikasi tersebut. Dengan demikian dapat membantu mengevaluasi tingkat akurasi speedometer kendaraan
3. Antarmuka Pengguna: Pengalaman pengguna dan respons antarmuka pengguna seperti layar LCD dan aplikasi SPEETER dapat diukur melalui pengukuran waktu respon, kejelasan informasi yang ditampilkan, dan tingkat kepuasan pengguna

3.1.5 Aspek Keteknikan dan K3

Terdapat beberapa aspek Keteknikan dan Keselamatan Kerja (K3) yang perlu diperhatikan. Berikut adalah beberapa aspek yang mungkin perlu Anda pertimbangkan:

Aspek Keteknikan :

- Presisi dan Akurasi: Pastikan bahwa perangkat memberikan pengukuran yang akurat dan presisi terkait kecepatan (speedometer) dan jarak tempuh (odometer).
- Stabilitas Koneksi *bluetooth*: Pastikan koneksi *bluetooth* stabil untuk memastikan data yang dikirimkan dari perangkat ke aplikasi atau perangkat penerima.
- Desain Antarmuka Pengguna: Rancang antarmuka pengguna yang dapat dengan mudah dipahami dan digunakan oleh pengguna motor listrik

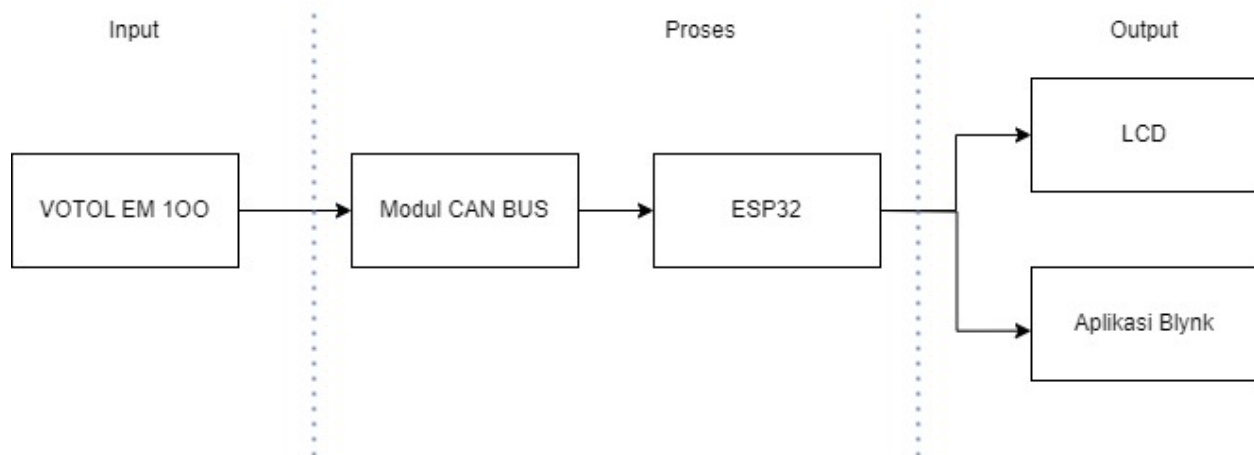
Aspek Keteknikan dan Keselamatan Kerja (K3) :

- Pemantauan Keselamatan Pengemudi: Pastikan bahwa pengemudi dapat dengan mudah membaca informasi dari perangkat tanpa mengalihkan perhatian mereka dari jalan.
- Peringatan Keselamatan: Pertimbangkan untuk menambahkan peringatan keselamatan, misalnya, peringatan kecepatan maksimum atau peringatan jika pengemudi telah berkendara terlalu lama.
- Perancangan Fisik: Pastikan perangkat memiliki perancangan fisik yang aman dan dapat diintegrasikan dengan baik pada kendaraan tanpa mengganggu fungsi utama atau mengakibatkan potensi bahaya.

3.2 Usulan Solusi 2

Usulan solusi 1 adalah sistem *Connected Vehicle* yang diusulkan dengan mendeteksi nilai keluaran speedometer dan odometer yang terintegritas dengan sistem *Internet of Things* yaitu menggunakan jaringan Wi-Fi yang ditampilkan di layar LCD dan aplikasi Blynk.

3.2.1 Desain Sistem 2

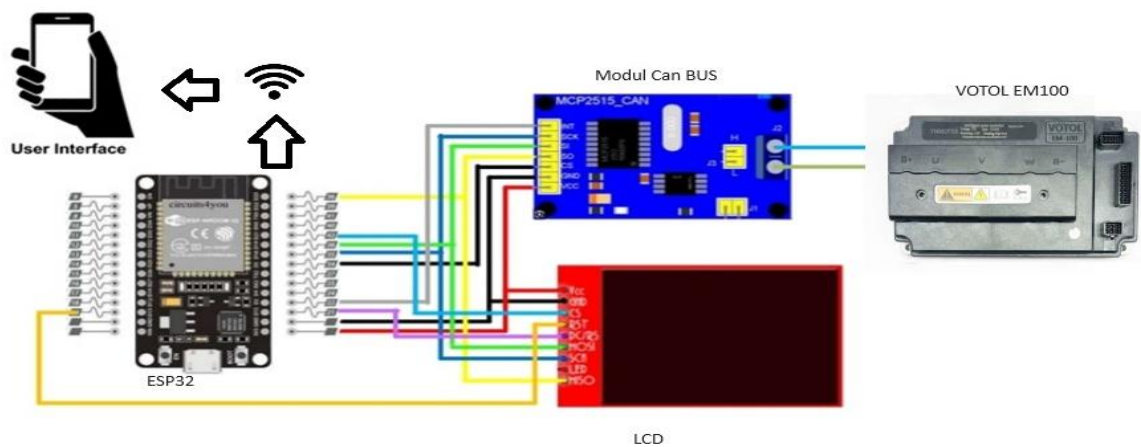


Gambar 3. 7 Diagram Blok Sistem

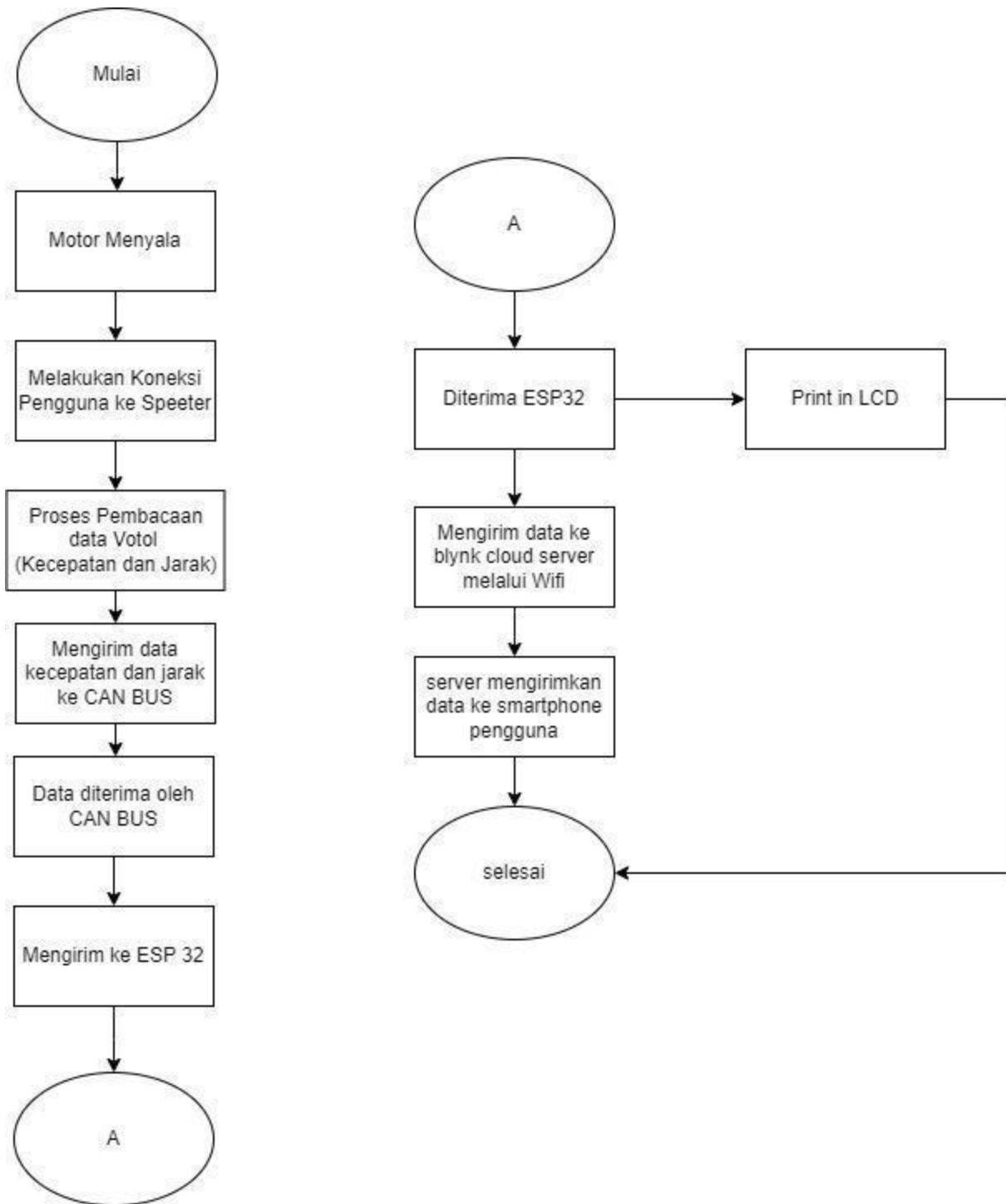
Cara kerja sistem ini melibatkan *controller* Votol EM100, modul CAN Bus, ESP32, LCD, dan Blynk. *Controller* Votol EM100 berfungsi sebagai otak utama pengendalian motor listrik yang dapat mengatur kecepatan dan jarak kendaraan. Modul CAN Bus menjadi perantara antara ESP32 dan sistem kendaraan. Hal ini memungkinkan ESP32 untuk berkomunikasi dengan CAN Bus.

ESP32 sebagai mikrokontroler untuk menerima informasi dari sensor-sensor pada speedometer dan odometer. Setelah menerima informasi, ESP32 akan memprosesnya sehingga pengguna dapat dengan mudah memonitor kecepatan dan jarak tempuh kendaraan secara *real-time* melalui tampilan yang diberikan oleh LCD.

Selanjutnya, aplikasi pada *smartphone* berfungsi untuk menampilkan informasi yang telah diproses oleh ESP32. Dengan hal ini, pengguna dapat dengan mudah menerima informasi dan pemantauan kondisi kendaraan secara lebih efektif. Berikut desain sistem elektronik dan diagram blok sistem yang ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Proses Cara Kerja Sistem



Gambar 3. 9 Diagram Alir Sistem

Cara kerja sistem ini akan berjalan ketika motor sudah menyala. Selanjutnya, pengguna melakukan koneksi dengan speedometer dan odometer (SPEETER) pada motor listrik untuk mengakses informasi tentang kecepatan dan jarak. Setelah melakukan koneksi, Votol mulai membaca data berupa kecepatan dan jarak tempuh. Data ini kemudian dikirim ke *Controller Area*

Network (CAN Bus), yang mana CAN Bus berperan sebagai media komunikasi untuk mengirim data kecepatan dan jarak tempuh. Setelah data diterima oleh CAN Bus, data tersebut akan dilanjutkan ke ESP32 untuk dilakukan pemrosesan data. ESP32 kemudian mengirim data ke *server* Blynk melalui koneksi Wi-Fi yang tersedia. *Server* Blynk mengirimkan data yang telah diproses dan diolah ke *smartphone* pengguna melalui aplikasi Blynk

Modul Wi-Fi memiliki kelebihan, terutama dalam hal kapasitas *bandwith* yang tinggi dan dapat memberikan kemampuan untuk *mentransfer* data dengan kecepatan yang signifikan. Selain itu, jangkauan sinyal Wi-Fi yang panjang menjadi salah satu ciri khasnya memungkinkan perangkat untuk tetap terhubung dalam area yang luas. Dengan kemampuan operasi pada frekuensi 2,4 GHz hingga 5 GHz, modul Wi-Fi memungkinkan adaptabilitas dan ketersediaan kanal yang lebih baik untuk menghindari interferensi. Faktor keamanan juga menjadi fokus utama dengan berbagai protokol keamanan Wi-Fi yang tersedia, seperti WPA2 dan WPA3 yang dapat memberikan perlindungan yang tinggi terhadap akses tanpa izin dan pengamanan data selama transmisi. Dengan keunggulan ini, modul Wi-Fi menjadi pilihan yang sangat efektif untuk menghadirkan konektivitas nirkabel yang handal dan aman dalam berbagai aplikasi [4].

Desain Gambar 3.10 dan 3.11 ini mencakup integrasi teknologi terkini untuk mengukur dan menyajikan informasi kecepatan serta jarak tempuh dengan akurasi tinggi. Selain itu, antarmuka pengguna dirancang untuk memberikan informasi bagi pengendara. Desain ini tidak hanya memastikan kinerja yang optimal dalam mengukur dan melaporkan data kendaraan, tetapi juga mempertimbangkan aspek estetika dan kenyamanan pengguna. Dengan demikian, desain ini dianggap sebagai solusi terbaik dan siap untuk diimplementasikan dalam produksi speedometer dan odometer yang memenuhi standar tinggi serta harapan pengguna.



Gambar 3. 10 Desain Sistem 2 Tampak Samping



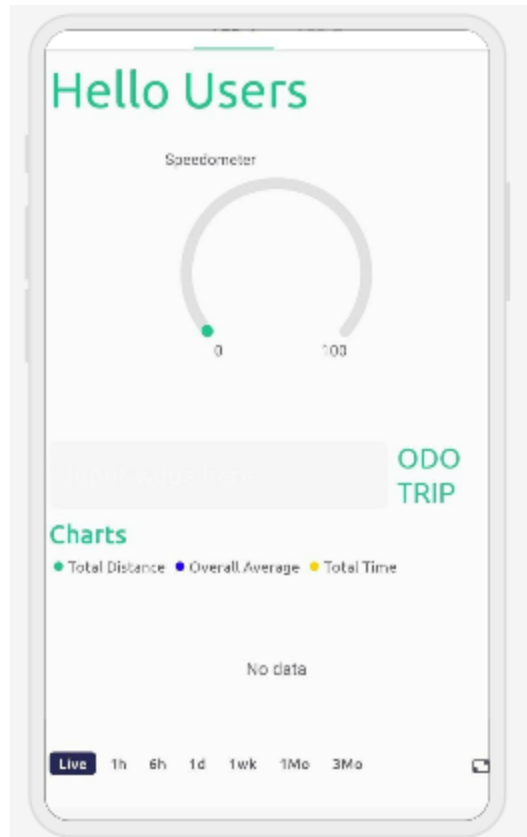
Gambar 3. 11 Desain Sistem 2 Tampak Samping

Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.3 menunjukkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3. 3 Inventarisasi kebutuhan usulan sistem perangkat keras SPEETER

No	Nama Alat	Keterangan
1	Votol EM 100	Merupakan kontroler motor listrik yang digunakan untuk mengendalikan motor listrik di kendaraan elektrik. EM100 menerima sinyal dari ESP32 untuk mengatur kecepatan dan arah motor.
2	Modul CAN Bus	Modul CAN Bus digunakan untuk berkomunikasi dengan berbagai perangkat dalam kendaraan, seperti sensor suhu, sensor kecepatan, sensor baterai, dan lainnya. Modul ini menerima dan mengirim data melalui protokol CAN (Controller Area Network) untuk mengumpulkan informasi tentang kondisi kendaraan.
3	ESP32	Mikrokontroler ESP32 digunakan untuk mengintegrasikan berbagai komponen dalam sistem. ESP32 berperan sebagai otak sistem yang mengambil data dari Modul CAN Bus dan mengirimnya ke aplikasi Blynk, serta mengendalikan Votol EM100 untuk mengatur motor.
4	LCD	Layar LCD digunakan untuk menampilkan informasi yang berguna kepada pengemudi atau pengguna kendaraan, seperti kecepatan, suhu, tingkat baterai, dan informasi lainnya yang diperoleh dari Modul CAN Bus. LCD ini terhubung ke ESP32 dan menerima data dari mikrokontroler untuk ditampilkan.
5	Blynk	Blynk adalah platform <i>Internet of Things</i> (IoT) yang memungkinkan Anda mengontrol dan memonitor perangkat melalui ponsel pintar atau perangkat lainnya. Aplikasi Blynk digunakan untuk berinteraksi dengan kendaraan elektrik, seperti mengontrol motor, melihat data kendaraan, atau mengatur parameter kendaraan melalui jaringan Wi-Fi atau seluler. ESP32 terhubung ke Blynk untuk berkomunikasi dengan perangkat eksternal.

Dikarenakan sistem ini tidak hanya menggunakan sistem perangkat keras tetapi juga menggunakan perangkat lunak maka dalam usulan perancangan ini, terdapat usulan sistem aplikasi yang digunakan. Aplikasi ini didesain untuk aplikasi Android dengan desain aplikasi dibuat *compatible* dengan *smartphone* dengan spesifikasi yang rendah, dengan desain tampilan seperti Gambar 3.12.



Gambar 3. 12 Usulan rancangan aplikasi untuk pengguna

3.2.2 Rencana Anggaran Desain Sistem 2

Rancangan anggaran pada desain 2 dapat ditunjukkan pada sub bab ini. Berikut rencana anggaran pada Tabel 3.4.

Tabel 3. 4 Rencana anggaran pengembangan sistem SPEETER

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	Modul CAN Bus MCP2515	Pcs	Rp. 37.500, -	1	Rp. 37.500, -
2	ESP-32	Pcs	Rp. 70.000, -	1	Rp. 70.000, -
3	LCD TFT	Pcs	Rp. 150.000, -	1	Rp. 150.000, -
4	Blynk	Bulan	Rp. 77.000, -	1	Rp. 77.000, -
5	Bahan Filament ABS dan PLA Printer 3D	Pcs	Rp. 172.000, -	1	Rp. 172.000, -
Total					Rp. 506.500, -

3.2.3 Analisis Risiko Desain 2

Sistem ini menggunakan berbagai komponen yaitu dari perangkat keras (Votol EM100, dan ESP32) dan perangkat lunak (aplikasi blynk). Usulan desain ini memiliki kelemahan seperti Modul Wi-Fi dan ESP32 yang memiliki konsumsi daya yang lebih tinggi dibandingkan dengan beberapa teknologi nirkabel rendah daya seperti *Bluetooth Low Energy* (BLE). Hal ini dapat menjadi kendala pada perangkat bertenaga baterai. Proses penyusunan jaringan dan pemberian alamat IP serta inisialisasi koneksi Wi-Fi harus diperhatikan agar Wi-Fi yang terhubung aman dan memiliki konektivitas yang bagus.

3.2.4 Pengukuran Performa

Untuk mengukur performa dari desain alat atau sistem, ada beberapa parameter dan metode yang dapat digunakan yaitu :

1. Keandalan: Performa dapat diukur dari sudut pandang keandalan. Hal ini bertujuan untuk mengetahui seberapa sering sistem mengalami kegagalan atau masalah teknis. Ini dapat diukur dengan menghitung waktu rata-rata antara kegagalan atau perbaikan yang diperlukan.
2. Akurasi: Performa dapat diukur menggunakan Aplikasi seperti "GPS Speedometer-Odometer For App" yang dapat membandingkan data kecepatan yang ditampilkan oleh speedometer kendaraan melalui server dengan data kecepatan yang diukur oleh aplikasi tersebut. Dengan demikian dapat membantu mengevaluasi tingkat akurasi speedometer kendaraan
3. Antarmuka Pengguna: Pengalaman pengguna dan respons antarmuka pengguna seperti layar LCD dan aplikasi Blynk dapat diukur melalui pengukuran waktu respon, kejelasan informasi yang ditampilkan, dan tingkat kepuasan pengguna.

3.2.5 Aspek Ketenikan dan K3

Terdapat beberapa aspek Keteknikan dan Keselamatan Kerja (K3) yang perlu diperhatikan. Berikut adalah beberapa aspek yang mungkin perlu Anda pertimbangkan:

Aspek Keteknikan :

- Presisi dan Akurasi: Pastikan bahwa perangkat memberikan pengukuran yang akurat dan presisi terkait kecepatan (speedometer) dan jarak tempuh (odometer).

- Stabilitas Koneksi Wi-Fi: Pastikan koneksi Wi-Fi stabil untuk memastikan data yang dikirimkan dari perangkat ke aplikasi atau perangkat penerima.
- Desain Antarmuka Pengguna: Rancang antarmuka pengguna yang dapat dengan mudah dipahami dan digunakan oleh pengguna motor listrik

Aspek Keteknikan dan Keselamatan Kerja (K3) :

- Pemantauan Keselamatan Pengemudi: Pastikan bahwa pengemudi dapat dengan mudah membaca informasi dari perangkat tanpa mengalihkan perhatian mereka dari jalan.
- Peringatan Keselamatan: Pertimbangkan untuk menambahkan peringatan keselamatan, misalnya, peringatan kecepatan maksimum atau peringatan jika pengemudi telah berkendara terlalu lama.
- Perancangan Fisik: Pastikan perangkat memiliki perancangan fisik yang aman dan dapat diintegrasikan dengan baik pada kendaraan tanpa mengganggu fungsi utama atau mengakibatkan potensi bahaya.

3.3 Analisis dan Penentuan Usulan Solusi/Desain Terbaik

Pemilihan usulan solusi desain terbaik dilakukan dengan menggunakan decision matrix analysis. Analisis ini dipilih karena dapat membandingkan solusi secara visual dengan mempertimbangkan faktor yang mempengaruhi. Tabel 3.5 menunjukkan perbedaan terhadap solusi 1 dan solusi 2. Berdasarkan Tabel 3.5 tim memilih usulan solusi desain 1 sebagai desain terbaik.

Tabel 3. 5 Perbandingan 2 Usulan Solusi

Faktor Yang Mempengaruhi	Usulan 1	Usulan 2
<i>Bandwidth</i>	Rendah	Tinggi
Jangkauan Sinyal	10 Meter	100 Meter
Frekuensi	2,4 GHz sampai 2,48GHz	2,4 GHz sampai 5GHz
Kebutuhan perangkat keras	Adaptor <i>bluetooth</i> pada semua perangkat yang kami sambungkan satu sama lain.	Adaptor nirkabel di semua perangkat dalam jaringan dan juga router nirkabel..

Faktor Yang Mempengaruhi	Usulan 1	Usulan 2
Kemudahan penggunaan	Cukup mudah digunakan dan menghubungkan perangkat secara nirkabel juga lebih mudah.	Lebih kompleks dan juga memerlukan konfigurasi perangkat keras dan perangkat lunak nirkabel.
Konsumsi energi	Rendah	Tinggi
Biaya Produksi	Rp. 538.500, -	Rp. 506.500, --
Penggunaan <i>Controller</i>	Votol EM100	Votol EM100
Protokol Komunikasi	CAN Bus	CAN Bus
Penggunaan <i>Microcontroller</i>	Arduino Uno dan HC-05	ESP-32
Sifat Alat	Fixed	Fixed
Bahan Dasar (Frame)	<i>Acrylonitrile Butadiene Styrene (ABS)</i>	<i>Acrylonitrile Butadiene Styrene (ABS)</i>
Dimensi Alat	3,6 inch	3,6 inch
Fitur Tambahan	<ul style="list-style-type: none"> ● Suhu ● Sein ● Indikator Lampu ● Baterai ● Waktu 	<ul style="list-style-type: none"> ● Suhu ● Sein ● Indikator Lampu ● Baterai ● Waktu
Indikator Keluaran Alat	Aplikasi Android	Aplikasi Blynk

3.4 Skenario Pengujian

Berikut ini adalah tabel pengujian sistem menggunakan metode *Black Box* dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Skenario Pengujian

No	Skenario Pengujian	Detail Pengujian
1	Persiapan awal	Motor menyala, koneksi <i>bluetooth</i> /WiFi ke <i>smartphone</i>
2	Pembacaan data	Data (kecepatan dan jarak tempuh) akan dibaca oleh <i>Votol</i> .

No	Skenario Pengujian	Detail Pengujian
3	Transfer data	Data (kecepatan dan jarak tempuh) dari Votol ke Arduino Uno lalu ke LCD dan Android.
4	Pemantauan <i>real time</i>	Spedometer dan odometer serta <i>smartphone</i> .
5	Login pengguna	Login user
6	Halaman utama	Menampilkan Fitur pemantauan kecepatan dan jarak tempuh
7	Halaman riwayat	Riwayat perjalanan pengguna

3.5 Gantt Chart

Diperlukan suatu perencanaan dan manajemen waktu untuk mempermudah pengerjaan sistem agar distribusi tugas dan target pencapaian dapat dipenuhi. Perencanaan meliputi kegiatan yang dilakukan beserta penanggung jawab (PIC : *Person In Charge*) setiap kegiatan pada setiap bulan menggunakan Gantt chart. Tabel 3.7 menunjukkan rincian kegiatan yang dimulai dari bulan September 2023 hingga bulan Juni 2024.

Tabel 3. 7 Gantt Chart Pelaksanaan Capstone Project Sistem

No.	Kegiatan/Capaian	Bulan ke -									
		2023				2024					
		Sept	Okt	Nov	Des	Jan	Feb	Mar	Apr	Mei	Juni
1	Identifikasi masalah	D,I									
2	Melakukan literatur dan mengumpulkan informasi untuk kebutuhan spesifikasi sistem	D,I	D,I	D,I							
3	Mengumpulkan ide solusi sistem sistem serta finalisasi usulan solusi			D,I	D,I						
4	Pengumpulan proposal tugas akhir dan seminar proposal				D,I						

No.	Kegiatan/Capaian	Bulan ke -									
		2023				2024					
		Sept	Okt	Nov	Des	Jan	Feb	Mar	Apr	Mei	Juni
5	Pembelian serta pembuatan alat dan bahan					D,I	D,I	D,I			
6	Perancangan dan pembangunan sistem					D,I	D,I	D,I	D,I		
7	Pengujian dan validasi									D,I	
8	Expo dan pengumpulan laporan akhir										D,I

Ket. : PIC – *Person in Charge* (Pihak yang bertanggung untuk kegiatan tersebut) D: Daffis Hesa Pratama, I : Ichiro Rizky Abdul Aziz

3.6 Realisasi Pelaksanaan Tugas Akhir 1

Pada proses penyusunan proposal TA 01 terdapat beberapa kendala yang dialami sehingga penyusunan proposal membutuhkan waktu yang lama dan berdasarkan *timeline* pada *Gantt chart* beserta peran dari masing-masing anggota seperti pada Tabel 3.7. Untuk kendala-kendala tersebut yaitu:

1. Kesulitan dalam menemukan jurnal terkait topik yang dibahas.
2. Kurangnya memahami sistem *controller* Votol.
3. Pada bulan November, salah satu anggota kelompok melaksanakan kerja praktik sehingga komunikasi secara langsung sedikit terkendala.

Tabel 3. 8 Realisasi aktivitas pelaksanaan tugas akhir 1

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Jumat, 15 September 2023, 3 jam	Pertemuan Pertama dengan dosen pembimbing	Daffis, Ichiro

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
2	Minggu, 17 September 2023, 3 jam	Mencari Referensi dan membuat pertanyaan untuk wawancara	Daffis, Ichiro
3	Selasa, 19 September 2023, 2 Jam	Wawancara Narasumber PT Fusi Global Teknologi.	Daffis, Ichiro
4	Selasa, 19 September 2023, 3 Jam	Mengerjakan Latar Bab 1 (Latar Belakang dan Rumusan Masalah)	Daffis, Ichiro
5	Selasa, 26 September 2023, 2 Jam	Bimbingan (Latar Belakang dan Rumusan Masalah)	Daffis, Ichiro
6	Selasa, 26 September 2023, 2 Jam	Melanjutkan Mengerjakan Proposal Bab 1 (Batasan masalah dan Batasan realistik)	Daffis, Ichiro
7	Senin, 2 Oktober 2023, 3 Jam	Melanjutkan Mengerjakan Proposal Bab 1 (Batasan masalah dan Batasan realistik)	Daffis, Ichiro
8	Selasa, 3 Oktober 2023, 3 Jam	Bimbingan (Batasan masalah dan Batasan realistik)	Daffis, Ichiro
9	Jumat, 6 Oktober 2023, 4 Jam	Revisi (Batasan masalah dan Batasan realistik)	Daffis, Ichiro
10	Minggu, 8 Oktober 2023, 4 Jam	Melanjutkan mengerjakan proposal Bab 2 (Studi literatur)	Daffis, Ichiro
11	Selasa, 10 Oktober 2023, 3 Jam	Bimbingan (Rumusan Masalah dan Batasan Realistik)	Daffis, Ichiro

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
12	Kamis, 12 Oktober 2023, 3 Jam	Melanjutkan mengerjakan proposal bab 2 (Dasar Teori)	Daffis, Ichiro
13	Sabtu, 14 Oktober 2023, 3 Jam	Melanjutkan mengerjakan proposal bab 2 (Dasar Teori dan analisis)	Daffis, Ichiro
14	Kamis, 19 Oktober 2023, 3 Jam	Bimbingan (Landasan Teori)	Daffis, Ichiro
15	Minggu, 22 Oktober 2023, 3 Jam	Revisi dan melanjutkan Bab 2 (spesifikasi sistem)	Daffis, Ichiro
16	Sabtu, 28 Oktober 2023, 4 Jam	Revisi dan merapihkan Bab 2	Daffis, Ichiro
17	Sabtu, 4 November 2023, 3 Jam	Memulai mengerjakan Bab 3	Daffis, Ichiro
18	Senin, 6 November 2023, 3 Jam	Bimbingan (Penjelasan dan Revisi dokomen TA 102)	Ichiro
19	Selasa, 7 November 2023, 3 Jam	Revisi Bab 2 dan melanjutkan Bab 3	Daffis, Ichiro
20	Minggu, 12 November 2023, 3 Jam	Merapihkan Bab 2 dan 3	Daffis, Ichiro
21	Selasa, 14 November 2023, 3 Jam	Pembahasan TA 103	Ichiro
22	Jumat, 17 November 2023, 2 Jam	Bimbingan Dosen Pembimbing 2	Ichiro

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
23	Sabtu, 18 November 2023, 4 Jam	Melanjutkan Bab 3 (Desain thinking)	Daffis, Ichiro
24	Minggu, 19 November 2023, 5 Jam	Mengerjakan desain sistem dan 3D	Daffis, Ichiro
25	Rabu, 22 November 2023, 3 Jam	Bimbingan (Pembahasan TA 103)	Ichiro
26	Jumat, 24 November 2023, 3 Jam	Mengerjakan Desain Aplikasi	Daffis, Ichiro
28	Selasa, 28 November 2023, 3 Jam	Bimbingan (Pembahasan usulan solusi)	Daffis, Ichiro
29	Kamis, 30 November 2023, 4 Jam	Revisi Bab 3	Daffis, Ichiro
30	Sabtu, 2 Desember 2023, 2 Jam	Revisi Flow Chart	Daffis, Ichiro
31	Selasa, 5 Desember 2023, 3 Jam	Bimbingan (Pembahasan Usulan Solusi Terbaik)	Daffis, Ichiro
32	Kamis, 7 Desember 2023, 3 Jam	Memberikan sitasi dan daftar pustaka	Daffis, Ichiro
33	Sabtu, 9 Desember 2023, 3 Jam	Revisi Flowchart	Daffis, Ichiro
34	Selasa, 12 Desember 2023, 3 Jam	Bimbingan (Revisi usulan solusi)	Daffis, Ichiro

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
34	Jumat, 15 Desember 2023, 3 Jam	Membuat loogbok dan kendala	Daffis, Ichiro
35	Senin, 25 Desember 2023, 3 Jam	Bimbingan (Rivisi Usulan Solusi terbaik)	Daffis, Ichiro
26	Kamis, 25 Desember 2023, 1 Jam	Ujian Proposal	Daffis, Ichiro

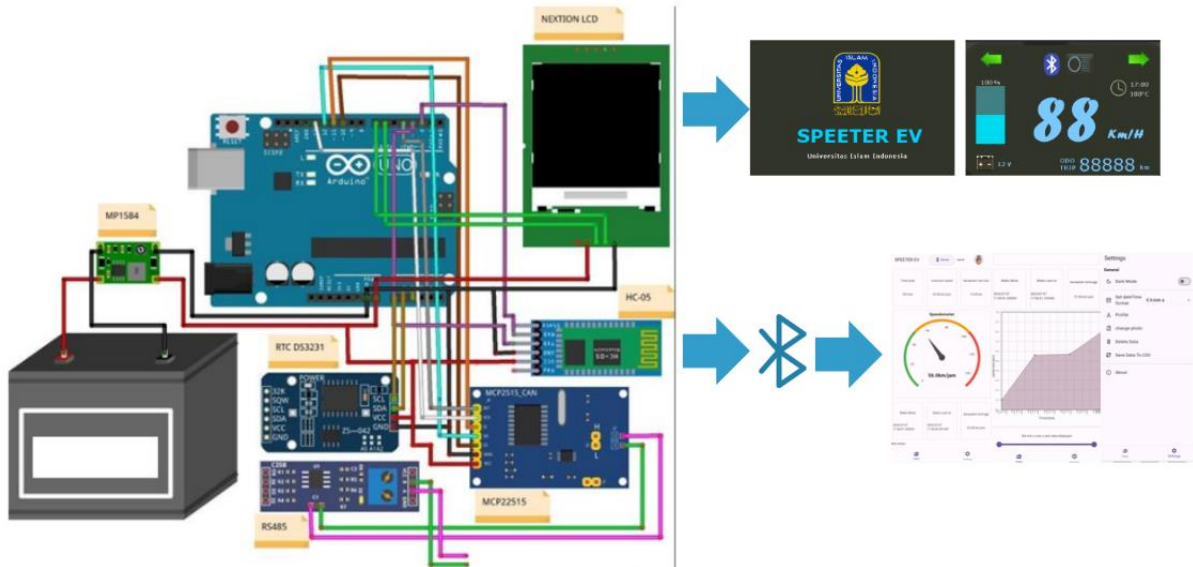
BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN

4.1 Hasil Rancangan Sistem

Implementasi desain sistem dilakukan melalui empat tahap untuk mencapai hasil yang optimal. Tahap pertama adalah merancang rangkaian listrik, yang merupakan langkah awal dan krusial, di mana komponen-komponen elektronik dipilih dengan cermat dan dihubungkan sesuai kebutuhan spesifik sistem agar dapat berfungsi dengan baik. Tahap kedua adalah pembuatan desain 3D, di mana bentuk fisik dari sistem serta tata letak setiap komponen dalam ruang tiga dimensi dirancang menggunakan perangkat lunak desain khusus untuk memastikan kesesuaian dan efisiensi ruang serta estetika produk akhir. Tahap ketiga adalah pengembangan perangkat lunak atau antarmuka pengguna, yang melibatkan pembuatan dan pengujian program-program yang diperlukan dalam mengoperasikan sistem. selain itu, memastikan bahwa perangkat lunak ramah dan mudah diakses oleh pengguna. Tahap keempat adalah dokumentasi dan presentasi hasil akhir perancangan, di mana foto-foto dari produk jadi diambil dan disertakan bersama penjelasan rinci tentang proses perancangan dan hasil akhir, memberikan gambaran lengkap mengenai produk yang telah dibuat serta menampilkan kualitas dan keunggulan desain yang telah dihasilkan melalui rangkaian tahapan yang komprehensif ini.

4.1.1 Rangkaian Elektronik

Pada tahap pertama dilakukan perancangan alat dengan komponen-komponen seperti LCD Nextion, MCP2515, Arduino Uno, *Bluetooth* HC-05, dan RTC DS3231 yang dimana setiap komponen akan diproses secara fungsionalitasnya. Perancangan rangkaian elektronik dapat dilihat di Gambar 4.1



Gambar 4. 1 Rangkaian Elektronik SPEETER

Gambar 4.1 merupakan rangkaian *receiver* yang bertugas membaca data dari *controller* motor listrik, termasuk pembacaan kecepatan dan jarak untuk speedometer motor listrik. Implementasi sistem ini menggabungkan berbagai komponen yang saling terintegrasi untuk menciptakan sebuah sistem kontrol yang komprehensif. Sistem dikendalikan oleh mikrokontroler yaitu Arduino Uno yang berperan menjalankan program dan mengatur operasi seluruh perangkat yang terhubung. Arduino Uno menerima daya dari sumber utama, yaitu Baterai *ion litium* (biasa disebut Baterai Li-ion atau LIB) berkapasitas 12V-84V-48Ah dan tegangannya diturunkan menjadi 5V menggunakan modul *buck converter* MP1584. Penurunan tegangan dilakukan untuk menyediakan daya yang sesuai bagi Arduino Uno dan komponen lainnya. Untuk menjaga waktu yang akurat, sistem ini menggunakan modul RTC DS3231 yang terhubung ke Arduino Uno. Komunikasi serial jarak jauh diimplementasikan melalui modul RS485. Sementara jaringan area pengendalian (CAN) diatur dengan bantuan MCP2515 yang digunakan untuk menerima data dari controller berbasis CAN Bus.

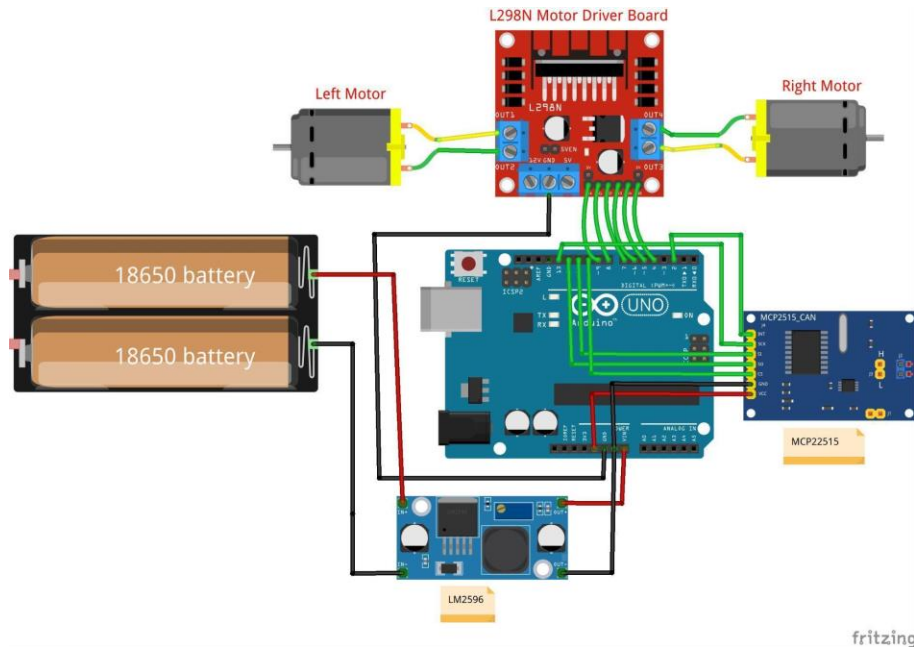
Komunikasi serial jarak jauh diimplementasikan melalui modul HC-05 yang mendukung komunikasi nirkabel antara perangkat mikrokontroler dan *smartphone* sehingga memungkinkan pengendalian atau pemantauan sistem dari jarak jauh dengan perangkat lain yang kompatibel. Antarmuka pengguna dibangun menggunakan layar sentuh yaitu LCD Nextion dan perangkat lunak berupa aplikasi buatan pada *smartphone* yang tidak hanya menampilkan informasi sistem dengan jelas tetapi juga memfasilitasi interaksi langsung dengan pengguna.

Setiap komponen dalam sistem ini terhubung dengan pin yang ditentukan pada Arduino Uno, memastikan komunikasi dan kontrol yang efisien sesuai dengan desain sistem yang telah dirancang. Dengan integrasi yang baik antara *hardware* dan *software*, sistem ini dapat memberikan kinerja yang optimal sesuai dengan kebutuhan aplikasi yang dituju, baik dalam lingkungan otomotif, industri, atau aplikasi lain yang memerlukan pengawasan dan kontrol yang cermat.

Pengalaman pengguna dari sistem ini tidak dapat diimplementasikan dikarenakan sistem ini tidak dapat terpasang pada motor listrik yang tersedia di Laboratorium Sistem Tenaga Listrik FTI UII. Terdapat tiga motor listrik, dua motor listrik menggunakan *controller* yang tidak mendukung sistem komunikasi CAN Bus dan terdapat satu motor yang mempunyai sistem komunikasi CAN Bus tetapi mengalami kerusakan terhadap IC MCP CAN Bus. Oleh karena itu, merubah rangkaian elektronis yang awalnya diterapkan pada motor listrik menjadi dua sistem yaitu *transmitter* dan *receiver* sebagai berikut.

A. *Transmitter*

Transmitter adalah perangkat atau alat yang digunakan untuk mengirimkan sinyal atau informasi dari satu tempat ke tempat lain. *Transmitter* mengubah informasi (seperti suara, video, atau data) menjadi bentuk sinyal yang dapat dikirimkan melalui berbagai media, seperti udara (gelombang radio), kabel, atau serat optik. Pada Tugas Akhir ini dibuat sistem *transmitter* sebagai simulator motor listrik untuk pengiriman data kecepatan dan jarak dengan menggunakan Arduino Uno, motor DC, L298N, LM2596 dan MCP2515 seperti rangkaian Gambar 4.2



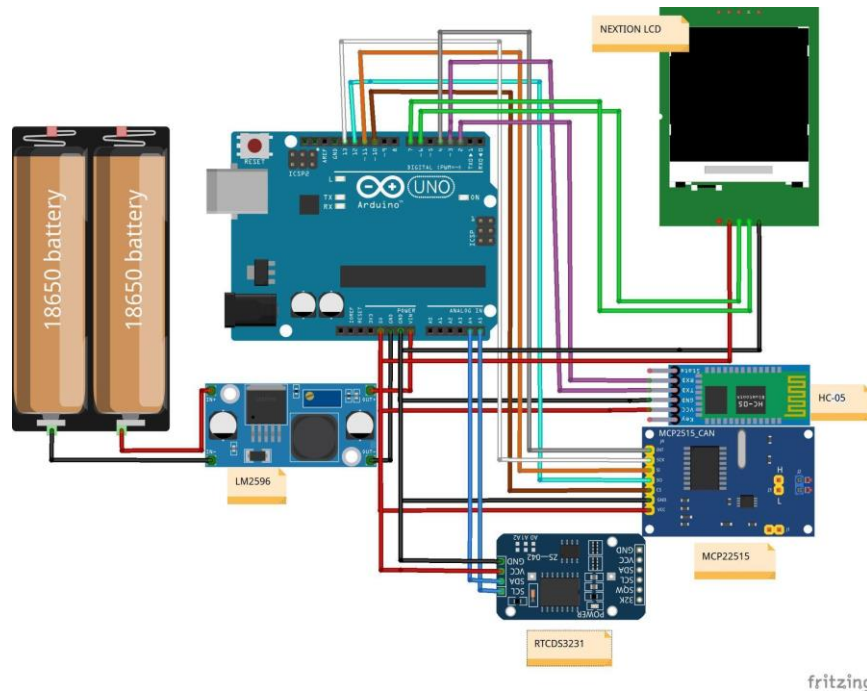
Gambar 4. 2 Rangkaian Elektronik Transmitter

Gambar 4.2 merupakan rangkaian transmitter. *Transmitter* adalah perangkat yang mengirimkan data dari satu tempat ke tempat lain. Arduino Uno sebagai pusat pengendali yang mengumpulkan data dari berbagai sensor dan modul, kemudian mengirimkan data tersebut melalui modul MCP2515 ke *receiver*. Modul RTC DS3231 menyediakan waktu yang sangat akurat untuk mencatat waktu dan tanggal saat data kecepatan dan jarak diukur dan dikirimkan. Modul L298N adalah motor driver yang memungkinkan Arduino Uno mengontrol dua motor DC yang dapat digunakan untuk mengontrol kecepatan dan arah motor. Modul *step down* LM2596 digunakan untuk menurunkan tegangan dari sumber daya utama ke tegangan yang diperlukan oleh Arduino Uno dan komponen lainnya. Potensiometer digunakan untuk mengatur kecepatan roda motor melalui input analog ke Arduino Uno. Sensor kecepatan (RPM) adalah perangkat yang digunakan untuk mengukur kecepatan putaran roda, dalam satuan putaran per menit (RPM). Nilai RPM yang dihasilkan dari pembacaan sensor kecepatan dan pengaturan roda oleh potensiometer akan dikonversi menjadi nilai kecepatan dan jarak tempuh. Data tersebut akan dipersiapkan oleh Arduino Uno lalu akan dikirimkan melalui MCP2515 ke *receiver*.

B. Receiver

Receiver adalah perangkat atau alat yang digunakan untuk menerima sinyal atau informasi yang dikirimkan oleh transmitter. *Receiver* menangkap sinyal yang diterima, mengubahnya

kembali ke bentuk informasi asli, dan kemudian memproses atau menampilkan informasi tersebut sesuai kebutuhan. Setelah *transmitter* mengirimkan data kecepatan dan jarak akan diterima oleh *receiver* untuk menerima data kecepatan dan jarak. Gambar 4.3 merupakan sistem elektronik yang terdapat Arduino Uno, LCD, HC-05, MCP2515, RTCDS3231 dan LM289.

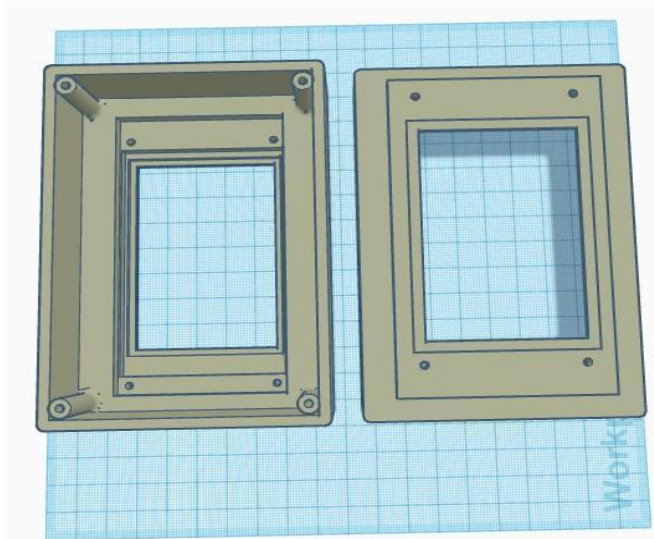


Gambar 4. 3 Rangkaian Elektronik Receiver

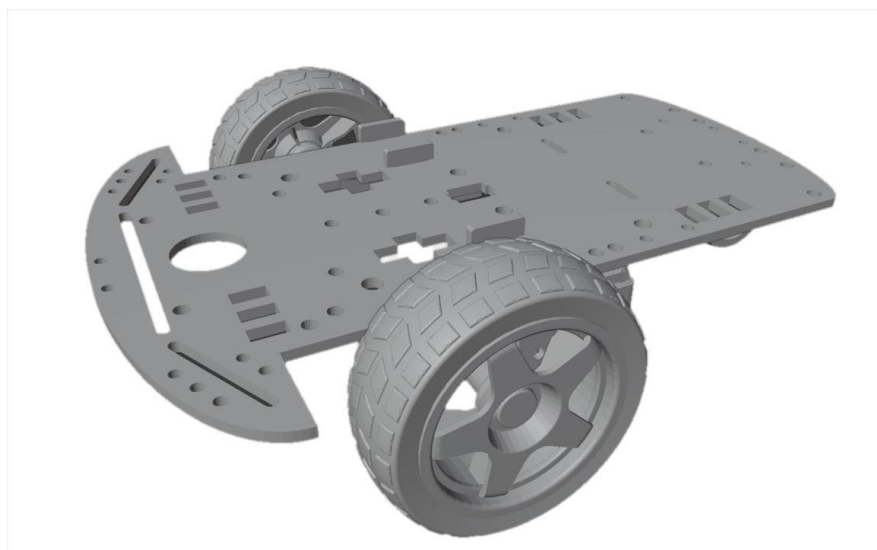
Gambar 4.3 merupakan rangkaian *Receiver* yang merupakan perangkat yang menerima data yang dikirimkan oleh transmitter. *Receiver* akan menerima data kecepatan dan jarak yang dikirimkan oleh *transmitter* melalui jaringan CAN. Arduino Uno sebagai penerima akan memproses data yang dikirimkan oleh *transmitter* melalui modul MCP2515. Data tersebut akan dikirimkan ke antarmuka pengguna yaitu *smartphone* dan LCD Nextion. Data yang dikirimkan ke *smartphone* menggunakan modul *bluetooth* HC-05 sebagai dan LCD Nextion menerapkan komunikasi serial. Modul *step down* LM2596 digunakan untuk menurunkan tegangan dari sumber daya utama ke tegangan yang diperlukan oleh Arduino Uno dan komponen lainnya. Modul RTC (*Real-Time Clock*) DS3231 digunakan untuk melacak waktu secara akurat.

C. Rancangan Akhir *Transmitter* dan *Receiver*

Skema pada Gambar 4.4 menunjukkan cara menggabungkan *receiver* dan *transmitter* untuk mengontrol motor melalui L298N dan CAN bus. *Transmitter* mengirimkan data nirkabel ke



Gambar 4. 6 Cover LCD SPEETER

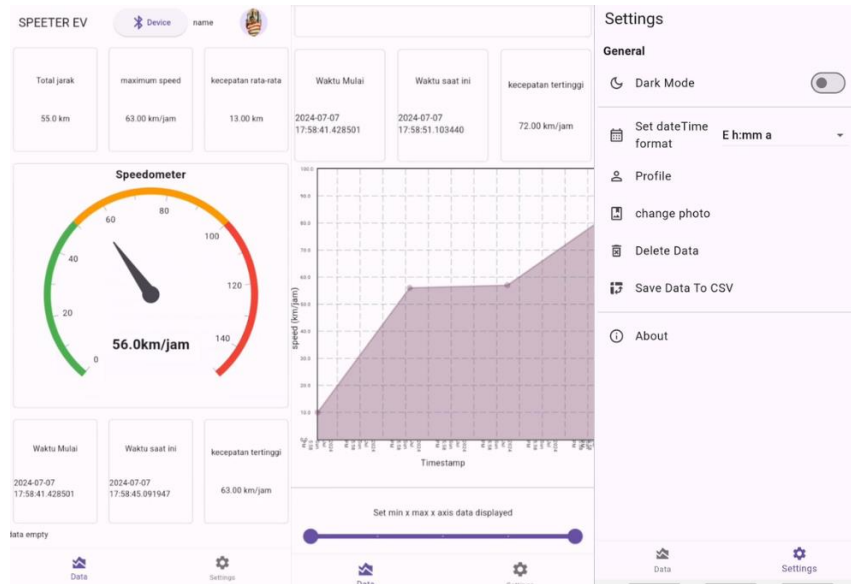


Gambar 4. 7 Rangkaian Car Kit

4.1.3 Antarmuka Pengguna

A. Aplikasi *Smartphone*

Pembuatan aplikasi *smartphone* untuk memonitor performa kendaraan listrik. Aplikasi ini dibuat menggunakan software MIT App Inventor, yang memungkinkan pengembangan aplikasi secara visual dengan blok pemrograman. Aplikasi ini menampilkan informasi penting seperti kecepatan, jarak tempuh, Gambar performa, kecepatan tertinggi, dan kecepatan rata-rata dalam antarmuka yang *user-friendly* seperti pada Gambar 4.8



Gambar 4. 8 Tampilan Aplikasi SPEETER

B. Instrument Display

Untuk membuat desain pada LCD Nextion menggunakan Nextion Editor yang merupakan perangkat lunak untuk merancang antarmuka pengguna (UI) pada LCD Nextion. Nextion adalah serangkaian layar sentuh HMI (*Human Machine Interface*) yang banyak digunakan dalam berbagai proyek elektronik karena kemudahannya dalam penggunaan dan kemampuannya untuk menampilkan antarmuka pengguna yang interaktif. Pada desain LCD Nextion, terdapat dua layer, pertama untuk display pembuka seperti Gambar 4.9. Kedua, merupakan desain utama yaitu untuk menampilkan kecepatan, jarak tempuh, jam, suhu motor, tegangan baterai, status baterai, indikator sein, dan indikator lampu serta *bluetooth* seperti pada Gambar 4.10.



Gambar 4. 9 Tampilan LCD Pertama



Gambar 4. 10 Tampilan LCD Utama

4.1.4 Pembuatan Kode Program

Tahap keempat dari implementasi desain adalah pembuatan kode program. Kode program dibuat berdasarkan flowchart kerja sistem sesuai Gambar Gambar menunjukkan logika berpikir pembuatan kode program yang dapat dibagi menjadi 2 subsistem sebagai berikut

A. Pembacaan Jarak dan Kecepatan

Kode program pembacaan jarak dan kecepatan dilakukan oleh Arduino Uno dengan memanfaatkan dua modul MCP2515 untuk membaca data dari sensor yang terhubung ke roda kendaraan. Data ini dikirimkan melalui protokol CAN (*Controller Area Network*) dan kemudian diproses untuk memperoleh informasi kecepatan dan jarak. Modul MCP2515 pertama berfungsi sebagai transmitter yang mengirimkan data dari sensor kecepatan dan jarak pada roda, sementara modul MCP2515 kedua berfungsi sebagai *receiver* yang menerima data tersebut. Data yang diterima kemudian diubah menjadi format yang dapat diproses oleh Arduino Uno dan ditampilkan pada layar speedometer sesuai dengan kebutuhan sistem. Pada program ini menggunakan program transmitter untuk pembacaan jarak dan kecepatan yang akan dikirimkan ke *receiver* untuk dapat diolah kembali.

```
void loop() {
    ...
    currentTime = millis();
    unsigned long elapsedTime = currentTime - previousTime;
    if (currentTime - previousPrintTime >= printInterval) {
        ...
        float jarakTempuhMM = (gerigiCount / (float)jumlahLubangGerigi) *
kelilingRodaMM;
        totalJarakTempuhMM += jarakTempuhMM;
        totalElapsedTime += elapsedTime;

        float jarakTempuhM = jarakTempuhMM / 1000.0;
        float kecepatanMS = jarakTempuhM / (elapsedTime / 1000.0);
        float kecepatanKMH = kecepatanMS * 3.6;
        ...
        if (kecepatanKMH > topSpeedKMH) { topSpeedKMH = kecepatanKMH; }
        ...
        gerigiCount = 0;
        previousTime = currentTime;
    }
}
```

```
previousPrintTime = currentTime;
}
}
```

B. Pengukuran Kecepatan

Pemrograman pengukuran kecepatan dilakukan dengan menggunakan fungsi `millis()` pada Arduino Uno. Fungsi ini akan dipanggil dua kali untuk mengukur selisih waktu. Selisih waktu ini, dikombinasikan dengan data jarak dari odometer, memungkinkan perhitungan kecepatan kendaraan berdasarkan putaran roda. Persamaan yang digunakan adalah sebagai berikut.

$$\text{Keliling Roda (mm)} = \pi \times \text{Diameter} \quad (4.1)$$

$$\text{Jarak Tempuh (mm)} = \text{Keliling Roda (mm)} \times \text{Jumlah Putaran} \quad (4.2)$$

$$\text{Jumlah Putaran} = \text{RPM} \times \left(\frac{\text{Waktu (detik)}}{60} \right) \quad (4.3)$$

$$\text{Kecepatan (mm/detik)} = \frac{\text{Jarak Tempuh (mm)}}{\text{Waktu (detik)}} \quad (4.4)$$

Pengukuran kecepatan dan jarak dilakukan dengan mendeteksi jumlah putaran roda menggunakan sensor yang terhubung. Setiap putaran roda akan menghasilkan data yang dikirim melalui modul MCP2515 dan diproses oleh Arduino Uno untuk menghitung kecepatan dan jarak yang telah ditempuh kendaraan. Hasil pengukuran ini kemudian ditampilkan secara *real-time* di layar speedometer, memungkinkan pengemudi untuk memantau kecepatan dan jarak tempuh dengan akurat.

4.1.5 Foto hasil akhir perancangan

Perancangan ini mencakup integrasi semua komponen dengan tepat, memastikan bahwa sistem bekerja secara optimal untuk perancangan Speedometer yang dapat mengukur kecepatan dan jarak dengan tingkat akurasi yang tinggi. Hasil akhir ini tidak hanya mencerminkan kemampuan teknis dari sistem, tetapi juga estetika dan keteraturan dalam penataan perangkat keras, menunjukkan tingkat profesionalisme dalam proses perancangan dan implementasi. Pada perancangan ini dibuat 2 sistem yang sistem pertama merupakan sistem perancangan pada motor listrik dan sistem yang kedua merupakan sistem pada pengukuran putaran roda yang terdapat di Car Kit sebagai berikut :

A. Hasil Akhir perancangan Motor Listrik

Pada perancangan ini, sistem dapat memonitor kecepatan dan jarak pada motor listrik melalui integrasi sistem *bluetooth* yang mampu mengirimkan data kecepatan dan jarak ke aplikasi Android. Sistem ini dapat diimplementasikan pada motor listrik yang memiliki sistem komunikasi CAN Bus. Gambar 4.11 merupakan hasil akhir rancangan pada motor listrik.

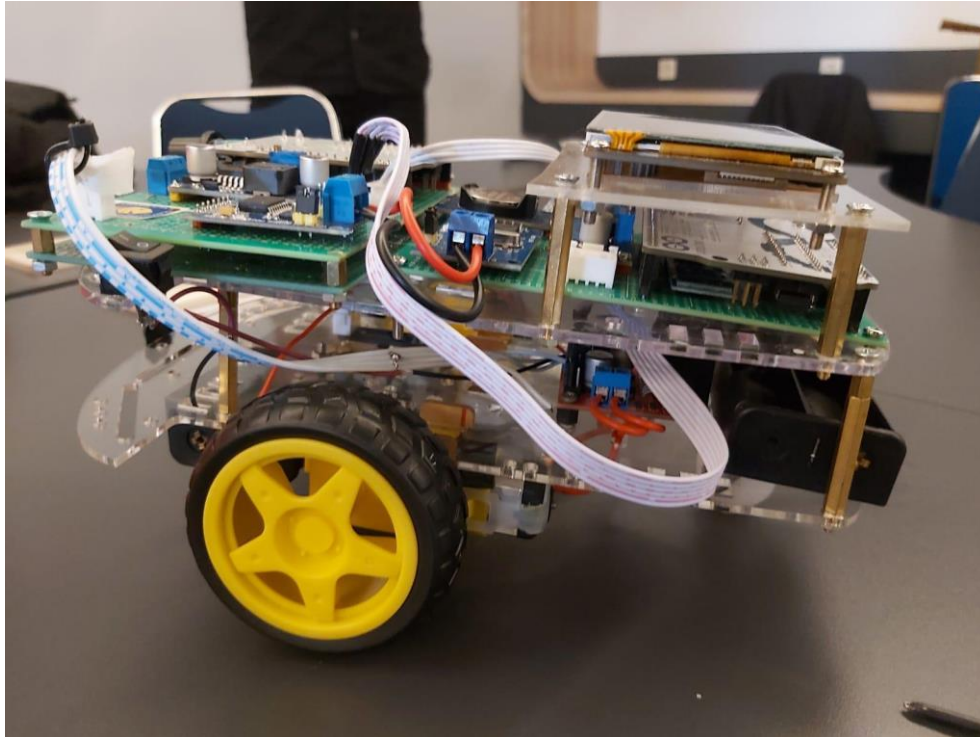


Gambar 4. 11 Hasil Akhir perancangan SPEETER

B. Hasil Akhir Perancangan Car Chassis

Pada perancangan ini, tim membuat sistem perancangan yaitu *transmitter* sebagai pengiriman data ke *receiver* dan *receiver* sebagai penerima data dari *transmitter*, dikarenakan motor listrik di Laboratorium Sistem Tenaga Listrik FTI UII mengalami kerusakan, sehingga solusinya adalah membuat perancangan di model *Car Kit* untuk dapat memonitoring kecepatan dan jarak seperti

4.12



Gambar 4. 12 Rangkaian Akhir *Car Kit*

4.2 Metode Pengukuran Kinerja Hasil Perancangan

Pengujian ini akan dilaksanakan secara eksperimen yang terbagi menjadi dua jenis yaitu pengujian sistem dan pengujian secara teoritis. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang telah dirancang dan untuk mengidentifikasi potensi kesalahan atau penyimpangan dalam pengukuran. Terdapat dua pengujian objek yaitu motor listrik dan prototipe dengan standar keteknikan yang sudah ada di regulasi nasional maupun internasional

A. Pengujian Motor Listrik

Pengujian pertama yaitu pada motor listrik akan dilakukan dengan memvariasikan kecepatan dan jarak yang akan ditempuh oleh motor listrik. Secara spesifik, pengujian akan dilakukan sebanyak 30 kali percobaan pada lima kecepatan yang berbeda, yaitu 10 km/jam, 20 km/jam, 30 km/jam, 40 km/jam, dan 50 km/jam. Untuk setiap kecepatan, pengujian akan diulang sebanyak enam kali guna memastikan konsistensi dan akurasi data yang diperoleh. Untuk mengukur jarak tempuh dilakukan dengan mengukur tempat awal motor berjalan hingga batas akhir motor dijalankan. Pengukuran dilakukan di jarak 1 km, 2 km, 3 km, 4 km, dan 5 km.

Proses pengujian akan dilakukan sebagai berikut pertama, kendaraan motor listrik akan dijalankan pada kecepatan yang ditentukan. Selama percobaan, data kecepatan dan jarak akan diambil melalui layar LCD speedometer yang telah terintegrasi dengan sistem pengukuran berbasis modul MCP2515 dan Arduino Uno. Pada saat yang bersamaan, aplikasi yang terintegrasi dengan *bluetooth* akan membantu dalam mengukur kecepatan dan jarak. selain itu, untuk media pembanding akan menggunakan aplikasi dari playstore yaitu “GPS Speedometer”.

Setelah data dari setiap percobaan diambil, hasilnya akan dianalisis untuk menentukan apakah terdapat *error* atau tidak. *Error* dihitung untuk mengetahui seberapa akurat sistem dalam mengukur kecepatan dan jarak yang ditempuh oleh kendaraan. Analisis ini akan mencakup perhitungan rata-rata *error* dan penyimpangan standar untuk setiap kecepatan yang diuji.

B. Pengujian pada Prototipe

Pengujian pada prototipe dilakukan dengan memvariasikan waktu dan kecepatan. Total Pengujian akan dilakukan sebanyak 24 kali percobaan pada pengujian kecepatan terdiri dari 2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam. Setiap kecepatan akan diuji dengan waktu yang berbeda yaitu 15 detik, 30 detik, 45 detik, 60 detik, 75 detik, dan 90 detik. Selama percobaan, akan menggunakan alat tachometer sebagai pembanding kecepatan yang dimana dengan menggunakan tachometer akan menghasilkan nilai RPM di setiap kecepatan yang berbeda. Nilai RPM tersebut akan digunakan untuk perhitungan secara teoritis. Selain pengukuran kecepatan, secara bersamaan akan dilakukan pengukuran jarak tempuh dengan cara setiap kecepatan terdapat delapan variasi waktu yang berbeda. Di setiap waktu yang berbeda, akan dilakukan pengukuran jarak tempuh. Setelah data dari setiap percobaan diambil, hasilnya akan dianalisis untuk menentukan nilai *error* dan akurasi sistem.

Pengujian ini diharapkan dapat memberikan gambaran yang jelas mengenai performa sistem dalam kondisi nyata. Dengan demikian, hasil pengujian akan menjadi dasar untuk melakukan perbaikan atau penyempurnaan pada sistem, jika diperlukan. Akhirnya, validasi keberhasilan sistem akan dicapai apabila data yang diperoleh menunjukkan tingkat akurasi yang tinggi dan konsistensi yang baik dalam berbagai kondisi kecepatan dan jarak yang diuji.

4.2.1 Standar Keteknikan

Pengujian ini dilakukan dengan mengikuti standar keteknikan dalam pembuatan Speedometer dan Odometer dalam beberapa sumber sebagai berikut :

- A. SAE J2976 : Standar ini memberikan panduan tentang akurasi sistem speedometer dan odometer untuk kendaraan jalan raya. Standar ini mencakup persyaratan akurasi dan offset sistem speedometer, serta cara pengukuran kecepatan dan jarak tempuh kendaraan menggunakan data rotasi roda. Standar ini diterbitkan oleh SAE International [14].
- Akurasi dan Offset: Speedometer harus memiliki tingkat akurasi tertentu dalam mengukur kecepatan kendaraan dan memiliki offset yang dapat diterima.
 - Integrasi Sistem: Sistem speedometer dan odometer harus terintegrasi dengan sistem elektronik kendaraan, menggunakan data rotasi roda untuk menghitung kecepatan dan jarak tempuh.
- B. AE J678: Standar ini mencakup persyaratan untuk speedometer dan odometer pada kendaraan penumpang, bus, dan truk, termasuk metode pengukuran dan akurasi[15].
- Akurasi dan Keandalan: Standar ini mencakup persyaratan akurasi untuk speedometer dan tachometer dalam kendaraan penumpang, bus, dan truk.
 - Metodologi Pengukuran: Menguraikan metodologi untuk menentukan revolusi roda per unit jarak dan variasi desain sistem keseluruhan.
- C. ECE R39: Regulasi dari Komisi Ekonomi untuk Eropa (ECE) yang mengatur tentang speedometer dan odometer untuk kendaraan bermotor, termasuk spesifikasi teknis dan persyaratan pengujian.[16]
- Akurasi dan Ketahanan: Menetapkan persyaratan untuk akurasi, ketahanan, dan keandalan speedometer.[16]
 - Desain dan Konstruksi: Speedometer harus dirancang untuk memberikan pembacaan kecepatan yang akurat di berbagai kondisi dan tahan terhadap pengaruh lingkungan.[16]
 - Prosedur Pengujian: Spesifikasi prosedur pengujian untuk verifikasi kepatuhan peralatan speedometer dengan regulasi.[16]
 - Keselamatan Jalan: Bertujuan untuk memastikan semua kendaraan memberikan informasi kecepatan yang andal dan akurat kepada pengemudi.

4.2.2 Parameter yang diukur

Parameter yang digunakan untuk menguji kinerja sistem ini terdiri dari dua parameter utama sebagai berikut:

A. Keberhasilan Pembacaan Data Speedometer dan Odometer

Keberhasilan pembacaan data speedometer dan odometer akan dinilai berdasarkan apakah data tersebut berhasil dikirim ke pengguna dan apakah informasi yang ditampilkan akurat serta konsisten saat kendaraan beroperasi. Untuk mengukur tingkat keberhasilan ini, akan dilakukan 32 kali percobaan. Setiap percobaan akan mengevaluasi apakah data kecepatan dan jarak yang diambil oleh sistem dapat dikirim secara efektif dan ditampilkan dengan benar pada layar LCD speedometer dan odometer. Tingkat keberhasilan akan dihitung sebagai persentase dari jumlah percobaan yang berhasil dibandingkan dengan total percobaan yang dilakukan. Pada pengujian pengambilan kecepatan dan jarak ada beberapa parameter-parameter yang akan diujikan seperti berikut :

a. Pengujian Akurasi Kecepatan dan Jarak

Metode perhitungan akurasi kecepatan berdasarkan kesalahan yang sudah dihitung sebelumnya. Tujuan perhitungan ini adalah untuk memastikan bahwa pembacaan jarak yang ditampilkan pada Aplikasi SPEETER akurat. Rumus perhitungan akurasi kecepatan adalah sebagai berikut:

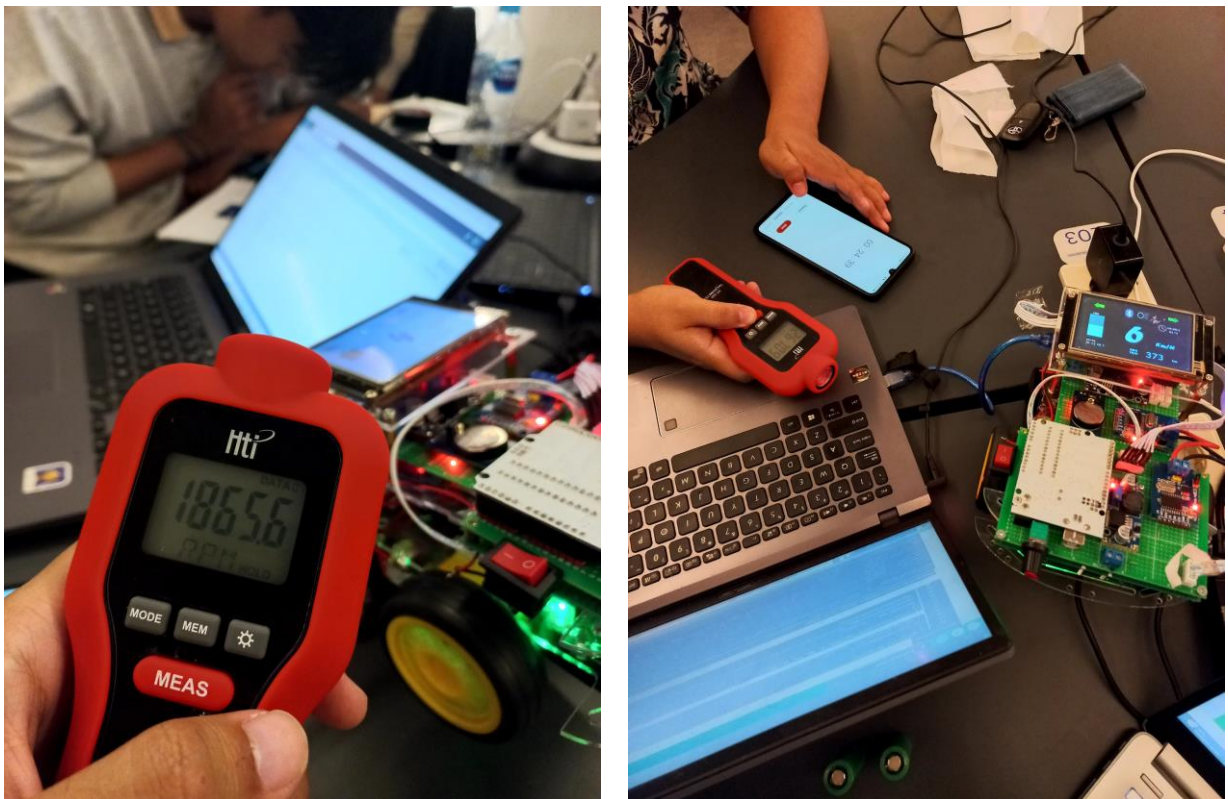
$$\text{Akurasi Kecepatan} = 100\% - \text{Error Kecepatan} \quad (4.5)$$

Metode perhitungan akurasi jarak berdasarkan kesalahan yang sudah dihitung sebelumnya. Tujuan perhitungan ini adalah untuk memastikan bahwa pembacaan jarak yang ditampilkan pada Aplikasi SPEETER akurat. Rumus perhitungan akurasi jarak adalah sebagai berikut:

$$\text{Akurasi Jarak \%} = 100\% - \text{Error Jarak} \quad (4.6)$$

b. Pengujian Menggunakan Tachometer

Metode yang digunakan adalah dengan memasang tachometer pada kendaraan untuk mengukur RPM (*Revolutions Per Minute*). Setelah itu, catat RPM pada berbagai kondisi kecepatan kendaraan. Gunakan rumus konversi RPM ke kecepatan untuk menghitung kecepatan kendaraan berdasarkan data RPM yang diperoleh. Bandingkan hasil perhitungan kecepatan ini dengan data kecepatan dari sistem yang ditampilkan pada instrument display atau aplikasi. Gambar 4.13 merupakan cara untuk pengujian *dengan Tachometer*.



Gambar 4. 13 Pengujian Tachometer

Tujuan pengujian ini adalah untuk memastikan bahwa kecepatan kendaraan yang diukur oleh sistem sesuai dengan kecepatan yang dihitung dari RPM menggunakan tachometer. Untuk menghitung kecepatan berada di persamaan 4.7.

● Konversi RPM ke Kecepatan

$$\text{Kecepatan} = \text{RPM} \times \text{Keliling roda (km)} \times 60 \quad (4.7)$$

Setelah mendapatkan nilai kecepatan, langkah selanjutnya dapat menghitung jarak tempuh seperti pada persamaan 4.5. jika jarak berupa meter maka dapat mengikuti persamaan 4.9.

$$Jarak (m) = Kecepatan \times \frac{detik}{3600 \text{ detik/jam}} \quad (4.8)$$

$$Jarak (Km) = (Kecepatan \times \frac{detik}{3600 \text{ detik/jam}}) \times 1000 \quad (4.9)$$

$$Keliling \text{ roda} = \pi \times d \quad (4.10)$$

Sebelum menghitung kecepatan, alangkah baiknya mengetahui diameter roda untuk menghitung keliling roda seperti persamaan 4.10.

B. Error Nilai Kecepatan dan Jarak

Error kecepatan akan dihitung berdasarkan percobaan hasil percobaan. Pada setiap percobaan, nilai kecepatan yang terbaca oleh sistem akan dibandingkan dengan kecepatan tachometer. Kecepatan tachometer didapatkan dari hasil perhitungan kecepatan dan jarak. Jika *Error* Jarak atau kecepatan mendekati 0%, maka Akurasi mendekati 100%. Perhitungan *Error* baik *error* kecepatan maupun jarak dapat dihitung dengan Persamaan (4.8).

$$Error = \left| \frac{Messured \ Value - True \ Value}{True \ Value} \right| \times 100 \% \quad (4.11)$$

BAB 5. HASIL PENGUKURAN DAN ANALISIS

5.1. Analisis Hasil

5.1.1 Hasil dan Analisis Pengujian Indikator

Hasil pengujian indikator didasarkan pada desain eksperimen yang sudah ditentukan pada pembuatan proposal Tugas Akhir . Pengujian ini dibagi menjadi 3 bagian, yaitu Uji kestabilan pada pembacaan Tachometer dan Serial Monitor, pembacaan LCD.

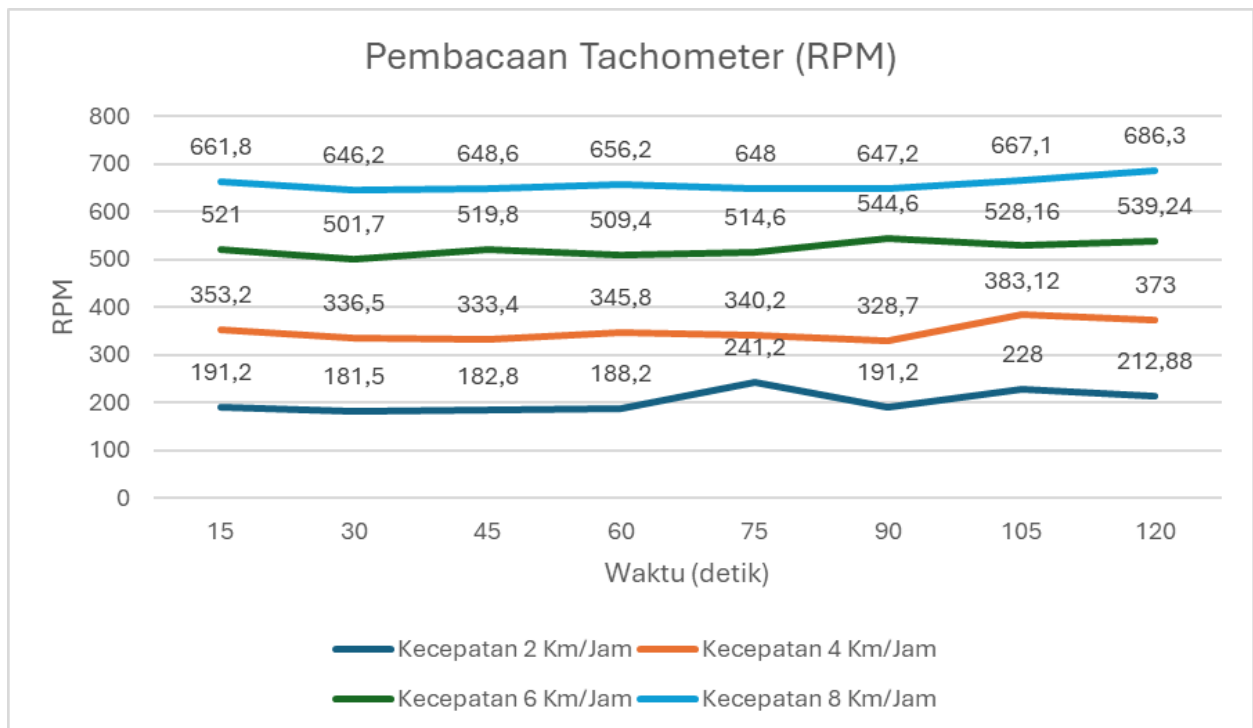
A. Uji Kestabilan pada Pembacaan Tachometer dan Serial Monitor

Pengukuran ini dilakukan dengan pembacaan dari Tachometer dan Serial Monitor dengan melakukan 32 kali percobaan. Tabel 5.1 merupakan data yang sudah didapatkan selama pengujian yang sudah dilakukan.

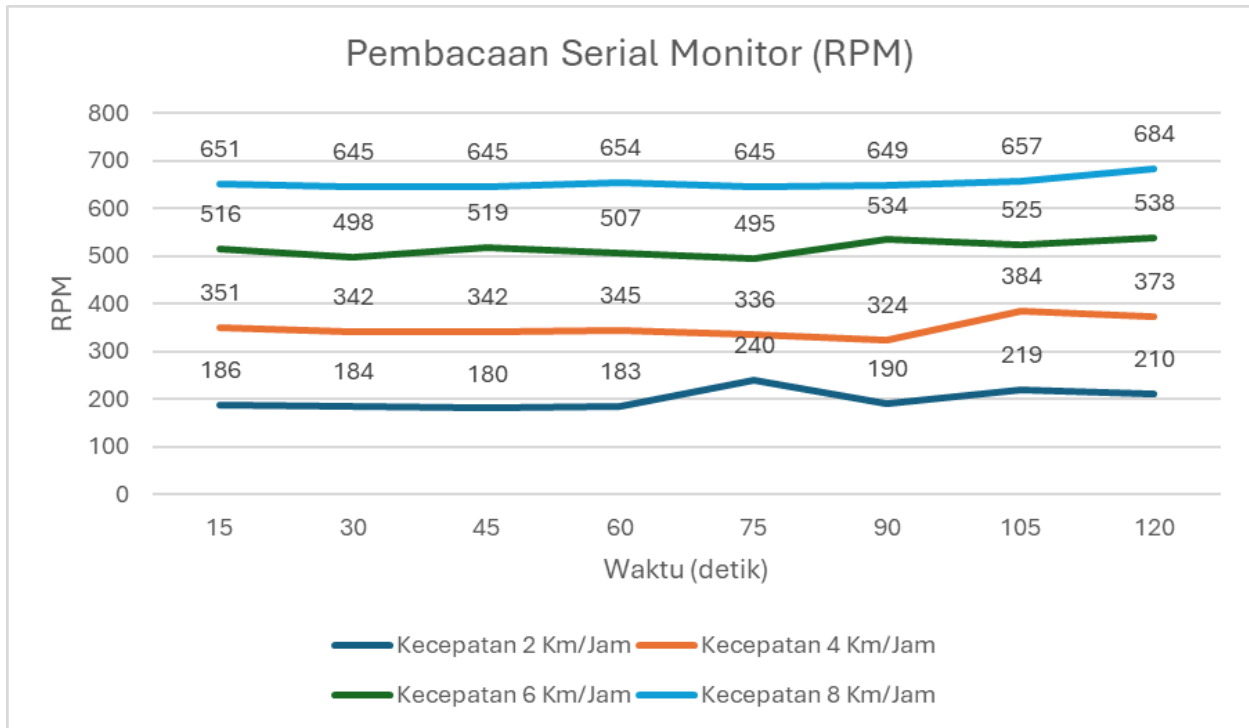
Tabel 5. 1 Pembacaan Tachometer dan Serial Monitor

Percobaan Ke -	Waktu (detik)	Pembacaan Tachometer	Serial Monitor
		RPM	RPM
Kecepatan 2 Km/jam			
1	15	191.20	186.00
2	30	181.50	184.00
3	45	182.80	180.00
4	60	188.20	183.00
5	75	241.20	240.00
6	90	191.20	190.00
7	105	228.00	219.00
8	120	212.88	210.00
Kecepatan 4 Km/jam			
9	15	353.20	351.00
10	30	336.50	342.00
11	45	333.40	342.00
12	60	345.80	345.00
13	75	340.20	336.00
14	90	328.70	324.00
15	105	383.12	384.00
16	120	373.00	373.00
Kecepatan 6 Km/jam			

17	15	521.00	516.00
18	30	501.70	498.00
19	45	519.80	519.00
20	60	509.40	507.00
21	75	514.60	495.00
22	90	544.60	534.00
23	105	528.16	525.00
24	120	539.24	538.00
Kecepatan 8 Km/jam			
25	15	661.80	651.00
26	30	646.20	645.00
27	45	648.60	645.00
28	60	656.20	654.00
29	75	648.00	645.00
30	90	647.20	649.00
31	105	667.10	657.00
32	120	686.30	684.00



Gambar 5. 1 Pembacaan Tachometer (RPM)



Gambar 5. 2 Pembacaan Serial Monitor

Dari Tabel 5.1 Gambar 5.1 Pengukuran dilakukan untuk empat kecepatan yang berbeda yaitu 2 Km/Jam, 4 Km/Jam, 6 Km/Jam, dan 8 Km/Jam, dengan delapan kali percobaan untuk masing-masing kecepatan di waktu yang telah ditentukan. Nilai RPM yang berbeda pada dikarenakan pengaturan kecepatan roda melalui potensiometer sehingga nilai RPM yang ditampilkan beraneka ragam. Selama pengukuran, tachometer digunakan untuk mengukur putaran roda, dengan tujuan memperoleh nilai RPM. Hasil pengukuran menunjukkan variasi RPM pada tiap kecepatan yang diukur. Data ini digunakan untuk menghitung kecepatan dan jarak tempuh roda.

Selain itu, Tabel 5.1 dan Gambar 5.2 merupakan pembacaan RPM pada serial monitor menggunakan sensor kecepatan yang terhubung dengan piringan roda. RPM dari serial monitor akan menjadi pembandingan dari pembacaan RPM dari tachometer, yang nantinya akan digunakan untuk mengukur kecepatan dan jarak tempuh. data RPM yang akurat diperoleh dan digunakan untuk menghitung kecepatan dan jarak tempuh.

Alat ukur tachometer lebih akurat dibandingkan dengan sensor kecepatan yang dipakai. Tachometer dapat membaca lebih akurat dalam pengukuran RPM karena metode pengukurannya yang langsung dan spesifik ke putaran roda. Walaupun sensor tersebut sudah dikalibrasi, RPM yang dihasilkan sudah bisa mendekati dengan nilai RPM yang dibaca oleh tachometer. Bisa dilihat

dari Tabel 5.1, perbedaan yang dihasilkan tidak begitu besar. Percobaan ini juga dilakukan tanpa beban sehingga motor dapat berputar dengan kecepatan penuh tanpa adanya pengaruh resistansi beban.

B. Pembacaan LCD

Pengukuran ini dilakukan dengan pembacaan dari LCD dengan melakukan 32 kali percobaan. Tabel 5.2 menunjukkan data yang telah diperoleh selama pengujian. Gambar 5.3 memperlihatkan proses pengujian dan pengambilan data pada alat simulasi.



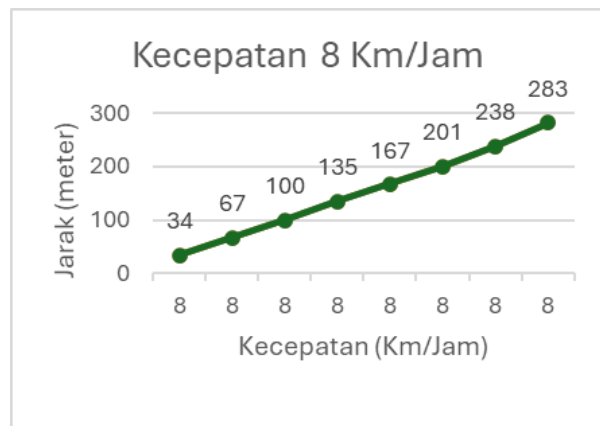
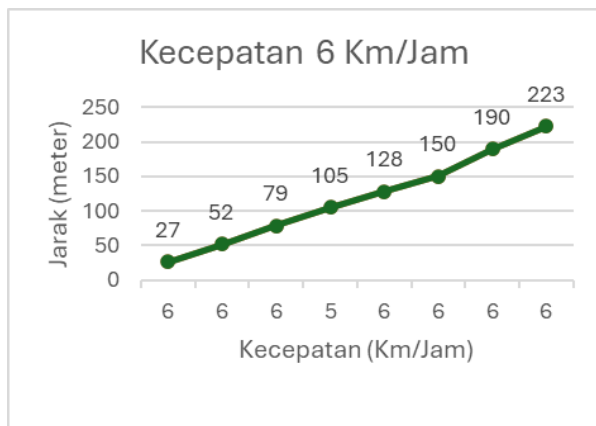
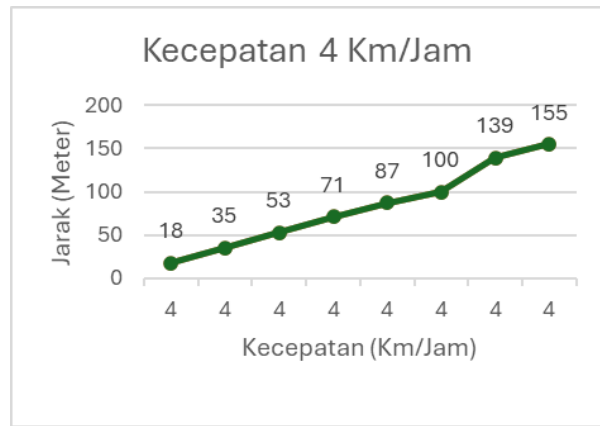
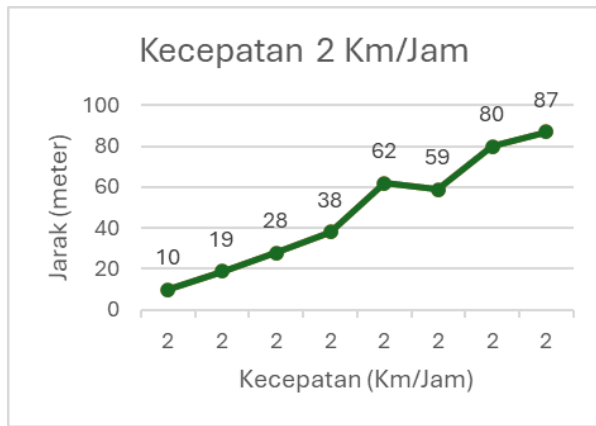
Gambar 5. 3 Pembacaan LCD

Tabel 5. 2 Data Pembacaan LCD

Pembacaan LCD		
Waktu (detik)	Kecepatan 2 Km/Jam	
	Kecepatan (Km/Jam)	Jarak Tempuh (m)
15	2.00	10.00
30	2.00	19.00
45	2.00	28.00

Pembacaan LCD		
60	2.00	38.00
75	2.00	62.00
90	2.00	59.00
105	2.00	80.00
120	2.00	87.00
Kecepatan 4 Km/Jam		
15	4.00	18.00
30	4.00	35.00
45	4.00	53.00
60	4.00	71.00
75	4.00	87.00
90	4.00	100.00
105	4.00	139.00
120	4.00	155.00
Kecepatan 6 Km/Jam		
15	6.00	27.00
30	6.00	52.00
45	6.00	79.00
60	5.00	105.00
75	6.00	128.00
90	6.00	150.00
105	6.00	190.00
120	6.00	223.00
Kecepatan 8 Km/Jam		
15	8.00	34.00
30	8.00	67.00
45	8.00	100.00
60	8.00	135.00

Pembacaan LCD		
75	8.00	167.00
90	8.00	201.00
105	8.00	238.00
120	8.00	283.00



Gambar 5. 4 Pembacaan LCD

Tabel 5.2 dan Gambar 5.4 merupakan hasil pembacaan dari LCD. Nilai jarak tempuh yang dihasilkan pada setiap kecepatan (2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam) menunjukkan kenaikan yang dikarenakan waktu tempuh pada setiap kecepatan memiliki rentang pengujian dari 15 detik – 120 detik. Semakin lama waktu yang ditempuh maka jarak tempuh yang dihasilkan

semakin jauh. Hal ini merupakan hubungan langsung antara waktu, kecepatan, dan jarak tempuh dalam hukum gerak linear yang dapat dirumuskan seperti pada persamaan (4.12)

$$s = v \times t \quad (4.12)$$

Artinya, jarak tempuh yang ditempuh (s) adalah hasil kali antara kecepatan (v) dan waktu (t). Jika kecepatan tetap konstan, maka semakin lama waktu yang dihabiskan untuk berjalan, semakin jauh jarak tempuh yang akan ditempuh.

C. Pengukuran Pembacaan Teoritis RPM dan Aplikasi

Pengukuran ini dilakukan dengan pembacaan dari Tachometer dan Aplikasi dengan melakukan 32 kali percobaan. Tabel 5.3 merupakan data yang sudah didapatkan selama pengujian yang sudah dilakukan. Gambar 5.5 merupakan saat pengujian pengambilan data pada alat simulasi.

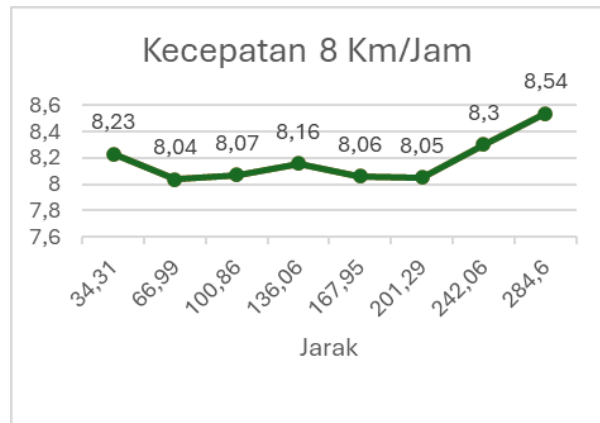
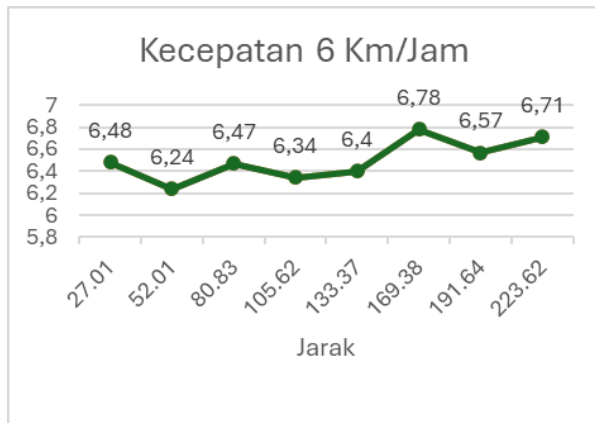
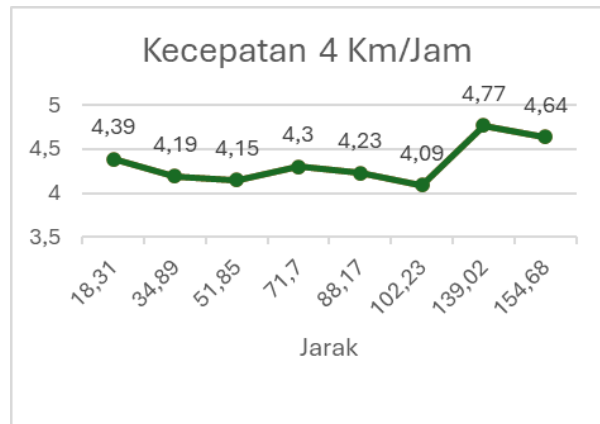
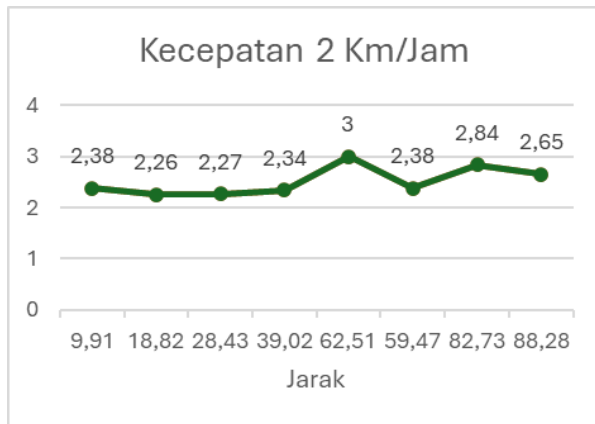


Gambar 5. 5 Hasil Pembacaan Aplikasi Android

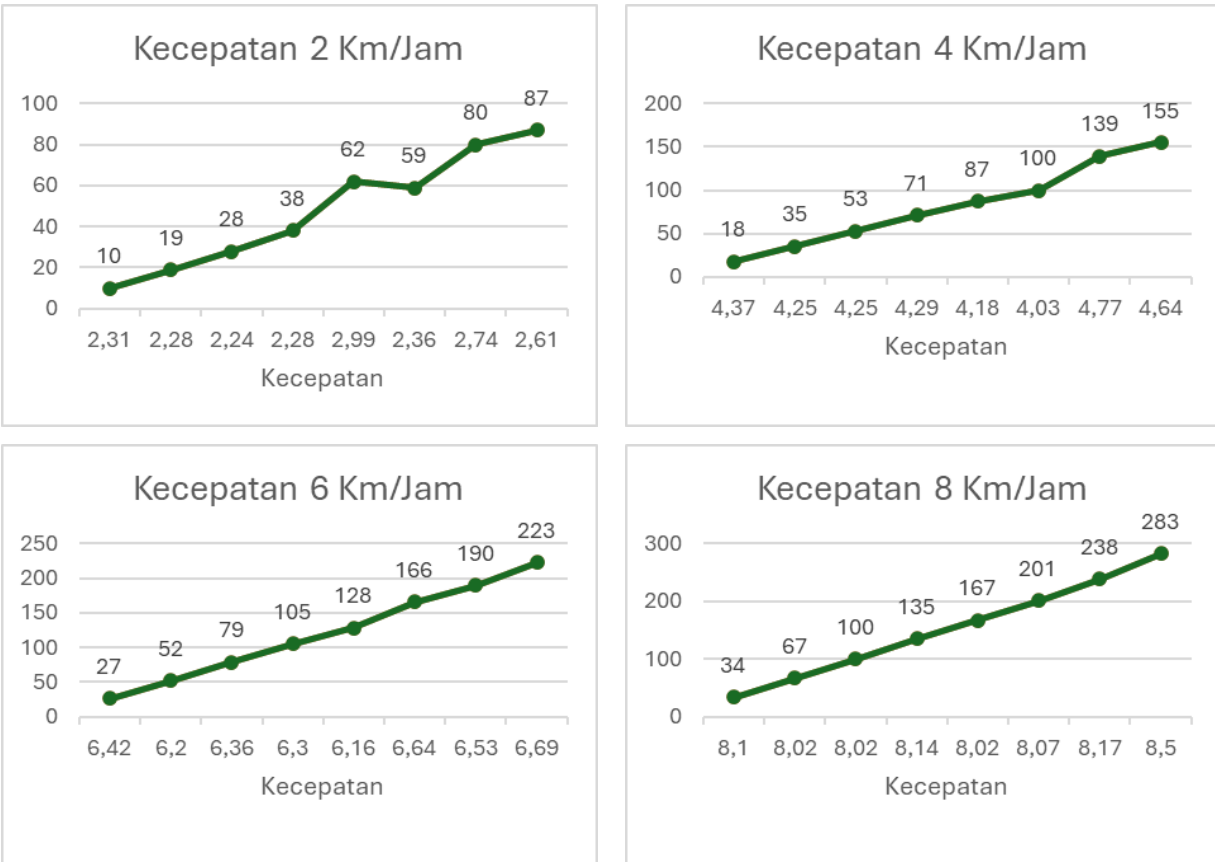
Tabel 5. 3 Data Pengukuran Pembacaan Teoritis RPM dan Aplikasi

Perhitungan Teoritis Tachometer		Aplikasi	
Kecepatan (Km/Jam)	Jarak Tempuh (m)	Kecepatan (Km/Jam)	Jarak Tempuh (m)
Kecepatan 2 Km/Jam			
2.38	9.91	2.31	10.00
2.26	18.82	2.28	19.00
2.27	28.43	2.24	28.00
2.34	39.02	2.28	38.00
3.00	62.51	2.99	62.00
2.38	59.47	2.36	59.00
2.84	82.73	2.74	80.00
2.65	88.28	2.61	87.00
Kecepatan 4 Km/Jam			
4.39	18.31	4.37	18.00
4.19	34.89	4.25	35.00
4.15	51.85	4.25	53.00
4.30	71.70	4.29	71.00
4.23	88.17	4.18	87.00
4.09	102.23	4.03	100.00
4.77	139.02	4.77	139.00
4.64	154.68	4.64	155.00
Kecepatan 6 Km/Jam			
6.48	27.01	6.42	27.00
6.24	52.01	6.20	52.00
6.47	80.83	6.36	79.00
6.34	105.62	6.30	105.00
6.40	133.37	6.16	128.00
6.78	169.38	6.64	166.00
6.57	191.64	6.53	190.00
6.71	223.62	6.69	223.00
Kecepatan 8 Km/Jam			

Perhitungan Teoritis Tachometer		Aplikasi	
Kecepatan (Km/Jam)	Jarak Tempuh (m)	Kecepatan (Km/Jam)	Jarak Tempuh (m)
8.23	34.31	8.10	34.00
8.04	66.99	8.02	67.00
8.07	100.86	8.02	100.00
8.16	136.06	8.14	135.00
8.06	167.95	8.02	167.00
8.05	201.29	8.07	201.00
8.30	242.06	8.17	238.00
8.54	284.60	8.50	283.00



Gambar 5. 6 Perhitungan Teoritis Tachometer



Gambar 5. 7 Pembacaan Aplikasi

Pada Tabel 5.3 dan Gambar 5.7 merupakan hasil perhitungan kecepatan dan jarak tempuh dari pembacaan RPM tachometer pada Tabel 5.3. RPM dari hasil pembacaan oleh tachometer digunakan untuk menghitung kecepatan dan jarak tempuh. Perhitungan ini menggunakan rumus perhitungan kecepatan dan jarak tempuh dalam Persamaan 4.7 dan 4.8. Didapatkan bahwa nilai RPM yang dihasilkan besar, maka nilai kecepatan yang dihasilkan juga besar. Nilai kecepatan yang telah didapatkan akan digunakan untuk menghitung jarak tempuh. Selain itu, hasil perhitungan pengukuran jarak tempuh berbanding lurus dengan waktu yang diterapkan, artinya semakin lama waktu yang ditempuh maka semakin jauh jarak yang ditempuh.

Dalam pengujian ini, dalam mengambil data dari dua sumber yaitu hasil pengukuran dari tachometer dan Aplikasi. Kedua hasil ini kemudian dibandingkan untuk mengevaluasi tingkat *error* dan akurasi dari kedua sumber. Dengan melakukan perbandingan, dapat menentukan seberapa akurat dan reliabel kedua sistem dalam mengukur kecepatan dan jarak tempuh. Jika kita bandingkan antara hasil perhitungan kecepatan dan jarak tempuh dengan hasil pembacaan pada tachometer dan aplikasi pada Tabel 5.4 menunjukkan hasil yang tidak jauh berbeda. Dapat dilihat

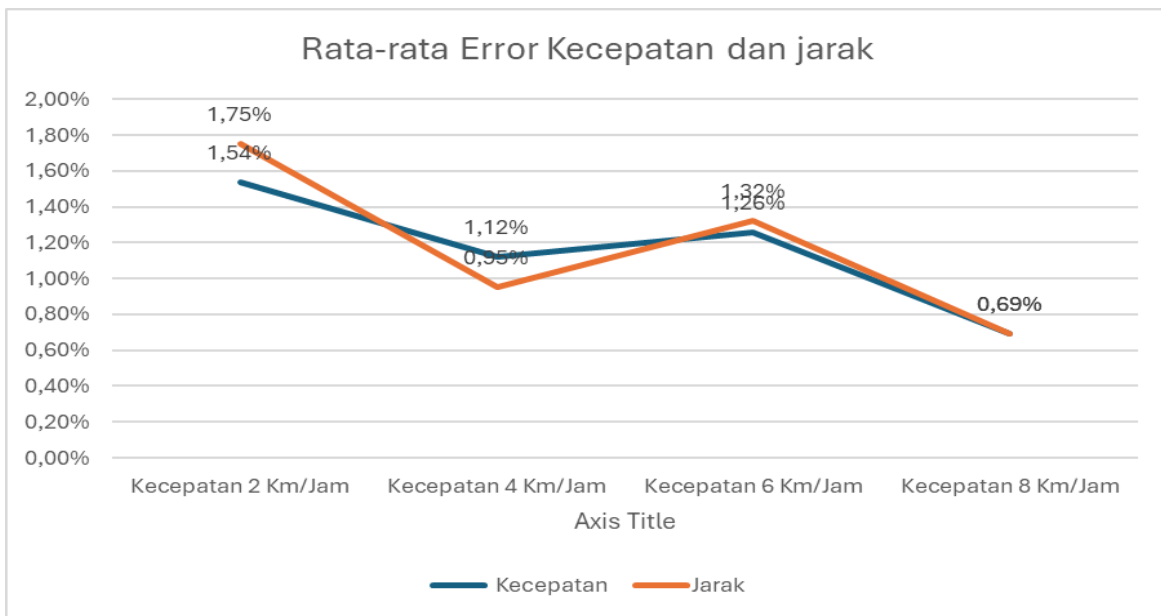
pada Tabel 5.3. hasil pembacaan di LCD dan aplikasi memiliki nilai yang hampir serupa dengan hasil perhitungan dengan tachometer.

D. Error dan Akurasi Kecepatan dan Jarak

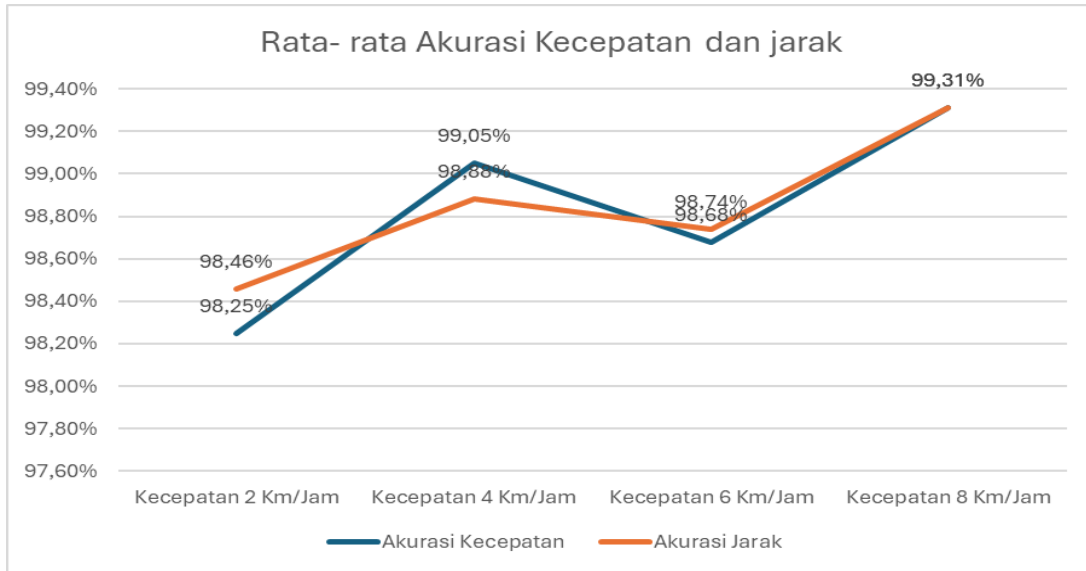
Tabel 5. 4 Data *Error* dan Akurasi Kecepatan dan Jarak

No	<i>Error</i> Jarak Tempuh (%)	<i>Error</i> Kecepatan (%)	Akurasi Kecepatan (%)	Akurasi Jarak Tempuh (%)
Kecepatan 2 km/jam				
1	0.90%	2.89%	97.11%	99.10%
2	0.98%	0.98%	99.02%	99.02%
3	1.50%	1.50%	98.50%	98.50%
4	2.62%	2.62%	97.38%	97.38%
5	0.82%	0.36%	99.64%	99.18%
6	0.78%	0.78%	99.22%	99.22%
7	3.30%	3.40%	96.60%	96.70%
8	1.45%	1.45%	98.55%	98.55%
Rata-rata	1.54%	1.75%	98.25%	98.46%
Kecepatan 4 km/jam				
9	1.68%	0.55%	99.45%	98.32%
10	0.33%	1.52%	98.48%	99.67%
11	2.23%	2.47%	97.53%	97.77%
12	0.98%	0.28%	99.72%	99.02%
13	1.33%	1.24%	98.76%	98.67%
14	2.18%	1.45%	98.55%	97.82%
15	0.01%	0.08%	99.92%	99.99%
16	0.21%	0.01%	99.99%	99.79%
Rata-rata	1.12%	0.95%	99.05%	98.88%
Kecepatan 6 km/jam				
17	0.02%	0.95%	99.05%	99.98%
18	0.02%	0.66%	99.34%	99.98%
19	2.27%	1.65%	98.35%	97.73%
20	0.59%	0.59%	99.41%	99.41%
21	4.03%	3.78%	96.22%	95.97%

No	Error Jarak Tempuh (%)	Error Kecepatan (%)	Akurasi Kecepatan (%)	Akurasi Jarak Tempuh (%)
22	2.00%	2.00%	98.00%	98.00%
23	0.86%	0.62%	99.38%	99.14%
24	0.28%	0.28%	99.72%	99.72%
Rata-rata	1.26%	1.32%	98.68%	98.74%
Kecepatan 8 km/jam				
25	0.89%	1.62%	98.38%	99.11%
26	0.01%	0.24%	99.76%	99.99%
27	0.86%	0.61%	99.39%	99.14%
28	0.78%	0.29%	99.71%	99.22%
29	0.56%	0.52%	99.48%	99.44%
30	0.14%	0.23%	99.77%	99.86%
31	1.68%	1.56%	98.44%	98.32%
32	0.56%	0.45%	99.55%	99.44%
Rata-rata	0.69%	0.69%	99.31%	99.31%
Rata-rata Keseluruhan	1.15%	1.17%	98.83%	98.85%



Gambar 5. 8 Rata-rata *Error* Kecepatan dan Jarak Tempuh



Gambar 5. 9 Rata-rata Akurasi Kecepatan dan Jarak Tempuh

Setelah melakukan pengujian dan perhitungan, didapatkan nilai *error* dan akurasi dari perhitungan yang telah dilakukan. Tabel 5.4. merupakan hasil *error* dan akurasi dari nilai kecepatan dan jarak tempuh pada masing-masing kecepatan 2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam. Perhitungan *error* diambil dari hasil pembacaan tachometer dengan pembacaan pada aplikasi, dengan hasil dari tachometer sebagai nilai referensi.

Nilai *error* yang dihasilkan didapatkan dari persamaan 4.11. yang dimana dari delapan percobaan pada setiap kecepatan dijumlahkan untuk mendapatkan nilai rata-rata *error* dengan tujuan mengetahui hasil *error* yang baik. Dari Tabel 5.5 dan Gambar 5.6 Setiap kecepatan menunjukkan nilai *error* yang cukup bagus, baik *error* kecepatan maupun *error* jarak. Bisa dilihat nilai *error* terendah berada di kecepatan 8 km/jam baik *error* kecepatan dan jarak menunjukkan nilai terendah yaitu 0.69%. Sedangkan *error* tertinggi berada di kecepatan 2 km/jam tetapi masih batas aman *error*. Dari keseluruhan nilai *error* yang dihasilkan baik kecepatan maupun jarak tempuh menunjukkan nilai yang aman yaitu kurang dari 5%.

Untuk mendukung hasil pengukuran, maka diperlukan perhitungan akurasi. Dapat dilihat pada Tabel 5.4. Bahkan akurasi terendah berada di kecepatan 2 km/jam yaitu 98.25% (kecepatan) dan 98.46% (jarak tempuh). Walaupun terdapat variasi nilai kecepatan yang diakibatkan pengaturan potensiometer seperti Gambar 5.9. yang menunjukkan menunjukkan rata-rata akurasi kecepatan dan jarak pada berbagai kecepatan (2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam).

Gambar ini memiliki dua kurva: akurasi kecepatan (biru), yang menunjukkan seberapa akurat kecepatan yang diatur sesuai dengan kecepatan yang diharapkan, dan akurasi jarak (*orange*), yang menunjukkan seberapa akurat jarak yang ditempuh sesuai dengan jarak yang diharapkan pada kecepatan tertentu.

Pada kecepatan 2 km/jam, baik akurasi kecepatan maupun jarak berada di bawah 99%, dengan akurasi kecepatan (98.46%) sedikit lebih tinggi daripada akurasi jarak (98.25%). Pada kecepatan 4 km/jam, terdapat peningkatan yang signifikan dalam akurasi baik untuk kecepatan (99.06%) maupun jarak (99.05%), dengan nilai di atas 99%, menunjukkan kinerja yang lebih bagus. Namun, pada kecepatan 6 km/jam, terdapat sedikit penurunan dalam akurasi baik untuk kecepatan (98.74%) maupun jarak (98.64%) dibandingkan dengan kecepatan 4 km/jam, namun masih cukup tinggi. Pada kecepatan 8 km/jam, baik akurasi kecepatan maupun jarak tempuh mencapai nilai tertinggi di atas 99% (99.31%), menunjukkan konsistensi dan presisi yang sangat baik pada kecepatan ini.

Secara keseluruhan, akurasi kecepatan dan jarak tempuh menunjukkan peningkatan yang signifikan pada kecepatan 4 km/jam, sedikit menurun pada kecepatan 6 km/jam, dan mencapai puncaknya pada kecepatan 8 km/jam. Kecepatan 4 km/jam dan 8 km/jam menunjukkan akurasi tertinggi, yang menunjukkan bahwa sistem lebih stabil pada kecepatan ini. Bahkan akurasi terendah berada di kecepatan 2 km/jam yaitu 98.25% (kecepatan) dan 98.46% (jarak tempuh).

Berdasarkan hasil pengukuran tersebut, akurasi pada setiap kecepatan menunjukkan angka yang bagus, baik akurasi kecepatan dan jarak tempuh yang menunjukkan nilai masih batas aman. Selain itu, Hal ini menandakan bahwa sistem memiliki kinerja yang bisa diandalkan dengan tingkat kesalahan yang kecil.

5.1.2 Perbandingan Performa dengan sistem Lain

Pengujian ini akan membandingkan performa antara *instrument display* SPEETER (Speedometer dan Odometer) dan tachometer. *Instrument display* SPEETER merupakan speedometer dan odometer yang terhubung ke poros roda kendaraan untuk mengukur kecepatan. Ketika kendaraan bergerak, putaran roda diterjemahkan menjadi kecepatan dan dapat menghasilkan nilai jarak tempuh yang nantinya akan ditampilkan pada *instrument display*. Di sisi lain, tachometer mengukur kecepatan kendaraan berdasarkan putaran mesin atau roda, biasanya melalui sensor yang mendeteksi jumlah putaran per menit (RPM). Tachometer umumnya memberikan data yang sangat akurat dan dapat digunakan pada berbagai jenis kendaraan. Dengan

membandingkan kedua alat ini, kita dapat mengevaluasi keakuratan dan keandalan masing-masing metode pengukuran kecepatan.

Tabel 5. 5 Perbandingan performa antara sistem yang dibuat dengan sistem lain

No	Fitur/Komponen	Speeter	Tachometer
1	Prinsip Kerja	Mengukur kecepatan berdasarkan putaran roda yang diterjemahkan melalui MCP2515	Mengukur kecepatan berdasarkan putaran mesin atau roda (RPM)
2	Akurasi	Akurat dalam kecepatan langsung, namun bisa terpengaruh oleh kondisi MCP2515	Sangat akurat dalam mengukur RPM, akurasi kecepatan tergantung pada konversi
3	Jenis Kendaraan	Biasanya digunakan pada kendaraan darat, terutama motor listrik berbasis komunikasi CAN Bus	Dapat digunakan pada berbagai jenis kendaraan dengan mesin atau roda berputar
4	Keandalan	Dapat terpengaruh oleh kerusakan MCP2515	Sangat andal dalam pengukuran RPM
5	Ketahanan Sistem	Ketahanan tergantung pada kondisi fisik microcontroller	Tahan lama jika sensor dan koneksi dirawat dengan baik
6	Kemampuan Tambahan	Memberikan data kecepatan dan jarak tempuh langsung yang mudah dibaca oleh pengemudi	Memberikan data RPM kendaraan
7	Kecepatan Respon	Cepat, namun dapat terpengaruh oleh sistem MCP2515	Sangat cepat dalam mendeteksi perubahan putaran roda
8	Aplikasi	Ada aplikasi monitoring	Tidak ada aplikasi monitoring
9	Harga	Rp. 1.159.000,00	Rp. 525.000,00

Tabel 5.5 menunjukkan perbandingan performa antara dua sistem pengukur kecepatan, yaitu SPEETER dan Tachometer. SPEETER mengukur kecepatan berdasarkan putaran roda yang diterjemahkan melalui MCP2515, sementara Tachometer mengukur kecepatan berdasarkan putaran mesin atau roda (RPM). Dalam hal akurasi, SPEETER akurat dalam kecepatan langsung namun bisa terpengaruh oleh kondisi MCP2515, sedangkan Tachometer sangat akurat dalam mengukur RPM dan akurasi kecepatan tergantung pada konversi.

SPEETER umumnya digunakan pada kendaraan darat, terutama motor listrik berbasis komunikasi CAN Bus, sementara Tachometer dapat digunakan pada berbagai jenis kendaraan

dengan mesin atau roda berputar. Dari segi keandalan, SPEETER dapat terpengaruh oleh kerusakan MCP2515, sedangkan Tachometer sangat andal dalam pengukuran RPM. Ketahanan sistem SPEETER bergantung pada kondisi fisik microcontroller, sementara tachometer tahan lama jika sensor dan koneksi dirawat dengan baik.

SPEETER dapat memberikan data kecepatan dan jarak tempuh yang mudah dibaca oleh pengemudi, sedangkan Tachometer hanya memberikan data RPM kendaraan tanpa adanya konversi ke kecepatan. Kecepatan respon SPEETER cepat, namun dapat terpengaruh oleh sistem MCP2515, sedangkan Tachometer sangat cepat dalam membaca RPM dari putaran roda. Dalam hal aplikasi, SPEETER memiliki aplikasi monitoring untuk membaca kecepatan dan jarak tempuh berkendara serta terdapat riwayat perjalanan para pengemudi, sementara tachometer tidak memiliki aplikasi monitoring. Dari segi harga, SPEETER memiliki harga jauh lebih mahal yakni Rp 1.159.000,-, dibandingkan dengan tachometer sebesar Rp 525.000,-. Dengan demikian, SPEETER berbasis IoT menawarkan berbagai fitur tambahan bagi kemudahan pengguna yang membuatnya lebih unggul.

5.1.3 Pemenuhan Spesifikasi Sistem

Tabel 5. 6 Perbandingan usulan dan hasil perancangan sistem

No	Spesifikasi	Usulan	Realisasi
1	Dimensi (panjang x lebar x tinggi)	12,5 x 8,5x5,3	Car chassis
2	Berat (gram)	327 gram	869 ggram
3	Fitur Tambahan	Suhu, Kecepatan, jarak, Sein, indikator lampu, baterai, waktu	Kecepatan, Jarak, Suhu, Waktu
4	Indikator Keluaran Alat	Aplikasi Android	Aplikasi Android
5	Sifat Alat	Fixed	Portable
6	Jangkauan Sinyal	10 Meter	≤ 30 meter
7	Konsumsi Energi	Rendah	Tinggi

Tabel 5.6 merupakan tabel data pemenuhan spesifikasi sistem dengan membandingkan spesifikasi yang diusulkan dan hasil realisasi dari perancangan sistem. Namun, terdapat dua realisasi, pertama realisasi untuk diterapkan ke motor listrik. Kedua, realisasi ke prototipe. Kedua

realisasi ini, tetap menerapkan komunikasi CAN Bus hanya saja sumber data transmitternya berbeda. Realisasi pertama, sumber data berasal dari *controller* votol berbasis CAN Bus pada motor listrik, sedangkan realisasi kedua dari motor DC yang terkoneksi dengan sensor kecepatan. Namun untuk saat ini yang dilakukan pengujian hanya realisasi kedua. Ketidaksesuaian antara spesifikasi usulan dan realisasi proyek ini terutama disebabkan oleh IC CAN Bus pada motor listrik yang ingin diuji tidak berfungsi. Selain itu, keterbatasan waktu dan biaya serta sumber daya juga belum bisa menerapkannya di motor listrik yang akan diuji. Oleh karena itu, pengujian kecepatan dan jarak tempuh akan menggunakan realisasi kedua.

Pendekatan implementasi juga berubah untuk memastikan alat tetap berfungsi dan memenuhi tujuan proyek. Dalam spesifikasi awal, alat diharapkan memiliki dimensi kompleks dan berbobot ringan tetapi dalam realisasi kedua, alat menggunakan *car chassis* yang lebih besar untuk stabilitas dan ruang yang cukup bagi semua komponen. Berat alat juga meningkat dalam realisasi 2 sebesar 869 gram. Untuk fitur tambahan, realisasi kedua hanya bisa memberikan fitur kecepatan, jarak tempuh, waktu, dan suhu. Untuk indikator keluaran atau *user interface-nya* berupa LCD Nextion dan aplikasi Android. Sifat alat realisasi kedua adalah *not fixed* yang berarti tidak sesuai desain yang telah ditentukan di awal perencanaan. Jangkauan sinyal dari alat ini, dikarenakan menggunakan komunikasi *bluetooth* hanya bisa menjangkau kurang lebih 10 meter. Konsumsi energi yang diusulkan adalah rendah. Pada realisasi pertaman konsumsi energi yang dihasilkan rendah karena tegangan yang dipakai hanya 5 Volt dan motor listrik tersebut didukung BMS sehingga konsumsi yang dihasilkan rendah. Namun dalam realisasi kedua, konsumsi energi menjadi tinggi dikarenakan hanya memakai baterai *cell* 8.4 Volt tanpa adanya BMS sehingga baterai cepat habis ketika alat dijalankan. Penyesuaian ini diperlukan untuk memastikan alat dapat berfungsi dengan baik meskipun konsumsi energi lebih tinggi.

Meskipun terdapat beberapa ketidaksesuaian, proyek ini tetap berfokus kepada tujuannya. Alat ini berhasil mengukur dan menampilkan kecepatan, jarak tempuh, suhu, dan waktu, serta mengirim data ke aplikasi Android melalui koneksi *bluetooth*. Implementasi dan pengujian alat pada model *car chassis* dan motor DC menunjukkan sistem bekerja dengan baik dan memenuhi spesifikasi dasar yang diusulkan. Perubahan dan penyesuaian ini memastikan alat tetap relevan dengan tujuan utama Tugas Akhir .

5.1.4 Pengalaman Pengguna

Pengalaman pengguna dari sistem ini tidak dapat diimplementasikan dikarenakan sistem ini tidak dapat terpasang pada motor listrik yang tersedia di Laboratorium Sistem Tenaga Listrik FTI UII. Terdapat tiga motor listrik, dua motor listrik menggunakan *controller* yang tidak mendukung sistem komunikasi CAN Bus dan terdapat satu motor yang mempunyai sistem komunikasi CAN Bus tetapi mengalami kerusakan terhadap IC MCP CAN Bus sehingga sistem atau realisasi pertama tidak bisa diterapkan.

Tabel 5. 7 Pengalaman Pengguna

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
1	Motor Listrik 1	Menggunakan <i>controller</i> Votol	Mengganti Sistem komunikasi CAN Bus
2	Motor Listrik 2	Menggunakan <i>controller</i> Votol	Mengganti Sistem komunikasi CAN Bus
3	Motor Listrik 3	Menggunakan <i>controller</i> Votol dan menerapkan komunikasi CAN Bus	Memperbaiki IC MCP CAN Bus pada <i>controller</i>
4.	Desain Hardware/realisasi 2	Semua modul terhubung dengan benar.	Uji setiap modul secara individual untuk memastikan fungsi yang benar.
5	Pengukuran Kecepatan	Potensiometer dan PWM berhasil digunakan untuk mengukur kecepatan motor.	Kalibrasi potensiometer dan menentukan PWM untuk akurasi pengukuran yang lebih baik.
6	Pengiriman Data	Data kecepatan dan jarak tempuh berhasil dikirim melalui CAN bus menggunakan MCP2515.	Pertahankan
7	Penerimaan Data	Data diterima dengan benar di receiver dan ditampilkan di Serial Monitor.	Verifikasi data sebelum pemrosesan lebih lanjut.
8	Penyimpanan Waktu	Pembacaan Waktu dari RTC DS321	Periksa sinkronisasi waktu secara berkala untuk memastikan akurasi.
9	Pengiriman ke Aplikasi	Data berhasil dikirim ke aplikasi smartphone melalui <i>bluetooth</i> HC-05.	Pertahankan
10	Pengujian Sistem	Sistem berhasil diuji dalam kondisi laboratorium dan berjalan sesuai dengan spesifikasi.	Uji sistem dalam kondisi lapangan untuk memastikan keandalan dan ketahanan. Optimalkan pengaturan

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
			daya untuk mengurangi konsumsi energi.

Potensiometer digunakan sebagai input untuk mengontrol kecepatan motor. Potensiometer menghasilkan nilai analog (0-1023) yang dibaca oleh Arduino Uno. Nilai analog dari potensiometer dipetakan ke nilai PWM (0-255) yang digunakan untuk mengatur kecepatan motor. Modul MCP2515 digunakan untuk mengirim dan menerima data melalui protokol CAN bus. CAN bus adalah protokol komunikasi yang andal dan tahan terhadap gangguan, banyak digunakan dalam sistem otomotif. Modul MCP2515 di receiver menerima frame CAN dari transmitter. Data diuraikan dari frame dan diolah oleh Arduino Uno. Modul HC-05 digunakan untuk mengirim data ke aplikasi smartphone melalui *bluetooth*. Hal ini memungkinkan pengguna untuk memantau kecepatan dan jarak tempuh secara nirkabel. Sistem diuji untuk memastikan semua fitur bekerja sesuai dengan spesifikasi, termasuk pengukuran kecepatan, pengiriman data, penerimaan data, dan pengiriman data ke aplikasi pada Android.

5.1.5 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya

Bagian ini memiliki peran penting dalam pengerjaan usulan pada rancangan sistem. Kesesuaian perancangan menjadi faktor penentu dalam mencapai tujuan dan hasil yang diharapkan. Secara umum, bagian ini bertujuan mengevaluasi kesesuaian antara perencanaan manajemen kerja tim dan realisasinya. Dengan demikian, dapat merencanakan langkah-langkah yang jelas untuk melaksanakan usulan. Rencana kegiatan ini sesuai dengan *gantt chart* pada pengerjaan laporan Tugas akhir.

Tabel 5. 8 Kesesuaian antara usulan dan realisasi timeline pengerjaan Tugas Akhir

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
1	Pembelian serta pembuatan alat dan bahan	Januari - Maret	Febuari – Maret
2	Perancangan dan pembangunan sistem	Januari - April	April - Juni
3	Pengujian dan validasi	Mei	Juni

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
4	Expo dan Pengumpulan Laporan Akhir	Juni	Juni

Dari Tabel 5.8 terdapat beberapa penundaan dalam pelaksanaan kegiatan dibandingkan dengan waktu yang diusulkan, terutama pada tahap perancangan dan pembangunan sistem. Meskipun demikian, semua kegiatan utama berhasil diselesaikan walaupun banyak kendala yang dihadapi oleh tim saat pengerjaan Tugas Akhir .

Tabel 5. 9 RAB SPEETER Motor Listrik

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	LCD Nextion	1 pcs	Rp 150.000,00	15 pcs	Rp 750.000,00
2	Modul MCP 2515	1 pcs	Rp 37.500,00	1 pcs	Rp 45.000,00
3	Bluetooth HC-05	1 pcs	Rp 44.000,00	1 pcs	Rp 50.000,00
4	Arduino Uno	1 pcs	Rp 135.000,00	1 pcs	Rp 135.000,00
5	PCB Lubang 8x12	-	-	1 pcs	Rp 12.000,00
6	RTC DS3231	-	-	1 pcs	Rp 70.000,00
7	MP 1584EN DC-DC	-	-	1 pcs	Rp 7.000,00
8	3D Printing			1 pcs	Rp 131.000,00
9	TTL to RS485	-	-	1 pcs	Rp. 9.000,00
Jumlah					Rp 1.209.000,00

Tabel 5. 10 RAB SPEETER Car Chasis

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	LCD Nextion	1 pcs	Rp 150.000,00	1 pcs	Rp 500.000,00
2	Box Baterai 2x Size 18650	-	-	2 Pcs	Rp 10.000,00

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
3	Modul MCP2515	1 pcs	Rp 37.500,00	2 pcs	Rp 90.000,00
4	Bluetooth HC-05	1 pcs	Rp 44.000,00	1 pcs	Rp 44.000,00
5	Car Chassis	-	-	1 pcs	RP 60.000,00
7	PCB Lubang 8x12	-	-	2 pcs	12.000,00
8	RTC DS3231	-	-	1 pcs	Rp 70.000,00
9	Motor DC	-	-	2 pcs	Rp 140.000,00
10	LM2596	-	-	2 pcs	Rp 30.000,00
11	Sensor Kecepatan	-	-	1 pcs	Rp 7.000,00
12	L298N	-	-	1 pcs	20.000,00
13	Arduino Uno	1 pcs	Rp 135.000,00	1 pcs	Rp 135.000,00
Jumlah					Rp1.118.000,00

Selama pembuatan alat Tugas Akhir 2, tim membuat dua alat, pertama diimplementasikan ke motor listrik dan kedua berupa prototipe untuk diterapkan sebagai simulasi kecepatan dan jarak tempuh berbentuk transmitter dan receiver. Dengan ini tim membuat dua RAB yang masing-masing dicantumkan pada Tabel 5.9 dan 5.10.

Pada Tabel 5.9, terdapat banyak tambahan komponen yang berbeda jauh dari komponen yang diusulkan dalam Proposal Tugas Akhir. Setelah menyelesaikan alat dan siap diimplementasikan pada motor listrik, tim menghadapi masalah yaitu terdapat kerusakan IC CAN Bus MCP pada motor listrik. Akibatnya, konsep alat diubah untuk memungkinkan simulasi kecepatan dan jarak tempuh dengan model *transmitter* dan *receiver* berbasis CAN Bus.

Perubahan ini menyebabkan pembengkakan biaya pada RAB akhir, yang tidak sesuai dengan RAB yang ditetapkan dalam Proposal Tugas Akhir 1. Tabel 5.9 dan 5.10 menunjukkan perbedaan signifikan antara usulan di Proposal Tugas Akhir 1 dan RAB yang terealisasi, dikarenakan tambahan komponen dan perbedaan harga saat pembelian. Sedangkan Tabel 5.10 menunjukkan RAB alat prototipe untuk simulasi kecepatan dan jarak tempuh yang menambah biaya pada komponen baru yang tidak diusulkan dalam RAB Proposal Tugas Akhir 1. Total biaya

yang telah dikeluarkan tim untuk membuat SPEETER dalam Tugas Akhir 2 sebesar .Rp 2.327.000,00.

Tabel 5. 11 Realisasi aktivitas pelaksanaan tugas akhir 2

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Selasa, 27 Februari 2024 (3 Jam)	Kuliah Perdana Tugas Akhir 2	Ichiro, Daffis
2	Jumat, 1 Maret 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
	Senin, 4 Maret 2024	Membuat timeline mengerjakan Tugas Akhir 2	Ichiro, Daffis
3	Kamis, 14 Maret 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
4	Sabtu, 23 Maret 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
5	Selasa, 4 April 2024	Design Elektronik	Ichiro, Daffis
6	Rabu, 3 April 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
7	Kamis, 25 April 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
8	26 April 2024	Design LCD Nextion	Ichiro, Daffis
9	6 Mei 2024	Desain Rangkaian dan Program	Ichiro, Daffis
10	Selasa, 7 Mei 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
11	Jumat, 17 Mei 2024 (6 Jam)	Desain Rangkaian dan Koding	Ichiro, Daffis
12	Senin, 20 Mei 2024 (6 Jam)	Pembuatan Aplikasi Android	Ichiro, Daffis
13	Selasa, 21 Mei 2024	Solder Rangkaian PCB	Ichiro, Daffis

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
	(6 Jam)		
14	Senin, 27 Mei 2024 (6 Jam)	Survei Motor Listrik SAHIN	Ichiro, Daffis
15	Kamis, 30 Mei 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
16	Rabu, 5 Juni 2024 (6 Jam)	Pemotongan box	Ichiro, Daffis
17	Kamis, 6 Juni 2024 (2 Jam)	Bimbingan	Ichiro, Daffis
18	Kamis, 13 Juni 2024 (6 Jam)	Pembelian Komponen	Ichiro, Daffis
19	Jumat, 14 Juni 2024 (6 Jam)	Pengecekan kembali Komponen di Motor Listrik	Ichiro, Daffis
20	Jumat, 21 Juni 2024 (2 Jam)	Bimbingan dan	Ichiro, Daffis
21	Sabtu, 22 Juni 2024 (6 Jam)	Melanjutkan Pembuatan final Aplikasi dan pengujian alat di motor listrik	Ichiro, Daffis
22	Minggu, 23 Juni 2024 (3 Jam)	Pembelian Komponen dan solder komponen	Ichiro, Daffis
23	Senin, 24 Juni 2024 (6 Jam)	Pembelian Komponen dan Cetak 3D Printing	Ichiro, Daffis
24	Selasa, 25 Juni 2024 (6 Jam)	Melakukan test data random pada LCD dan microcontroller	Ichiro, Daffis
25	Rabu, 26 Juni 2024 (6 Jam)	Perakitan Car Kit	Ichiro, Daffis
26	Kamis, 27 Juni 2024	Bimbingan dan Pengujian Alat	Ichiro, Daffis

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
	(2 Jam)		
27	Jumat, 28 Juni 2024 (6 Jam)	Pengujian Alat dan Pengambilan data	Ichiro, Daffis
28	Senin, 1 Juli 2024 (6 Jam)	Pengujian Alat dan Pengambilan data	Ichiro, Daffis
29	Selasa, 2 Juli 2024 (6 Jam)	Pengujian Alat, Pengambilan data dan membuat laporan	Ichiro, Daffis
30	Rabu, 3 Juli 2024 (6 Jam)	Pengujian Alat, Pengambilan data dan membuat laporan	Ichiro, Daffis
31	Kamis, 4 Juli 2024 (6 Jam)	Membuat Laporan	Ichiro, Daffis
32	Jumat, 5 Juli 2024 (6 Jam)	Membuat Laporan	Ichiro, Daffis
33	Sabtu, 6 Juli 2024 (10 Jam)	Membuat Laporan	Ichiro, Daffis
34	Minggu, 7 Juli 2024 (10 Jam)	Membuat Laporan	Ichiro, Daffis
35	Senin, 8 Juli 2024 (10 Jam)	Membuat Laporan	Ichiro, Daffis
36	Selasa, 9 Juli 2024 (10 Jam)	Membuat Laporan	Ichiro, Daffis

5.2 Dampak Implementasi Sistem

Adapun Implementasi sistem SPEETER berbasis *bluetooth* pada motor listrik memberikan berbagai dampak yang signifikan dalam berbagai bidang. Berikut adalah penjelasan lengkap dampak dari implementasi sistem ini:

a) Aspek Teknologi dan Keterbaruan

Implementasi sistem ini menggunakan teknologi CAN bus dan *bluetooth* dalam pengukuran kecepatan dan jarak, yang dapat meningkatkan keakuratan dan keandalan data.. Dengan menggunakan *bluetooth* untuk mengirim data ke aplikasi smartphone, sistem ini menjadi bagian dari ekosistem IoT, memungkinkan integrasi dengan perangkat pintar lainnya. Dengan menggunakan CAN bus, data dari berbagai sensor dan modul di dalam kendaraan dapat dikirim melalui satu jalur komunikasi, sehingga mengurangi kebutuhan akan kabel yang banyak dan kompleks. Pengurangan jumlah kabel membuat instalasi sistem menjadi lebih mudah dan cepat. Ini juga mengurangi kompleksitas dalam perawatan dan perbaikan, karena lebih sedikit kabel yang perlu diperiksa atau diganti jika terjadi masalah.

b) Aspek Sosial, Kenyamanan dan Keselamatan

Dengan adanya informasi *real-time* mengenai kecepatan dan jarak, pengguna dapat berkendara dengan lebih aman dan berhati-hati dalam berkendara untuk mengurangi risiko kecelakaan. Selain itu, sistem ini dapat memberikan kenyamanan dan kemudahan bagi pengguna motor listrik karena sistem dapat terhubung ke smartphone mereka berupa kecepatan dan jarak tempuh yang mudah untuk digunakan dalam pemantauan riwayat perjalanan.

c) Aspek Politik dan Regulasi

Implementasi teknologi ini bisa membantu pengguna dan produsen dalam memenuhi regulasi pemerintah mengenai keselamatan kendaraan dan emisi, mendukung inisiatif ramah lingkungan. Sistem speedometer harus dirancang untuk memastikan bahwa kecepatan yang ditunjukkan selalu berada dalam batas yang diizinkan, yaitu tidak lebih dari $110\% + 4 \text{ km/jam}$ dari kecepatan asli kendaraan. Ini berarti jika kendaraan bergerak dengan kecepatan 100 km/jam, speedometer tidak boleh menunjukkan lebih dari 114 km/jam. Selain mematuhi standar internasional, perancangan speedometer dan odometer juga harus mematuhi regulasi yang ditetapkan oleh pemerintah Indonesia. Pemerintah Indonesia dapat mengadopsi atau menyesuaikan regulasi internasional ini untuk memastikan keselamatan di jalan raya. Pemerintah dan produsen kendaraan dapat bekerja sama untuk memberikan edukasi kepada pengemudi mengenai pentingnya memahami cara kerja speedometer dan menjaga kecepatan dalam batas yang diizinkan.

5.3 Kelebihan dan Kekurangan Sistem

Berikut adalah Tabel 5.12 merupakan perbandingan keseluruhan antara speedometer analog, speedometer digital, speedometer digital yang terhubung IoT, dan speedometer dengan teknologi CAN Bus dengan *Bluetooth*.

Tabel 5. 12 Perbandingan keseluruhan Speedometer

Aspek	Speedometer Analog	Speedometer Digital	Speedometer Digital IoT	Speedometer CAN Bus
Harga	Tejangkau	Lebih Mahal	Lebih Mahal	Paling Mahal
Keakuratan	Cukup Baik, Tapi terbatas	Tinggi	Sangat Tinggi	Sangat Tinggi
Kemudahan Pengguna	Sangat Mudah	Mudah	Mudah	Mudah
Perawatan	Mudah dan Murah	Lebih rumit dan Mahal	Lebih rumit dan mahal	Paling Rumit dan Mahal
Pengurangan Kabel	Tidak	Tidak	Ya	Ya
Interoperabilitas	Tidak	Ya	Ya	Ya
Kemampuan Diagnostik	Tidak	Terbatas	Tinggi	Sangat Tinggi
Tampilan	Jarum	Layar Digital	Layar digital	Layar digital
Fitur Tambahan	Minimal	Banyak	Sangat Banyak	Sangat Banyak
Kompleksitas Instalasi	Rendah	Sedang	Tinggi	Tinggi
Keandalan	Tinggi, tergantung kondisi mekanis	Tinggi, rentan masalah elektronik	Tinggi, rentan masalah elektronik	Sangat tinggi, rentan serangan siber
Konektivitas	Tidak	Terbatas	Sangat tinggi	Sangat tinggi
Efisiensi Energi	Tinggi	Sedang	Tinggi	Sangat tinggi

Speedometer analog cocok untuk pengguna yang mengutamakan kesederhanaan dan biaya rendah. Speedometer digital cocok bagi mereka yang menginginkan akurasi dan tampilan modern. Speedometer digital IoT dan speedometer dengan teknologi CAN Bus dan Bluetooth adalah pilihan terbaik untuk keakuratan, keandalan, dan fitur canggih, terutama bagi pengguna yang menginginkan integrasi dengan perangkat pintar dan ekosistem IoT, meskipun dengan biaya dan kompleksitas yang lebih tinggi.

Teknologi CAN Bus menawarkan berbagai keuntungan yang signifikan untuk redesain sistem speedometer. Pertama, CAN Bus meningkatkan keakuratan dan keandalan pengukuran kecepatan dan jarak melalui komunikasi yang cepat dan efisien antara sensor dan modul kontrol

kendaraan. Ini mengurangi gangguan elektromagnetik dan memastikan data yang dikumpulkan *real-time* dan akurat.

Kedua, CAN Bus mengurangi kompleksitas kabel fisik dengan menggunakan dua kabel untuk seluruh jaringan komunikasi, menyederhanakan instalasi dan mengurangi potensi masalah kabel. Integrasi dengan perangkat lain menjadi lebih mudah karena banyak kendaraan modern sudah menggunakan jaringan ini, memungkinkan komunikasi yang mulus dengan sistem lain seperti unit kontrol mesin (ECU) dan sistem navigasi.

Keuntungan lain adalah kemampuan diagnostik yang lebih baik, memungkinkan pemantauan kondisi komponen secara *real-time* dan deteksi kesalahan segera, yang membantu dalam perawatan preventif. Selain itu, protokol keamanan yang kuat melindungi jaringan dari gangguan atau serangan siber, dengan enkripsi dan otentikasi data yang memastikan integritas data.

Selain itu, CAN Bus mendukung konektivitas dengan perangkat IoT melalui modul Bluetooth atau Wi-Fi, membuka peluang untuk pemantauan jarak jauh dan integrasi dengan aplikasi smartphone. Pengguna dapat mengakses informasi kendaraan secara *real-time*, meningkatkan pengalaman dan kontrol pengguna.

Terakhir, penggunaan CAN Bus memudahkan kepatuhan terhadap regulasi pemerintah terkait keselamatan dan emisi kendaraan, karena sudah banyak diadopsi sebagai standar industri otomotif. Dengan berbagai keuntungan ini, teknologi CAN Bus meningkatkan keakuratan, keandalan, keamanan, dan konektivitas sistem speedometer, serta mendukung efisiensi operasional kendaraan.

BAB 6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Pada sistem SPEETER untuk motor listrik sudah dapat diimplementasikan, sehingga sebagian besar sesuai dengan spesifikasi yang telah diusulkan, tetapi rancangan motor listrik mengalami kerusakan pada IC CAN Bus MCP. Hal ini mengakibatkan ketidakmampuan alat untuk bekerja sesuai dengan yang diharapkan. Karena itu, untuk dapat membaca kecepatan dan jarak tempuh untuk dapat ditampilkan pada LCD dengan membuat alat simulasi kecepatan dan jarak tempuh menggunakan putaran Motor DC. Alat ini menggunakan sistem *transmitter* dan *receiver*. *Transmitter* dapat mengirimkan data kecepatan dan jarak tempuh dari putaran motor DC dan akan mengirimkan data kecepatan dan jarak tempuh ke *receiver* lalu *receiver* untuk menerima dapat kecepatan dan jarak tempuh yang nantinya dikirimkan ke LCD dan Aplikasi SPEETER. Sistem ini memiliki kemampuan dalam memantau kecepatan dan jarak tempuh untuk pengguna motor listrik. Walaupun demikian, adanya beberapa komponen yang tidak sesuai dengan spesifikasi yang telah diusulkan sebelumnya. Kurangnya alat simulasi ini tidak mempunyai fitur pembacaan lampu sein, lampu utama, dan baterai. Alat ini mempunyai fitur pembacaan kecepatan, jarak tempuh, suhu, dan jam Walaupun akhirnya menggunakan simulasi pada motor DC untuk mengukur kecepatan dan jarak tempuh, alat ini dapat menggantikan alat pada motor listrik (IC CAN Bus MCP) yang rusak.

Meskipun terdapat beberapa perbedaan dari spesifikasi yang diusulkan, namun tugas akhir ini sudah terpenuhi dan tercapai. Tujuan tugas Akhir ini sudah meliputi sebagai perangkat yang memberikan informasi kecepatan dan jarak tempuh yang akurat. Rata-rata *error* keseluruhan yang dihasilkan dari setiap kecepatan (2 km/jam, 4 km/jam, 6 km/jam, dan 8 km/jam) sebesar 1.15% untuk *error* jarak dan 1.17% untuk *error* kecepatan. Dari kedua *error* tersebut, rata-rata akurasi jarak sebesar 98.85% dan kecepatan sebesar 98.83%, menjadikannya sangat andal untuk memantau kecepatan dan jarak tempuh yang membutuhkan presisi dalam pengukuran kecepatan dan jarak tempuh. Dengan adanya alat ini, diharapkan dapat memberikan manfaat yang luar biasa untuk pengguna motor listrik dalam mengurangi resiko kecelakaan dalam berkendara.

6.2 Saran

Berdasarkan hasil pengujian alat, terdapat beberapa saran yang masih perlu dikembangkan kedepannya yaitu:

1. Mengganti IC CAN Bus MCP pada motor listrik yang ada di laboratorium Sistem Tenaga Listrik FTI UII.
2. Membuat notifikasi pada Aplikasi Android dan *instrument display* untuk memberikan peringatan jika kecepatan kendaraan melebihi batas kecepatan yang telah ditentukan.
3. Menambahkan indikator suhu motor dan status baterai baik di aplikasi Android maupun di *instrument display*.
4. Melakukan pengujian sistem di berbagai kondisi nyata dan variasi kecepatan serta jarak tempuh yang lebih luas untuk memastikan keandalan dan konsistensi sistem dalam berbagai situasi.

DAFTAR PUSTAKA

- [1] W. Asmarini, "Pakai Motor Listrik, Kamu Bisa Hemat Ongkos Sampai 75%," CNBC Indonesia, May 2019. [Online]. Available: <https://www.cnbcindonesia.com/news/20230519130321-4-438731/pakai-motor-listrik-kamu-bisa-hemat-ongkos-sampai-75>. [Använd 2023].
- [2] C. M. Annur., "Riset Deloitte dan Foundry: Penggunaan Motor Listrik di Indonesia Naik 13 Kali Lipat dalam Dua Tahun," Databoks, Sept 2019. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2023/09/15/riset-deloitte-dan-foundry-penggunaan-motor-listrik-di-indonesia-naik-13-kali-lipat-dalam-dua-tahun#:~:text=Menurut%20hasil%20riset%20Deloitte%20dan,menjadi%2025.782%20unit%20pada%202022>. [Använd 2023].
- [3] K. Chahyono., "Angka Kecelakaan Tinggi, Kemenhub Minta Pemotor Beralih ke Transportasi Umum," GOOTO, Juli 2023. [Online]. Available: <https://www.gooto.com/read/1746296/angka-kecelakaan-tinggi-kemenhub-minta-pemotor-beralih-ke-transportasi-umum>. [Använd 2023].
- [4] M. H. Sulsitiawan, "Sensor Kelembaban Tanah Multi Point Nirkabel," *Laporan Tugas Akhir Universitas Sanata Dharma*, pp. i -59, 2017.
- [5] U. Suriepto, "Telemonitoring Elektro Kardiografi Portabel," *Laporan Tugas Akhir Teknik Elektro Universitas Komputer Indonesia*, pp. i - 105, 2014.
- [6] Z. M. H. K. T. R. H. Ari Permana L, "Rancang Bangun Sistem Kontrol Nirkabel On-Off Perlatan Listrik dengan Perintah Suara Menggunakan Smartphone Android," *Jurnal Simetrik*, vol. 11, nr 1, pp. 388-397, 2021.
- [7] D. A. Pamungkas, "Perancangan Prototype Speedometer Digital menggunakan Atmega dan Sensor Hall Effect," *Tugas Akhir*, pp. i - 54, 2016.
- [8] E. D. P. G. C. E. Samsul Ma, "Studi Perencanaan Pengaturan Kecepatan Motor BLDC pada Gerobak Listrik dengan Driver Votol EM-30S,," *Jurnal Ilmiah Teknik Elektro*, vol. 5, nr 2, pp. 137-144, 2017.
- [9] A. M. A. T. Arief Wisnu Wardhana, "A Controller Area Network (CAN) Bus Temperature and Humidity Data Monitoring System," *Jurnal Ilmiah Teknik Elektro*, vol. 25, nr 2, pp. 115 - 125, 2019.
- [10] H. K. Ngizzah, "Konsep, Teori dan Prosedur Design Thinking," *S1 Pendidikan Sejarah, Fakultas Ilmu Sosial*, 2018.
- [11] P. A. a. D. A. U. N. Dharmaputra, "Desain dan Implementasi Sistem Instrumentasi Sepeda Motor Listrik serta Uji Coba Kinerjanya,," *Jurnal Teknik Elektro*, vol. 05, nr 01, pp. 7-13, 2022.
- [12] N. R. R. Ridwan, "Rancang Bangun Sistem Akuisisi Data Untuk Dashboard Android Pada Sepeda Motor Listrik," *G-Tech: Jurnal Teknologi Terapan*, vol. 7, nr 2, pp. 618-626, 2023.
- [13] Suzuki, "Perbedaan Speedometer, Odometer dan Tachometer" Wired, 29 Januari 2021, [Online]. Tersedia: <https://www.suzuki.co.id/tips-trik/ternyata-ini-perbedaan-speedometer-odometer-dan-tachometer?pages=all> [Diakses: 23 Juli 2024].

- [14] SAE International, "Speedometer and Odometer Systems for On-Highway Vehicles," SAE J2976-2022, vol. 1, no. 1, pp. 1-3, May 2022. doi: 10.4271/J2976_202205.
- [15] SAE International. "Speedometers and Tachometers - Automotive." *SAE J678*, 1988. doi: [10.4271/J678_198812](https://doi.org/10.4271/J678_198812) ([SAE Mobility Mission](#)) ([SAE Mobilus](#)) ([Engineer Documents Center](#)).
- [16] United Nations Economic Commission for Europe. "ECE Regulation No. 39: Speedometer Equipment." 2018. Link ECE R39.

LAMPIRAN – LAMPIRAN

TABEL PERBAIKAN LAPORAN AKHIR CAPSTONE

MAHASISWA #1 : 20524015 Daffis Hesa Pratama
MAHASISWA #2 : 20524017 Ichiro Rizky Abdul Aziz
JUDUL/TOPIK : Connected Vehicle: Speedometer dan Odometer Berbasis Bluetooth

No	Saran penguji	Perbaikan oleh mahasiswa	Halaman	Status
1	Pertajam urgensi pembuatan alat karena dalam produk pasaran sudah banyak diterapkan bluetooth speedmeter (meskipun pada non EV). Jelaskan perbedaan dengan teknologi yang sudah ada.	menambahkan penulisan di latar belakang untuk menjelaskan perbedaan dengan teknologi yang sudah ada di pasaran. Menekankan kelebihan sistem yang diusulkan dibandingkan dengan teknologi yang sudah ada di pasaran.	10-11	Approved
2	Jelaskan lebih rinci standar keteknikan yang diacu pada desain sistem	Menambahkan penjelasan rinci mengenai standar keteknikan yang digunakan dalam desain sistem	84	Approved
3	Perbaiki bagian aspek-aspek yang mempengaruhi desain sistem. Aspek yang mempengaruhi desain sistem bukan merupakan dampak dari sistem yang dibuat melainkan penyebab mengapa sistem dibuat dengan desain tertentu	Menguraikan aspek-aspek yang mempengaruhi desain sistem dan menjelaskan penyebab mengapa sistem dibuat dengan desain tertentu.	92-95	Approved
4	Pertajam kelebihan sistem sebagai gap dari sistem yang sudah ada di pasaran	Menekankan kelebihan sistem yang diusulkan dibandingkan dengan sistem yang sudah ada di pasaran.	94-95	Approved
5	Jelaskan kekurangan sistem dan uraikan bagaimana membuat redesign sistem untuk memperbaiki kekurangan tersebut	Mengidentifikasi kekurangan sistem dan memberikan solusi untuk memperbaiki kekurangan tersebut dalam bentuk redesign sistem.	94-95	Approved
6	Fitur utama dalam speedometer adalah indikator SoC baterai. Sebaiknya sistem dilengkapi dengan hal tersebut	dalam fitur SoC sudah kami tambahkan pada sistem receiver di motor listrik, tidak hanya SoC saja tetapi ada beberapa fitur yang ditambahkan seperti suhu, jam, lampu sein dan lampu utama. dalam sistem receiver di simulator tidak ditambahkan tapi kami masukkan saran di laporan	84 97	Approved
7				Not started
8				Not started
9				Not started
10				Not started

Yogyakarta, 16 Juli 2024

Menyetujui,
Penguji

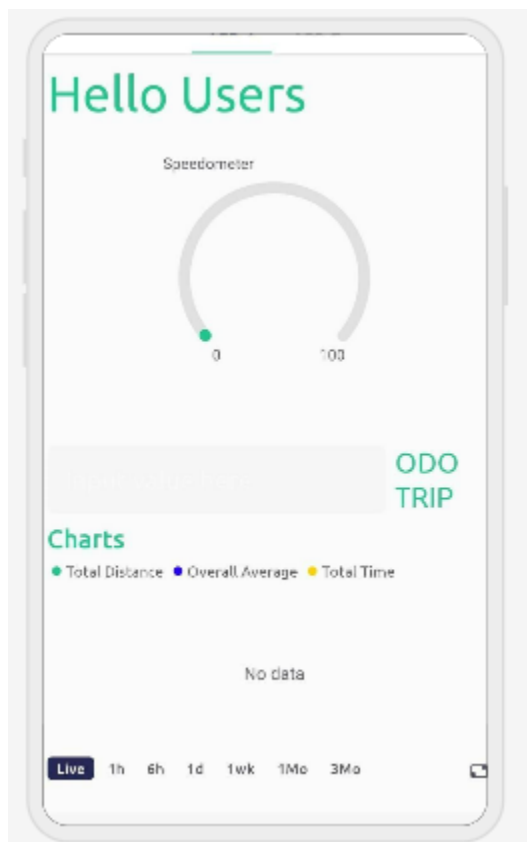
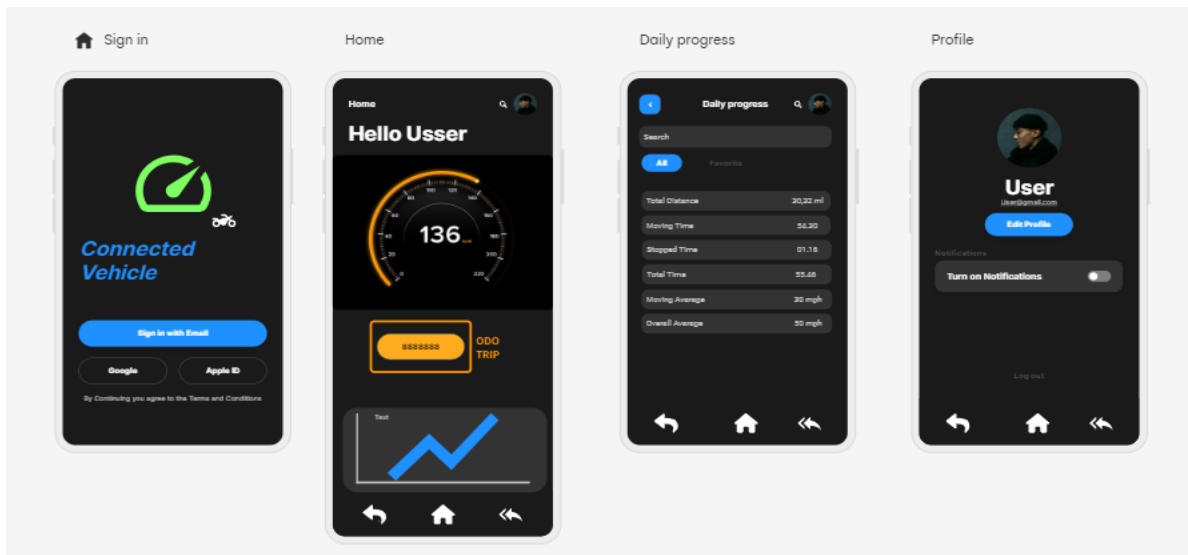


(Husein Mubarak, S.T., M.Eng.)

Desain Sistem Solusi 1 dan 2



Desain Aplikasi Solusi 1 dan 2





MCP2515

Stand-Alone CAN Controller with SPI Interface

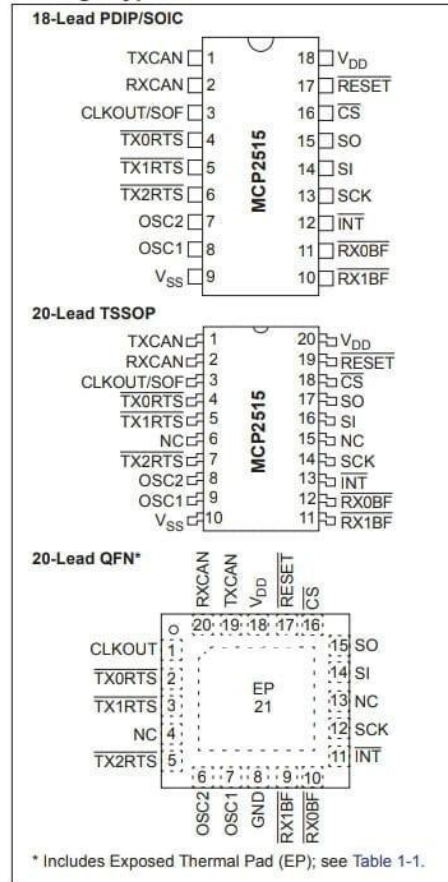
Features

- Implements CAN V2.0B at 1 Mb/s:
 - 0 to 8-byte length in the data field
 - Standard and extended data and remote frames
- Receive Buffers, Masks and Filters:
 - Two receive buffers with prioritized message storage
 - Six 29-bit filters
 - Two 29-bit masks
- Data Byte Filtering on the First Two Data Bytes (applies to standard data frames)
- Three Transmit Buffers with Prioritization and Abort Features
- High-Speed SPI Interface (10 MHz):
 - SPI modes 0,0 and 1,1
- One-Shot mode Ensures Message Transmission is Attempted Only One Time
- Clock Out Pin with Programmable Prescaler:
 - Can be used as a clock source for other device(s)
- Start-of-Frame (SOF) Signal is Available for Monitoring the SOF Signal:
 - Can be used for time slot-based protocols and/or bus diagnostics to detect early bus degradation
- Interrupt Output Pin with Selectable Enables
 - Buffer Full Output Pins Configurable as:
 - Interrupt output for each receive buffer
 - General purpose output
- Request-to-Send (RTS) Input Pins Individually Configurable as:
 - Control pins to request transmission for each transmit buffer
 - General purpose inputs
- Low-Power CMOS Technology:
 - Operates from 2.7V-5.5V
 - 5 mA active current (typical)
 - 1 μ A standby current (typical) (Sleep mode)
- Temperature Ranges Supported:
 - Industrial (I): -40°C to +85°C
 - Extended (E): -40°C to +125°C

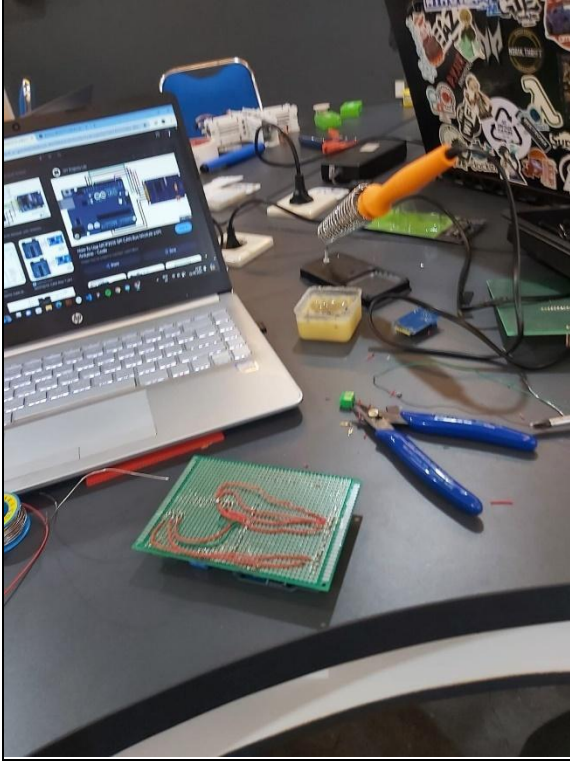
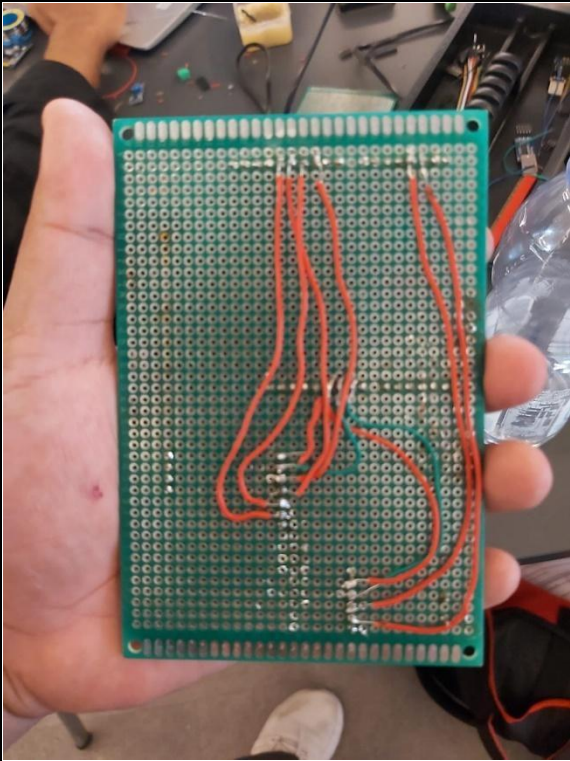
Description

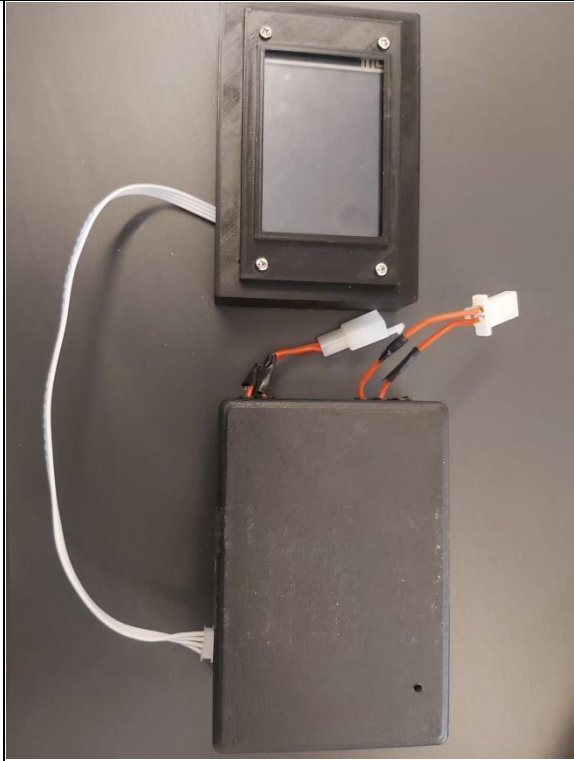
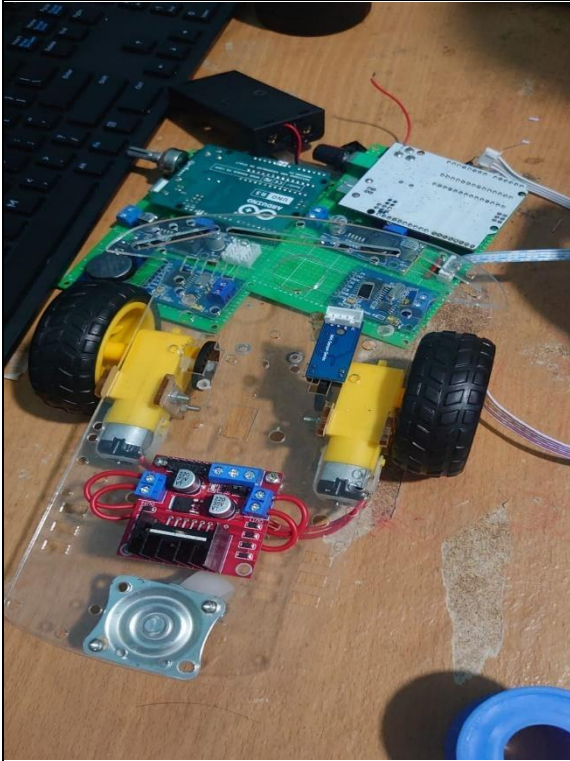
Microchip Technology's MCP2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, Version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCU's overhead. The MCP2515 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI).

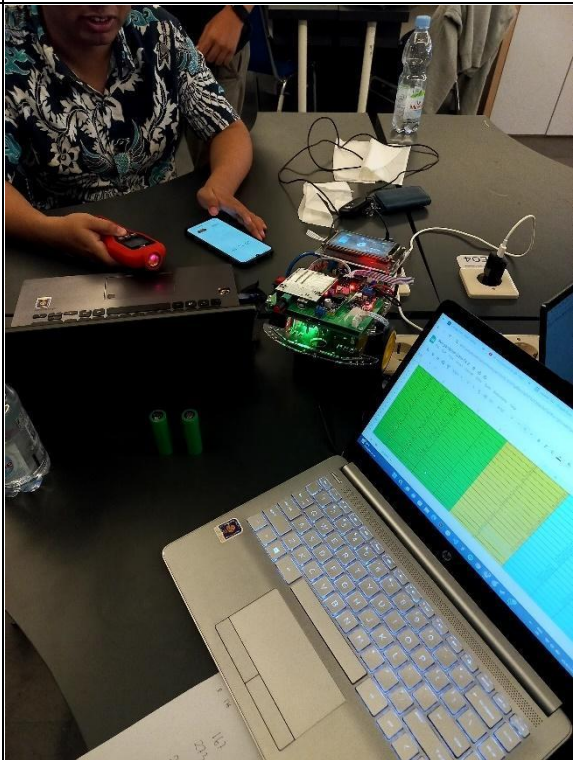
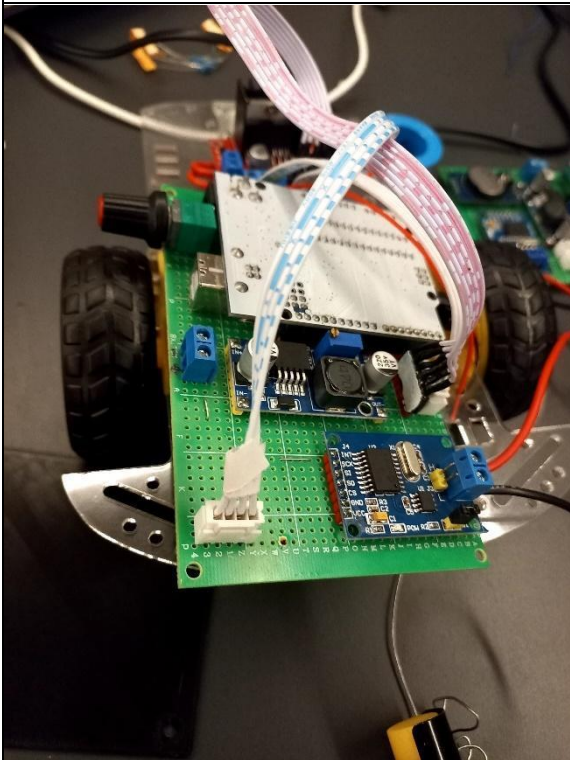
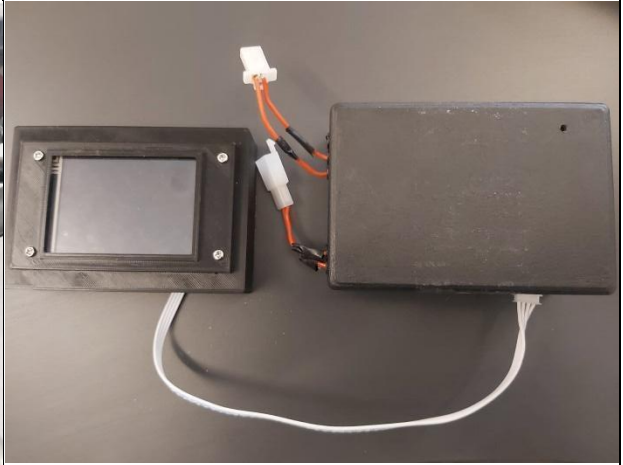
Package Types

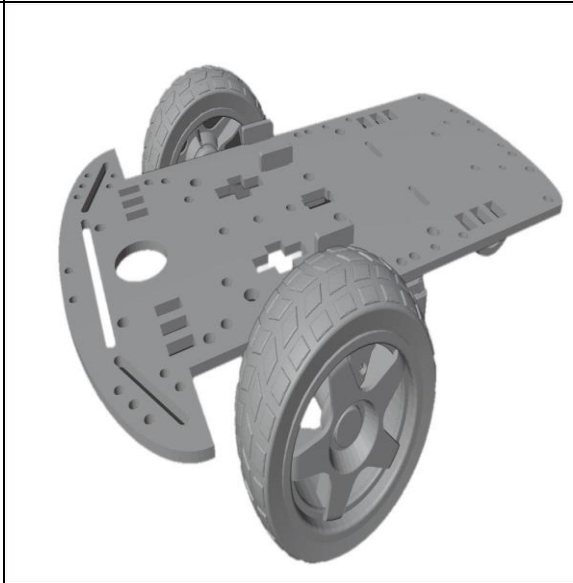
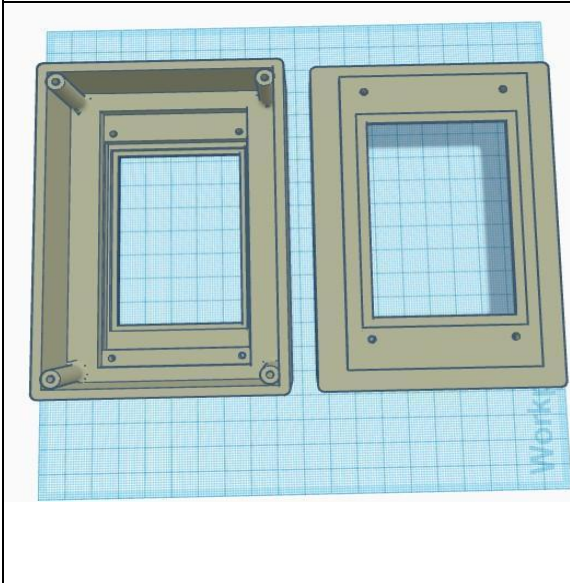


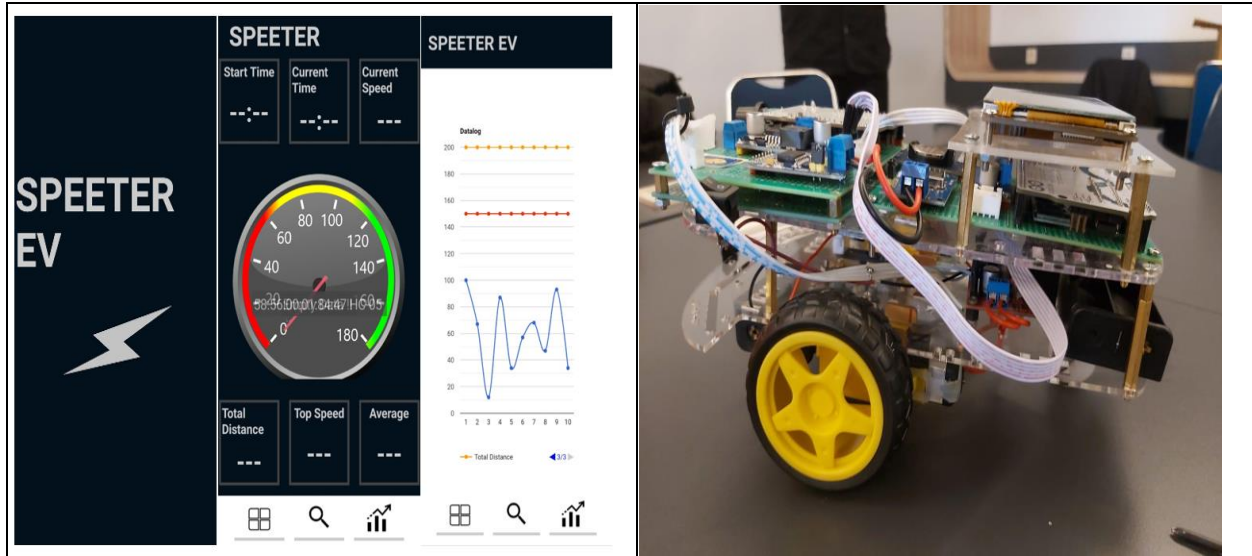
Tugas Akhir 2











Code Receiver Motor Listrik

```
#include <SPI.h>
#include <mcp2515.h>
#include <SoftwareSerial.h>
#include <RTClib.h>
```

```
SoftwareSerial HMIserial(6, 7);
SoftwareSerial BTSerial(2, 3);
struct can_frame canMsg;
MCP2515 mcp2515(10);
RTC_DS3231 rtc;
```

```
int speedRead = 0;
int tempValue = 0;
int currentValue = 0;
int voltageIn = 0;
int voltaseMaksimal = 84;
int voltaseMinimal = 63;
int odometerValue = 0; // Nilai odometer
int previousMinute = 0;
int jam, menit ;
int jumlah_data = 0;
float total_jarak = 0;
float kecepatan_tertinggi = 0;
float kecepatan_rata_rata = 0;
```

```
void setup() {
  Serial.begin(115200);
  HMIserial.begin(9600); // Baudrate for Nextion LCD
```

```

mcp2515.reset();
mcp2515.setBaudrate(CAN_250KBPS,MCP_8MHZ);
mcp2515.setNormalMode();
BTSerial.begin(9600); // Baudrate for Bluetooth
rtc.begin();

// Cek koneksi RTC
if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
}

// Cek status RTC
if (rtc.lostPower()) {
    Serial.println("RTC lost power, let's set the time!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}

// Mengirim inisialisasi ke Nextion
HMISerial.print("t2.txt=\"00:00\"");
HMISerial.write(0xff);
HMISerial.write(0xff);
HMISerial.write(0xff);
}

void loop() {
    if (mcp2515.readMessage(&canMsg) == MCP2515::ERROR_OK) {
        if (canMsg.can_id == 0x80000926) {
            // Speed code
            long decimalValueSpeed = convertHexToDecimalSpeed(canMsg.data);
            speedRead = 32 * (decimalValueSpeed / 10) * 0.001885;
            Serial.print("Speed (Km/h): ");
            Serial.println(abs(speedRead));
            HMISerial.print("speed.val=");
            HMISerial.print(abs(speedRead));
            HMISerial.write(0xff);
            HMISerial.write(0xff);
            HMISerial.write(0xff);
            BTSerial.print("Speed: ");
            BTSerial.print(abs(speedRead));
            BTSerial.println(" Km/h");
            BTSerial.print("Speed: ");
            BTSerial.println(abs(speedRead));
            BTSerial.println(); // Baris baru untuk memisahkan data

```

```

total_jarak += (speedRead / 3.6) / 1000 ;
jumlah_data += 1;
kecepatan_rata_rata = (total_jarak * 3.6) / jumlah_data;

if (kecepatan_tertinggi <= speedRead) {
    kecepatan_tertinggi = speedRead;
}

} else if (canMsg.can_id == 0x80001026) {
    // Temperature code
    long decimalValueTemp = convertHexToDecimalTemp(canMsg.data);
    tempValue = decimalValueTemp;
    Serial.print("Temperature (Celsius): ");
    Serial.println(abs(tempValue));
    HMISerial.print("temp.val=");
    HMISerial.print(abs(tempValue));
    HMISerial.write(0xff);
    HMISerial.write(0xff);
    HMISerial.write(0xff);

    //odometer code
    // Baca waktu dari RTC3231
    DateTime now = rtc.now();
    jam    = now.hour();
    menit  = now.minute();

    // Kirim waktu ke LCD Nextion
    HMISerial.print("t2.txt=\");
    if (jam < 10) HMISerial.print("0");
    HMISerial.print(jam);
    HMISerial.print(":");
    if (menit < 10) HMISerial.print("0");
    HMISerial.print(menit);
    HMISerial.print("\");
    HMISerial.write(0xff);
    HMISerial.write(0xff);
    HMISerial.write(0xff);

    // Simulasikan jarak yang ditempuh
    int timeElapsed = (menit + jam * 60) - previousMinute;
    if (timeElapsed < 0) {
        timeElapsed += 1440; // Koreksi jika melintasi tengah malam
    }
}

```

```

// Simulasikan jarak yang ditempuh
odometerValue += (speedRead * timeElapsed / 60.0);

// Tampilkan nilai odometer
Serial.print("Odometer: ");
Serial.println(odometerValue);
HMISerial.print("odo.val=");
HMISerial.print(abs(odometerValue));
HMISerial.write(0xff);
HMISerial.write(0xff);
HMISerial.write(0xff);

// Simpan nilai odometer
previousMinute = now.minute();

} else if (canMsg.can_id == 0x80001B26) {
// Voltage code
long decimalValueVoltage = convertHexToDecimalVoltage(canMsg.data);
voltageIn = decimalValueVoltage / 10;
Serial.print("Voltage (V): ");
Serial.println(abs(voltageIn));
HMISerial.print("voltage.val=");
HMISerial.print(abs(voltageIn));
HMISerial.write(0xff);
HMISerial.write(0xff);
HMISerial.write(0xff);

// battery code
float persentase = ((voltageIn - voltaseMinimal) / (voltaseMaksimal -
voltageMinimal)) * 100;

HMISerial.print("persen.val=");
HMISerial.print("j0.val=");
HMISerial.print(abs(persentase));
HMISerial.write(0xff);
HMISerial.write(0xff);
HMISerial.write(0xff);

// Sending data to HC-05
String output = String(odometerValue) + ',' + String(tempValue) + ',' +
String(speedRead) + ',' + String(total_jarak) + ',' + String(kecepatan_rata_rata)
+ ',' + String(kecepatan_tertinggi);
BTSerial.println(output);

```

```

    }
}
}
long convertHexToDecimalSpeed(byte hexData[]) {
    String hexString = "";
    for (int i = 0; i < 4; i++) {
        hexString += String(hexData[i], HEX);
    }
    return strtol(hexString.c_str(), NULL, 16);
}

long convertHexToDecimalTemp(byte hexData[]) {
    String hexString = "";
    for (int i = 0; i < 2; i++) {
        hexString += String(hexData[i], HEX);
    }
    return strtol(hexString.c_str(), NULL, 16);
}

long convertHexToDecimalVoltage(byte hexData[]) {
    String hexString = "";
    for (int i = 4; i < 6; i++) {
        hexString += String(hexData[i], HEX);
    }
    return strtol(hexString.c_str(), NULL, 16);
}

```

Code Transmitter

```

#include <CAN.h>

#define sensorPin 3
#define enaKa 4
#define in1Ka 5
#define in2Ka 6
#define enaKi 9
#define in1Ki 8
#define in2Ki 7
#define potensio A0

volatile unsigned long gerigiCount = 0;
unsigned long previousTime = 0;
unsigned long currentTime = 0;
unsigned long previousPrintTime = 0;

```

```

const int jumlahLubangGerigi = 20;
const float diameterRodaMM = 66.0;
const int OFFSET1 = 5;
const int OFFSET2 = -30;

float totalJarakTempuhM = 0.0; // Menggunakan meter
unsigned long totalElapsedTime = 0;
const unsigned long printInterval = 1000;
float topSpeedKMH = 0.0;

void setup() {
  pinMode(enaKa, OUTPUT);
  pinMode(in1Ka, OUTPUT);
  pinMode(in2Ka, OUTPUT);
  pinMode(enaKi, OUTPUT);
  pinMode(in1Ki, OUTPUT);
  pinMode(in2Ki, OUTPUT);
  digitalWrite(enaKa, LOW);
  digitalWrite(in1Ka, LOW);
  digitalWrite(in2Ka, LOW);
  digitalWrite(enaKi, LOW);
  digitalWrite(in1Ki, LOW);
  digitalWrite(in2Ki, LOW);
  Serial.begin(115200);
  if (!CAN.begin(500E3)) {
    Serial.println("Starting CAN failed!");
    while (1);
  }
  pinMode(sensorPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(sensorPin), gerigiDetected, FALLING);
  previousTime = millis();
  previousPrintTime = millis();
}

void loop() {
  int potValue = analogRead(potensio);

  // Mengkonversi nilai potensiometer (0-1023) ke nilai PWM (0-255)
  int pwmOutputKa = map(potValue, 0, 1023, 0, 255) + OFFSET1;
  int pwmOutputKi = map(potValue, 0, 1023, 0, 255) + OFFSET2; // Tambahkan offset
  untuk kalibrasi

  // Mengeluarkan sinyal PWM ke motor

```

```

    analogWrite(enaKa, constrain(pwmOutputKa, 0, 255)); // Pastikan nilai PWM tidak
melebihi batas
    analogWrite(enaKi, constrain(pwmOutputKi, 0, 255));

    // Mengatur arah motor
    digitalWrite(in1Ka, HIGH);
    digitalWrite(in2Ka, LOW);
    digitalWrite(in1Ki, HIGH);
    digitalWrite(in2Ki, LOW);

    currentTime = millis();
    unsigned long elapsedTime = currentTime - previousTime;
    if (currentTime - previousPrintTime >= printInterval) {
        float kelilingRodaMM = 3.14159 * diameterRodaMM;
        float jarakTempuhMM = (gerigiCount / (float)jumlahLubangGerigi) *
kelilingRodaMM;

        float jarakTempuhM = jarakTempuhMM / 1000.0; // Mengubah dari mm ke m

        totalJarakTempuhM += jarakTempuhM;
        totalElapsedTime += elapsedTime;

        float kecepatanKMH = (jarakTempuhM / (elapsedTime / 3600000.0)); // Mengubah
ms ke jam

        if (kecepatanKMH > topSpeedKMH) { topSpeedKMH = kecepatanKMH; }

        float kecepatanRataRataKMH = (totalJarakTempuhM / (totalElapsedTime /
3600000.0));

        // Menghitung RPM
        float RPM = (gerigiCount / (float)jumlahLubangGerigi) * (60000.0 /
elapsedTime); // Menghitung RPM
        int RPMAdjusted = map(RPM, 0, 1000, 600, 6000); // Menyesuaikan RPM ke
rentang 600 - 6000

        Serial.println("=====");
        Serial.println("Jumlah Gerigi: " + String(gerigiCount));
        Serial.println("Jarak Tempuh (m): " + String(totalJarakTempuhM, 1)); //
Output jarak tempuh dalam meter
        Serial.println("Kecepatan (km/jam): " + String(kecepatanKMH, 1));
        Serial.println("Kecepatan Rata-rata (km/jam): " +
String(kecepatanRataRataKMH, 1));
        Serial.println("Top Speed (km/jam): " + String(topSpeedKMH, 1));
        Serial.println("RPM: " + String(RPMAdjusted)); //

```

```

    Serial.println("=====");

    String message = String(totalJarakTempuhM, 1) + "," + String(kecepatanKMH, 1)
+ "," + String(kecepatanRataRataKMH, 1) + "," + String(topSpeedKMH, 1) + "," +
String(RPMAdjusted) + "\n";
    sendCANMessage(message);

    gerigiCount = 0;
    previousTime = currentTime;
    previousPrintTime = currentTime;
}
}

void gerigiDetected() {
    gerigiCount++;
}

void sendCANMessage(String message) {
    int len = message.length();
    int index = 0;
    int canId = 0x123;
    while (index < len) {
        CAN.beginPacket(canId);
        for (int i = 0; i < 8 && index < len; i++) {
            CAN.write(message[index++]);
        }
        CAN.endPacket();
        delay(10);
    }
    Serial.println("Message Sent: " + message);
}

```

Code Receiver

```

#include <CAN.h>
#include <SoftwareSerial.h>
#include <RTClib.h>

SoftwareSerial HMISerial(3, 4); // RX, TX for HMI
SoftwareSerial BTSerial(9, 8); // RX, TX for HC-05
RTC_DS3231 rtc;
String receivedData = "";
String masterData = "";
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

```

```

char buffStartTime[10];
char buffCurrentTime[10];

void setup() {
  Serial.begin(115200);
  HMISerial.begin(115200);
  BTSerial.begin(9600);
  while (!Serial);
  if (!CAN.begin(500E3)) {
    Serial.println("Starting CAN failed!");
    while (1);
  }
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    Serial.flush();
    while (1) delay(10);
  }
  if (rtc.lostPower()) {
    Serial.println("RTC lost power, let's set the time!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    //rtc.adjust(DateTime(2024, 6, 24, 23, 18, 30));
  }
  DateTime now = rtc.now();
  sprintf(buffStartTime, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
  CAN.onReceive(onReceive);
}

void loop() {
  DateTime now = rtc.now();
  sprintf(buffCurrentTime, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
  Serial.println("=====");
  Serial.println("Waktu Mulai      : " + String(buffStartTime));
  Serial.println("Waktu saat ini   : " + String(buffCurrentTime));
  Serial.println("Temperature      : " + String(rtc.getTemperature()) + " C");
  Serial.println("Total jarak      : " + String(getValue(masterData, ',', 0)) +
"m");
  Serial.println("kecepatan tertinggi : " + String(getValue(masterData, ',', 1))
+ "Km/h");
  Serial.println("kecepatan rata-rata : " + String(getValue(masterData, ',', 2))
+ "Km/h");
  Serial.println("maximum speed     : " + String(getValue(masterData, ',', 3)) +
"Km/h");
}

```

```

    Serial.println("RPM          : " + String(getValue(masterData, ',', 4))); //
Display RPM
    Serial.println("=====");
    HMISerial.print("speed.val="); HMISerial.print(getValue(masterData, ',',
1).toInt()); kirim();
    HMISerial.print("temp.val="); HMISerial.print(int(rtc.getTemperature()));
kirim();
    HMISerial.print("t22.txt=\"); HMISerial.print(String(buffCurrentTime));
HMISerial.print("\"); kirim();
    HMISerial.print("odo.val="); HMISerial.print(getValue(masterData, ',',
0).toInt()); kirim();
    HMISerial.print("voltage.val="); HMISerial.print("5"); kirim();
    HMISerial.print("persen.val="); HMISerial.print("100"); kirim();

// Sending data to HC-05
String output = String(getValue(masterData, ',', 1)) + ',' +
                String(getValue(masterData, ',', 0)) + ',' +
                String(getValue(masterData, ',', 2)) + ',' +
                String(getValue(masterData, ',', 3)) + ',' +
                String(getValue(masterData, ',', 4));
BTSerial.println(output);

delay(1000);
}

void onReceive(int packetSize) {
    if (packetSize) {
        int canId = CAN.packetId();
        while (CAN.available()) {
            receivedData += (char)CAN.read();
        }
        if (receivedData.length() > 0 && receivedData.endsWith("\n")) {
            Serial.print("Received CAN Message, ID = 0x");
            Serial.print(canId, HEX);
            Serial.print(", Data: ");
            Serial.println(receivedData);
            masterData = receivedData;
            receivedData = "";
        }
    }
}

String getValue(String data, char separator, int index) {
    int found = 0;
    int strIndex[] = { 0, -1 };

```

```

int maxIndex = data.length() - 1;
for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
        found++;
        strIndex[0] = strIndex[1] + 1;
        strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void kirim() {
    HMISerial.write(0xff);
    HMISerial.write(0xff);
    HMISerial.write(0xff);
}

```