RANCANG BANGUN PROTOTYPE CRUISE CONTROL PADA KENDARAAN LISTRIK DENGAN METODE KENDALI PID

LAPORAN PENELITIAN

Disusun oleh:
Ir. Edy Supriyadi, MT.
Alwi Hamzah, ST.

LEMBAGA PENELITIAN DAN PENGABDIAN MASYARAKAT
INSTITUT SAINS DAN TEKNOLOGI NASIONAL
JAKARTA
2023

LEMBAR IDENTITAS DAN PENGESAHAN
LAPORAN AKHIR HASIL PENELITIAN

1.	Judul Penelitian	:	Rancang Bangun Prototype Cruise Control Pada Kendaraan Listrik Dengan Metode Kendali PID
2.	Bidang Ilmu	:	Teknik Elektro/Sistem Kendali
3.	Jumlah Tim Peneliti	:	2 Orang
4.	Ketua Peneliti		
	a Nama (lengkap dengan gelar)	:	Ir. Edy Supriyadi, MT
	b NIDN	:	0319106301
	c Jenis Kelamin	:	Laki-laki
	d Jabatan Akademik	:	Lektor Kepala
	e Fakultas	:	Fakultas Teknologi Industri
	f Program Studi	:	Teknik Elektro
	g Bidang Keahlian	:	Teknik Kendali
5.	Anggota Peneliti		
	a Nama (lengkap dengan gelar)	:	Alwi Hamzah, ST.
	b Jenis Kelamin	:	Laki-laki
	c Fakultas	:	Fakultas Teknologi Industri
	d Program Studi	:	Teknik Elektro
	e Perguruan Tinggi	:	Institut Sains dan Teknologi Nasional
	f Bidang Keahlian	:	Sistem Kendali
6.	Lokasi Penelitian	:	Institut Sains dan Teknologi Nasional
7.	Jangka Waktu Penelitian	:	6 Bulan
8.	Biaya Penelitian	:	**Rp 2,750,000.00**

Jakarta, 09 Februari 2023

Mengetahui,
Dekan Fakultas Teknologi Industri

Dr. Musfirah Cahya Fajrah, S. Si., M.Si
NIDN : 0004067109

Ketua Peneliti

Ir. Edy Supriyadi, MT.
NIDN : 0319106301

Disetujui Oleh,
Kepala Lembaga Penelitian dan Pengabdian Masyarakat

Ir. Syahril Taufik, M. Sc. Eng., Ph. D
NIDN : .....

PRAKATA

Puji dan syukur kami panjatkan kehadirat Allah SWT, yang telah melimpahkan berkat, rahmat dan hidayah-Nya, sehingga penelitian ini dapat berjalan dengan baik, lancar dan selesai tepat pada waktu yang telah ditentukan.

Laporan penelitian ini merupakan salah salah satu tugas dari tenaga pendidikan dalam melaksanakan Tridharma Perguruan Tinggi dan suatu kewajiban yang harus dilaksanakan sebagai dosen lingkungan program studi Teknik Elektro Fakultas Teknologi Industri ISTN. Penelitian ini dilaksanakan dikawasan kampus ISTN, untuk semester ganjil 2022-2023 mulai dari bulan September 2022 sampai dengan bulan Februari 2023 dengan judul "Rancang Bangun Prototype Cruise Control Pada Kendaraan Listrik Dengan Metode Kendali PID".

Pada kesempatan ini para peneliti mengucapkan terima kasih kepada tim peneliti yang telah membantu dalam pelaksanaan dilapangan, fasilitas, proses data maupun dana sehingga penelitian ini dapat dilaksanakan dengan baik.

Semoga hasil penelitian ini berguna untuk kemajuan ilmu dan teknologi komunikasi data dan juga komunikasi bidang teknik elektro

Jakarta, 09 Februari 2023

Peneliti

ABSTRAK

Sistem kendali Cruise Control pada kendaraan listrik dengan metode kendali PID dalam prototype ini, menambahkan fitur manuver dan menggunakan input sensor jarak Ultrasonik dan Tof (Time of Flight) untuk membaca jarak dengan kendaraan terhadap objek. Tidak hanya diletakkan didepan, sensor ini juga akan diletakkan di sebelah kanan dan kiri mobil untuk mendeteksi letak pembatas jalan supaya manuver yang akan dilakukan tidak menyebabkan kendaraan menabrak pembatas. Untuk mendapatkan jarak stabil, sistem kendali PID akan digunakan untuk menstabilkan kecepatan mobil sesuai setpoint dari jarak yang diinginkan. Jika objek didepan sudah kurang dari batas setpoint, maka kecepatan akan dikurangi secara proporsional. Namun jika objek terlalu dekat (jarak sangat kurang dari setpoint) maka manuver akan dilakukan. Hal ini bertujuan untuk menghindari tabrakan dikarenakan rem yang kurang cepat (pakem). Mikrokontroler akan membaca sensor jarak yang terdapat pada sisi kiri dan kanan dan memilih jarak terbesar yang berarti sisi tersebut bebas pembatas atau hambatan. ESP32 wemos lolin32 lite akan memproses sistem kendali PID untuk mencapai setpoint yang diinginkan.

Kata Kunci : *Cruise control, PID, ESP32. Ultrasonik.*

ABSTRACT

The Cruise Control control system on electric vehicles with the PID control method in this prototype, adds maneuver features and uses Ultrasonic and Tof (Time of Flight) sensor inputs to read the distance from the vehicle to the object. Not only placed in front, this sensor will also be placed on the right and left of the car to detect the location of the road divider so that the maneuvers that will be carried out do not cause the vehicle to crash into the barrier. To get a stable distance, the PID control system will be used to stabilize the car's speed according to the setpoint of the desired distance. If the object in front is less than the setpoint limit, then the speed will be reduced proportionally. However, if the object is too close (distance is very less than the setpoint) then the maneuver will be performed. This aims to avoid collisions due to brakes that are not fast enough (grip). The microcontroller will read the proximity sensors on the left and right and choose the largest distance, which means that the side is free of barriers or obstacles. ESP32 wemos lolin32 lite will process the PID control system to reach the desired setpoint.

Keywords : Cruise control, PID, ESP32, Ultrasonic.

DAFTAR ISI

LEMBAR IDENTITAS DAN PENGESAHAN I

PRAKATA II

ABSTRAK III

ABSTRACT IV

DAFTAR ISI V

DAFTAR GAMBAR VIII

DAFTAR TABEL XI

BAB I PENDAHULUAN 1

1.1. LATAR BELAKANG 1
1.2. TUJUAN PENELITIAN 2
1.3. RUMUSAN MASALAH 2
1.4. BATASAN MASALAH 2
1.5. METODE PENELITIAN 2
1.6. SISTEMATIKA PENELITIAN 3

BAB II TINJAUAN PUSTAKA 5

2.1. DESKRIPSI SINGKAT 5
2.2. MIKROKONTROLER 6
2.2.1. ESP32 WeMos LoLin32 Lite 7
2.3. PID (PROPORTIONAL–INTEGRAL–DERIVATIVE CONTROLLER) 10
2.4. CRUISE CONTROL: PID CONTROLLER DESIGN 12
2.5. PULSE WIDTH MODULATION 18
2.6. SENSOR ULTRASONIK HC-SR04 19
2.7. TIME OF FLIGHT SENSOR 20
2.8. DRIVER MOTOR L298N 21
2.8.1. Tabel Logika Driver Motor L298N 22
2.9. MOTOR DC 23

2.9.1. Simbol Motor DC *24*
2.9.2. Bagian Atau Komponen Utama Motor DC *24*
2.10. MOTOR SERVO 25
2.10.1. Prinsip Kerja Motor Servo *26*
2.11. BATTERY LITHIUM ION 18650 27
2.12. STEP DOWN DC LM2596 27
2.13. APLIKASI BLYNK 28
BAB III PERANCANGAN ALAT 30
3.1. DIAGRAM ALUR KERJA 30
3.2. DIAGRAM BLOK 31
3.3. BLOK SISTEM KENDALI PID 33
3.4. FLOWCHART 33
3.5. TATA LETAK KOMPONEN 37
3.6. PERANCANGAN APLIKASI BLYNK IOT 39
3.7. PERANCANGAN PERANGKAT KERAS 40
3.8. RANGKAIAN SISTEM ESP32 WEMOS LOLIN32 LITE 41
3.8.1. Rangkaian ESP32 WeMos LoLin32 Lite dengan Sensor Ultrasonik *43*
3.8.2. Rangkaian ESP32 WeMos LoLin32 Lite dengan Sensor ToF *43*
3.8.3. Rangkaian ESP32 WeMos LoLin32 Lite dengan Driver Motor dan Motor DC *44*
3.8.4. Rangkaian ESP32 WeMos LoLin32 Lite dengan Motor Servo *45*
3.8.5. Rangkaian ESP32 WeMos LoLin32 Lite dengan Aplikasi Blynk IoT *45*
3.8.6. Rangkaian ESP32 WeMos LoLin32 Lite dengan Lampu Depan *46*
3.8.7. Rangkaian ESP32 WeMos LoLin32 Lite dengan Lampu Belakang .. *46*
BAB IV PENGUJIAN DAN ANALISA 47
4.1. PENGUJIAN CATU DAYA 47
4.1.1. Pengujian Tanpa Beban *47*
4.1.2. Pengujian Dengan Beban *48*
4.2. PENGUJIAN SENSOR ULTRASONIK HC-SR04 DAN SENSOR TIME OF FLIGHT VL53L0X 50
4.3. PENGUJIAN MOTOR SERVO 52

4.4. PENGUJIAN JAGA JARAK DENGAN PID 54
4.5. PENGUJIAN MANUVER 54
4.6. PENGUJIAN PWM SEBAGAI PENGATUR KECEPATAN MOTOR DC 55
4.7. MENENTUKAN NILAI PID DENGAN TUNING KONTROL MATLAB 56
4.7.1. Menentukan Nilai Kp 57
4.7.2. Menentukan Nilai Kp dan Ki 58
4.7.3. Menentukan Nilai Kp, Ki dan Kd 59
4.8. PENGUJIAN KESTABILAN KECEPATAN MOTOR DC 59
BAB V KESIMPULAN 61
DAFTAR PUSTAKA 62
LAMPIRAN 64

DAFTAR GAMBAR

Gambar 2. 1 Blok diagram sistem kontrol 5
Gambar 2. 2 Blok diagram sistem kontrol loop terbuka 6
Gambar 2. 3 Blok diagram sistem kontrol loop tertutup..... 6
Gambar 2. 4 Skema pinout wemos lolin32 lite 8
Gambar 2. 5 Blok diagram PID..... 10
Gambar 2. 6 Diagram Blok Sistem PID..... 12
Gambar 2. 7 Grafik Respon Kendali Proposional Dengan Nilai Kp = 100 Menggunakan PID Tuner Matlab..... 15
Gambar 2. 8 Grafik Respon Kendali Proposional Dengan Nilai Kp = 5000 Menggunakan PID Tuner Matlab..... 16
Gambar 2. 9 Bentuk Pulsa PWM 18
Gambar 2. 10 Sensor ulltrasonik..... 20
Gambar 2. 11 Sensor time of flight..... 20
Gambar 2. 12 Driver motor L298N..... 22
Gambar 2. 13 Motor DC 24
Gambar 2. 14 Simbol Motor DC..... 24
Gambar 2. 15 Bagian-bagian motor DC 25
Gambar 2. 16 Motor Servo..... 26
Gambar 2. 17 Pulse wide modulation servo..... 26
Gambar 2. 18 Battery lithium ion 18650 27
Gambar 2. 19 Step down dc LM2596 28
Gambar 2. 20 Skema antarmuka blynk 29

gambar 3. 1 Diagram Alur Kerja..... 30
Gambar 3. 2 Diagram Blok 31
Gambar 3. 3 Blok sistem kendali PID..... 33
gambar 3. 4 Flowchart keseluruhan kerja alat 34
gambar 3. 5 Flowchart input navigasi 35
gambar 3. 6 Flowchart perhitungan PID 36
gambar 3. 7 Tata letak komponen 38
gambar 3. 8 Tampilan aplikasi blynk..... 39

gambar 3. 9 Perancangan skematik perangkat keras alat 41
gambar 3. 10 Rangkaian yang menghubungkan esp 32 wemos lolin32 lite dengan sensor ultrasonik..... 43
gambar 3. 11 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan sensor ToF 44
gambar 3. 12 Rangkaian yang menghubungkan esp32 wemos lolin32 lite dengan driver motor dan motor dc..... 44
gambar 3. 13 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan motor servo..... 45
gambar 3. 14 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan blynk IoT 45
gambar 3. 15 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan lampu depan 46
gambar 3. 16 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan lampu belakang 46

Gambar 4. 1 Rangkaian Pengukuran Tegangan Tanpa Beban Menggunakan Multimeter……..……47
Gambar 4. 2 (a) Pengujian Output Catu Daya 5VDC Tanpa Beban, dan (b) Pengujian Output Catu Daya 8.4VDC Tanpa Beban 48
Gambar 4. 3 Rangkaian Pengukuran Tegangan Dengan Beban Menggunakan Multimeter..... 48
Gambar 4. 4 (a) Pengujian Output Catu Daya 5VDC Dengan Beban, dan (b) Pengujian Output Catu Daya 8.4VDC Dengan Beban,..... 49
Gambar 4. 5 Pengujian sensor Ultrasonik dan ToF Menggunakan Penggaris dengan Jarak 20 cm 51
Gambar 4. 6 Data Jarak Sensor Ultrasonik dan ToF yang terbaca dengan Jarak 20 cm 51
Gambar 4. 7 a) Sudut belok kiri motor servo b) Sudut belok kanan motor servo 53
Gambar 4. 8 a) Sudut servo yang terbaca saat belok kanan b) Sudut servo yang terbaca saat belok kiri..... 53
Gambar 4. 9 KP = 1 57
Gambar 4. 10 KP = 2 57

Gambar 4. 11 KP = 3 57
Gambar 4. 12 KP = 2 dan KI = 0.5 58
Gambar 4. 13 KP = 2 dan KI = 1 58
Gambar 4. 14 KP = 2 KI = 0.5 KD = 0.2 59
Gambar 4. 15 Grafik RPM Terhadap Jarak 60

DAFTAR TABEL

Tabel 2. 1 Spesifikasi wemos lolin32 lite 8
Tabel 2. 2 Logika input motor driver L298N..... 23

Tabel 3. 1 Keterangan fitur aplikasi blynk..... 40
Tabel 3. 2 Koneksi pin ESP32 wemos lolin32 lite..... 42

Tabel 4. 1 Hasil Rata-Rata Pengujian Tegangan Catu Daya 5 VDC 49
Tabel 4. 2 Hasil Rata-Rata Pengujian Tegangan Catu Daya 8,4 VDC 50
Tabel 4. 3 Pengukuran sensor ultrasonik dan ToF 52
Tabel 4. 4 Pengujian Jaga Jarak dengan PID 54
Tabel 4. 5 Pengujian Manuver Mobil 55
Tabel 4. 6 Pengujian PWM Pada Motor DC..... 56
Tabel 4. 7 Pengujian RPM Motor DC..... 60

BAB I
PENDAHULUAN

1.1. Latar Belakang

Kasus kecelakaan lalu lintas dalam berkendara masih menjadi masalah serius di Indonesia, kasus ini paling banyak disebabkan karena kelalaian manusia (human error) seperti hilang konsentrasi, mengantuk, dan kelelahan lalu di dukung dengan perilaku berkendara dengan kecepatan tinggi yang tidak terkontrol.

Hal ini menjadikan beberapa perusahaan mobil mengeluarkan teknologi cruise control sebagai upaya mengatasi kelalaian manusia dalam berkendara yang menjadi penyebab utama terjadinya kecelakaan.

Cruise control adalah fitur teknologi pada mobil untuk mengatur kecepatan mobil secara konstan. Namun cruise control umumnya terfokus pada penstabilan kecepatan dan menjaga jarak dengan objek yang di depan. Apabila terdapat kasus rem mendadak, kendaraan akan bergantung sepenuhnya kepada kecepatan pengereman yang tidak setiap saat akan berhenti sebelum menabrak objek didepannya. Pada penelitian sebelumnya, untuk rancang bangun sistem konvoi atau *robot follower* dimana mobil dapat mengikuti objek yang ada di depan (Sirojul 2017), namun dari hasil pengujian penelitian tersebut masih belum maksimal pada manuver mobil dan kecepatan motor masih stagnan. Karena itu perlu dikembangkan fitur lain pada sistem cruise control seperti manuver secara cepat untuk menghidar dari objek yang ada di depan dan kecepatan mobil yang lebih terkendali.

Fitur manuver yang dirancang akan menggunakan input sensor jarak untuk membaca jarak dengan kendaraan terhadap objek. Tidak hanya diletakkan di depan, sensor ini juga akan diletakkan di sebelah kanan dan kiri mobil untuk mendeteksi kendaraan atau pembatas jalan supaya manuver yang akan dilakukan tidak menyebabkan kendaraan menabrak pembatas.

Dengan rancangan prototype cruise control pada kendaraan listrik dengan metode kendali PID masalah tersebut dapat teratasi sebagai upaya untuk

mengurangi angka kecelakaan dalam berkendara, sehingga pengendara mobil bisa lebih safety dalam berkendara di jalan raya.

1.2. Tujuan Penelitian

Dalam bahasan materi ini akan merancang Prototype Cruise Control Pada Kendaraan Listrik Dengan Metode PID. Dimana tujuan dari prototype sistem ini sebagai upaya untuk meminimalisir angka kecelakaan dalam berkendara sekaligus mengembangkan dari penelitian sebelumnya.

1.3. Rumusan Masalah

Rumusan masalah yang diangkat dalam Penelitian ini adalah sebagai berikut:

1. Bagaimana cara prototype sistem mobil dapat menjaga jarak dengan objek di depannya tanpa dikendalikan oleh pengendara
2. Bagaimana cara prototype sistem mobil melakukan manuver menghindari tabrakan ke kiri atau kanan, saat mobil sudah terlalu dekat dengan objek yang ada di depan
3. Bagaimana sistem kerja alat prototype cruise control pada kendaraan listrik dengan metode kendali PID secara menyeluruh

1.4. Batasan Masalah

Batasan masalah ini dibuat bertujuan agar pembahasan penelitian tidak menyimpang dari yang dirumuskan pada penelitian ini. Adapun batasan masalahnya yaitu:

1. Menggunakan metode kendali PID sebagai perhitungan kecepatan untuk jaga jarak mobil dengan objek.
2. Mikrokontroler yang digunakan yaitu ESP32 WeMos LoLin32 Lite.
3. Manuver pada mobil hanya untuk menghindar dari tabrakan.
4. Perancangan sistem pada alat ini hanya dapat di gunakan pada jalanan lurus tanpa tikungan.
5. Kontrol untuk mengoperasikan keseluruhan prototype sistem ini menggunakan aplikasi blynk iot.

1.5. Metode Penelitian

Untuk menyelesaikan penelitian ini, dilakukan langkah-langkah sebagai berikut:

1. Penentuan judul
2. Studi kepustakaan

   Studi kepustakaan ini dilakukan untuk menambah pengetahuan penulis dan untuk mencari referensi bahan dengan membaca literature maupun bahan – bahan teori baik berupa buku dan data dari internet yang dapat menunjang pembuatan skripsi serta penulisan laporan skripsi.
3. Perencanaan perangkat keras

   Berisikan desain alat yang akan digunakan pembuatan "Alat prototype cruise control pada kendaraan listrik dengan metode kendali PID"
4. Melakukan survey komponen

   Berupa pengetesan komponen yang akan digunakan.
5. Pembuatan alat

   Berisikan proses pembuatan alat yang bisa bekerja.
6. Melakukan pengujian alat

   Berisikan tentang pengujian alat yang telah selesai dibuat apakah sudah beroperasi sesuai rencana atau belum.
7. Penyempurnaan alat

   Berisikan tentang penyempurnaan alat jika masih belum dapat beroperasi dengan baik.
8. Menyusun buku laporan

   Berisikan penyusunan hasil dari penelitian dalam bentuk laporan penelitian.

1.6. Sistematika Penelitian

Sistematika penulisan digunakan untuk memudahkan penulis dalam menyusun laporan penelitian dan memudahkan pembaca untuk mengikuti alur laporan penelitian ini, maka sistematika penulisannya adalah sebagai berikut:

BAB I PENDAHULUAN Berisi latar belakang, tujuan penulisan, rumusan masalah, batasan masalah, metodologi penelitian, serta sistematika penelitian yang memuat susunan penulisan penelitian ini. **BAB II TEORI DASAR** Dasar Teori, bab ini menguraikan teori - teori yang mendukung dari penelitian, yang digunakan pada proses perancangan dan realisasi dari penelitian ini. **BAB III PERENCANAAN DAN PERANCANGAN** Perancangan Alat, disini penulis

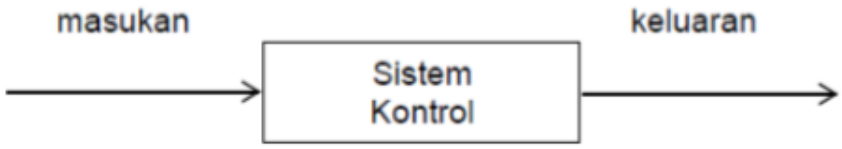
menjabarkan konsep perancangan alat prototype cruise control pada kendaraan listrik dengan metode kendali PID. **BAB IV PENGUJIAN DAN ANALISA** Pengujian Dan Analisis Alat, dalam rancang bangun alat prototype cruise control pada kendaraan listrik dengan metode kendali PID. **BAB V KESIMPULAN** Berisi kesimpulan untuk melengkapi serta menambahkan rekayasa secara lebih lanjut mengenai penelitian yang terjadi. **DAFTAR PUSTAKA** Daftar Pustaka, berisi referensi – referensi acuan sumber data untuk acuan Penelitian. **LAMPIRAN** Lampiran, terdiri dari data – data yang dilampirkan

BAB II
TINJAUAN PUSTAKA

2.1. Deskripsi singkat

Sistem kontrol (control system) merupakan suatu kumpulan cara atau metode yang dipelajari dari kebiasaan-kebiasaan manusia dalam bekerja, dimana manusia membutuhkan suatu pengamatan kualitas dari apa yang telah mereka kerjakan sehingga memiliki karakteristik sesuai dengan yang diharapkan pada mulanya. Perkembangan teknologi menyebabkan manusia selalu terus belajar untuk mengembangkan dan mengoperasikan pekerjaan-pekerjaan kontrol yang semula dilakukan oleh manusia menjadi serba otomatis (Irfan Permana 2013).

Dalam aplikasinya, suatu sistem kontrol memiliki tujuan/sasaran tertentu. Sasaran sistem kontrol adalah untuk mengatur keluaran (output) dalam suatu sikap / kondisi / keadaan yang telah ditetapkan oleh masukan (input) melalui elemen sistem kontrol.



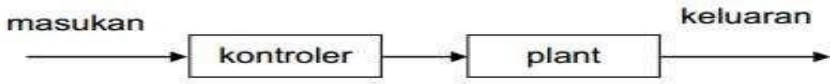
Gambar 2. 1 Blok diagram sistem kontrol

Dengan adanya sasaran ini, maka kualitas keluaran yang dihasilkan tergantung dari proses yang dilakukan dalam sistem kontrol ini.

Pada pembahasan skripsi ini sistem kontrol yang digunakan yaitu meliputi sistem menjaga kesetabilan jarak dengan metode kendali PID dan manuver mobil menggunakan sensor ultrasonik dan ToF (Time of Flight) dengan mikrokontroler ESP 32.

- **Sistem Kendali Loop Terbuka**

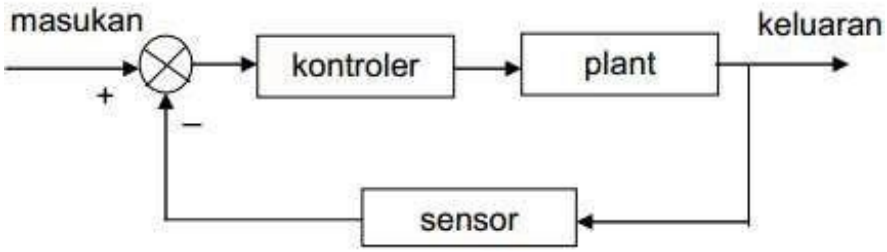
Sistem kendali Loop Terbuka adalah Sistem Kendali yang kinerjanya tidak berpengaruh terhadap keluarannya atau tidak berpengaruh terhadap umpan balik dari prosesnya. Sistem Kendali Loop Terbuka Menggunakan peralatan penggerak untuk mengontrol proses secara langsung.



Gambar 2. 2 Blok diagram sistem kontrol loop terbuka

- **Sistem Kendali Loop Tetutup**

Sistem Kendali Loop Tertutup adalah Sistem Kendali Yang kinerjanya memiliki pengaruh terhadap keluarannya, dan memiliki umpan balik terhadap proses yang berjalan.



Gambar 2. 3 Blok diagram sistem kontrol loop tertutup

2.2. Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja Mikrokontroler sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Andapun bisa pula menulis hal-hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda dapat membuat program untuk membuat suatu sistem pengaturan otomatik menggunakan microcontroller sesuai keinginan Anda (Syahwil, Muhammad 2013).

Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut "pengendali kecil" dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti *IC TTL* dan *CMOS* dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh Mikrokontroler ini.

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara automatis, seperti sistem kontrol mesin, *remote control*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat input output yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan Mikrokontroler ini maka:

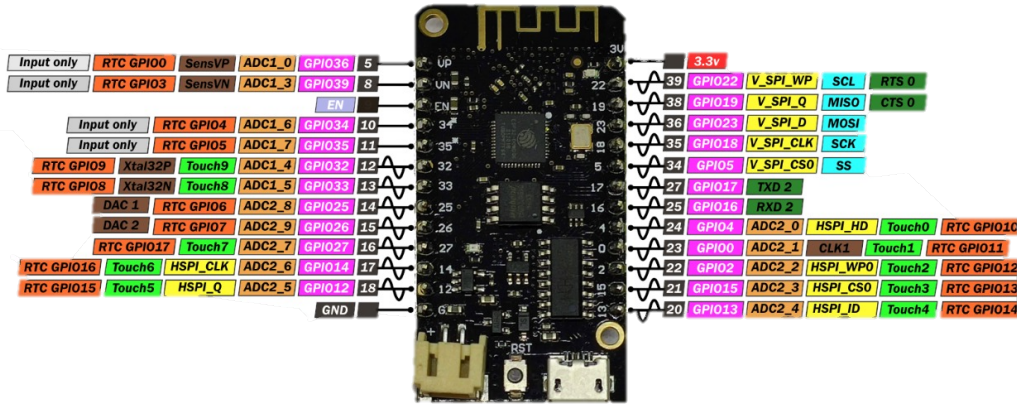
1. Sistem elektronik akan menjadi lebih ringkas
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Agar sebuah mikrokontroler dapat berfungsi, maka Mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem clock dan reset, walaupun pada beberapa Mikrokontroler sudah menyediakan sistem clock internal, sehingga tanpa rangkaian eksternal pun Mikrokontroler sudah beroperasi.

2.2.1. ESP32 WeMos LoLin32 Lite

Mikrokontroler pada dasarnya adalah komputer yang terdapat mikroporsesor, memori, dan perangkat lainya. Fungsi dari mikrokontroler adalah menerima, mengolah dan mengirim sinyal (Wahyuni, 2015). ESP32 adalah mikrokontroler yang menjadi suksesor dari mikrokontroler keluaran Espressif sebelumnya yaitu ESP8266. Mikrokontroler ESP32 juga memiliki beberapa macam versi seperti ESP32 DEV KIT DOIT, ESP32 Lolin32,

Adafruit HUZZAH32, ESP32 Thing. ESP32 LoLin32 Lite sendiri adalah versi mini dari ESP32 Lolin32 dimana ia kehilangan beberapa pin supaya lebih ringkas (Santos, 2021). Tampak mikrokontroler ESP32 Lolin32 Lite dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Skema pinout wemos lolin32 lite

IoT dengan *bluetooth* dan WiFi yang sudah tergabung didalamnya (Muliadi dkk, 2020). ESP32 LoLin32 Lite memiliki *processors* dengan *cores* yang berjumlah 2 yang memungkinkan mikrokontroler bekerja dengan cepat. Mikrokontroler ini menggunakan tegangan 3.3V di setiap pinnya dan 5V pada konektor USB. Informasi detail mengenai spesifikasi ESP32 LoLin32 Lite dapat dilihat pada tabel 2.1.

Tabel 2. 1 Spesifikasi wemos lolin32 lite

No	Spesifikasi	Keterangan
1	Prosesor	Tensilica Xtensa LX6 Dual Core Processor
2	WiFi	WiFi 802.11 b/g/n
3	Bluetooth	Bluetooth LE
4	Baterai	LiPo 500mA
5	Pin	2 x 13 for UART, I2C, SPI, VP / VN, DAC
6	Ukuran keseluruhan	40 x 25 mm

Antarmuka periferal:

- 12-bit SAR ADC hingga 18 saluran
- 2 x 8-bit DAC

- 10 x sensor sentuh (GPIO penginderaan kapasitif)
- 4 x SPI
- 2 x antarmuka I2S
- 2 x antarmuka I2C
- 3 x UART
- Pengontrol host SD/SDIO/CE-ATA/MMC/eMMC
- Pengontrol budak SDIO/SPI
- Antarmuka Ethernet MAC dengan dukungan DMA dan IEEE 1588 Precision Time Protocol khusus
- CAN bus 2.0
- Pengendali jarak jauh inframerah (TX/RX, hingga 8 saluran)
- PWM motor
- PWM LED (hingga 16 saluran)
- Hall effect sensor
- Pre-amplifier analog berdaya sangat rendah

Keamanan:

- Semua fitur keamanan standar IEEE 802.11 didukung, termasuk WFA, WPA/WPA2, dan WAPI
- Boot aman
- Enkripsi kilat
- OTP 1024-bit, hingga 768-bit untuk pelanggan
- Akselerasi perangkat keras kriptografi: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

Manajemen daya:

- Internal low-dropout regulator
- Domain daya individual untuk RTC
- 5μA deep sleep current
- Bangun dari interupsi GPIO, timer, pengukuran ADC, interupsi sensor sentuh kapasitif

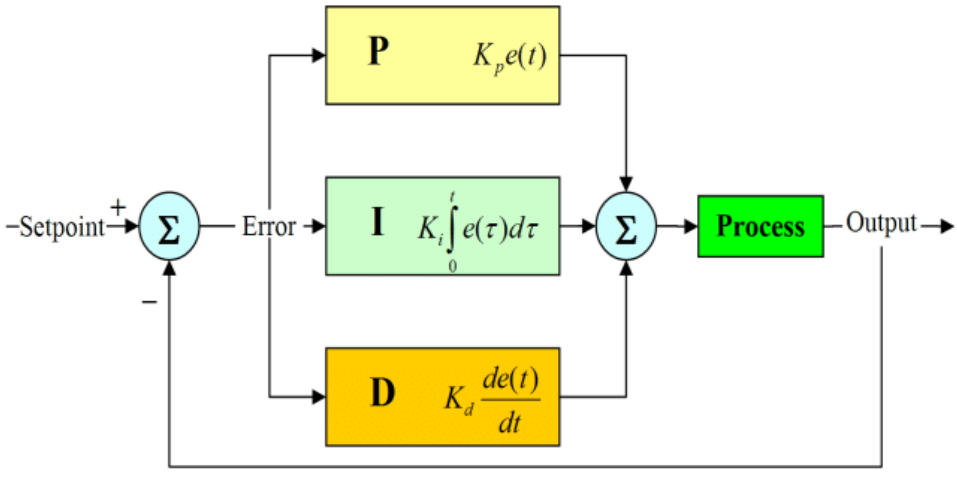
Manajemen baterai

- Konektor untuk baterai 3,7v (seperti 18650).

2.3. PID (Proportional–Integral–Derivative controller)

Intrumentasi dan control industri tentu tidak lepas dari sistem instrumentasi sebagai pengontrol yang digunakan dalam keperluan pabrik. Sistem kontrol pada pabrik tidak lagi manual seperti dahulu, tetapi saat sekarang ini telah dibantu dengan perangkat kontroler sehingga dalam proses produksinya suatu pabrik bisa lebih efisien dan efektif. Kontroler juga berfungsi untuk memastikan bahwa setiap proses produksi terjadi dengan baik (Putra Eka 2013).

PID *(Proportional–Integral–Derivative controller)* merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Pengontrol PID akan memberikan aksi kepada Motor dc berdasarkan besar error yang diperoleh. Motor dc akan menjadi akuator yang mengatur putaran motor dc untuk mencapai set point yang diinginkan. Error adalah perbedaan dari Set Point dengan nilai aktual. PID Blok Diagram dapat dilihat pada gambar dibawah :



Gambar 2. 5 Blok diagram PID

Adapun persamaan Pengontrol PID adalah :

$$mv(t) = K_P\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right) \quad 2.1$$

Keterangan :

mv(t) = output dari pengontrol PID atau *Manipulated Variable*

Kp = konstanta Proporsional

Ti = konstanta Integral

Td = konstanta Detivatif

e(t) = error (selisih antara set point dengan level aktual)

Persamaan Pengontrol PID diatas dapat juga dituliskan sebagai berikut :

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad 2.2$$

dengan :

$$K_i = K_p \times \frac{1}{T_i} \text{ dan } K_d = K_p \times Td \quad 2.3$$

Untuk lebih memaksimalkan kerja pengontrol diperlukan nilai batas minimum dan maksimum yang akan membatasi nilai *Manipulated Variable* yang dihasilkan.

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant.

2.4. Cruise Control: PID Controller Design

Perintah utama MATLAB yang digunakan dalam tutorial ini adalah: tf , step , feedback.

- **Model dan parameter sistem**

Model fungsi transfer untuk masalah cruise control diberikan di bawah ini. Silakan lihat halaman Cruise Control: Pemodelan Sistem untuk turunannya.

$$P(s) = \frac{V(s)}{U(s)} = \frac{1}{ms + b} \qquad [\frac{m/s}{N}] \tag{2.4}$$

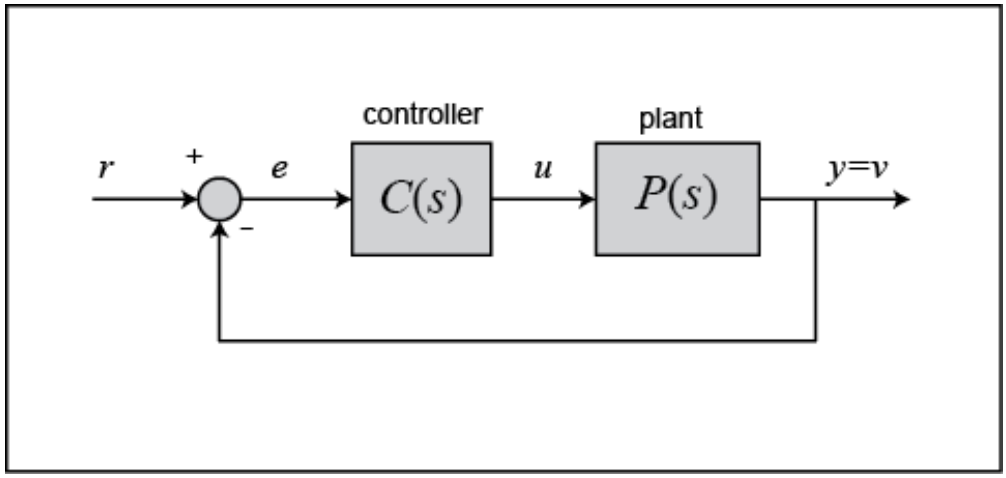
Parameter yang digunakan dalam contoh ini adalah sebagai berikut:

(m)	vehicle mass	1000 kg
(b)	damping coefficient	50 N.s/m
(r)	reference speed	10 m/s

- **Spesifikasi kinerja**
  - Rise time < 5 s
  - Overshoot < 10%
  - Steady-state error < 2%

- **Ringkasan PID**

Diagram blok dari sistem umpan balik kesatuan yang khas ditunjukkan di bawah ini.



Gambar 2. 6 Diagram Blok Sistem PID

Ingat dari Pendahuluan: Halaman Desain Pengontrol PID, fungsi transfer pengontrol PID adalah

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad 2.5$$

Kita dapat mendefinisikan pengontrol PID di MATLAB menggunakan fungsi transfer secara langsung:

```
Kp = 1;

Ki = 1;

Kd = 1;



s = tf('s');
C = Kp + Ki/s + Kd*s
```

```
C =


  s^2 + s + 1

  -----------

       s


Continuous-time transfer function.
```

Atau, kami dapat menggunakan objek pengontrol pid MATLAB untuk menghasilkan pengontrol waktu kontinu yang setara sebagai berikut:

```
C = pid(Kp,Ki,Kd)
```

```
C =


             1

  Kp + Ki * --- + Kd * s
```

```
          s

  with Kp = 1, Ki = 1, Kd = 1

Continuous-time PID controller in parallel form.
```

- **Kontrol proporsional**

Hal pertama yang harus dilakukan dalam soal ini adalah menemukan fungsi transfer loop tertutup dengan kontrol proporsional (C = Kp) ditambahkan. Dengan mengurangi diagram blok umpan balik kesatuan, fungsi transfer loop tertutup dengan pengontrol proporsional menjadi:

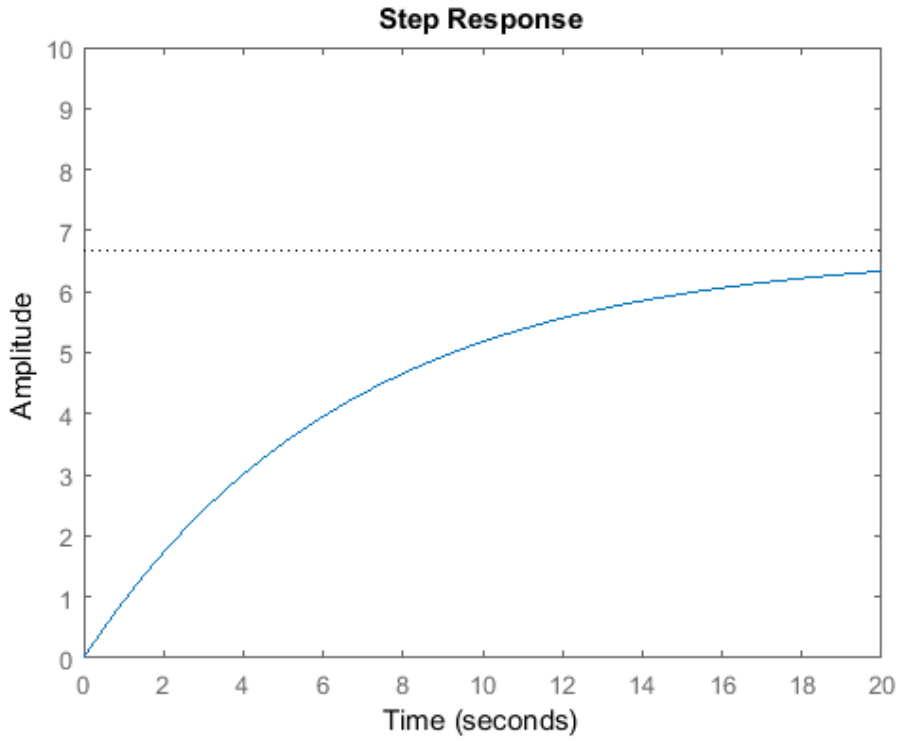
$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{K_p}{ms + b + K_p} \qquad 2.6$$

Ingat dari Pendahuluan: Halaman Desain Pengontrol PID, pengontrol proporsional, Kp , mengurangi waktu naik, yang diinginkan dalam kasus ini.

Untuk saat ini, gunakan Kp sama dengan 100 dan kecepatan referensi 10 m/s. Buat m-file baru dan masukkan perintah berikut.

```
M = 1000;

b = 50;

r = 10;



s = tf('s');
P_cruise = 1/(m*s + b);

Kp = 100;
C = pid(Kp);

T = feedback(C*P_cruise,1)

t = 0:0.1:20;
step(r*T,t)
axis([0 20 0 10])
```

```
T =

      100

  ------------

  1000 s + 150


Continuous-time transfer function.
```



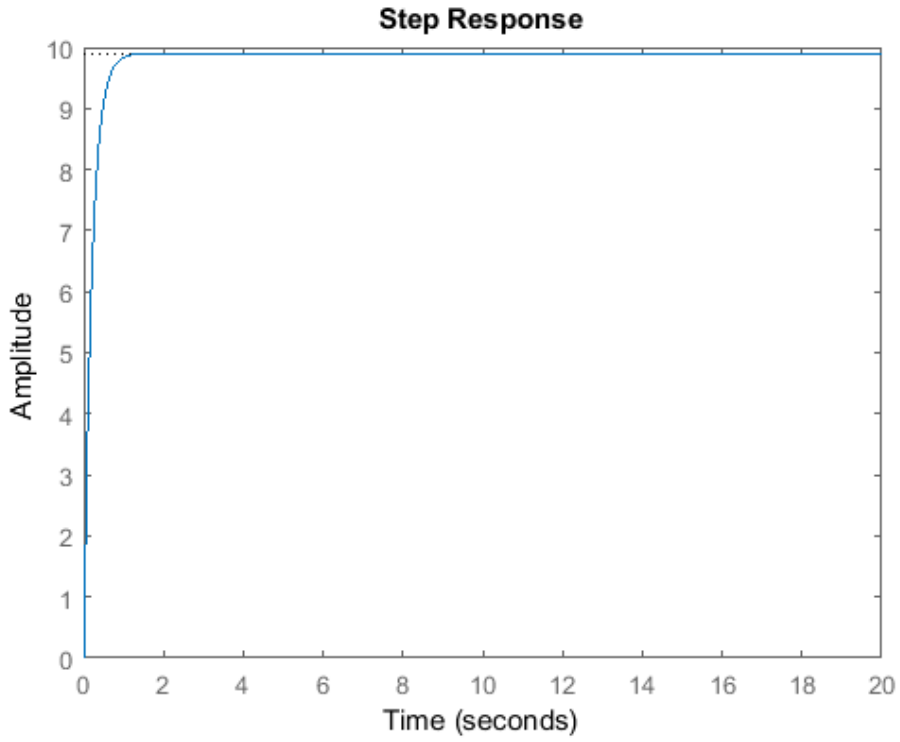
Gambar 2. 7 Grafik Respon Kendali Proposional Dengan Nilai Kp = 100 Menggunakan PID Tuner Matlab

Perhatikan bahwa kami telah menggunakan perintah umpan balik MATLAB untuk menyederhanakan pengurangan diagram blok dari sistem loop tertutup. Harap verifikasi sendiri bahwa hasilnya sesuai dengan fungsi transfer loop tertutup, T, yang diturunkan di atas.Menjalankan m-file di MATLAB akan memberi Anda respons langkah di atas. Seperti yang dapat Anda lihat dari plot,

baik kesalahan kondisi mapan maupun waktu naik tidak memenuhi kriteria desain kami.

Anda dapat meningkatkan keuntungan proporsional, Kp , untuk mengurangi waktu naik dan kesalahan keadaan tunak.
Ubah m-file yang ada sehingga Kp sama dengan 5000 dan jalankan kembali di jendela perintah MATLAB. Anda harus melihat plot berikut.

```
Kp = 5000;

C = pid(Kp);

T = feedback(C*P_cruise,1);



step(r*T,t)

axis([0 20 0 10])
```



Gambar 2. 8 Grafik Respon Kendali Proposional Dengan Nilai Kp = 5000 Menggunakan PID Tuner Matlab

Kesalahan kondisi tunak sekarang pada dasarnya nol, dan waktu naik telah berkurang secara substansial. Namun, respon ini tidak realistis karena sistem cruise control yang sebenarnya pada umumnya tidak dapat mengubah kecepatan kendaraan dari 0 menjadi 10 m/s dalam waktu kurang dari 0,5 detik karena keterbatasan tenaga mesin dan drivetrain.

Keterbatasan 17ctuator sangat sering ditemui dalam praktek di bidang rekayasa sistem kontrol, dan akibatnya, tindakan kontrol yang diperlukan harus selalu dipertimbangkan ketika mengusulkan kontroler baru. Kami akan membahas masalah ini lebih lanjut di tutorial berikutnya.

Solusi untuk masalah ini dalam hal ini adalah memilih gain proporsional yang lebih rendah, Kp , yang akan memberikan waktu naik yang masuk akal, dan menambahkan pengontrol integral untuk menghilangkan kesalahan keadaan tunak

- **PID Kontrol**

Untuk contoh khusus ini, tidak diperlukan penerapan pengontrol turunan untuk mendapatkan keluaran yang diperlukan. Namun, Anda mungkin ingin melihat cara bekerja dengan kontrol PID untuk referensi di masa mendatang. Fungsi transfer loop tertutup untuk sistem cruise control ini dengan pengontrol PID (C = Kp + Ki/s +Kds) adalah:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{K_d s^2 + K_p s + K_i}{(m + K_d)s^2 + (b + K_p)s + K_i} \quad 2.7$$

Biarkan Kp = 1, Ki = 1, dan Kd = 1 dan masukkan perintah berikut ke dalam m-file baru.

```
Kp = 1;

Ki = 1;

Kd = 1;

C = pid(Kp,Ki,Kd);
```

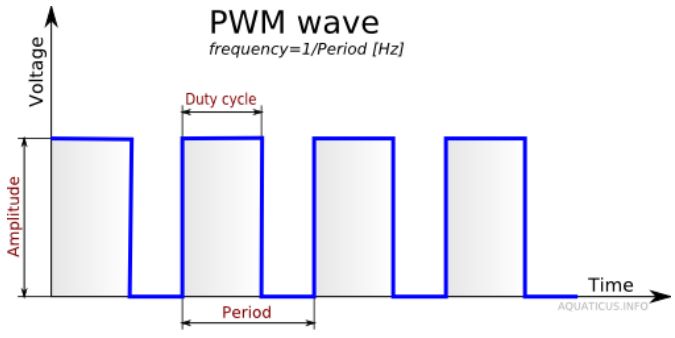
```
T = feedback(C*P_cruise,1);
```

Plot respon langkah dan sesuaikan semua KP, Kd, dan Ki hingga mendapatkan hasil yang memuaskan. Kami akan meninggalkan ini sebagai latihan untuk Anda kerjakan.

Saran : Biasanya memilih gain yang sesuai membutuhkan proses trial and error. Cara terbaik untuk menyerang proses yang membosankan ini adalah dengan menyesuaikan satu variabel (Kp, Ki, atau Kd) pada satu waktu dan mengamati bagaimana perubahan satu variabel memengaruhi keluaran sistem. Karakteristik Kp, Ki, dan Kd dirangkum pada halaman Pendahuluan: Perancangan Kontroler PID.

2.5. Pulse Width Modulation

Pulse Width Modulation disingkat PWM adalah suatu teknik untuk menghasilkan bentuk sinyal analog yang berbentuk pulsa (pulse) dengan menggunakan proses digital. PWM juga dikenal dengan PDM atau Pulse Duration Modulation dikarenakan sinyal yang dihasilkan berbentuk pulsa yang dapat diatur lebar dan sempitnya sinyal tersebut dengan memanipulasi durasi sinyalnya. Kelebihan penggunaan PWM dibanding dengan penguatan linier yaitu PWM menggunakan sinyal biner (digital) sehingga pengendalian dapat dilakukan oleh pengendali digital tanpa memerlukan DAC (Digital to Analog Convertion). Pada PWM, transistor bekerja hanya pada mode operasi saturasi dan cut off, maka hanya sedikit kerugian daya berupa panas. Bentuk pulsa yang dibangkitkan oleh PWM dapat dilihat pada gambar 1.1.



Gambar 2. 9 Bentuk Pulsa PWM

PWM pada dasarnya hanya memiliki dua kondisi pada sinyal PWM yaitu sinyak aktif (1) dan sinyal non-aktif (0). Sinyal aktif terjadi saat mencapai puncak amplitudo dan menjadi non-aktif saat mencapai titik bawah sinyal. Parameter yang ada pada gambar 2.1 dapat dijelaskan sebagai berikut :

1. Duty Cycle. Duty Cycle adalah perbandingan antara waktu ketika sinyal mencapai kondisi ON dan ketika mencapai OFF dalam satu periode sinyal. Contoh misalkan suatu sinyal PWM memiliki duty cycle sebesar 75% maka itu berarti bahwa sebanyak 75% dari waktu periode sinyal merupakan sinyal aktif (ON) dan 25% sisanya adalah sinyal nonaktif (OFF).
2. Periode. Satu periode sinyal adalah satu satuan waktu yang ditetapkan di awal. Nilainya dapat ditentukan sendiri tergantung kebutuhan sinyal yang diinginkan. Namun, sebagian besar perancang menentukan nilainya pada orde milisekon (ms).
3. Amplitudo. Besar nilai sinyal saat mencapai keadaan aktif.

2.6. Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek atau benda tertentu didepan frekuensi kerja pada daerah diatas gelombang suara dari 20 kHz hingga 2 MHz (Arief 2011). Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima struktur unit pemancar dan penerima.

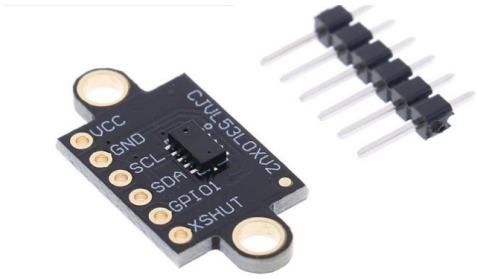
Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal piezoelectric dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz hingga 400 KHz (Arief 2011). Struktur atom dari Kristal piezoelectric menyebabkan berkontraksi mengembang atau menyusut, sebuah polaritas tegangan yang diberikan dan ini disebut dengan efek piezoelectric pada sensor ultrasonik. Pantulan gelombang ultrasonik terjadi bila ada objek tertentu dan pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek piezoelectric menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama.

Besar amplitudo sebuah sinyal elektrik yang dihasilkan sensor penerima tergantung dari jauh dekatnya sebuah objek yang akan dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensoring yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan objek sasaran.

Gambar 2. 10 Sensor ulltrasonik

2.7. Time of Flight Sensor

VL53L0X adalah sensor jarak laser time-of-flight (ToF). VL53L0X dari ST Microelectronics adalah sistem rentang waktu penerbangan yang terintegrasi ke dalam modul ringkas. Papan ini adalah pembawa untuk VL53L0X, jadi kami sarankan untuk membaca lembar data VL53L0X dengan cermat (pdf 1 MB) sebelum menggunakan produk ini.



Gambar 2. 11 Sensor time of flight

VL53L0X menggunakan teknologi FlightSense ST untuk mengukur secara tepat berapa lama waktu yang dibutuhkan pulsa pancaran sinar laser inframerah untuk mencapai objek terdekat dan dipantulkan kembali ke detektor, sehingga dapat dianggap sebagai sistem lidar mandiri yang kecil. Pengukuran time-of-flight (TOF) ini memungkinkannya untuk secara akurat menentukan jarak absolut ke target tanpa reflektansi objek yang sangat memengaruhi pengukuran. Sensor dapat melaporkan jarak hingga 2 m (6,6 kaki) dengan resolusi 1 mm, tetapi jangkauan

efektif dan akurasi (noise) sangat bergantung pada kondisi sekitar dan karakteristik target seperti reflektansi dan ukuran, serta konfigurasi sensor. (Akurasi sensor ditentukan untuk berkisar dari 3% terbaik hingga lebih dari 10% dalam kondisi yang kurang optimal.)

Pengukuran rentang tersedia melalui antarmuka I2C (TWI) sensor, yang juga digunakan untuk mengonfigurasi pengaturan sensor, dan sensor menyediakan dua pin tambahan: input shutdown dan output interupsi. Adapun Pin pada sensor Tof, sebagai berikut :

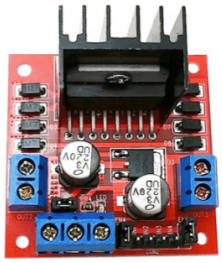
1. VIN Ini adalah sambungan catu daya utama 2,6 V hingga 5,5 V. Pemindah level SCL dan SDA menarik garis I2C tinggi ke level ini.
2. GND Koneksi ground (0 V) untuk catu daya Anda. Sumber kontrol I2C Anda juga harus memiliki kesamaan dengan board ini.
3. SDA Level-shifted I2C jalur data: TINGGI adalah VIN, RENDAH adalah 0 V
4. SCL Level-shifted I2C garis jam: TINGGI adalah VIN, RENDAH adalah 0 V
5. XSHUT Pin ini merupakan input shutdown aktif-rendah; papan menariknya ke VDD untuk mengaktifkan sensor secara default. Mengemudikan pin ini rendah menempatkan sensor ke siaga perangkat keras. Input ini tidak berubah level.

2.8. Driver Motor L298N

Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC.

IC L298 merupakan sebuah IC tipe H-bridge yang mampu mengendalikan beban beban induktif seperti relay, solenoid, motor DC dan motor stepper. Pada IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang NAND yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor stepper (Agus 2017).

Untuk dipasaran sudah terdapat modul driver motor menggunakan IC L298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah tersusun dengan rapi dan mudah digunakan. Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol.



Gambar 2. 12 Driver motor L298N

Adapun spesifikasi pada Driver motor L298N, sebagai berikut

- Tegangan Masukan: 3.2V~40Vdc.
- Driver: Driver Motor DC L298N Dual H Bridge
- Catu Daya: DC 5 V - 35 V
- Arus puncak: 2 Amp
- Jangkauan pengoperasian saat ini: 0 ~ 36mA
- Kisaran voltase input sinyal kontrol:
- Rendah: -0,3V ≤ Vin ≤ 1,5V .
- Tinggi: 2.3V ≤ Vin ≤ Vss.
- Aktifkan rentang voltase masukan sinyal :
    - Rendah: -0,3 ≤ Vin ≤ 1,5V (sinyal kontrol tidak valid).
    - Tinggi: 2.3V ≤ Vin ≤ Vss (sinyal kontrol aktif).
- Konsumsi daya maksimum: 20W (saat suhu T = 75 °C).
- Suhu penyimpanan: -25 °C ~ +130 °C.
- Catu Output teregulasi +5V on-board (pasokan ke papan pengontrol atau mikrokontroller).
- Ukuran: 3,4cm x 4,3cm x 2,7cm

2.8.1. Tabel Logika Driver Motor L298N

Di bawah ini adalah tabel logika input dari Driver Motor L298N.

Tabel 2. 2 Logika input motor driver L298N

IN	IN2	PUTARAN MOTOR
Low	Low	Motor tidak jalan
High	Low	Maju
Low	High	Mundur
High	High	Rem

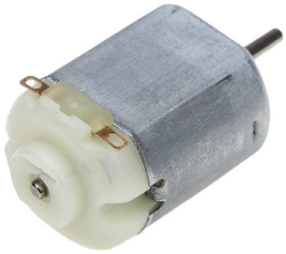
2.9. Motor DC

Motor DC Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/directunidirectional (Agus Purnama 2022).

Motor DC adalah piranti elektronik yang mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (double pole, double throw switch). Motor DC bekerja berdasarkan prinsip gaya Lorentz, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus. Kecepatan putar motor DC (N) dirumuskan dengan Persamaan berikut.

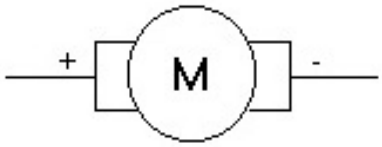
$$N = \frac{V_{TM} - I_A R_A}{K\phi} \qquad 2.8$$

Keterangan:
V_{TM} : Tegangan terminal
I_A : Arus jangkar motor
R_A : Hambatan jangkar motor
K : Konstanta motor
ϕ : Fluk magnet yang terbentuk pada motor.



Gambar 2. 13 Motor DC

2.9.1. Simbol Motor DC



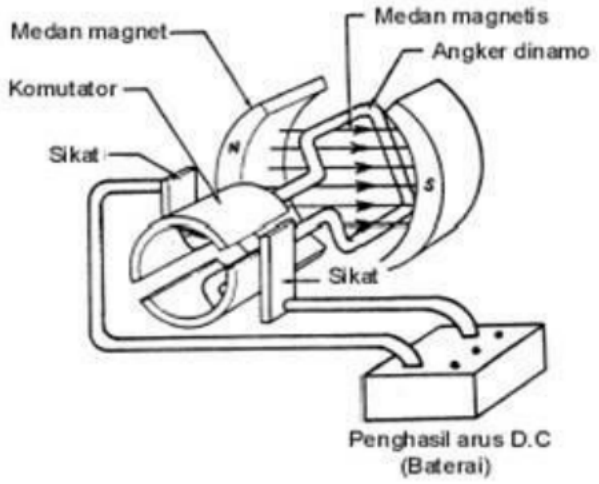
Gambar 2. 14 Simbol Motor DC

Motor DC tersusun dari dua bagian yaitu bagian diam (stator) dan bagian bergerak (rotor). Stator motor arus searah adalah badan motor atau kutub magnet (sikat-sikat), sedangkan yang termasuk rotor adalah jangkar lilitanya. Pada motor, kawat penghantar listrik yang bergerak tersebut pada dasarnya merupakan lilitan yang berbentuk persegi panjang yang disebut kumparan.

2.9.2. Bagian Atau Komponen Utama Motor DC

1. **Kutub medan.** Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih komplek terdapat satu atau lebih elektromagnet.
2. **Kumparan Motor DC.** Bila arus masuk menuju kumparan motor DC, maka arus ini akan menjadi elektromagnet. kumparan motor DC yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakan beban. Untuk kasus motor DC yang kecil, kumparan motor DC berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan kumparan motor DC.

3. **Komutator Motor DC**. Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam 9 kumparan motor DC dan juga membantu dalam transmisi arus antara kumparan motor DC dan sumber daya.



Gambar 2. 15 Bagian-bagian motor DC

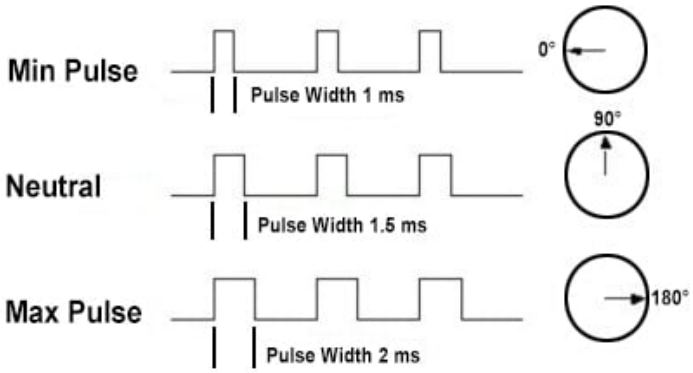
2.10. Motor Servo

Motor servo menggunakan dengan sistem umpan balik tertutup, di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energy mekanik, maka magnit permanent motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnit. Salah satu medan dihasilkan oleh magnit permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnit tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Gambar 2. 16 Motor Servo

2.10.1. Prinsip Kerja Motor Servo

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel arah. Lebar pulsa sinyal arah yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90º. Bila pulsa lebih pendek dari 1,5 ms maka akan berputar kearah posisi 0º atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar kearah posisi 180º atau ke kanan (searah jarum jam). Pulse Wide Modulation servo ditunjukkan dalam Gambar 2.13



Gambar 2. 17 Pulse wide modulation servo

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan

mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya.

2.11. Battery Lithium Ion 18650

Baterai 18650 adalah baterai sel khusus yang dapat diisi ulang dengan kemampuan yang tinggi. Baterai tersebut merupakan satu dari sederetan baterai berbahan ‘lithium-ion’. Sebagaimana umumnya baterai-baterai sel lithium-ion, tegangan yang dihasilkannya adalah sebesar 3,6V atau 3,7V. Namun baterai 18650 ini mampu mempunyai kapasitas hingga 3500mAH. Karena itu baterai ini banyak diandalkan untuk menjalankan peralatan-peralatan listrik kecil yang membutuhkan energi ekstra. Baterai ini disebut sebagai ‘18650 cell’. Angka-angka 18650 menggambarkan bentuk dan ukurannya, yaitu :

18 = ukuran diameter baterai, yaitu 18mm
65 = ukuran panjang baterai, yaitu 65mm.
0 = faktor bentuk baterai yang silindris.

Gambar 2. 18 Battery lithium ion 18650

2.12. Step Down DC LM2596

Step down dc LM2596 adalah rangkaian terpadu monolitik yang ideal untuk desain regulator step-down switching (buck converter) yang mudah dan aman. Modul ini mampu mencatu beban hingga 3A dengan metode pengaturan tegangan. Module LM2596 adalah sebuah konverter catu daya dengan sistem switch mode, Module ini memiliki efisiensinya jauh lebih tinggi dibandingkan dengan pengatur linier tiga terminal umumnya. LM2596 beroperasi pada

frekuensi switching 150 kHz sehingga memungkinkan komponen filter berukuran lebih kecil.

Spesifikasi teknis dari LM2596 adalah sebagai berikut:

- Properti Modul: regulator switching step-down (BUCK) yang tidak terisolasi
- Tegangan input: DC 3.0 - 35V
- Tegangan Output: Adjustable 1,5 - 35V DC (Input harus 1,5V lebih besar dari output)
- Output Saat Ini: Dinilai 2A, maks. 3A (Diperlukan Heat Sink tambahan)
- Efisiensi konversi: Hingga 92% (Tegangan output lebih tinggi, semakin tinggi efisiensinya)
- Tegangan dropdown minimum: 1.5V
- Pengaturan tegangan: 0,5%
- Kecepatan respons dinamis: 5% 200uS
- Frekuensi switching: 150KHz
- Perlindungan sirkuit: SS36
- Suhu pengoperasian: Tingkat industri (-40 hingga +85 ° C) (daya output 10W atau kurang)
- Regulasi beban: 0,5%
- Ukuran: 50mm x 23 mm x 14mm

Gambar 2. 19 Step down dc LM2596

2.13. Aplikasi Blynk

Blynk adalah platform aplikasi yang dapat diunduh secara gratis untuk iOS dan Android yang berfungsi mengontrol Arduino, Raspberry Pi dan sejenisnya melalui Internet. Blynk dirancang untuk Internet of Things dengan tujuan dapat mengontrol hardware dari jarak jauh, dapat menampilkan data sensor, dapat

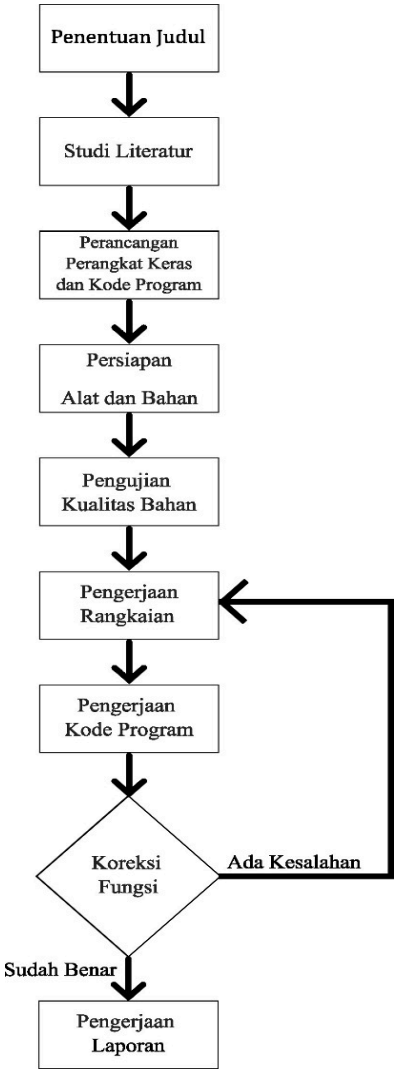
menyimpan data, visual dan melakukan banyak hal canggih lainnya. Ada tiga komponen utama dalam platform yaitu Blynk App, Blynk Server, dan Blynk Library.

Gambar 2. 20 Skema antarmuka blynk

BAB III
PERANCANGAN ALAT

Pada bab ini akan di jelaskan tentang perancangan dan langkah-langkah pembuatan Alat prototype cruise control pada kendaraan listrik dengan metode kendali PID, baik secara hardware maupun software. Rancangan dimulai dari perancangan sistem, perancangan perangkat keras *(Hardware*) sampai tahapan perancangan perangkat lunak (*Software*) secara umum system ini menggunakan control loop tertutup atau kendali tertutup.

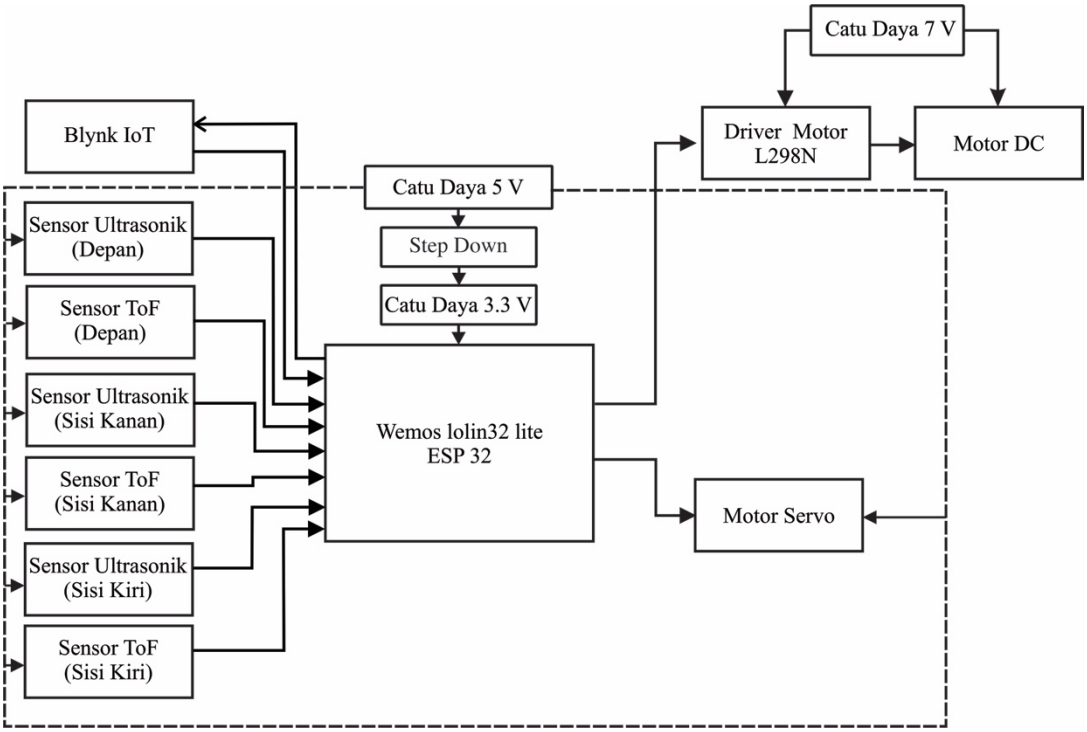
3.1. Diagram Alur Kerja



gambar 3. 1 Diagram Alur Kerja

3.2. Diagram Blok

Diagram blok sistem berfungsi untuk menggambarkan sistem pengontrolan alat Prototype Cruise Control Pada kendaraan Listrik Dengan Metode Kendali PID. dimana alat ini akan mengontrol untuk mendapatkan jarak stabil, sistem kendali PID akan digunakan untuk mencapai setpoint dari jarak yang diinginkan. Jika objek didepan sudah kurang dari batas setpoint, maka kecepatan akan dikurangi secara proporsional. Namun ada kalanya jika jarak sangat kurang dari setpoint (objek terlalu dekat) maka manuver akan dilakukan. Hal ini bertujuan untuk menghindari tabrakan dikarenakan rem yang kurang cepat (pakem). Mikrokontroler akan membaca sensor jarak yang terdapat pada sisi kiri dan kanan dan memilih jarak terbesar yang berarti sisi tersebut bebas pembatas atau hambatan. ESP32 Wemos lolin32 lite akan memproses sistem kendali PID untuk terhubung dengan sensor dan motor.



Gambar 3. 2 Diagram Blok

Dalam blok diagram yang disajikan pada Gambar 3.2 akan dijelaskan secara bertahap sebagi berikut:

a. Catu Daya

   Berfungsi sebagai sumber listrik pada alat sistem cruise control , baik suplay kontrol maupun suplay motor DC.

b. Step Down

   Berfungsi sebagai menurunkan tegangan untuk mikrokontroller.

c. Aplikasi Blynk IoT

   Sebagai kontrol untuk mengoperasikan keseluruhan alat ini.

d. Sensor Ultrasonik (Depan)

   Sebagai input data berupa hasil pembacaan jarak mobil dengan objek yang ada di depan.

e. Sensor ToF (Depan)

   Sebagai input data berupa hasil pembacaan jarak mobil dengan objek yang ada di depan.

f. Sensor Ultrasonik (Kanan)

   Sebagai input untuk membaca jarak terbesar yang bebas hambatan di sisi kanan mobil ketika mobil akan manuver.

g. Sensor ToF (Kanan)

   Sebagai input untuk membaca jarak terbesar yang bebas hambatan di sisi kanan mobil ketika mobil akan manuver.

h. Sensor Ultrasonik (Kiri)

   Sebagai input untuk membaca jarak terbesar yang bebas hambatan di sisi kiri mobil ketika mobil akan manuver.

i. Sensor ToF (Kiri)

   Sebagai input untuk membaca jarak terbesar yang bebas hambatan di sisi kiri mobil ketika mobil akan manuver.

j. Motor Driver

   Berfungsi sebagai kontrol arah putar dan kecepatan pada motor dc yang terhubung ke ESP 32 wemos lolin32 lite.

k. Motor DC

   Berfungsi untuk menjalankan roda belakang pada mobil.

l. Motor Servo

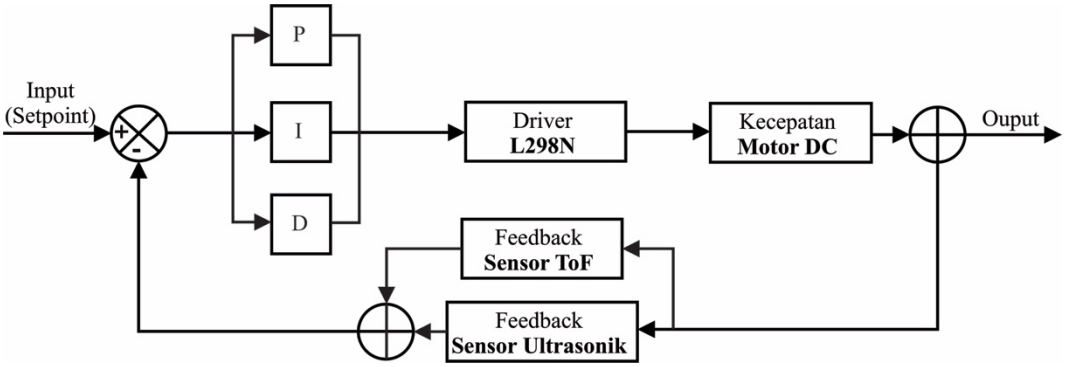
Berfungsi sebagai berbelok arah kanan dan kiri serta manuver mobil ketika jarak sudah kurang dari set point.

m. Sistem Kendali PID

Sebagai sistem kendali untuk perhitungan kecepatan dan pengereman pada mobil pada saat mobil sampai batas set point maupun kurang dari set point.

3.3. Blok Sistem Kendali PID

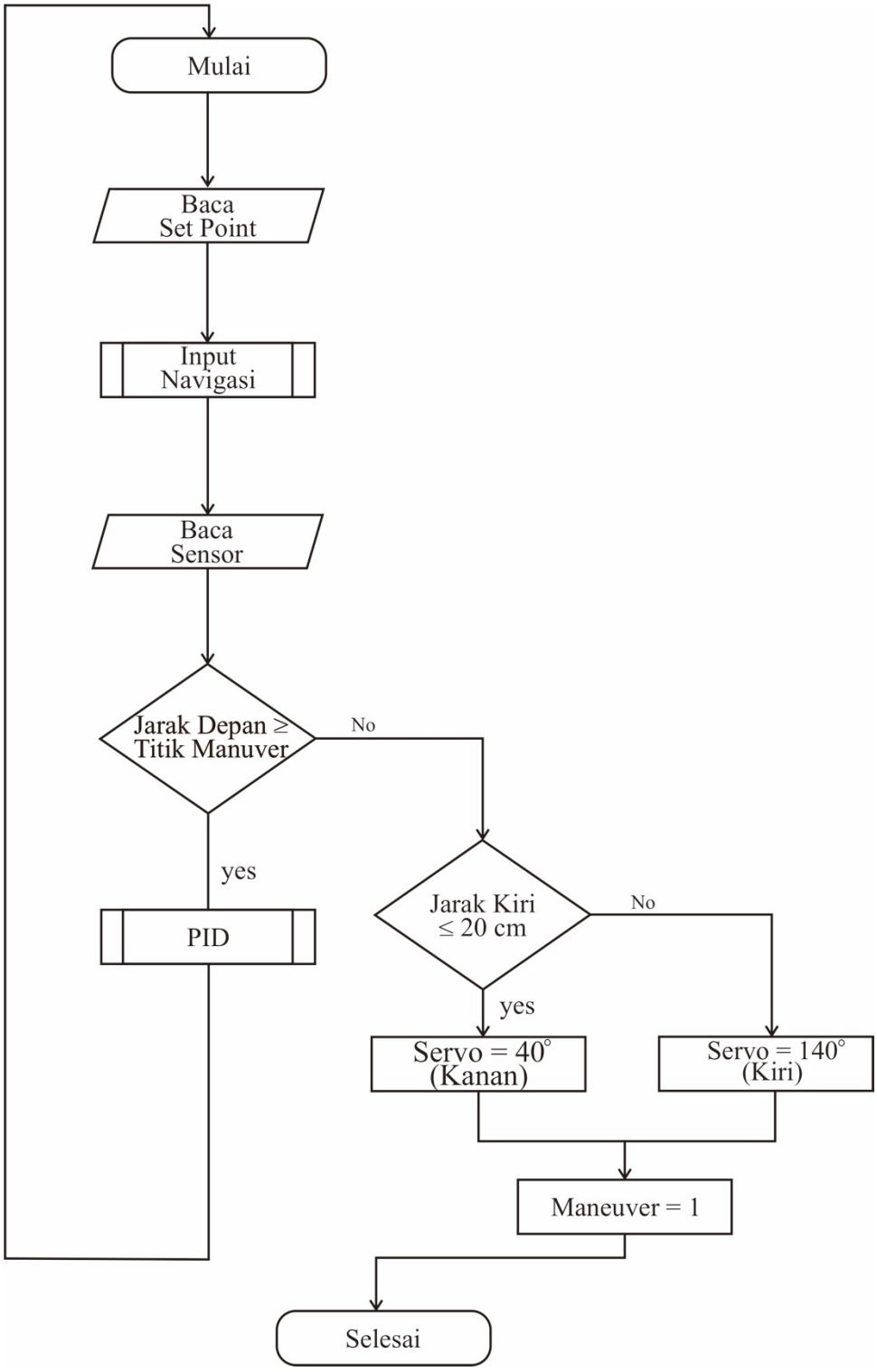
Blok sistem kendali PID ini sebagai gambaran proses kendali kecepatan motor untuk kendali kecepatan mobil. Mulai dari menghitung nilai error kemudian memasukan nilai konstanta P, konstanta I dan Konstanta D untuk didapat nilai output masing-masing dan kemudian dijumlahkan dari masing-masing output tadi dengan keluaran berupa PWM. Kemudian PWM akan menjadi masukan untuk motor driver L298N yang selanjutnya menjadi tegangan untuk motor dc. Selanjutnya terjadi pengulangan proses sampai mencapai nilai setpoint yang ditentukan. Seperti gambar 3.3 berikut.



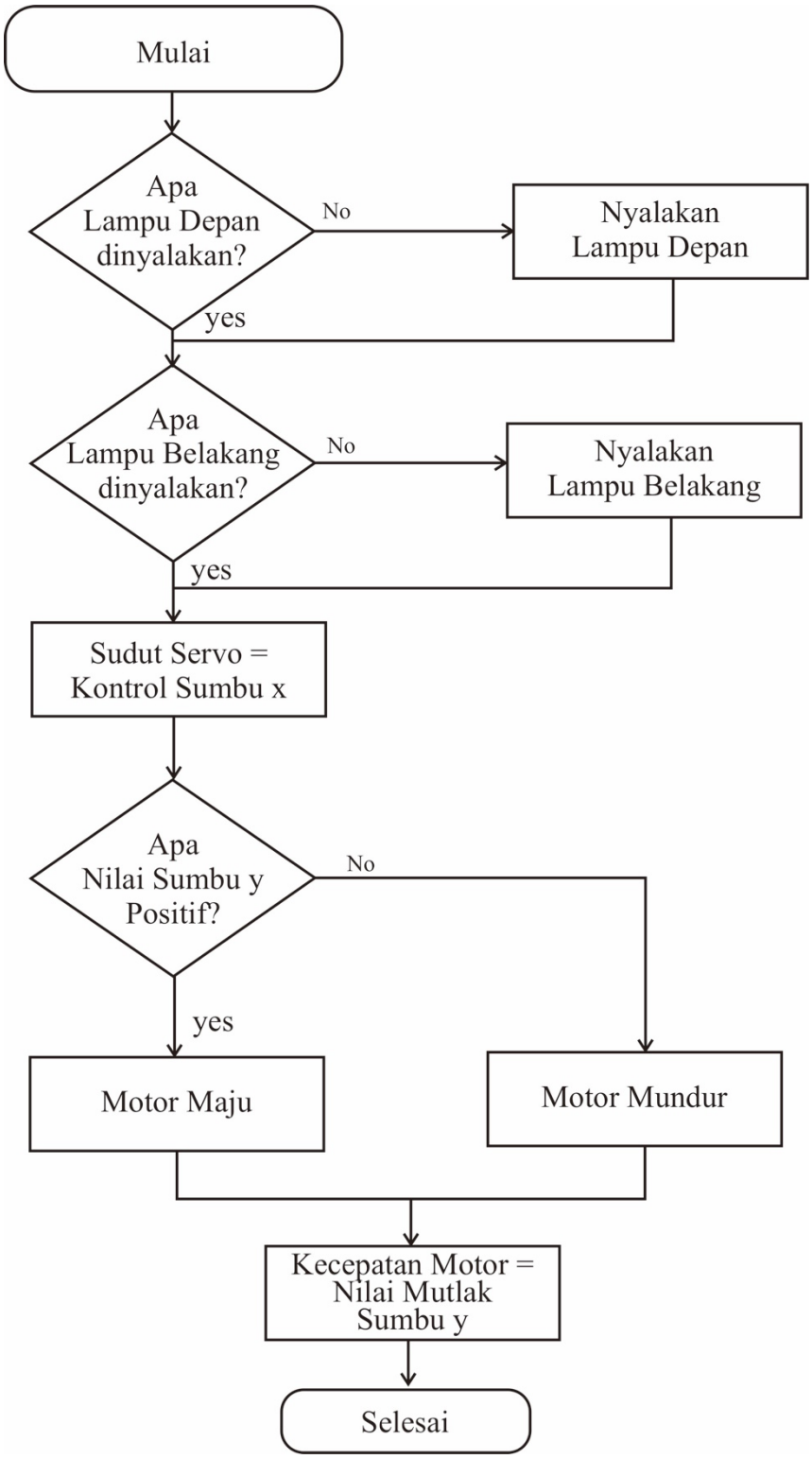
Gambar 3. 3 Blok sistem kendali PID

3.4. Flowchart

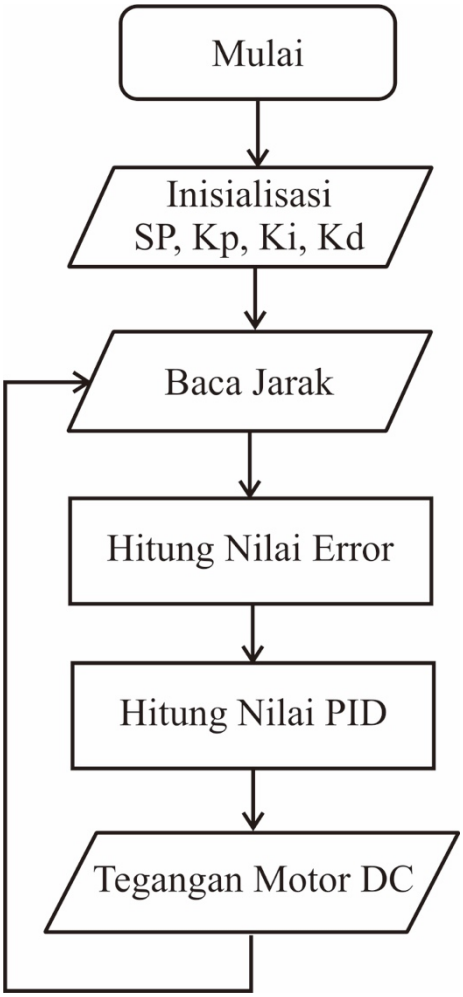
Untuk menggambarkan keseluruhan alat prototype cruise control pada kendaraan listrik dengan metode kendali PID, maka pada flowchart ini akan menjelaskan tahapan-tahapan dari sistem keseluruhan yang terdiri dari beberapa proses yaitu :



gambar 3. 4 Flowchart keseluruhan kerja alat



gambar 3. 5 Flowchart input navigasi



gambar 3. 6 Flowchart perhitungan PID

1. Mulai

Kondisi yang menyatakan titik awal seluruh sistem umtuk melakukan proses.

2. Baca Set Point

Setelah alat telah dimulai, set point akan terbaca sesuai dengan yang telah diinput di program.

3. Input Navigasi

Proses dimana alat masih dioperasikan dengan mode manual atau dalam kata lain alat atau mobil masih dikendarai oleh manusia. Mulai dari menyalakan lampu depan dan lampu belakang, kemudian belok kanan dan kiri dan maju mundurnya mobil.

4. Baca sensor

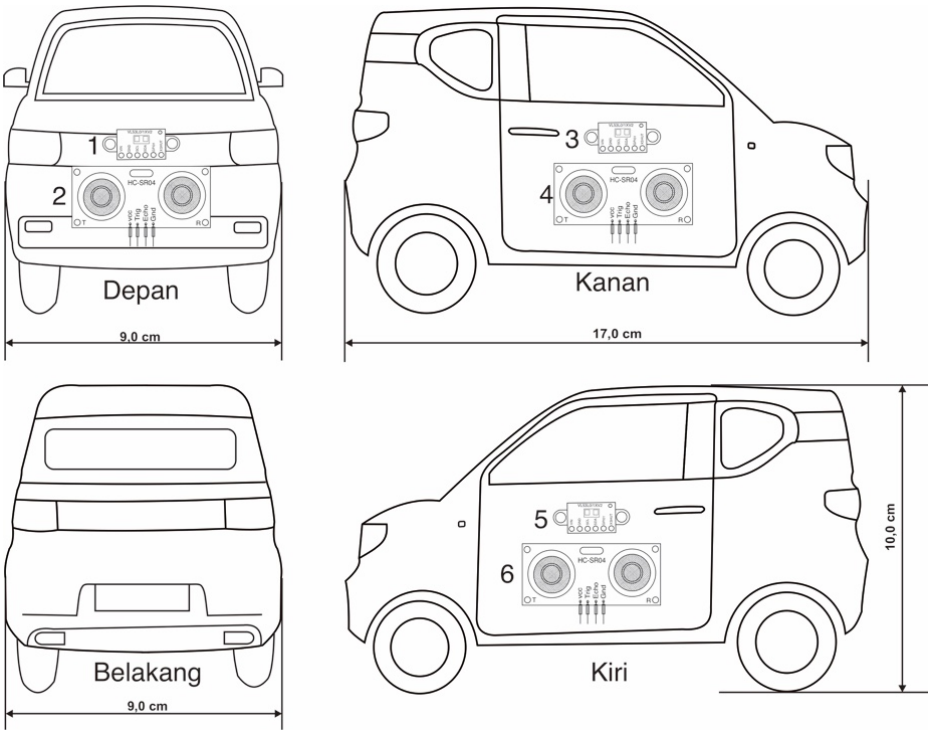
Sensor akan membaca jarak terhadap objek yang berada di depan, kanan dan kiri. Jika jarak depan mobil terhadap objek masih lebih besar dari titik manuver maka kecepatan mobil akan terkendali otomatis dengan perhitungan PID. Namun jika jarak depan mobil terhadap objek kurang dari titik manuver, maka, mobil akan melakukan manuver, dengan cara apabila jarak kiri kurang dari 20 cm maka mobil akan manuver ke kanan dan apabila jarak kiri lebih dari 20 cm, maka mobil akan manuver ke kiri.

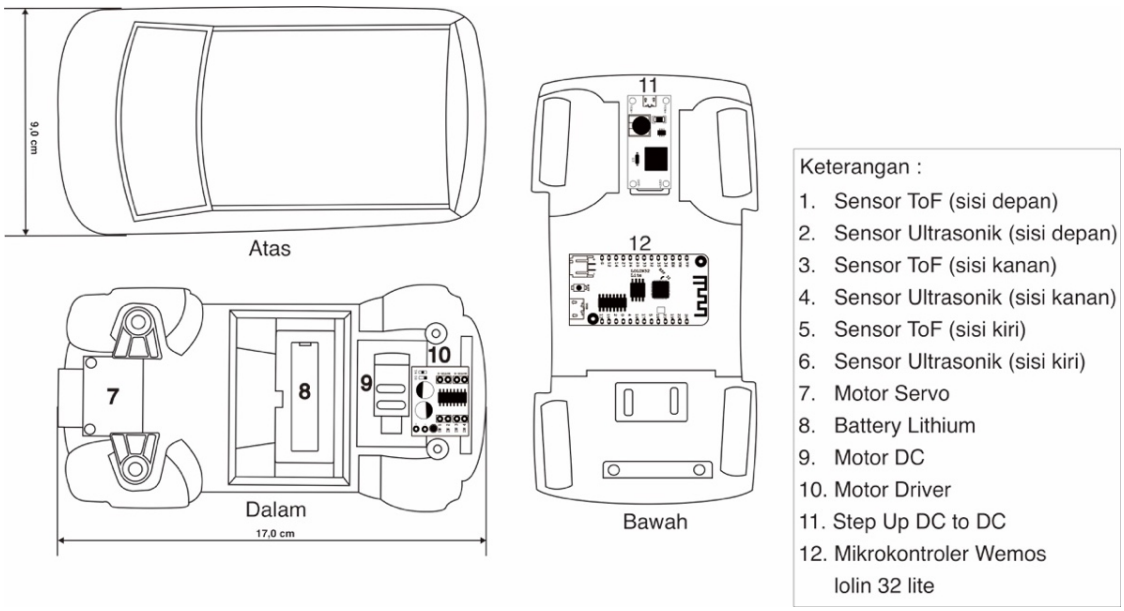
5. Perhitungan PID

Sebagai kendali Jarak mobil jika jarak depan mobil terhadap objek masih lebih besar dari titik manuver. Mulai dari menghitung nilai error kemudian memasukan nilai konstanta P, konstanta I dan Konstanta D untuk didapat nilai output masing-masing dan kemudian dijumlahkan dari masing-masing output tadi.

3.5. Tata Letak Komponen

Tata letak komponen alat Prototype cruise control pada kendaraan listrik dengan metode kendali PID. Peletakan komponen elektronika dilakukan agar dapat mempermudah penginstalan pada tiap-tiap bagian dari sistem ini.





gambar 3. 7 Tata letak komponen

a) Nomor 1 adalah sensor ToF sisi depan sebagai pembaca jarak depan mobil terhadap objek.

b) Nomor 2 adalah sensor Ultrasonik sisi depan sebagai pembaca jarak depan mobil terhadap objek.

c) Nomor 3 adalah sensor ToF sisi kanan sebagai pembaca jarak pada sisi kanan mobil.

d) Nomor 4 adalah sensor ultrasonik sisi kanan sebagai pembaca jarak pada sisi kanan mobil.

e) Nomor 5 adalah sensor ToF sisi kiri sebagai pembaca jarak pada sisi kiri mobil.

f) Nomor 6 adalah sensor ultrasonik sisi kiri sebagai pembaca jarak pada sisi kiri mobil.

g) Nomor 7 adalah motor servo untuk mobil belok kanan dan kiri.

h) Nomor 8 adalah battery lithium sebagai power supply pada perangkat ini.

i) Nomor 9 adalah motor DC sebagai penggerak pada roda mobil.

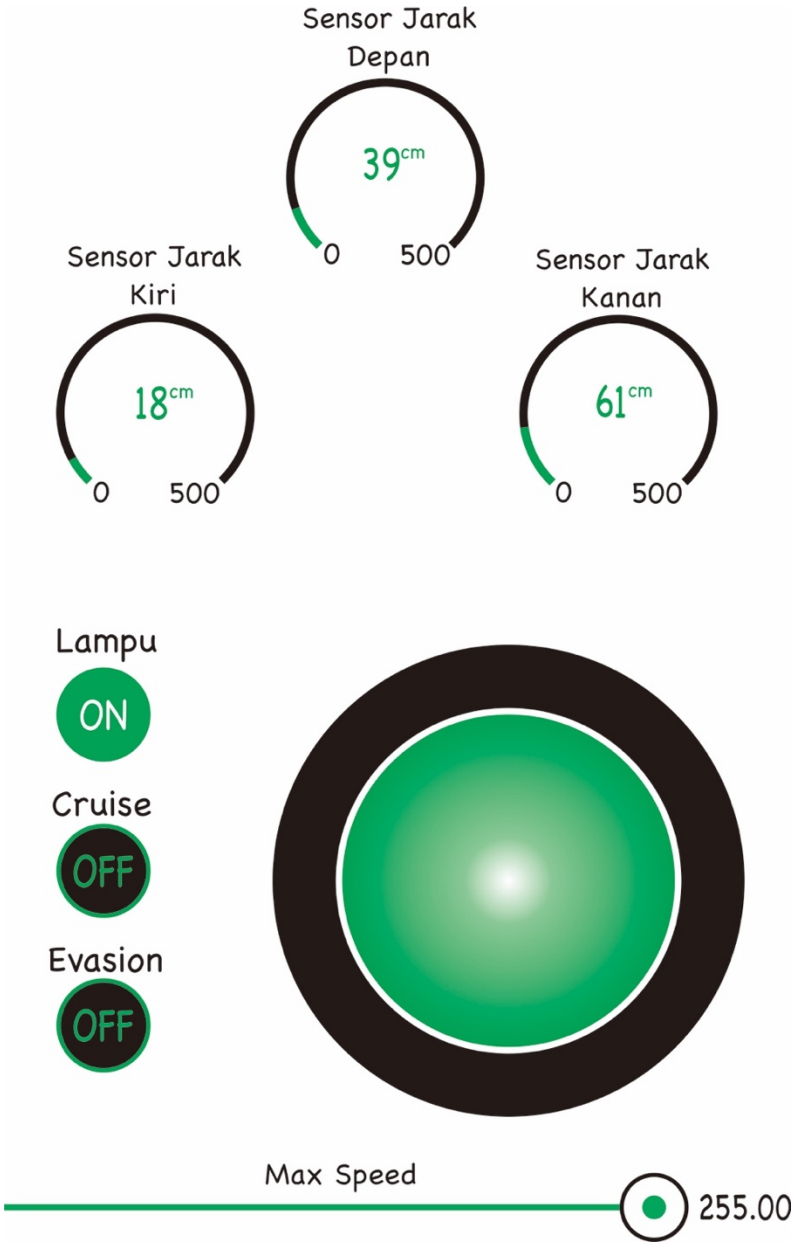
j) Nomor 10 adalah driver motor sebagai pengatur arah putar dan kecepatan pada motor DC.

k) Nomor 11 adalah step up untuk menaikan tegangan dari battery yang terhubung langsung ke battery.

l) Nomor 12 adalah mikrokontroler ESP32 WeMos Lolin32 Lite

3.6. Perancangan Aplikasi Blynk IoT

Perancangan Aplikasi ini berfungsi sebagai input kontrol untuk pengoperasian keseluruhan pada alat ini, seperti mengatur gas, kecepatan dan fitur-fitur pada cruise control serta dapat memonitor langsung jarak mobil pada objek. Seperti terlihat pada gambar 3.8 di bawah ini.



gambar 3. 8 Tampilan aplikasi blynk

Pada tampilan aplikasi blynk IoT terdapat beberapa menu, sebagai berikut

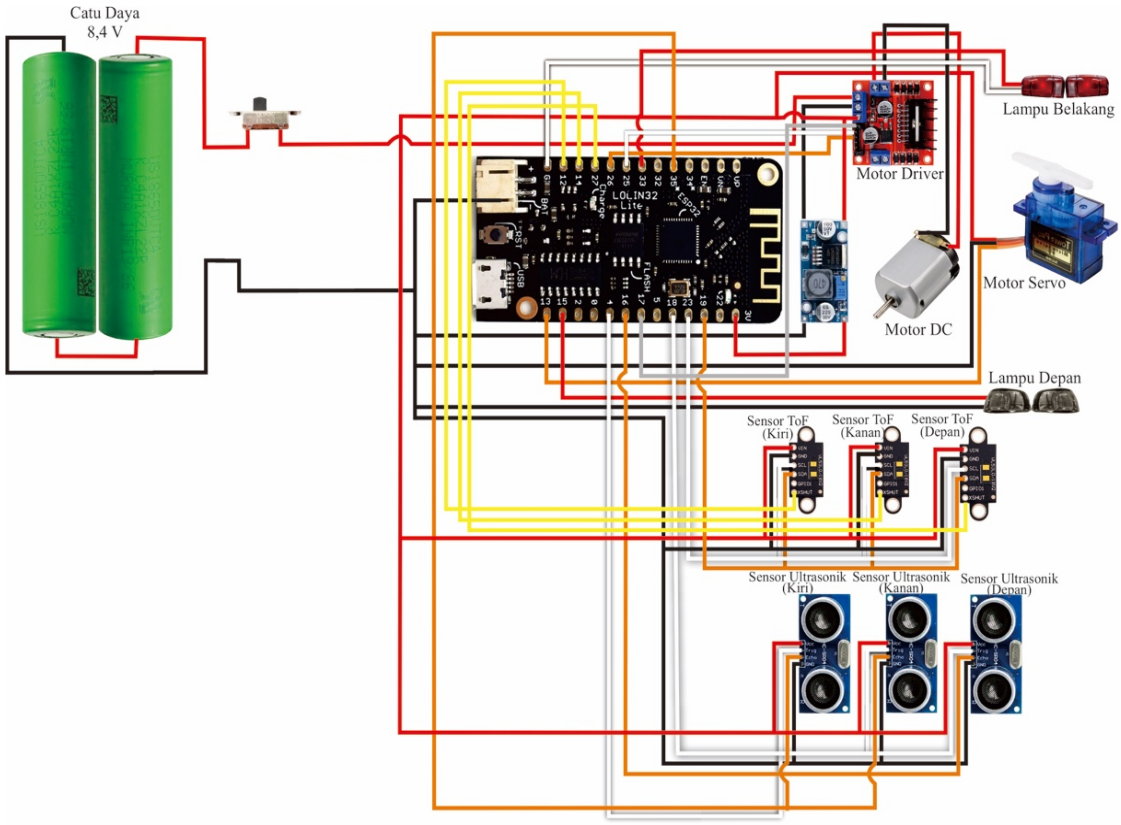
Tabel 3. 1 Keterangan fitur aplikasi blynk

No	Menu	Keterangan
1	Sensor jarak	Sebagai monitor jarak terhadap objek
2	Lampu	Untuk menghidupkan lampu depan mobil
3	Cruise	Fitur cruise control adalah fitur keseluruhan untuk kendali kecepatan mobil secara otomatis dan manuver mobil secara otomatis
4	Evasion	Bagian dari fitur cruise, fitur ini sebagai manuver otomatis mobil.

3.7. Perancangan Perangkat Keras

Perancangan perangkat keras ini meliputi suatu sistem Womos lolin32 lite ESP 32 dengan metode kendali PID sebagai pengandali utama sistem cruise control pada kendaraan listrik. Dimana ESP 32 Womos lolin32 lite mengendalikan seluruh komponen yang terdapat di dalam rangkaian.

Pada sistem Cruise control pada kendaraan listrik ini terdiri dari Womos lolin32 lite ESP 32 + Sensor ToF, Womos lolin32 lite ESP 32 + Sensor Ultrasonik, Womos lolin32 lite ESP 32 + Driver motor + Motor DC, Womos lolin32 lite ESP 32 + Motor Servo dan Womos lolin32 lite ESP 32 + Blynk IoT.



gambar 3. 9 Perancangan skematik perangkat keras alat

3.8. Rangkaian Sistem ESP32 WeMos LoLin32 Lite

ESP32 WeMos LoLin32 Lite yang digunakan sebagai mikrokontroler akan dihubungkan dengan catu daya 5V. sedangkan pin-pin pada ESP32 WeMos LoLin32 Lite dihubungkan pada pin data input maupun output. Koneksi masing-masing pin ESP32 WeMos LoLin32 Lite dapat dijabarkan dengan jelas melalui Tabel 3.2.

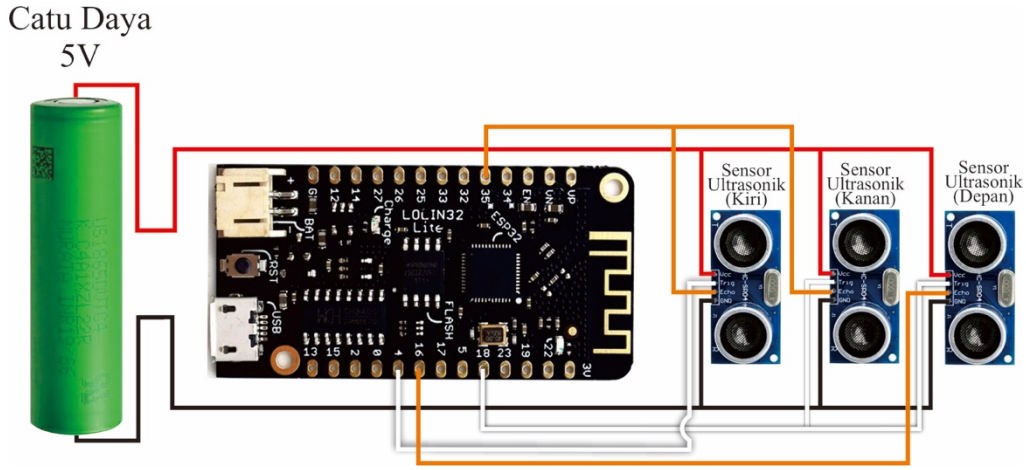
Tabel 3. 2 Koneksi pin ESP32 wemos lolin32 lite

No	Keterangan	Pin	ESP32 WeMos LoLin32 Lite	Catu Daya
1	Driver Motor	IN 1	25	7V
		IN 2	26	7V
		Enable A	17	7V
2	ToF (Depan)	SDA	19	5V
		SCL	23	5V
		XSHUT	27	5V
3	ToF (Kanan)	SDA	19	5V
		SCL	23	5V
		XSHUT	14	5V
4	ToF (Kiri)	SDA	19	5V
		SCL	23	5V
		XSHUT	12	5V
5	Ultrasonik (Depan)	Trig	18	5V
		Echo	16	5V
6	Ultrasonik (Kanan)	Trig	18	5V
		Echo	35	5V
7	Ultrasonik (Kiri)	Trig	4	5V
		Echo	35	5V
8	Lampu Depan		15	3.3V(ESP32)
9	Lampu Belakang		33	3.3V(ESP32)
10	Motor Servo		13	5V

3.8.1. Rangkaian ESP32 WeMos LoLin32 Lite dengan Sensor Ultrasonik

Sensor ultrasonik HC-SR04 pada rangkaian ini berfungsi untuk mengukur jarak antara mobil dengan objek. Pada sistem ini menggunakan 3 buah sensor ultrasonik yang terpasang di sisi depan, kanan, dan kiri mobil.

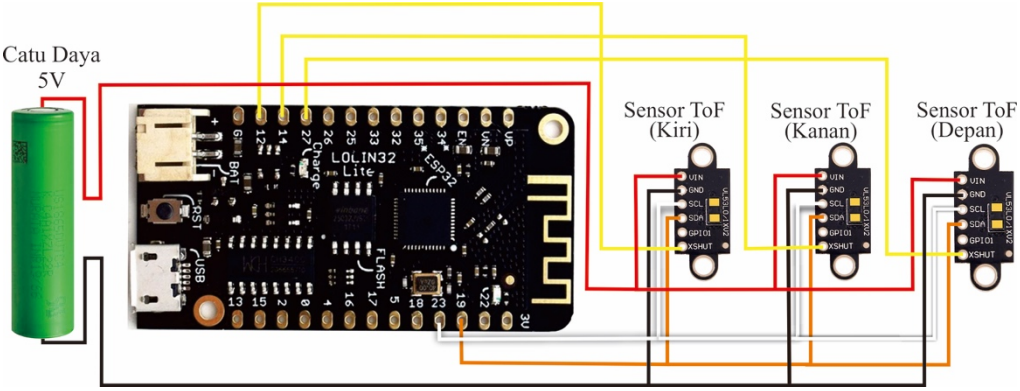
Untuk mendapatkan kecepatan stabil, Mikrokontroler akan mengambil data jarak dari Sensor ultrasonik yang berada di depan sebagai perhitungan kecepatan pada motor DC dengan metode kendali PID. dan sensor ultrasonik yang terdapat pada sisi kanan dan kiri akan memilih jarak terbesar yang terbebas pembatas atau hambatan untuk mobil bermanuver ketika mobil sudah dekat dengan objek. Gambar 3.10 menggambarkan terhubungnya ESP32 WeMos LoLin32 Lite dengan sensor Ultrasonik.



gambar 3. 10 Rangkaian yang menghubungkan esp 32 wemos lolin32 lite dengan sensor ultrasonik

3.8.2. Rangkaian ESP32 WeMos LoLin32 Lite dengan Sensor ToF

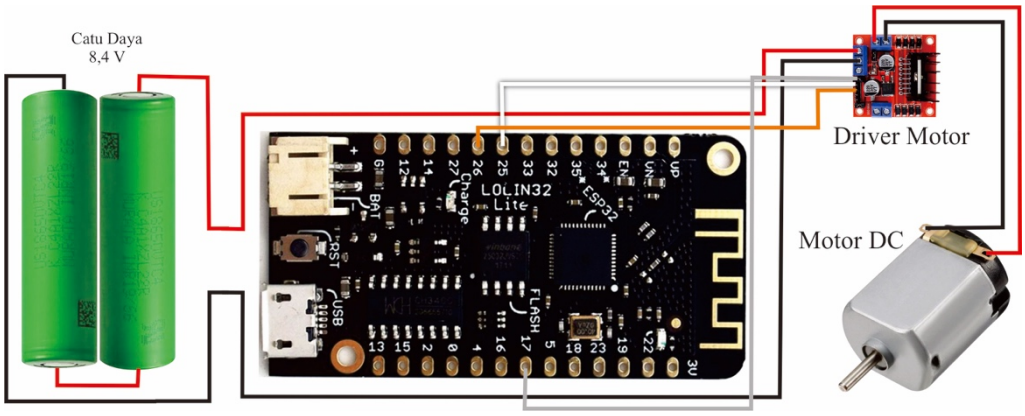
Sensor Tof *(Time of Flight)* VL53L0X pada sistem ini berfungsi sama seperti sensor ultrasonik yaitu membaca jarak, dimana mikrokontroler akan membandingkan data jarak dari sensor ultrasonik dengan sensor ToF dan akan dipilih data jarak yang lebih akurat.



gambar 3. 11 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan sensor ToF

3.8.3. Rangkaian ESP32 WeMos LoLin32 Lite dengan Driver Motor dan Motor DC

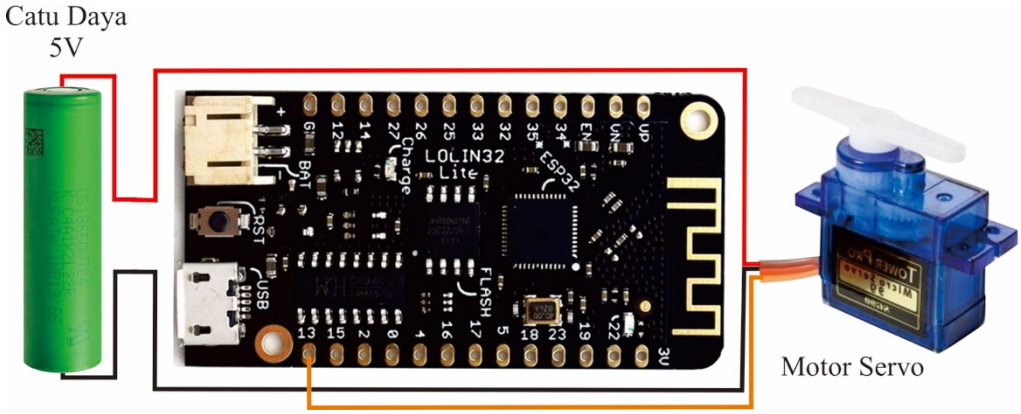
Pada Rangkaian sistem alat ini, Motor DC berfungsi sebagai akuator yang menghasilkan putaran untuk menggerakan roda belakang pada mobil. Untuk mengatur kecepatan atau putaran pada motor DC sampai dapat stabil, sistem kendali PID digunakan dengan cara memasukan nilai error atau nilai selisih antara setpoint dengan actualpoint ke persamaan PID dan keluaran PID berupa PWM *(Pulse Width Modulation)* yang menjadi masukan untuk driver motor L298N melalui mikrokontroller ESP32 WeMos LoLin32 Lite. Selanjutnya terjadi pengulangan proses sampai mencapai nilai setpoint yang ditentukan



gambar 3. 12 Rangkaian yang menghubungkan esp32 wemos lolin32 lite dengan driver motor dan motor dc

3.8.4. Rangkaian ESP32 WeMos LoLin32 Lite dengan Motor Servo

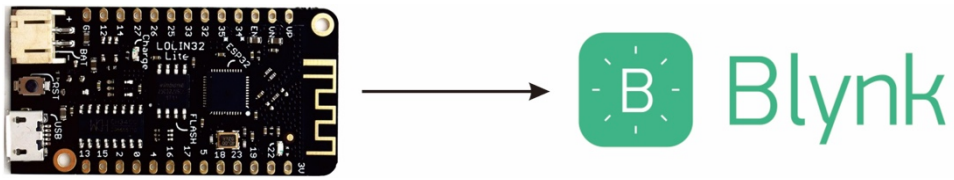
Motor servo pada sistem ini berfungsi ketika mobil ingin manuver atau berbelok kanan atau kiri. Saat apabila mobil sudah terlalu dekat dengan objek atau kendaraan yang berada di depan dan pengereman sudah tidak dapat dilakukan, maka servo akan bergerak ke kanan atau ke kiri untuk memilih jarak terbesar sesuai data yang diterima dari sensor melalui ESP32 WeMos LoLin32 Lite.



gambar 3. 13 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan motor servo

3.8.5. Rangkaian ESP32 WeMos LoLin32 Lite dengan Aplikasi Blynk IoT

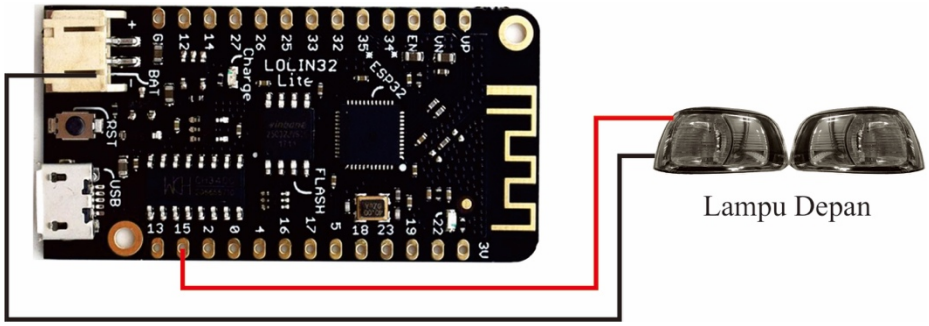
Aplikasi Blynk IoT berfungsi sebagai Input untuk mengoperasikan atau mengendarakan mobil. Seperti, mengatur gas, kecepatan, dan mengaktifkan fitur-fitur yang terdapat di cruise control. Wemos lolin32 lite di hubungkan ke wifi agar terhubung ke aplikasi blynk iot.



gambar 3. 14 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan blynk IoT

3.8.6. Rangkaian ESP32 WeMos LoLin32 Lite dengan Lampu Depan

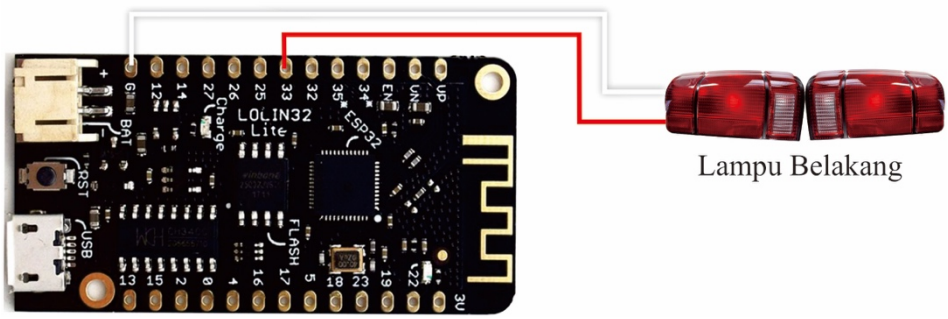
Lampu depan pada sistem ini berfungsi sebagai penerang jalan untuk kondisi pada jalan gelap. Untuk menyalakan dan mematikan lampu ini dapat diatur melalui aplikasi Blynk IoT.



gambar 3. 15 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan lampu depan

3.8.7. Rangkaian ESP32 WeMos LoLin32 Lite dengan Lampu Belakang

Lampu belakang pada sistem ini berfungsi sebagai lampu peringatan untuk memberi sinyal kepada kendaraan di belakang mobil untuk mengurangi kecepatan dan juga berfungsi sebagai lampu hazard atau lampu darurat, dimana lampu hazard ini aktif ketika mobil berjalan mundur.



gambar 3. 16 Rangkaian yang menghubungkan ESP32 wemos lolin32 lite dengan lampu belakang

BAB IV
PENGUJIAN DAN ANALISA

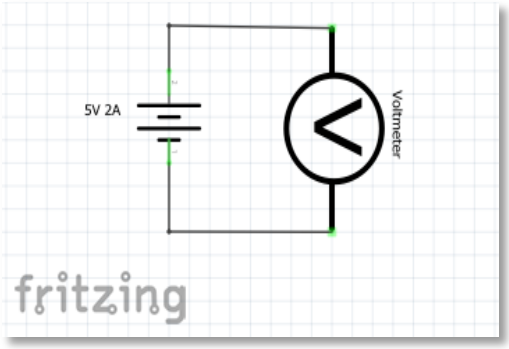
Pengujian alat ini dilakukan agar dapat mengetahui kinerja setiap komponen dan tahapan – tahapan proses apakah dapat berjalan sesuai dengan yang telah direncanakan. Dari hasil pengujian akan dianalisa setiap proses dan tahapan-tahapan yang ada pada alat cruise control pada kendaraan listrik dengan metode kendali PID.

4.1. Pengujian Catu Daya

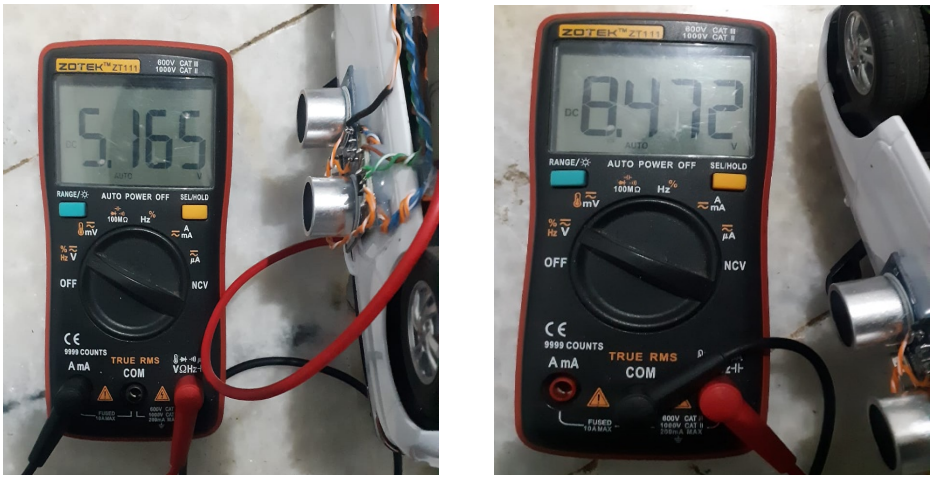
Catu daya atau power supplay merupakan bagian yang tidak dapat dipisahkan dari rancang bangun alat untuk sistem cruise control pada kendaraan listrik dengan metode kendali PID. Pada pengujian catu daya dibagi menjadi 2 pengukuran, yaitu pengukuran tanpa beban dan pengukuran dengan beban.

4.1.1. Pengujian Tanpa Beban

Sebagai sumber daya pada mikrokontroler dan komponen lainnya yang menunjang sistem kendali cruise control pada kendaraan listrik dengan metode kendali PID. maka diperlukan catu daya atau power supply untuk mengoperasikannya. Tahap awal pengukuran tanpa beban pada Gambar 4.2 berikut:



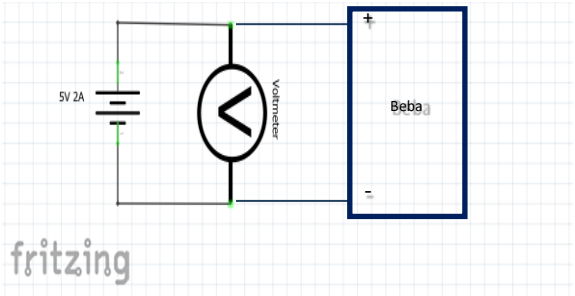
Gambar 4. 1 Rangkaian Pengukuran Tegangan Tanpa Beban Menggunakan Multimeter



Gambar 4. 2 (a) Pengujian Output Catu Daya 5VDC Tanpa Beban, dan (b) Pengujian Output Catu Daya 8.4VDC Tanpa Beban

4.1.2. Pengujian Dengan Beban

Pengujian Catu daya dengan beban dilakukan saat motor di pada kecepatan putaran maksimal. Pengukuran catu daya dengan beban dapat dilihat pada Gambar 4.4. Hal ini untuk menganalisa kestabilan dari catu daya.



Gambar 4. 3 Rangkaian Pengukuran Tegangan Dengan Beban Menggunakan Multimeter

(a) (b)

Gambar 4. 4 (a) Pengujian Output Catu Daya 5VDC Dengan Beban, dan (b) Pengujian Output Catu Daya 8.4VDC Dengan Beban,

Pengujian Catu daya dengan beban dilakukan pengukuran diulang sebanyak 5 kali dengan jeda waktu 5 detik dengan mengamati jatuh tegangan arus yang mengalir saat sensor dan aktuator menyala. Tabel 4.1 merupakan hasil dari 5 kali pengukuran catu daya 5 VDC dengan dan tanpa beban.

Tabel 4. 1 Hasil Rata-Rata Pengujian Tegangan Catu Daya 5 VDC

Percobaan	**Output tanpa beban**	**Output dengan beban**	**Error (%)**
1	5,16 VDC	5,12 VDC	0,77 %
2	5,16 VDC	5,13 VDC	0,58 %
3	5,16 VDC	5,12 VDC	0,77 %
4	5,16 VDC	5,13 VDC	0,58 %
5	5,16 VDC	5,12 VDC	0,77 %
Rata-rata	**5,16 VDC**	**5,12 VDC**	**0,77 %**

Dari hasil pengujian catu daya 5 VDC dan yang telah dilakukan, didapat nilai rata-rata tegangan **5,16 VDC** tanpa beban dan **5,12 VDC** dengan beban dengan persentase kesalahan **0,77%**.

Pengujian pengukuran catu daya 8,4 VDC dengan beban, dapat dilihat pada tabel 4.2 berikut

Tabel 4. 2 Hasil Rata-Rata Pengujian Tegangan Catu Daya 8,4 VDC

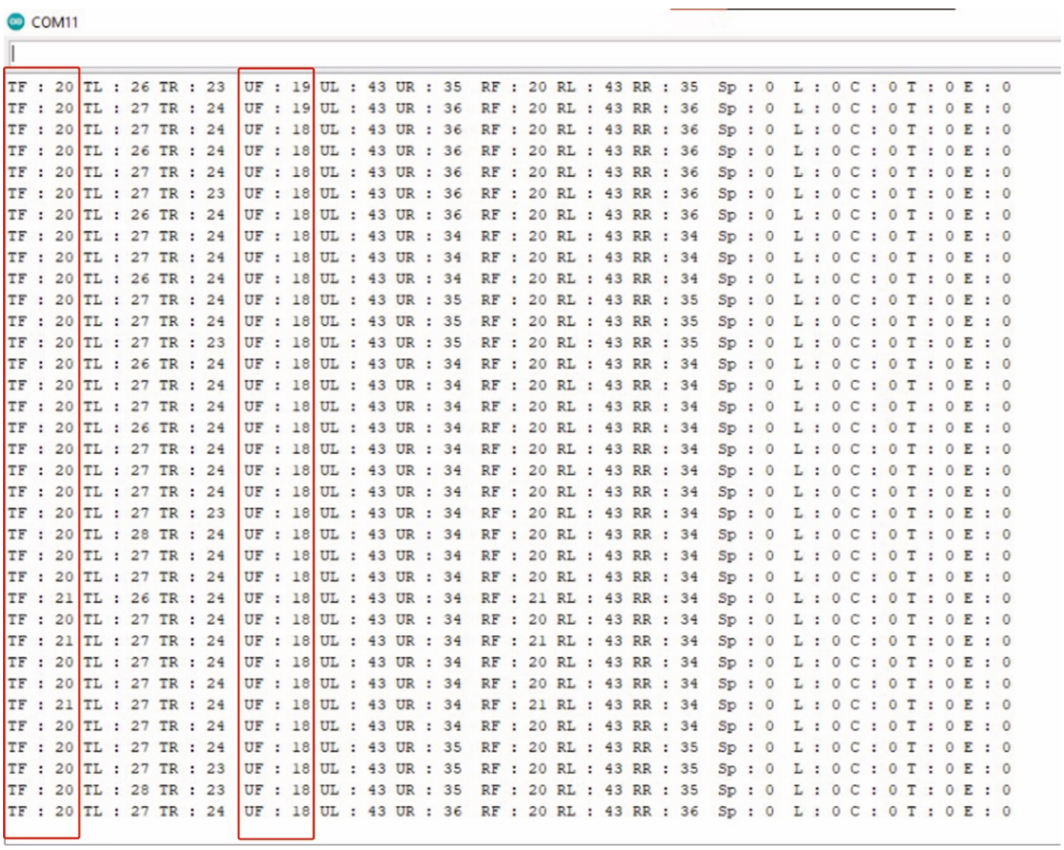
Percobaan	Output tanpa beban	Output dengan beban	Error (%)
1	8,47 VDC	8,45 VDC	0,23 %
2	8,47 VDC	8,46 VDC	0,11 %
3	8,47 VDC	8,45 VDC	0,23 %
4	8,47 VDC	8,45 VDC	0,23 %
5	8,47 VDC	8,46 VDC	0,11 %
Rata-rata	**8,47 VDC**	**8,45 VDC**	**0,23 %**

Dari hasil pengujian catu daya 8,4 VDC yang telah dilakukan, didapat nilai rata-rata tegangan **8,47 VDC** tanpa beban dan **8,45 VDC** dengan beban dengan persentase kesalahan **0,23%**.

4.2.Pengujian Sensor Ultrasonik HC-SR04 dan Sensor Time of Flight VL53L0X

Sensor ultrasonik HC-SR04 dan Sensor Time of Flight VL53L0X adalah sensor yang digunakan dalam rancang bangun prototype cruise control pada kendaraan listrik dengan metode kendali PID. Kedua sensor ini berfungsi sebagai pengukuran jarak mobil terhadap objek. Dimana mikrokontroller akan membandingkan kedua data tersebut untuk dipilih data jarak yang paling akurat. Pengujian dilakukan dengan cara pengambilan sampel menggunakan penggaris untuk menentukan berapa jarak jangkauan sensor dalam mendeteksi adanya objek. Seperti gambar yang terlihat di bawah ini:

Gambar 4. 5 Pengujian sensor Ultrasonik dan ToF Menggunakan Penggaris dengan Jarak 20 cm



Gambar 4. 6 Data Jarak Sensor Ultrasonik dan ToF yang terbaca dengan Jarak 20 cm

Pada sensor yang terletak pada sisi depan, kanan dan kiri. Masing-masing dilakukan 3 pengukuran dengan jarak yang berbeda-beda dapat dilihat hasil pengukuran pada sensor ultrasonik dan sensor ToF pada tabel 4.3.

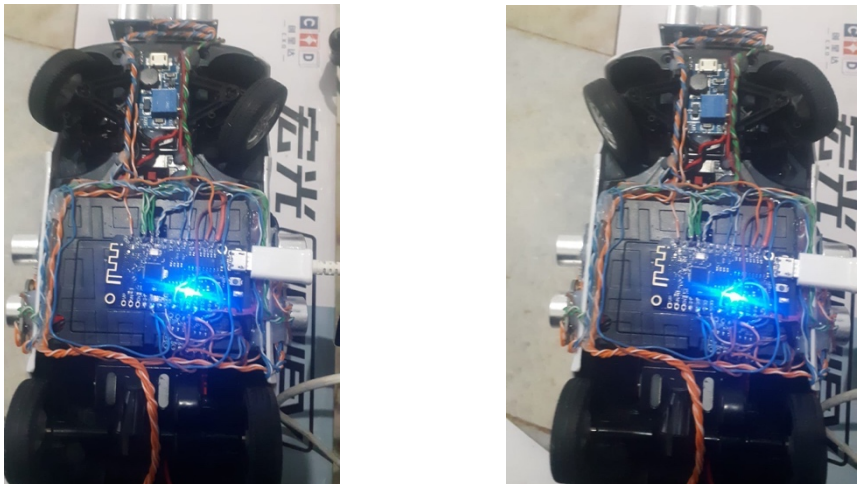
Tabel 4. 3 Pengukuran sensor ultrasonik dan ToF

Posisi Sensor	**Pengukuran Pada Penggaris**	**Hasil Pembacaan Pada Program**	
		Sensor Ultrasonik	**Sensor ToF**
Depan	30 cm	28 cm	30 cm
	20 cm	18 cm	20 cm
	10 cm	8 cm	10 cm
Kanan	25 cm	23 cm	25 cm
	20 cm	18 cm	20 cm
	15 cm	13 cm	15 cm
Kiri	25 cm	23 cm	25 cm
	20 cm	18 cm	20 cm
	15 cm	13 cm	15 cm

Hasil pembacaan pada sensor ultrasonik memiliki pembacaan yang sedikit berbeda dengan sensor ToF dari hasil pembacaan pada program. Dengan nilai perbedaan 2 cm. Hal ini dikarenakan letak sensor ultrasonik sedikit ke depan. Namun pada program dari data jarak yang terbaca, data jarak sensor ultrasonik ditambahkan 2 cm untuk mendapatkan hasil jarak yang lebih akurat. pembacaan dan pengukuran sensor ultrasonik dan sensor ToF lebih detail dapat dilihat pada **Lampiran 1**

4.3. Pengujian Motor Servo

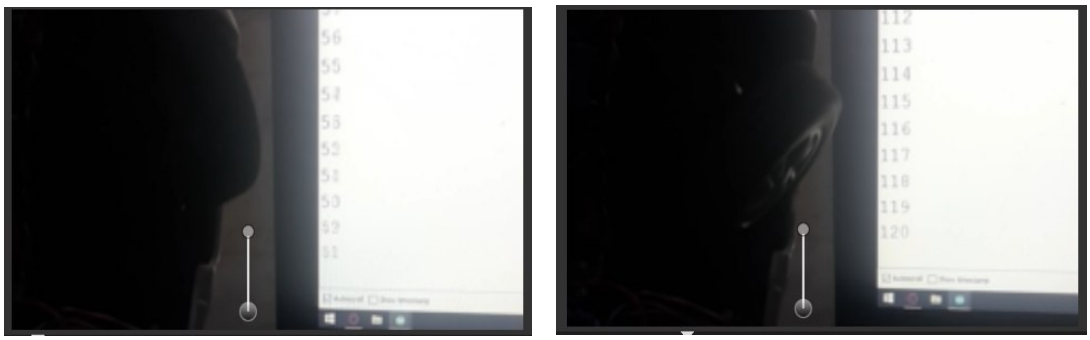
Motor servo pada sistem alat ini digunakan sebagai belok kanan dan kiri serta manuver pada rancang bangun prototype cruise control pada kendaraan listrik dengan metode kendali PID.



(a) (b)

Gambar 4. 7 a) Sudut belok kiri motor servo b) Sudut belok kanan motor servo

Pengukuran motor servo dilakukan dengan cara pembacaan sudut servo pada program saat servo belok kiri dan saat servo belok kanan. Pada pengujian ini di dapat dilihat sebagai berikut:



Gambar 4. 8 a) Sudut servo yang terbaca saat belok kanan b) Sudut servo yang terbaca saat belok kiri

Dari hasil pengujian motor servo di dapat di dapat nilai sudut saat belok kanan **51** dan nilai sudut saat belok kiri **120**.

4.4. Pengujian Jaga Jarak Dengan PID

PID Kontrol pada sistem ini sebagai perhitungan kendali kecepatan pada motor dc, Dimana mobil dapat menjaga jarak atau mengikuti dengan objek di depan sesuai setpoint, yaitu 35 cm. Pada pengujian sistem ini di lakukan sebanyak 5 kali percobaan, dimana mobil akan mengikuti objek yang kemudian akan diberhentikan mendadak. nilai Konstanta PID yang dimasukan adalah Kp = 2 Ki = 0,5 Kd = 0,2. Hasil pengujian jaga jarak dengan PID dapat dilihat pada tabel 4.4.

Tabel 4. 4 Pengujian Jaga Jarak dengan PID

No	Jarak Lintasan (cm)	Hasil	Simpangan
1	75 cm	Berhasil mengikuti objek	-4 cm
2	100 cm	Berhasil mengikuti objek	-3 cm
3	125 cm	Berhasil mengikuti objek	-2,5 cm
4	150 cm	Berhasil mengikuti objek	-2 cm
5	175 cm	Berhasil mengikuti objek	2 cm

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa mobil dapat menjaga jarak terhadap objek dengan kecepatan yang stabil, namun pada saat objek di berhentikan mobil masih mendapati benturan dengan objek dengan nilai jarak simpangan atau *overlaps* yang berbeda beda.

4.5. Pengujian Manuver

Manuver pada sistem ini sebagai penghindar dari objek yang ada di depan saat mobil sudah terlalu dekat dengan objek. Dimana setpoint manuver yang di input yaitu 15 cm. Manuver yang dilakukan berdasarkan pedeteksian sensor yang terdapat pada kanan dan kiri mobil. Pada pengujian sistem ini dilakukan sebanyak 6 kali, dimana 3 dilakukan dengan meletakan pembatas di kiri dan 3 dilakukan dengan meletakan pembatas di kanan. Hasil pengujian dapat di lihat pada tabel 4.5.

Tabel 4. 5 Pengujian Manuver Mobil

No	Letak Pembatas	Start Jarak	Hasil	Simpangan
1	Kiri	40 cm	Berhasil Manuver	13 cm
2		30 cm	Berhasil Manuver	12 cm
3		25 cm	Berhasil Manuver	14 cm
4	Kanan	40 cm	Berhasil Manuver	6 cm
5		30 cm	Berhasil Manuver	10 cm
6		25 cm	Berhasil Manuver	12 cm

Dari hasil pengujian manuver yang telah dilakukan, dinyatakan berhasil. Dimana mobil diberikan kecepatan maksimum dan pembatas diletakan pada sisi kanan dan kiri mobil. Dimana mobil dapat manuver sebelum terbentur objek dibanding dengan saat pengujian jaga jarak dengan PID.

4.6. Pengujian PWM Sebagai Pengatur Kecepatan Motor DC

Pada pengujian ini akan dijelaskan mengenai pengaruh sinyal PWM terhadap perubahan kecepatan motor DC. Pengujian PWM bertujuan untuk mengetahui apakah program pengatur kecepatan motor DC yang telah dibuat dapat mengatur motor DC. Metode pengujian dilakukan dengan cara memberikan nilai *duty cycle* PWM secara bertahap mulai dari nilai terkecil sampai terbesar dan mengamati hasilnya langsung dan menentukan tegangan keluaran sesuai dengan *duty cycle* yang diberikan. Pengujian dilakukan dengan cara membandingkan hasil pengukuran langsung dengan alat ukur multimeter dengan hasil perhitungan. Pengujian dilakukan sebanyak 10 kali, dengan V_{FULL} 7V, *duty cycle* PWM 10%, 20%, 30% sampai 100%.

Tabel 4. 6 Pengujian PWM Pada Motor DC

Pengujian Ke-	V Full (volt)	Duty Cycle PWM %	Nilai PWM	V out (Volt)
1	7 Volt	10 %	25	0,7 V
2		20 %	51	1,4 V
3		30 %	76	2,1 V
4		40 %	102	2,8 V
5		50 %	127	3,5 V
6		60 %	153	4,2 V
7		70 %	178	4,9 V
8		80 %	204	5,6 V
9		90 %	229	6,3 V
10		100 %	255	7 V

Dari hasil pengujian dapat disimpulkan bahwa semakin besar duty cycle pada sinyal PWM yang diberikan, maka tegangan yang masuk ke motor akan semakin besar. Dari data tersebut terbukti bahwa putaran motor menjadi semakin cepat. Begitu juga sebaliknya semakin kecil duty cycle pada sinyal PWM yang diberikan, makan tegangan yang masuk ke motor semakin kecil. Dari data tersebut terbukti bahwa putaran motor akan semakin lambat.

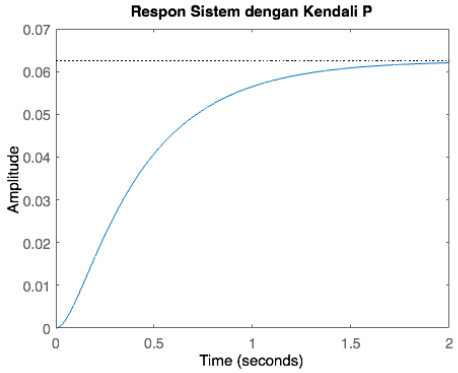
4.7. Menentukan Nilai PID dengan Tuning Kontrol Matlab

Dalam sistem ini yang dikendalikan adalah kecepatan. Parameter yang diukur atau dipantau adalah jarak hasil dari pembacaan sensor . Parameter tersebut diolah dan dikendalikan menggunakan metode pengendalian PID. *software* yang digunakan untuk menampilkan uji coba nilai gain adalah Matlab. Software ini memudahkan untuk melakukan perhitungan – perhitungan tertentu. Metode yang digunakan untuk menentukan nilai gain dari masing – masing komponen PID pun sama yaitu metode *tuning "Trial and Error"* yang merupakan sebuah metode yang digunakan untuk menentukan nilai gain dari masing – masing komponen

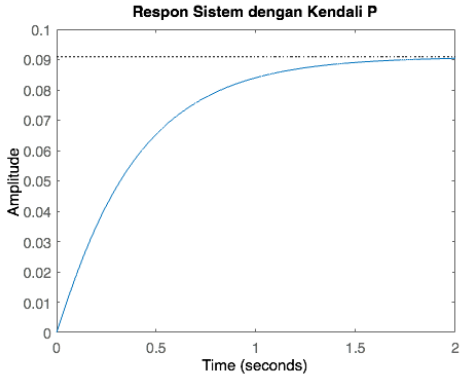
PID berdasarkan hasil uji coba dengan menetapkan sembarang nilai sebagai nilai gain untuk masing – masing komponen PID tersebut hingga didapatkan hasil keluaran terbaik.

4.7.1. Menentukan Nilai Kp

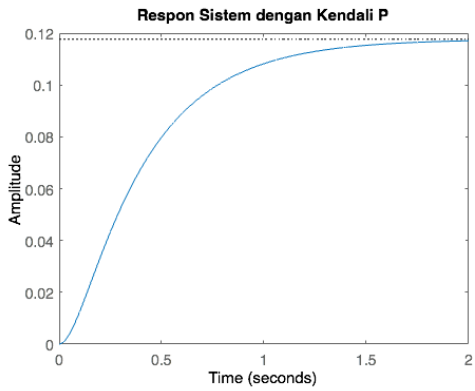
Dalam percobaan ini dilakukan penentuan nilai gain *Proportional* (Kp) dengan nilai 1 sampai 3. Dari percobaan tersebut didapatkan hasil seperti terlihat pada Gambar 4.9, Gambar 4.10, Gambar 4.11, berikut ini.



Gambar 4. 9 KP = 1



Gambar 4. 10 KP = 2

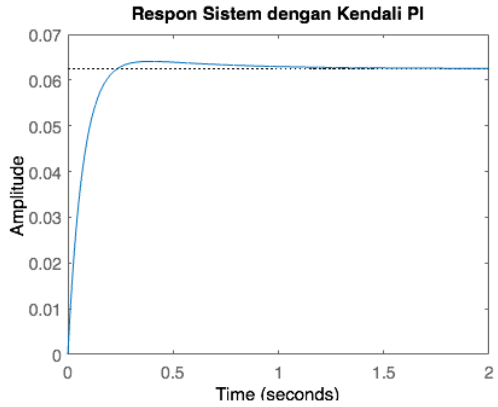


Gambar 4. 11 KP = 3

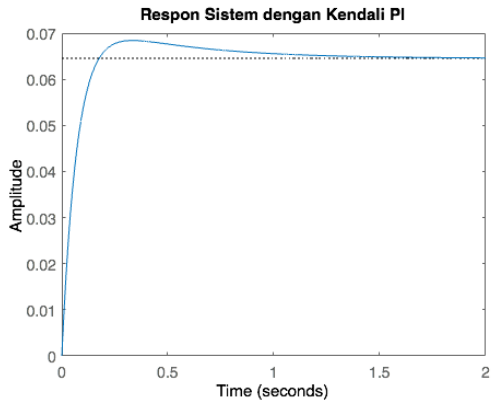
Dari 3 hasil pengujian tersebut, Nilai gain Kp = 2 memiliki repon yang cukup baik dibandingkan dengan respon dengan nilai gain Kp = 1 dan Kp = 3.

4.7.2. Menentukan Nilai Kp dan Ki

Dalam percobaan ini dilakukan penentuan nilai gain *Proportional* (Kp) dengan nilai 2 dan nilai gain *Integral* 0.5 sampai 1. Dari percobaan tersebut didapatkan hasil seperti terlihat pada Gambar 4.12, Gambar 4.13. berikut ini.



Gambar 4. 12 KP = 2 dan KI = 0.5

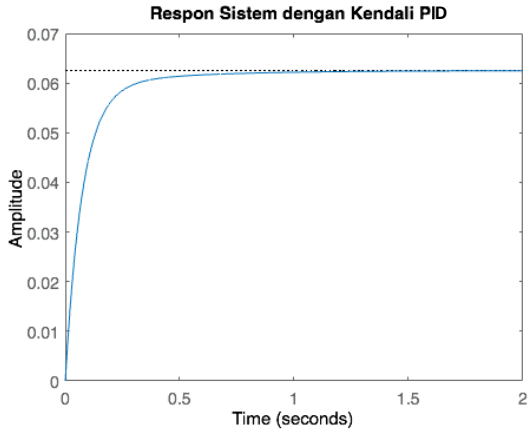


Gambar 4. 13 KP = 2 dan KI = 1

Hasil dari percobaan penentuan nilai *Proportional* (Kp) dan *Integral* (Ki) penggunaan kontrol Kp dan Ki didapat nilai gain Ki = 0.5 dimana dapat meminimalisir *overshoot* dan *settling time*.

4.7.3. Menentukan Nilai Kp, Ki dan Kd

Dalam percobaan ini dilakukan penentuan nilai gain Proportional (Kp) dengan nilai 2, nilai gain integral (Ki) 0.5 dan nilai gain Derivative (Kd) 0.2. Dari percobaan tersebut didapatkan hasil seperti terlihat pada Gambar 4.14. berikut ini.



Gambar 4. 14 KP = 2 KI = 0.5 KD = 0.2

Hasil dari percobaan penentuan nilai *Proportional* (Kp), *Integral* (Ki) dan Derivative (Kd) penggunaan kontrol Kp, Ki dan Kd didapat respon yang baik yang dapat menghilangkan *overshoot* dan *settling time* yaitu dengan nilai Kp = 2 Ki = 0,5 dan Kd = 0.2.

4.8. Pengujian Kestabilan Kecepatan Motor DC

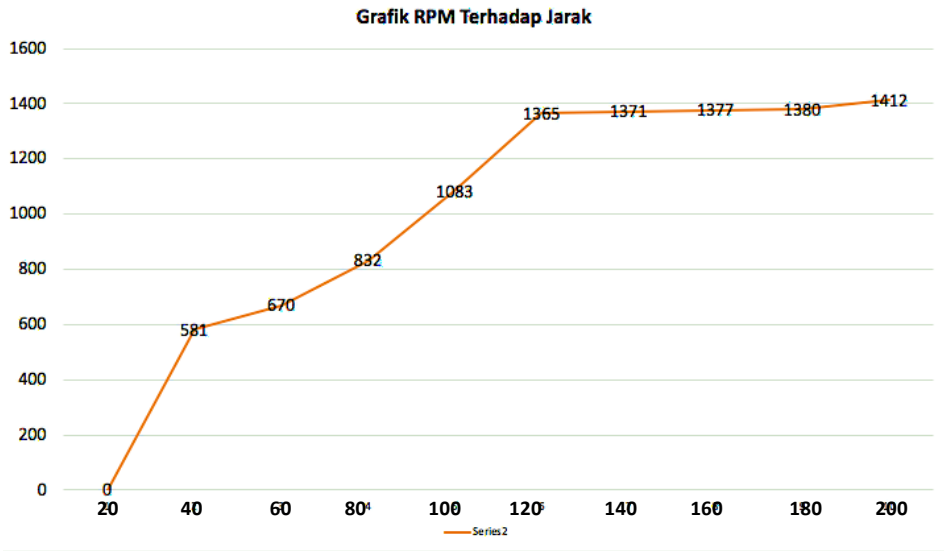
Pada prototype sistem cruise control pada kendaraan listrik dengan metode kendali PID ini, yang di kendalikan adalah kecepatan pada motor dc agar mendapatkan kecepatan stabil dengan parameter jarak. Dimana mobil akan menjaga jarak secara stabil dengan objek yang berada didepan sesuai setpoint yang telah ditentukan.

Pada pengujian ini, untuk mengetahui kestabilan motor dc, dilakukan dengan membaca nilai RPM dari masing masing jarak yang berbeda-beda dengan dilakukan sebanyak 10 kali, untuk mendapatkan nilai kestabilan pada motor dc. Adapun hasil pengujian dapat dilihat pada tabel 4.7.

Tabel 4. 7 Pengujian RPM Motor DC

Pengujian Ke	Jarak	RPM
1	20 cm	0
2	40 cm	581
3	60 cm	670
4	80 cm	832
5	100 cm	1083
6	120 cm	1365
7	140 cm	1371
8	160 cm	1377
9	180 cm	1380
10	200 cm	1412

Pada hasil pengujian tersebut, dapat disimpulkan bahwa kestabilan kecepatan mobil ada pada nilai rata rata **1007 Rpm**. Untuk melihat grafik pada pengujian ini, dapat dilihat pada gambar 4.15 berikut.



Gambar 4. 15 Grafik RPM Terhadap Jarak

BAB V
KESIMPULAN

Berdasarkan hasil perancangan, analisis data dan pengujian yang telah dilakukan dapat diperoleh kesimpulan tentang kinerja dari sistem yang telah dibuat, yaitu sebagai berikut :

1. Sistem cruise control pada alat ini, mobil dapat menjaga jarak dengan objek yang berada di depan dengan kecepatan stabil 1007 RPM melalui perhitungan PID dengan setpoint 35 cm dan nilai gain Kp = 2 Ki = 0,5 Kd = 0,2.
2. Sistem manuver mampu bekerja pada jarak 15 cm dengan pendeteksian objek pada sisi kiri mobil, saat jarak kiri mobil terhadap objek < 20 cm maka mobil akan manuver ke kanan dan saat jarak kiri > 20 cm mobil akan manuver ke kiri.
3. Sistem cruise control pada keseluruhan alat ini mampu berjalan baik, dengan kecepatan stabil 1007 RPM pada prototype yang telah di rancang, kontrol keseluruhan sistem alat ini menggunakan aplikasi blynk IoT yang terhubung dengan WeMos LoLin32 Lite.

DAFTAR PUSTAKA

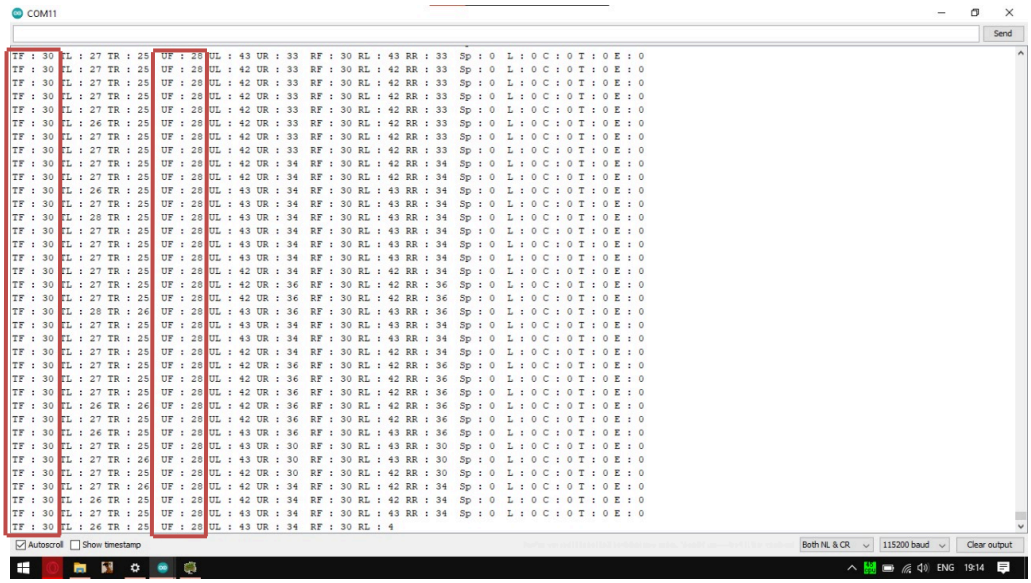
1. Putra Eka Purnama. (2013). PID (Proportional–Integral–Derivative) controller. https://putraekapermana.wordpress.com/2013/11/21/pid/
2. Agus Faudin. (2017). Tutorial Arduino mengakses driver motor L298N. https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/
3. Politeknik Negeri Sriwijaya. Sensor ultrasonik: Pengertian dan prinsip kerja. http://eprints.polsri.ac.id/1803/3/BAB%20II.pdf
4. Renzo Mischianti. (Juli 2021). ESP32 Wemos LoLin32 Lite pinout dan spesifikasi resolusi tinggi**.** https://www.mischianti.org/2021/07/30/esp32-wemos-lolin32-lite-high-resolution-pinout-and-specs/
5. Simanjuntak. (2017). Motor Dc: Pengertian, Prinsip kerja dan Bagian - bagian motor DC**.** http://eprints.polsri.ac.id/4649/4/BAB%20II%20%20LA.pdf
6. Fakhmi. M (2021) Rancang Bangun Alat Pengantrian Puskesmas Menggunakan Rfid Berbasis Iot. Institut Sains dan Teknologi Nasional.
7. Febriansyah. M (2022) Implementasi arm robot pada smart farming berbasis Internet of Things. Jurnal Techno.com Jilid 21 No.4 hal 927-934.
8. Sirujul (2017) Rancang Bangun Prototipe Sistem Konvoi Otomatis Pada Robot Mobile Dengan Sistem Linear Platooning Dan Adaptive Cruise Control Berbasis Mikrokontroler. ISTN Jakarta
9. Syahwil, Muhammad (2013). Mengenal Mikrokontroller. https://derfaule.wordpress.com/artikel/mengenal-mikrokontroler/
10. Irfan Permana (2013). Sistem kontrol, Diagram sistem kontrol dan contoh-contoh sistem kontrol http://insyaansori.blogspot.com/2013/02/sistem-kontrol.html
11. Fina Supegina (2017). Aplikasi Blynk IoT. Jurnal teknologi elektro. Universitas Mercu Buana. http://insyaansori.blogspot.com/2013/02/sistem-kontrol.html
12. Fahmizal (2019). Servo controller circuit using IC NE555. Sekolah Vokasi

Universitas Gajah Mada. https://otomasi.sv.ugm.ac.id/2019/12/25/servo-controller-circuit-using-ic-ne555/

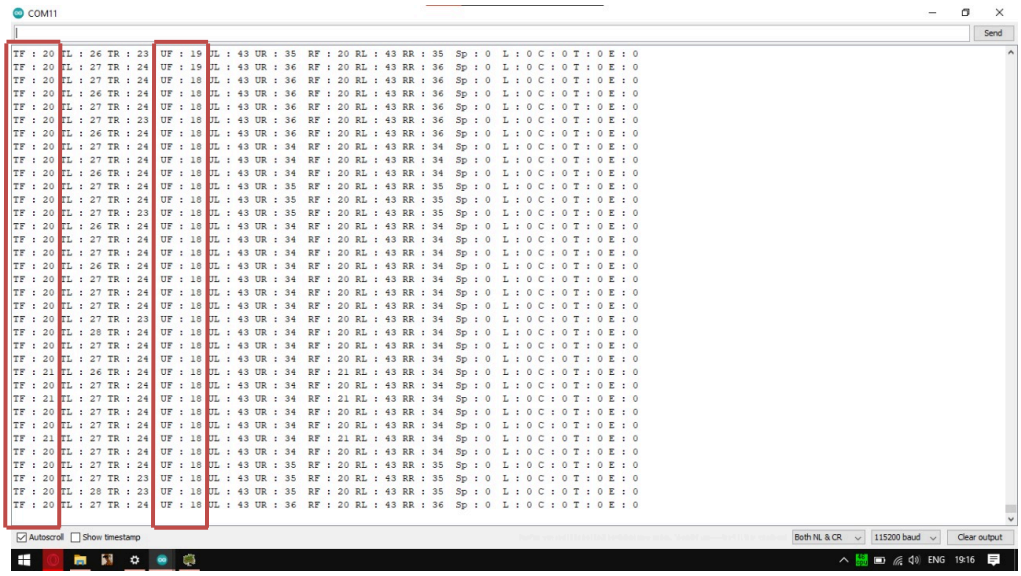
13. Katsuhiko Ogata (2010). Modern Control Engineering.
14. Reza Muhardian (2020). Kendali kecepatan motor dc dengan controller PID dan Antarmuka Visual Basic. Teknik elektro. Universitas Negeri Padang. http://ejournal.unp.ac.id/index.php/jtev/article/view/108034/103133
15. Sujarwata. (2013). PENGENDALI MOTOR SERVO BERBASIS MIKROKONTROLER. *Angkasa*, 8.
16. Fajar Lutfialayubi (2015). Perancangan sistem kontrol pengereman motor dc secara eleltris pada mobil listrik. Teknik elektro. Universitas brawijaya malang.
17. Karl J. Astrom and Tore Hagglund, 1934. 2nd edition. *PID Controllers: Theory, Design and Tuning.*
18. Elang Sakti. (2015, Desember 13). *Cara Kerja Sensor Ultrasonik, Rangkaian, & Aplikasinya*. Diambil kembali dari Elangsakti.com: https://www.elangsakti.com/2015/05/sensor-ultrasonik.html
19. Dr. Zikri Noer, S. M. ((Oktober 2021)). *Buku Sistem Kontro.* Indonesia: GUEPEDIA.
20. Kho, D. (2021, 09 27). *Pengertian Mikrokontroler (Microcontroller) dan Strukturnya*. Diambil kembali dari Teknik Elektronika: https://teknikelektronika.com/pengertian-mikrokontroler-microcontroller-struktur-mikrokontroler/
21. Cruise control: PID Controller design https://ctms.engin.umich.edu/CTMS/index.php?example=CruiseControl&section=ControlPID
22. R. H. Sianipar (2019). Dasar Sistem Kontrol Matlab. Model matematis, Sistem kontrol linear, Simulink: Sistem Tak-linear dan Kontroler berbasis model.

Lampiran 1

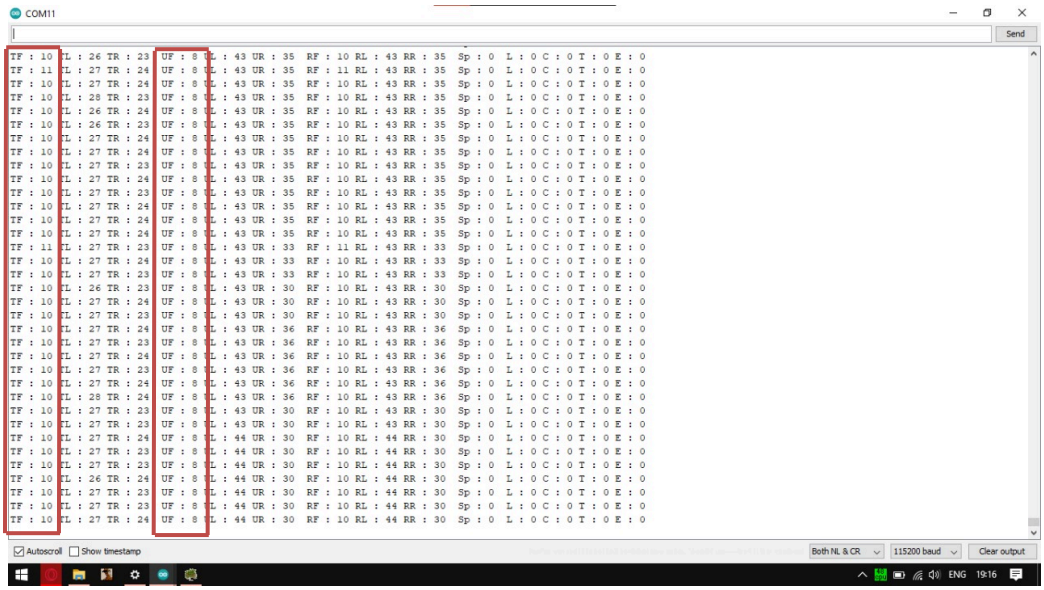
Hasil Pengujian Sensor Ultrasonik dan ToF Depan 30 cm



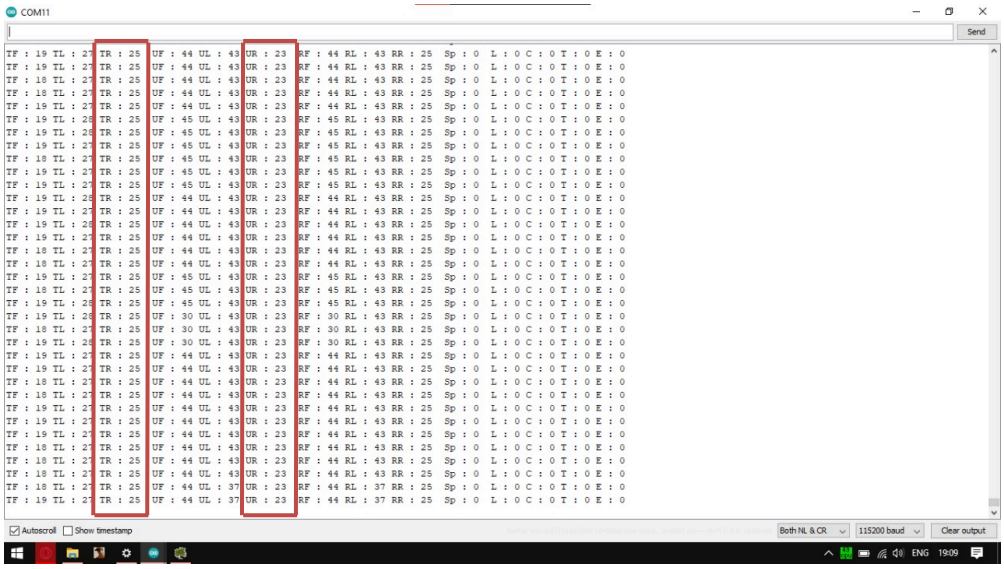
Hasil Pengujian Sensor Ultrasonik dan ToF Depan 20 cm



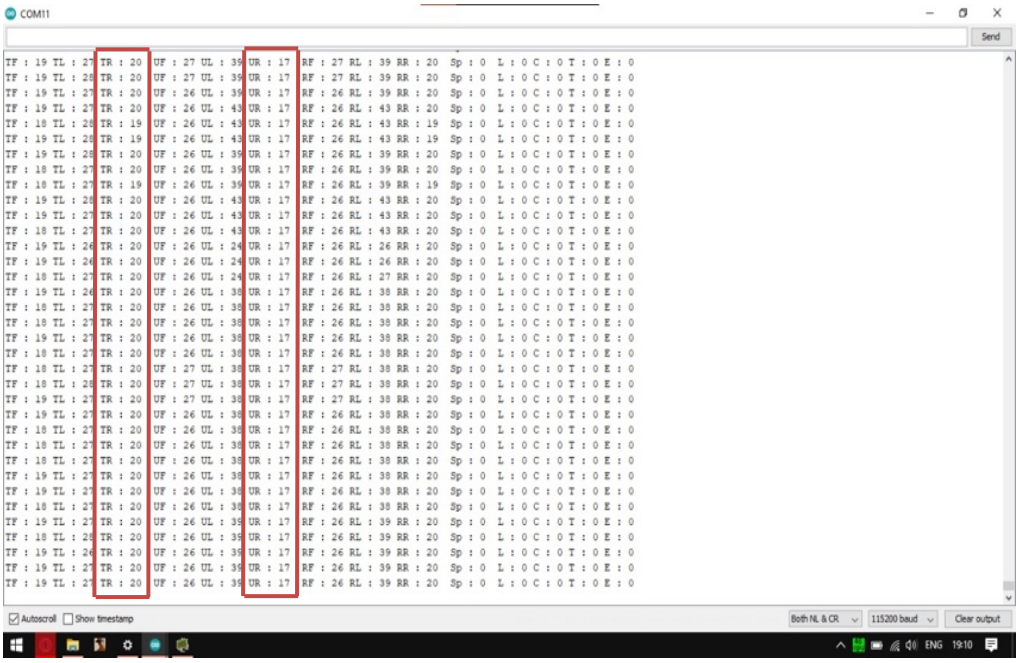
Hasil Pengujian Sensor Ultrasonik dan ToF Depan 10 cm



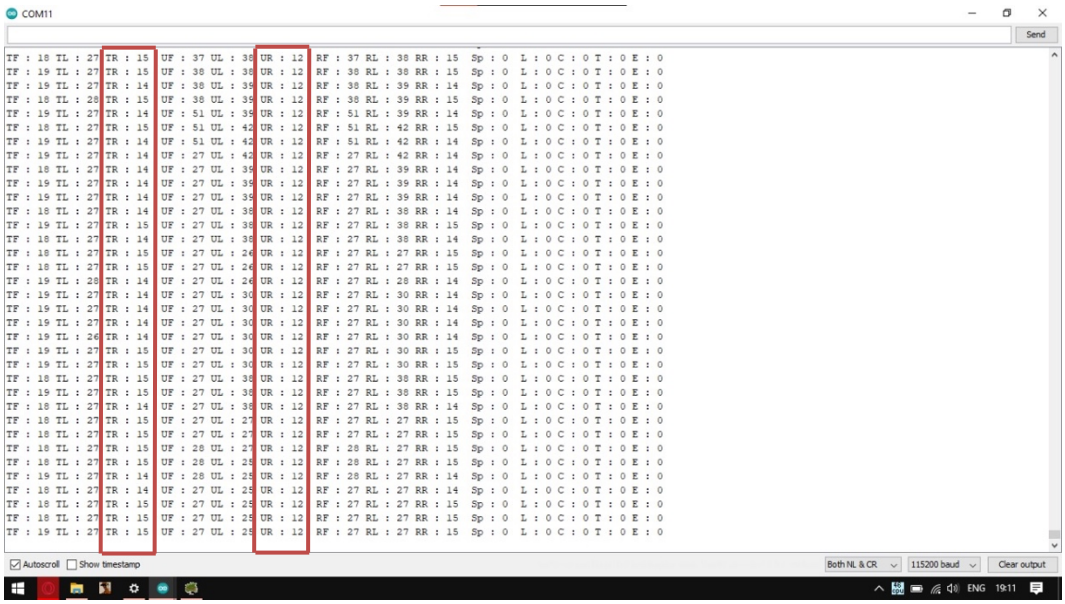
Hasil Pengujian Sensor Ultrasonik dan ToF Kanan 25 cm



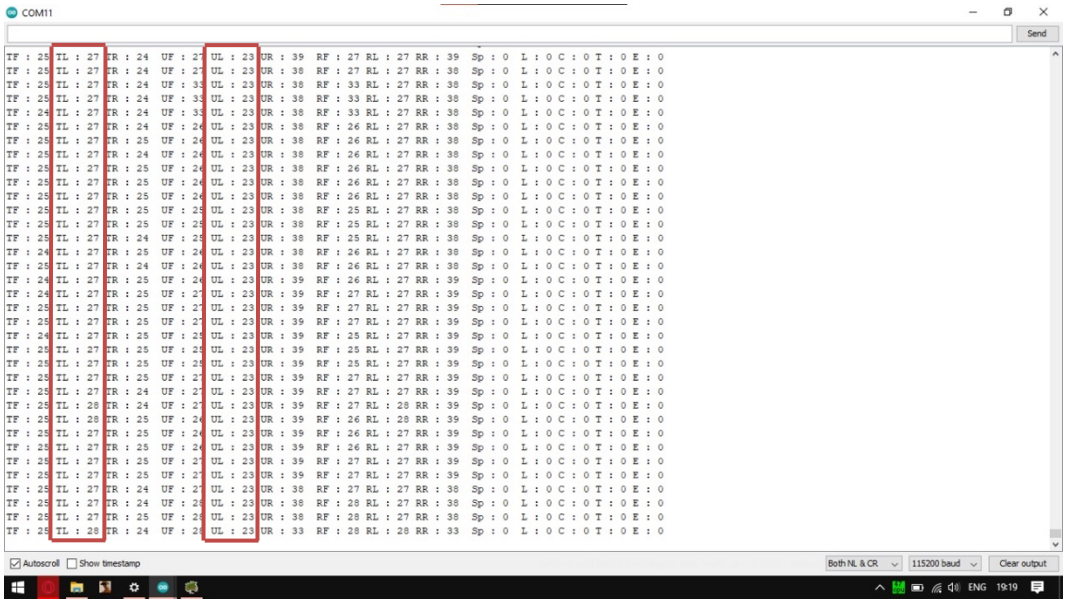
Hasil Pengujian Sensor Ultrasonik dan ToF Kanan 20 cm



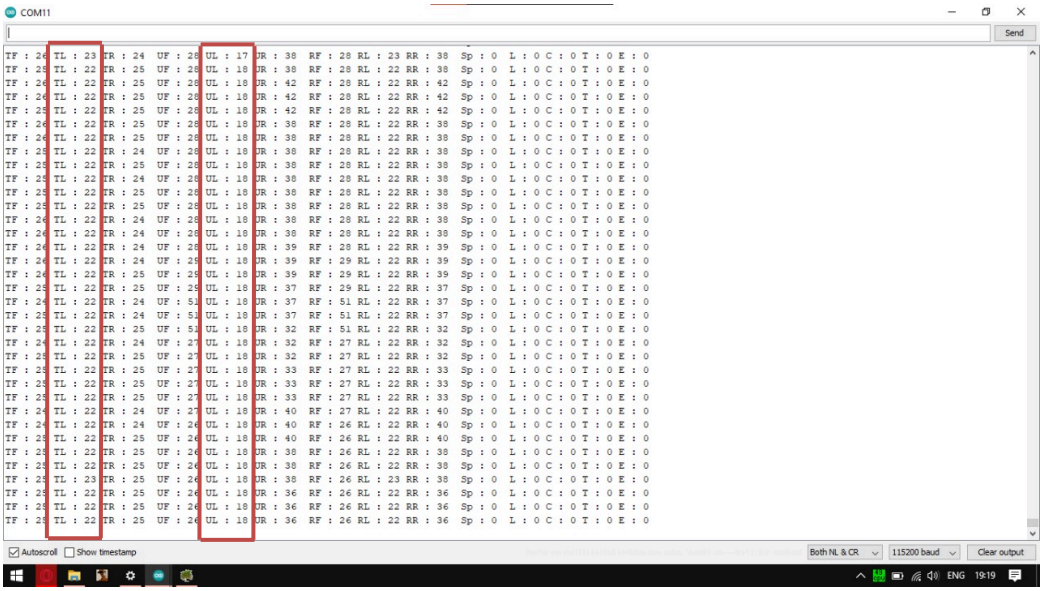
Hasil Pengujian Sensor Ultrasonik dan ToF Kanan 15 cm

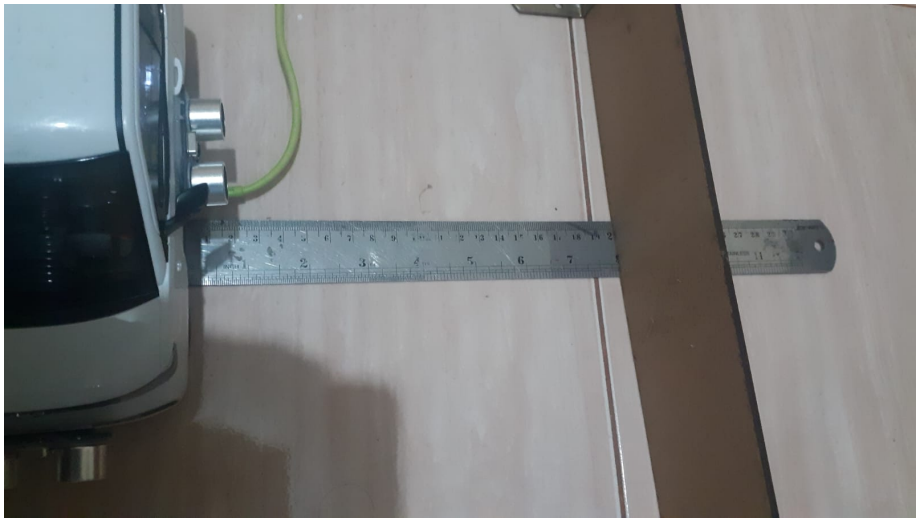


Hasil Pengujian Sensor Ultrasonik dan ToF Kiri 25 cm

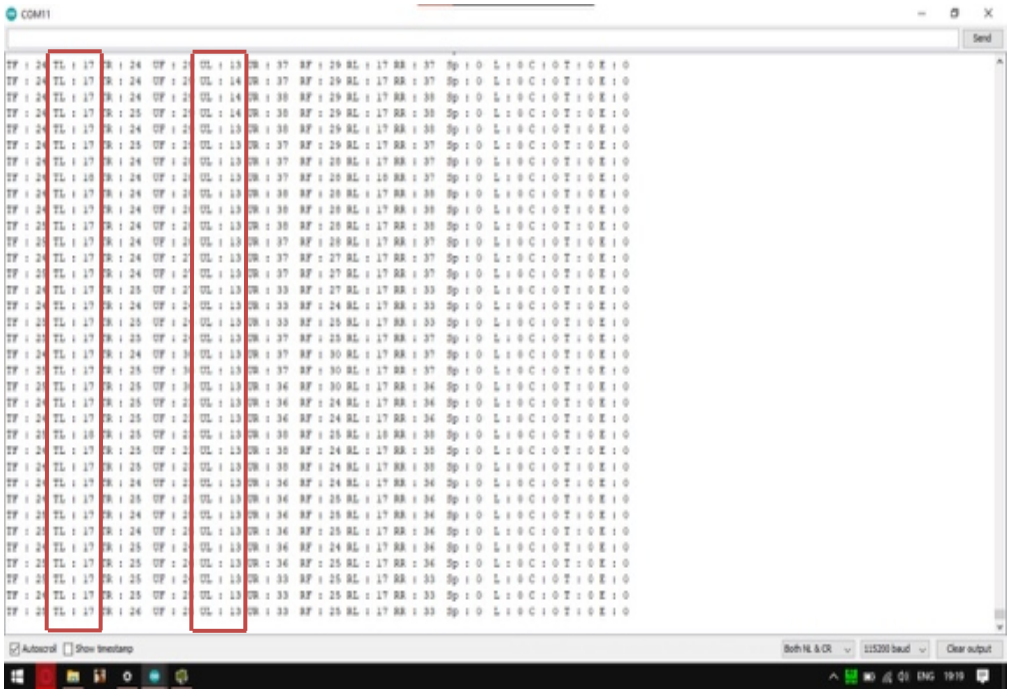


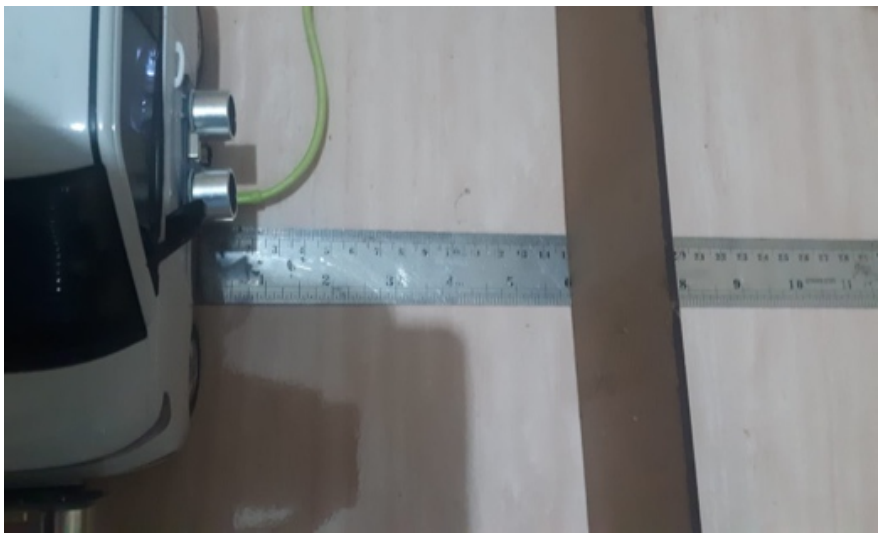
Hasil Pengujian Sensor Ultrasonik dan ToF Kiri 20 cm





Hasil Pengujian Sensor Ultrasonik dan ToF Kiri 15 cm





Lampiran 2

Program PID

```
#include <Preferences.h>
Preferences preferences;
#include "ArduPID.h"
ArduPID myController;
double input;
double output;
double setpoint = 17;
double p = 2;
double i = 0.5;
double d = 0.2;
 #define L_Depan 15
#define L_Belakang 33
#define M1 25
#define M2 26

#include <NewPing.h>
#define SONAR_NUM 3
#define MAX_DISTANCE 200
NewPing sonar[SONAR_NUM] = {
  NewPing(18, 16, MAX_DISTANCE),
  NewPing(4, 35, MAX_DISTANCE),
  NewPing(18, 35, MAX_DISTANCE)
};
void setup() {
Serial.begin(115200);
pinMode(L_Depan, OUTPUT);
pinMode(L_Belakang, OUTPUT);
pinMode(M1, OUTPUT);
pinMode(M2, OUTPUT);
preferences.begin("Settings", false);
```

```
setpoint = preferences.getDouble("setpoint", 0);
//p = preferences.getDouble("p", 0);
//i = preferences.getDouble("i", 0);
//d = preferences.getDouble("d", 0);
myController.begin(&input, &output, &setpoint, p, i, d);
myController.reverse();
 // Uncomment if controller output is "reversed"   //
myController.setSampleTime(10);
// OPTIONAL - will ensure at least 10ms have past between successful compute()
calls   myController.setOutputLimits(0, 255);
myController.setBias(255.0 / 255.0);
myController.setWindUpLimits(-10, 10);
// Groth bounds for the integral term to prevent integral wind-up
myController.start();
 // myController.reset();
            // Used for resetting the I and D terms - only use this if you know what
you're doing   // myController.stop();
 // Turn off the PID controller (compute() will not do anything until start() is
called)   digitalWrite(L_Depan, OUTPUT);
 d\igitalWrite(L_Belakang, OUTPUT);
   }

void loop(){
   input = sonar[0].ping_cm() + 2;
   myController.compute();
//myController.debug(&Serial, "myController", PRINT_INPUT    |
// Can include or comment out any of these terms to print   //
PRINT_OUTPUT   |
// in the Serial plotter   //

PRINT_SETPOINT);
//|
```

```
 //PRINT_BIAS     |
//PRINT_P        |
//PRINT_I        |
//PRINT_D
 //analogWrite(3, output);
   analogWrite(M1, output);
   digitalWrite(M2, LOW);
   readSerial();
   Serial.println(output);
 }
  void readSerial(){
   if(Serial.available()){
     String incoming = Serial.readString();
     incoming.trim();
     Serial.println(incoming);
     String par;
     int num;
     if(incoming.charAt(0)=='`'){
       par = incoming.charAt(1);
       num = String(incoming.substring(3, incoming.length())).toInt();
       Serial.printf("Setting %s to : %d", par, num);
       Serial.println();       if(par=="p"){         p = num;
preferences.putDouble("p", num);
         delay(10);
         ESP.restart();
       }
       else if(par=="i"){         i = num;
         preferences.putDouble("i", num);
         delay(10);
         ESP.restart();
       }
       else if(par=="d"){         d = num;
```

```
      preferences.putDouble("d", num);
      delay(10);
      ESP.restart();
     }
     else if(par=="s"){
      setpoint = num;
      preferences.putDouble("setpoint", num);
     delay(10);
      ESP.restart();
     }       else{
      Serial.println("Unknown Parameter");
     }
    }
    else if(incoming.equalsIgnoreCase("show parameter")){
     double pp = preferences.getDouble("p", 0);
     double ip = preferences.getDouble("i", 0);
     double dp = preferences.getDouble("d", 0);
     Serial.printf("P = %d, I = %d, D = %d", pp, ip, dp);
    }
   }
}

#include "Adafruit_VL53L0X.h"
#define LOX1_ADDRESS 0x30
#define LOX2_ADDRESS 0x31
#define LOX3_ADDRESS 0x32
#define SHT_LOX1 14
#define SHT_LOX2 12
#define SHT_LOX3 27
Adafruit_VL53L0X lox1 = Adafruit_VL53L0X();
Adafruit_VL53L0X lox2 = Adafruit_VL53L0X();
```

```
Adafruit_VL53L0X lox3 = Adafruit_VL53L0X();
VL53L0X_RangingMeasurementData_t measure1;
VL53L0X_RangingMeasurementData_t measure2;
VL53L0X_RangingMeasurementData_t measure3;


 #include <NewPing.h>
 #define SONAR_NUM 3
 #define MAX_DISTANCE 200
NewPing sonar[SONAR_NUM] = {
   NewPing(18, 16, MAX_DISTANCE),
   NewPing(4, 35, MAX_DISTANCE),
   NewPing(18, 35, MAX_DISTANCE)
 };
   #define BLYNK_TEMPLATE_ID
          "TMPLoWigD85C" #define BLYNK_DEVICE_NAME
          "Cruzz" #define BLYNK_AUTH_TOKEN
"h0P2hSBhsgP9pnoFeXqI2kFBG9xh1xK7"
 //#define BLYNK_PRINT Serial
// Comment this out to disable prints and save space

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Min";
char pass[] = "yeyeyeye";

#include <Servo.h>
Servo servo;
#define L_Depan 15
#define L_Belakang 33
```

```
#define M1 25
#define M2 26
#define SPD 17   int v1, v2, v3, v4, v5, v6, v8;
int th; int son; int UF,UR,UL, TF,TR,TL, RF, RR, RL;
bool blinkenable, blinkstat;
BLYNK_WRITE(V1){   v1 = param.asInt();
  digitalWrite(L_Depan, v1);
 }
BLYNK_WRITE(V2){   v2 = param.asInt();
 }BLYNK_WRITE(V5){   v5 = param.asDouble();
  v5 = map(v5, 0, 255, 160, 20);
 }BLYNK_WRITE(V6){   v6 = param.asDouble();
 }BLYNK_WRITE(V8){   v8 = param.asInt();
 }BLYNK_CONNECTED() {   Blynk.syncAll();
 }
   void setID() {
  digitalWrite(SHT_LOX1, LOW);
     digitalWrite(SHT_LOX2, LOW);
   digitalWrite(SHT_LOX3, LOW);
  delay(10);
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, HIGH);
  digitalWrite(SHT_LOX3, HIGH);
 delay(10);
  digitalWrite(SHT_LOX1, HIGH);
  digitalWrite(SHT_LOX2, LOW);
  digitalWrite(SHT_LOX3, LOW);
  if(!lox1.begin(LOX1_ADDRESS)) {
    Serial.println(F("Failed to boot first VL53L0X"));
  }
  delay(10);
  digitalWrite(SHT_LOX2, HIGH);
```

```
  delay(10);
  if(!lox2.begin(LOX2_ADDRESS)) {
    Serial.println(F("Failed to boot second VL53L0X"));
  }
  digitalWrite(SHT_LOX3, HIGH);
  delay(10);
  if(!lox3.begin(LOX3_ADDRESS)) {
    Serial.println(F("Failed to boot third VL53L0X"));
  } } void readTOF() {
    lox1.rangingTest(&measure1, false);
  lox2.rangingTest(&measure2, false);
  lox3.rangingTest(&measure3, false);
  if(measure1.RangeStatus != 4) {
       TR = measure1.RangeMilliMeter/10;
  } else {
    Serial.print(F("Out of range"));
  }
if(measure2.RangeStatus != 4) {
    TL = measure2.RangeMilliMeter/10;
  } else {
    Serial.print(F("Out of range"));
  }
  if(measure3.RangeStatus != 4) {
    TF = measure3.RangeMilliMeter/10;
  } else {
    Serial.print(F("Out of range"));
  }
 }


 #include <TaskScheduler.h>
void readUltra(){
```

```
  if(son==0){UF=sonar[0].ping_cm() + 2;
}else if(son==1){UL=sonar[1].ping_cm() + 2;
}else if(son==2){     UR=sonar[2].ping_cm() + 2;
  }   son++;
 if(son>=SONAR_NUM){     son=0;
  } } void serialreport(){   Serial.print("TF : ");
Serial.print(TF);Serial.print(" ");
  Serial.print("TL : ");
Serial.print(TL);
Serial.print(" ");
  Serial.print("TR : ")
;Serial.print(TR);
Serial.print("  ");
  Serial.print("UF : ");
Serial.print(UF);
Serial.print(" ");
  Serial.print("UL : ");
Serial.print(UL);
Serial.print(" ");
  Serial.print("UR : ");
Serial.print(UR);
Serial.print("  ");
  Serial.print("RF : ");
Serial.print(RF);
Serial.print(" ");
  Serial.print("RL : ");
Serial.print(RL);
Serial.print(" ");
  Serial.print("RR : ");
Serial.print(RR);
Serial.print("  ");
Serial.print("Sp : ");
```

```
Serial.print(th);S
erial.print("  ");
Serial.print("L : ");
Serial.print(v1);
Serial.print(" ");
Serial.print("C : ");
Serial.print(v2);
Serial.print(" ");
Serial.print("T : ");
Serial.print(v3);
Serial.print(" ");
  Serial.print("E : ");
Serial.print(v4);
Serial.print("  ");
   Serial.println();
 }
 void Blinking(){
  if(blinkenable==1){
    digitalWrite(L_Belakang, blinkstat);
    blinkstat = !blinkstat;
  }
 }
 void Timer10(){
  readUltra();
  //readTOF();
  if((UF!=0)&&(TF!=0)){
    if(UF>=TF){RF=UF;}else{RF=TF;}
    if(UL>=TL){RL=UL;}else{RL=TL;}
if(UR>=TR){RR=UR;}else{RR=TR;}
  }
 }
 void Timer100(){
```

```
  Blinking();
  serialreport();
}
void Timer1000(){
  Blynk.virtualWrite(V20, RF);
  Blynk.virtualWrite(V22, RL);
  Blynk.virtualWrite(V21, RR);
}
void moveit(){
  if((v2==1)&&(RF>=10)){
  //Cruise On     int maxspeed = map(v8, 0, 255, 255, 75);
    analogWrite(SPD, maxspeed);
    digitalWrite(M1, HIGH);
    digitalWrite(M2, LOW);
    blinkenable=0;
    blinkstat = 0;
    digitalWrite(L_Belakang, blinkstat);
  }
   else if((v6>130)&&(RF>=10)){
              //Y Forward     int maxspeed = map(v8, 0, 255, 200, 100);
    th = map(v6, 130, 255, 100, maxspeed);
    analogWrite(SPD, th);
    digitalWrite(M1, HIGH);
    digitalWrite(M2, LOW);
    blinkenable=0;
    blinkstat = 0;
    digitalWrite(L_Belakang, blinkstat);
  }else{
    digitalWrite(M1, 0);
    digitalWrite(M2, 0);
    blinkenable=0;
    blinkstat = 1;
```

```
    th = 0;
    digitalWrite(L_Belakang, blinkstat);
  }
    if(v6<100){
    int maxspeed = map(v8, 0, 255, 100, 125);
    th = map(v6, 100, 0, 100, maxspeed);
    analogWrite(SPD, th);
    digitalWrite(M1, LOW);
    digitalWrite(M2, HIGH);
    blinkenable=1;
  }
  else if((v6<=130)&&(v6>=100)){
    analogWrite(M1, 0);
    analogWrite(M2, 0);
    blinkenable=0;
    blinkstat = 1;
    th = 0;
    digitalWrite(L_Belakang, blinkstat);
  }
  servo.write(v5);
}
Task timer10(10, TASK_FOREVER, &Timer10);
Task timer100(100, TASK_FOREVER, &Timer100);
Task timer1000(1000, TASK_FOREVER, &Timer1000);
Scheduler runner;
 void setup() {   Serial.begin(115200);
  Serial.println("Connecting...");
  Blynk.begin(auth, ssid, pass);
    Serial.println("Setting up PWM");
  Serial.println("Center the servo");
  servo.attach(13);
  servo.write(90);
```

```
  Serial.println("Setting pinMode");
 pinMode(L_Depan, OUTPUT);
 pinMode(L_Belakang, OUTPUT);
 pinMode(M1, OUTPUT);
 pinMode(M2, OUTPUT);
 pinMode(SPD, OUTPUT);
 digitalWrite(M1, LOW);
 digitalWrite(M2, LOW);
 pinMode(SHT_LOX1, OUTPUT);
 pinMode(SHT_LOX2, OUTPUT);
 pinMode(SHT_LOX3, OUTPUT);
   digitalWrite(L_Depan, HIGH);
 digitalWrite(L_Belakang, HIGH);
 delay(100);   digitalWrite(L_Depan, LOW);
 digitalWrite(L_Belakang, LOW);
 delay(100);
  Serial.println("Initializing TOF");
 //setID();
 Serial.println("Initializing timer");
 runner.init();
 runner.addTask(timer10);
 timer10.enable();
 runner.addTask(timer100);
 timer100.enable();
 runner.addTask(timer1000);
 timer1000.enable();
  Serial.println("Sync with Blynk server");
 Blynk.syncAll();
  v5=90;   v6=128;
  Serial.println("Ready");
}
 void loop() {
```

```
  Blynk.run();
  runner.execute();
  moveit();
 }
```

Lampiran 3

Foto Hasil Pengujian