

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Saat ini perpustakaan telah merealisasikan banyak aspek teknologi informasi seperti pembelian buku, koleksi buku, pembaca meminjam dengan barcode, RFID, database dan teknologi lainya yang meningkatkan manajemen dan tingkat layanan perpustakaan. Salah satu teknologi yang digunakan adalah robot perpustakaan, Robot perpustakaan adalah integrasi otomatisasi dan informasi dalam aplikasi harian perpustakaan, yang bisa meningkatkan tingkat otomatisasi perpustakaan.

Pengelolaan perpustakaan adalah suatu tindakan untuk memfasilitasi segala sesuatu yang berhubungan dengan pelayanan perpustakaan, seperti peminjaman buku, pengembalian buku, pengkategorian dan lain-lain. Semua proses tersebut kini dapat berjalan secara otomatis dengan bantuan teknologi yang ada, dan disebut dengan LMS (Library Management System). Library Management System adalah sistem manajemen perpustakaan yang dibantu oleh sistem komputer untuk membantu pelanggan perpustakaan atau pustakawan. Meskipun teknologi LMS telah banyak diterapkan pada perpustakaan dan membantu dalam bidang otomasi manajemen perpustakaan, namun masih terdapat sistem otomasi yang dapat membantu meringankan pekerjaan pustakawan.

Namun saat ini perpustakaan masih kekurangan teknologi untuk membantu dalam otomatisasi di perpustakaan, misalnya penyortiran buku seperti, pemindaian barcode, klasifikasi dan pemeriksaan masih mengandalkan system manual. Kelebihan dari perpustakaan yang menerapkan robot adalah untuk meningkatkan efisiensi pada layanan perpustakaan sehari-hari salah satu contohnya adalah untuk mengurangi tenaga kerja pustakawan dan untuk meningkatkan efisiensi pemilahan buku dan juga penataan buku.

Berdasarkan permasalahan diatas menggunakan robot pada perpustakaan merupakan hal yang sangat penting untuk management perpustakaan yang lebih baik. Dengan adanya robot penata buku otomatis dapat meningkatkan tingkat efisiensi perpustakaan yang lebih baik. Salah satunya adalah penataan buku yang otomatis dan

pasti sesuai dengan tempatnya, salah satu contoh efisiensi lainnya adalah dapat mengurangi tenaga kerja pustakawan. Dan bahkan pada masa depan nantinya diharapkan terdapat perpustakaan dengan sistem otomatis penuh yang sudah tidak memerlukan lagi petugas perpustakaan.

Saat ini, robotika merupakan salah satu sistem yang digunakan di banyak sektor seperti *smarthome*, industri, keamanan, dan kesehatan. Dalam sistem robotika terdapat beberapa aspek penting di dalamnya seperti sistem mekanik, sistem kelistrikan, sistem kecerdasan buatan, dan juga beberapa sistem yang dapat membantu dalam kehidupan manusia. Robot *pick and place* adalah salah satu sistem robotika yang paling populer, *robot pick and place* adalah sistem robotik yang digunakan untuk membantu dalam bidang pengambilan dan penempatan barang. *Robot pick and place* banyak digunakan di industri untuk meningkatkan efektivitas produksi. Dengan banyaknya manfaat dari sistem *robot pick and place* ini, banyak hal yang dapat dikembangkan untuk membantu kehidupan manusia dari sistem robotika ini .

Ada banyak karakteristik robot pick and place yang membuat sistem robot ini sangat populer dan digunakan di banyak sektor industri. Beberapa contoh dari karakteristik ini adalah kualitas dan akurasi yang tepat dari sistem robot pick and place. Keuntungan lain dari robot pick and place ini adalah sistem robot ini dapat diprogram ulang dan dapat digunakan untuk banyak layanan kepada konsumen dengan peningkatan perkembangan teknologi, semakin banyak industri dan konsumen lain yang menggunakan robot pick and place ini untuk sistem otomatisasi mereka di untuk meningkatkan kualitas produksi mereka.

Mobile robot juga merupakan salah satu sistem robotika yang populer saat ini, karena mobile robot merupakan salah satu bidang yang perkembangannya sangat pesat dalam bidang penelitian ilmiah. Karena mobile robot dapat menggantikan beberapa pekerjaan manusia di berbagai bidang seperti keamanan, industri, militer, transportasi, kesehatan, keamanan, pelayanan publik, penjelajahan planet di luar angkasa, dan lain-lain. Salah satu jenis mobile robot yang sangat populer adalah AGV atau *Automatic Guided Vehicle* yang merupakan mobile robot yang dapat bergerak secara otomatis tanpa bantuan

manusia, AGV sangat banyak digunakan di industri dalam membantu pengangkutan barang selama proses produksi, hal ini digunakan untuk mendapatkan hasil produksi yang maksimal dengan waktu proses produksi yang efisien.

### **1.2 Rumusan Masalah**

Berdasarkan penjabaran latar belakang diatas maka ada beberapa rumusan masalah yang dapat diuraikan sebagai berikut :

- 1) Bagaimana sebuah system dapat mengenali kode buku yang berbeda beda?
- 2) Bagaimana seorang pengunjung perpustakaan tidak lagi salah dalam mengembalikan buku pada rak?
- 3) Bagaimana cara sebuah robot dapat otomatis menata buku pada tempatnya yang sesuai dengan kode buku tersebut?

### **1.3 Tujuan Penelitian**

- 1) Membuat system otomatis mobile robot penata buku otomatis pada perpustakaan.
- 2) Menambah efisiensi penataan buku pada perpustakaan
- 3) Penerapan teknologi robotika dalam system management perpustakaan

### **1.4 Luaran Yang Diharapkan**

Luaran yang diharapkan dari pelaksanaan kegiatan ini yaitu terciptanya sistem otomatis robot penata buku di perpustakaan yang dapat memudahkan pembaca dan petugas perpustakaan saat mengembalikan buku pada rak.

### **1.5 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah :

- 1) Membantu tugas seorang petugas perpustakaan dalam bidang management penataan buku
- 2) Memudahkan pengunjung perpustakaan saat pengembalian buku pada rak.
- 3) Dapat meningkatkan kualitas pelayanan pada perpustakaan.

## **1.6 Batasan Masalah**

Adapun batasan masalah penelitian ini adalah sebagai berikut :

- 1) Ukuran maksimal buku yang dapat dibawa robot ini adalah 20 cm x 15 cm x 5cm dengan berat maksimal 1 kg.
- 2) Maksimal ketinggian rak yang dapat dicapai oleh robot ini adalah 1 meter.
- 3) Maksimal buku yang dapat dibawa robot adalah satu buah buku.

## **1.7 Metodologi**

Metodologi dalam pembuatan proyek akhir ini meliputi :

### **1.7.1 Studi Literatur**

### **1.7.2 Perancangan dan Pembuatan Desain Mekanik**

### **1.7.3 Perancangan dan Pembuatan Hardware dan Software**

### **1.7.4 Pengujian Alat**

### **1.7.5 Evaluasi Alat**

## **1.8 Sistematika Penulisan**

Sistematika pembahasan dalam penyusunan buku proyek akhir ini adalah sebagai berikut :

Menguraikan secara singkat latar belakang, tujuan proyek akhir, perumusan masalah, batasan masalah, metodologi dan sistematika penulisan.

## **BAB I PENDAHULUAN**

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Menjelaskan tentang teori – teori yang berkaitan dengan penyelesaian proyek akhir, yang didapatkan dari berbagai sumber – sumber terkait yang berhubungan dengan pembuatan proyek akhir ini.

### **BAB III PERANCANGAN DAN PEMBUATAN SISTEM**

Menjelaskan tentang perancangan sistem, perancangan pengambilan data, serta proses pengolahan data.

### **BAB IV PENGUJIAN DAN ANALISA**

Menjelaskan tentang seluruh hasil mulai dari pengukuran data dari sensor, proses pengiriman data, pengaktifan actuator, hingga visualisasi data ke pengguna dari proyek akhir yang kemudian akan dilakukan analisa untuk dapat disempurnakan pembuatan proyek akhir ini.

### **BAB V PENUTUP**

Menjelaskan kesimpulan dan saran – saran dari hasil penelitian.

### **DAFTAR PUSTAKA**

Pada bagian ini berisi tentang referensi-referensi yang telah dipakai oleh penulis sebagai acuan dan penunjang serta parameter yang mendukung penyelesaian proyek akhir ini baik secara praktis maupun teoritis.



## **BAB II**

# **TINJAUAN PUSTAKA**

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Perpustakaan Umum**

Secara bahasa, kata perpustakaan berasal dari kata pustaka yang berarti kitab, buku, kitab primbon. Kata Pustaka mendapat awalan per dan akhiran-an menjadi perpustakaan. Perpustakaan dalam bahasa inggris dikenal dengan nama library. Library berasal dari kata latin liber atau libri yang artinya adalah buku. Sedangkan dalam bahasa asing lainnya.

Menurut Undang-Undang Perpustakaan No 43 Tahun 2007 pada pasal (1), Perpustakaan umum adalah perpustakaan yang diperuntukkan bagi masyarakat luas sebagai sarana pembelajaran sepanjang hayat tanpa membedakan umur, jenis kelamin, suku, ras, agama, dan status sosial lainnya.



**Gambar 2.1 Perpustakaan Umum**

Sumber : <https://panduanteknisi.com/>

#### **2.2 Pelayanan pada Perpustakaan Umum**

Pelayanan pada hakikatnya adalah serangkaian kegiatan dalam proses pemenuhan kebutuhan melalui aktifitas orang lain, oleh karena itu pelayanan merupakan proses. Sebagai proses, pelayanan berlangsung secara rutin dan berkesinambungan (Moenir, 1995: 27). Ini berarti bahwa pelayanan berlangsung tidak hanya satu kali melainkan berkali-kali, sehingga pengguna dapat merasakan perbedaan antara pelayanan ketika pertama datang ke perpustakaan dengan pelayanan untuk yang kedua kali dan seterusnya.

Terdapat tujuh layanan perpustakaan yaitu layanan silang layan, layanan peminjaman dan pengembalian, layanan rujukan, layanan internet, layanan terjemahan, layanan bimbingan, dan layanan konsultasi. Layanan silang layan adalah hubungan timbal balik antara perpustakaan dengan pengunjung.



**Gambar 2.2 Pelayanan pada perpustakaan**

Sumber :

<https://donyprisma.files.wordpress.com/2012/06/library-services.jpg>

Layanan peminjam dan pengembalian adalah layanan yang minimal harus ada di sebuah perpustakaan. Layanan rujukan adalah layanan yang berhubungan dengan buku yang dijadikan koleksi sebagai referensi untuk dibaca di tempat atau di perpustakaan, dan tidak boleh dipinjamkan. Layanan internet adalah layanan yang diberikan oleh perpustakaan agar pengunjung lebih mudah dalam mencari sebuah informasi. Layanan terjemahan adalah layanan yang terjadi apabila pengunjung mengalami kesulitan dalam bahasa di dalam koleksi buku. Layanan bimbingan adalah layanan untuk memenuhi kebutuhan pengunjung akan informasi.

### **2.3 RFID (Radio Frequency Identification)**

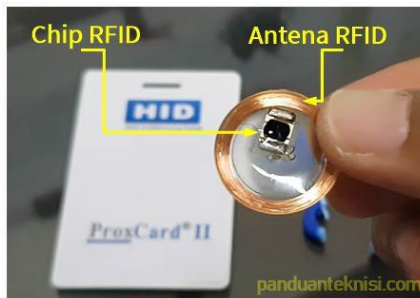
RFID atau Pengenal Frekuensi Radio adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau transponder untuk menyimpan dan mengambil data jarak jauh. Label atau kartu RFID adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio. Label RFID berisi informasi



yang disimpan secara elektronik dan dapat dibaca hingga beberapa meter jauhnya. Sistem pembaca RFID tidak memerlukan kontak langsung seperti sistem pembaca pada barcode.

Pada system RFID terdapat beberapa hardware yang diperlukan. Jika sistem dibuat dengan benar dengan peralatan yang tepat dan telah diujicoba berulang kali, maka sistem bisa mencapai akurasi sampai 100% meskipun benda/objek yang di tag bergerak. Bagian utama pada RFID adalah sebagai berikut :

1. **Reader** - RFID reader adalah perangkat alat yang sangat penting pada sistem RFID. Reader bisa diprogram untuk hanya membaca tag tertentu, waktu tertentu, tergantung dari masing-masing kebutuhan pemakai.
2. **Antenna** – Antenna RFID adalah bagian dari RFID reader dan tag, berfungsi sebagai penerima dan pengirim radio frekuensi dari/ke RFID tag. Pada RFID reader biasanya Antenna terhubung dengan menggunakan kabel coaxial.
3. **Tag** – RFID tag biasanya ditempatkan pada objek yang kita ingin identifikasi, ketika menerima Radio Frekuensi dari RFID reader tag akan membalas sinyal tersebut dengan radio frekuensi juga yang sudah disertai ID dari tag tersebut.

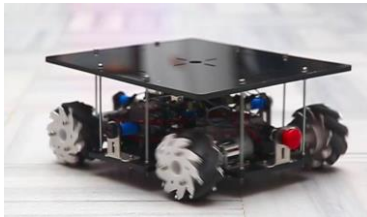


**Gambar 2.3 Bagian-bagian RFID**

Sumber : <https://panduanteknisi.com/pengertian-rfid-adalah.html>

## 2.4 Mobile Robot

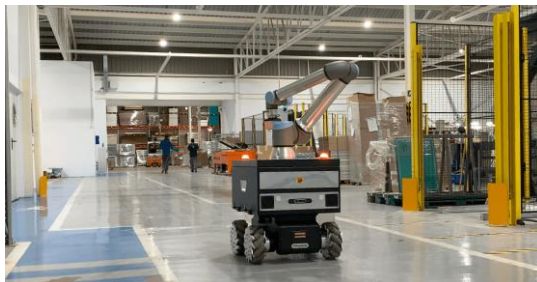
Robot Mobil atau Mobile Robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Base robot mobil dapat dengan mudah dibuat dengan menggunakan plywood /triplek, akrilik sampai menggunakan logam ( aluminium ). Robot mobil dapat dibuat sebagai pengikut garis ( Line Follower ) atau pengikut dinding ( Wall Follower ) ataupun pengikut cahaya.



**Gambar 2.4 Mobile Robot**

Sumber: <https://www.roboticgizmos.com/namla/>

Salah satu system *mobile robot* yang populer saat ini adalah *autonomous mobile robot*, *mobile robot* adalah jenis robot yang dapat bergerak secara otomatis tanpa bantuan manusia. AMR (automatic mobile robot) berbeda dari pendahulunya yaitu AGV (Automatic guided vehicle) yang mengandalkan trek atau jalur yang telah ditentukan sebelumnya dan seringkali memerlukan pengawasan operator.



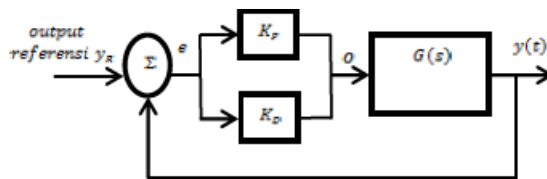
**Gambar 2.5 Automatic Mobile Robot**

Sumber: <https://robotnik.eu/what-is-an-autonomous-mobile-robot-amr-what-robotnik-arm-brings-to-your-business/>

AMR (automatic mobile robot) menggunakan serangkaian sensor canggih, *Artificial Intelligence*, *machine learning*, dan komputasi untuk perencanaan jalur guna menafsirkan dan menavigasi melalui lingkungan mereka. Karena AMR dilengkapi dengan kamera dan sensor, jika mereka mengalami hambatan tak terduga saat menavigasi lingkungan mereka, seperti kotak jatuh atau kerumunan orang, mereka akan menggunakan teknik navigasi seperti menghindari tabrakan untuk memperlambat, menghentikan, atau mengubah rute jalan mereka. objek dan kemudian melanjutkan tugas mereka.

## 2.5 **Kontroller PD (Proportional–Derivative)**

Kontrol proporsional-derivatif merupakan kombinasi antara dua kontrol yaitu kontrol proporsional dan kontrol derivatif. Kontrol proporsional memiliki kelebihan dapat menstabilkan sistem yang tidak stabil dan mempercepat keadaan stabil, tetapi menyebabkan adanya lonjakan (overshoot). Sedangkan kontrol derivatif memiliki sifat meredam lonjakan (overshoot). Oleh karena itu, kontrol proporsional dan kontrol derivatif sangat baik dikombinasikan karena dapat mempercepat keadaan stabil dengan lonjakan (overshoot) yang kecil. Kontrol proporsional-derivatif digambarkan pada diagram berikut :



**Gambar 2.6 Diagram Blok Pengendali PD**

Sumber:

<http://anaksistemkontrol.blogspot.com/2016/04/implementasi-sistem-kendali-pd.html>

Bentuk umum dari kontrol proporsional-derivatif sebagai berikut :

$$y(t) = K_p e(t) + K_D \frac{de(t)}{dt}$$

Dalam transformasi Laplace dapat dinyatakan sebagai berikut :

$$\frac{Y(s)}{E(s)} = K_p + K_D s$$

dengan  $Y(s)$  dan  $E(s)$  adalah transformasi Laplace dari output  $y(t)$  dan  $e(t)$  error yaitu selisih output referensi dan output aktual, serta  $K_p$  dan  $K_d$  adalah gain kontrol proporsional dan gain kontrol derivatif.

## 2.6 Metode A-Star

Algoritma A\* (Astar) merupakan salah satu algoritma yang termasuk dalam kategori metode pencarian yang memiliki informasi (informed search method). Algoritma ini sangat baik sebagai solusi proses path finding (pencari jalan). Algoritma ini mencari jarak rute terpendek yang akan ditempuh suatu point awal (starting point) sampai ke objek tujuan. Teknik pencarian yang digunakan dalam simulasi ini adalah menggunakan Algoritma A\* dengan fungsi heuristic. Tujuan utama penelitian ini mempelajari cara kerja algoritma A\* dalam mencari jarak tercepat, yang disimulasikan seperti kondisi ketika seorang mencari rute dalam keadaan jalanan macet. Simulasi ini memberikan gambaran yang lebih realistis terhadap perilaku algoritma A\* dalam pencarian jarak rute terpendek. Notasi dari algorithm a-star adalah

$$f(n) = g(n) + h(n)$$

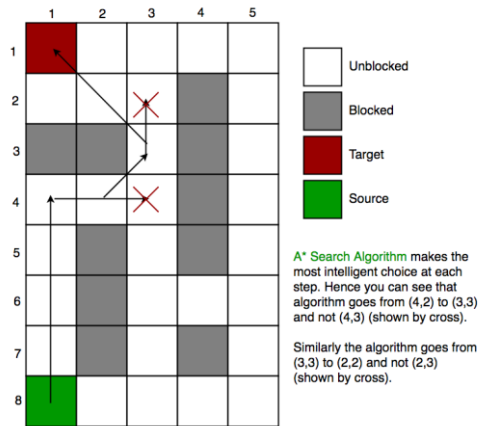
**f(n)** adalah jumlah dari g(n) dan h(n). ini adalah perkiraan jalur terpendek sementara. maka f(n) adalah jalur terpendek yang sebenarnya yang tidak ditelusuri sampai Algoritme A-Star (A\*) diselesaikan.

**g(n)/Geographical Cost** adalah total jarak yang didapat dari verteks awal ke verteks sekarang (halangan).

**h(n)/Heuristic Cost** adalah perkiran jarak dari vertek sekarang (yang sedang dikunjungi) ke vertek tujuan. sebuah fungsi heuristic digunakan

untuk membuat perkiraan seberapa jauh lintasan yang akan diambil ke vertek tujuan.

Jadi misalkan seperti pada gambar di bawah ini jika kita ingin mencapai sel target dari sel sumber, lalu A\* Search algoritma akan mengikuti jalan seperti yang ditunjukkan di bawah ini. Perhatikan bahwa gambar di bawah ini dibuat dengan mempertimbangkan Jarak *Euclidean* sebagai heuristik :



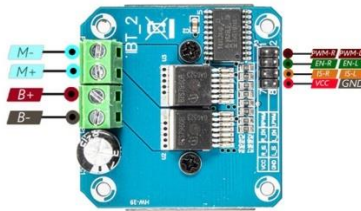
**Gambar 2.7 Algoritma A-Star**

Sumber: <https://www.geeksforgeeks.org/a-search-algorithm/>

## 2.7 Driver motor BTS7960

Driver motor merupakan suatu sistem yang mengontrol tegangan yang akan diteruskan ke motor dan juga dapat merubah arah putaran dari motor. Misalkan suplay motor 12V maka kita dapat mengatur tegangan dari suplay untuk masuk ke motor dengan driver motor, dengan driver motor kita dapat mengontrol hanya dengan tegangan 0-5V. Driver motor BTS7960 adalah rangkaian H bridge arus tinggi yang dapat mengontrol tegangan yang akan diteruskan ke motor dan mengarah arah putaran motor. Driver moor ini dilengkapi dengan fitur diagnosis arus , dan perlindungan terhadap temperature tinggi, tegangan berlebih, arus berlebih, dan *short circuit*. Jadi motor ini dapat mengontrol suatu motor dengan arus tinggi dengan aman.

POWER  
GND  
MOTOR PINS  
MOTOR VOLTAGE  
ENABLE PINS  
CURRENT ALARM



**Gambar 2.8 Driver motor BTS7960**

Sumber: <https://electropeak.com/learn/interfacing-bts7960-43a-high-power-motor-driver-module-with-arduino/>

## BTS7960 43A Motor Driver Features

The BTS7960 is a high-current full-bridge motor driver module. The Key features are:

- Input voltage: 6V to 27V
- Maximum allowable current: 43 A
- PWM capability: up to 25 kHz
- Two PWM output pins for speed control in direct and reverse directions
- Two EN output pins to control motors
- Two IS input pins to protect against high current and heat

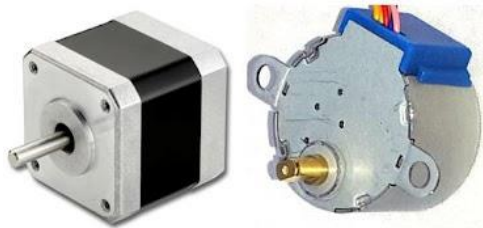
**Gambar 2.9 Spesifikasi Driver motor BTS7960**

Sumber: <https://electropeak.com/learn/interfacing-bts7960-43a-high-power-motor-driver-module-with-arduino/>

## 2.8 Motor Stepper

motor Stepper adalah jenis motor yang putarannya berdasarkan langkah (step) diskrit. Input pada motor stepper berasal dari pulsa-pulsa digital, berbeda dengan motor DC konvensional yang bekerja berdasarkan komutasi pada komponen brush (sikat) nya. Step yang mengendalikan motor berasal dari konstruksi kumparan yang disusun menjadi beberapa kelompok yang disebut fase. Motor dapat berputar apabila diberikan

energi pada fase secara berurutan. Motor Stepper mengubah sinyal-sinyal listrik menjadi gerakan mekanis diskrit. Motor Stepper bergerak dalam langkah (step) secara teratur. Anda dapat mengendalikan langkah pada motor menggunakan mikrokontroler maupun rangkaian digital. Torsi dari motor Stepper tidak sebesar motor DC. Namun, motor jenis ini memiliki tingkat presisi yang tinggi dalam putarannya. Kecepatan gerak pada stepper dinyatakan dalam step per second atau jumlah step per detik.

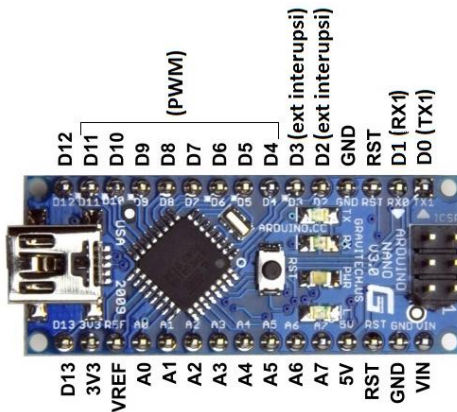


**Gambar 2.10 Motor Stepper**

Sumber: <https://www.andalanelektro.id/2021/01/mengenal-motor-stepper.html>

## **2.9 Arduino Nano**

Arduino Nano adalah salah satu varian dari produk board mikrokontroler keluaran Arduino. Arduino Nano adalah board Arduino terkecil, menggunakan mikrokontroler Atmega 328 untuk Arduino Nano 3.x dan Atmega168 untuk Arduino Nano 2.x. Varian ini mempunyai rangkaian yang sama dengan jenis Arduino Duemilanove, tetapi dengan ukuran dan desain PCB yang berbeda. Arduino Nano tidak dilengkapi dengan soket catudaya, tetapi terdapat pin untuk catu daya luar atau dapat menggunakan catu daya dari mini USB port. Arduino Nano didesain dan diproduksi oleh Gravitech.



**Gambar 2.11 Arduino Nano**

Sumber :

<https://djukarna4arduino.wordpress.com/2015/01/19/arduino-nano/>

Dan Arduino nando memiliki spesifikasi seperti berikut :

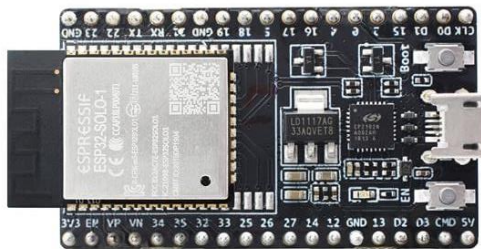
Mikrokontroller	: Atmel ATmega328
Tegangan kerja	: 5 Volt
Tegangan input	: Optimal : 7 – 12 Volt
Minimum	: 6 Volt
Maksimum	: 20 Volt
Digital pin I/O	: 14 pin yaitu pin D0 sampai pin D13
Analog pin	: 8 pin yaitu pin A0 sampai pin A7
Arus listrik maksimum	: 40 mA
Flash memori	: 32 Mbyte
SRAM	: 2 kbyte (ATmega328)
EEPROM	: 1 kbyte (Atmega328)
Kecepatan clock	: 16 MHz
Ukuran board	: 4,5 mm x 18 mm
Berat	: 5 gram



## 2.9 ESP32 Microcontroller

ESP32 adalah salah satu keluarga mikrokontroler yang dikenalkan dan dikembangkan oleh Espressif System. ESP32 ini merupakan penerus dari mikrokontroler ESP8266. Mikrokontroler satu ini compatible dengan Arduino IDE. Pada mikrokontroler ini sudah tersedia modul WiFi dan ditambah dengan BLE (Bluetooth Low Energy) dalam chip sehingga sangat mendukung dan dapat menjadi pilihan bagus untuk membuat sistem aplikasi Internet of Things. Spesifikasinya adalah sebagai berikut :

- Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.
- Memori: 520 KB SRAM.
- Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).



**Gambar 2.12 : ESP32 Wifi Module**

Sumber : [https://www.mouser.co.id/ProductDetail/Espressif-Systems/ESP32-DevKitC-](https://www.mouser.co.id/ProductDetail/Espressif-Systems/ESP32-DevKitC-S1?q=s%252BEw9%252B0nqrCx7FWvXX3A%3D%3D)

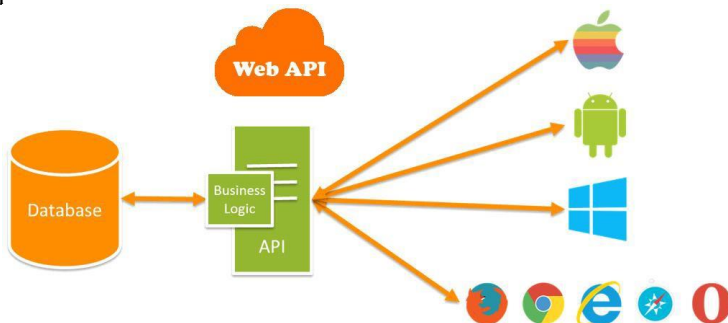
[S1?q=s%252BEw9%252B0nqrCx7FWvXX3A%3D%3D](https://www.mouser.co.id/ProductDetail/Espressif-Systems/ESP32-DevKitC-S1?q=s%252BEw9%252B0nqrCx7FWvXX3A%3D%3D)

## 2.10 REST API

API adalah singkatan dari Application Programming Interface yaitu sebuah software yang memungkinkan para developer untuk mengintegrasikan dan mengizinkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain..

Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah function yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa.

Istilah “API” sebetulnya tidak ada hubungannya dengan hal-hal yang berkaitan dengan web, karena istilah tersebut sudah ada sebelum web. Hal Ini semacam dikooptasi yang berarti “pemanggilan web service”. Tapi secara tradisional, Pengertian API bukan seperti itu. Tapi lebih berkaitan dengan fungsi-fungsi yang disediakan oleh Sistem Operasi.



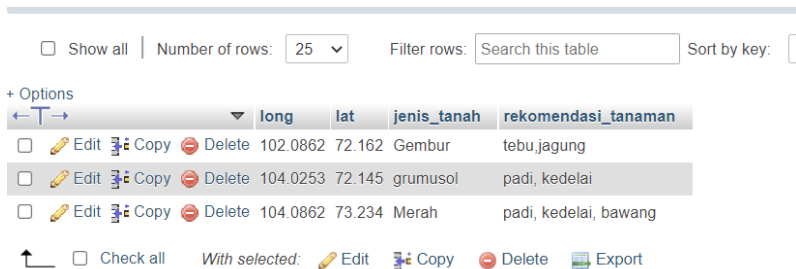
**Gambar 2.13 : REST API**

Sumber : <https://medium.com/jagoanhosting/perbedaan-antara-api-rest-api-dan-restful-api-6a66d655a6c2>

### 2.11 *My SQL Database*

MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (Relational Database Management System). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel.



	long	lat	jenis_tanah	rekomendasi_tanaman
<input type="checkbox"/>	102.0862	72.162	Gembur	tebu, jagung
<input type="checkbox"/>	104.0253	72.145	grumusol	padi, kedelai
<input type="checkbox"/>	104.0862	73.234	Merah	padi, kedelai, bawang

**Gambar 2.14 : Tabel *MySql Database***

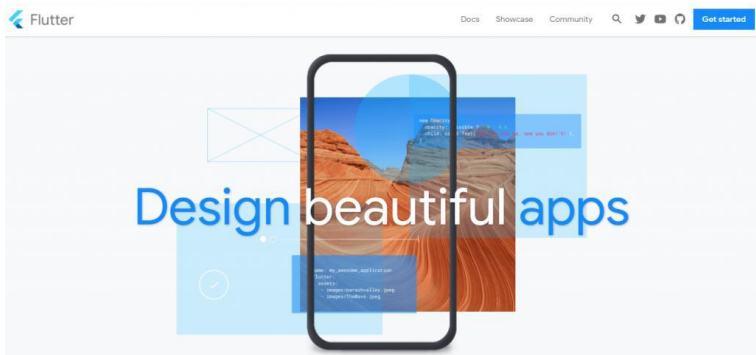
Sumber : Penulis

## 2.12 **Flutter Framework**

Flutter adalah platform yang digunakan para developer untuk membuat aplikasi multiplatform hanya dengan satu basis coding (codebase). Artinya, aplikasi yang dihasilkan dapat dipakai di berbagai platform, baik mobile Android, iOS, web, maupun desktop.

Flutter menyediakan user interface widgets yang dikembangkan oleh Google dengan kualitas coding yang tinggi. Widgets ini bisa Anda kustomisasi dengan mudah, tanpa perlu menulis coding dari nol.

Tak hanya itu, Flutter juga memiliki fitur Hot Reload yang akan menampilkan hasil coding dengan kilat. Jadi, Anda bisa bereksperimen menambahkan fitur dan memperbaiki bug lebih cepat.



**Gambar 2.15 : Flutter Framework**

Sumber : <https://flutter.dev>



# **BAB III**

## **PERANCANGAN DAN**

## **PEMBUATAN SISTEM**

## BAB III

### PERANCANGAN DAN PEMBUATAN SISTEM

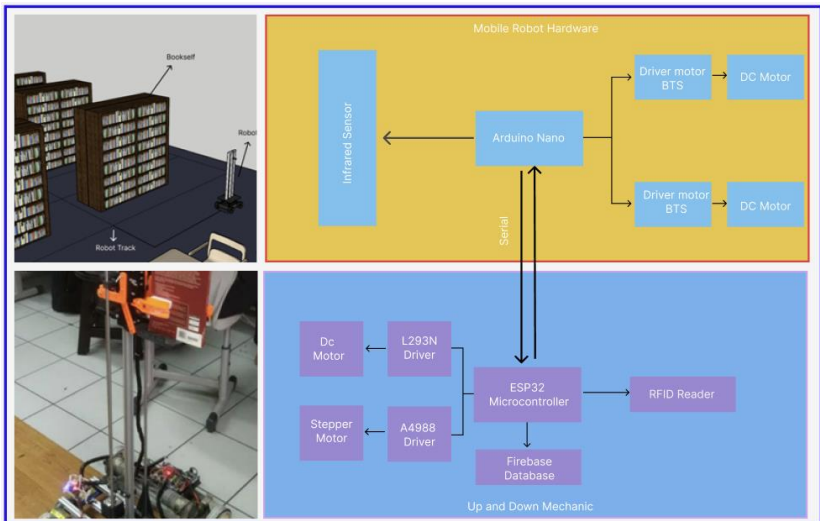
Adapun tahap pelaksanaan yang digunakan untuk mendukung pelaksanaan penelitian ini adalah sebagai berikut :

#### 3.1 Studi Literatur

Pada tahap ini akan dilakukan pencarian materi yang berkaitan dengan penelitian, mulai dari konsep dasar, metode, sistem sampai ke peralatan dan komponen yang akan digunakan dalam mendukung penelitian.

#### 3.2 Perancangan Sistem

Perancangan sistem yang dilakukan meliputi pembuatan blok diagram sistem. Blok diagram ditunjukkan pada Gambar 3.2.



**Gambar 3. 2** Blok Diagram Sistem  
Sumber : Penulis

Sesuai dengan Gambar 3.1 perancangan sistem ini secara keseluruhan terdiri dari dua bagian dasar, yaitu bagian perangkat keras (hardware) dan bagian perangkat lunak (software). Pembuatan hardware disini terdiri dari perancangan mikrokontroler dan modul-modul komponen untuk *mobile robot* dan mekanik atas. Untuk pembuatan software yang dilakukan adalah membuat program menggunakan bahasa pemrograman C++ dengan menggunakan software Arduino IDE. Pembuatan program dibagi menjadi dua yaitu untuk *mobile robot* yang dimana program akan menjalankan sistem pergerakan mobile robot dengan metode *line follower* yang akan mengikuti garis dari titik start menuju titik finish atau titik rak buku tujuan. Dan untuk program kedua adalah untuk pembuatan program pada mekanik atas, pembuatan program disini adalah untuk pergerakan mekanik naik turun, pergerakan pendorong buku, pembacaan sensor RFID *reader*, request data buku pada server dan juga update data monitoring pada database.

Dalam proyek akhir ini sistem keseluruhan yang dikerjakan adalah merancang sistem otomatis mobile robot yang dapat melakukan sistem pengembalian buku secara otomatis pada rak buku, disini sistem pergerakan mobile robot berjalan dengan metode mengikuti garis atau *line follower* dengan bantuan PID untuk respon dan pergerakan robot yang baik saat mengikuti jalur garis mulai dari *start* sampai *finish*. Untuk dalam sistem mekanik atas adalah sistem pembacaan rfid dari buku, naik turun buku, dan juga mendorong buku pada rak buku.

Berikut merupakan uraian singkat mengenai kinerja setiap komponen yang akan digunakan pada proyek akhir ini:

#### **Mobile Robot :**

1. Sensor infrared sebagai sensor utama untuk mengikuti garis dan mengembalikan nilai *error* untuk diolah oleh *mikrokontroller* sebagai acuan pergerakan dc motor pada mobile robot.
2. Arduino Nano *Mikrokontroller* sebagai kontroler utama untuk mengontrol pergerakan mobile robot. Pada arduino nano disini logika PID dijalankan untuk pergerakan mobile robot.
3. DC Motor yang merupakan akuator utama untuk pergerakan mobile robot, disini digunakan DC Motor Mitsubishi dengan spesifikasi 3000 RPM

4. Driver motor tipe BTS sebagai driver utama untuk pergerakan kedua motor sebagai akuator utama. Driver motor BTS memiliki batas arus yang tinggi hingga 60 A.
5. Regulator 5v sebagai pembatas tegangan dari baterai yang digunakan untuk di turunkan ke 5v supaya bisa untuk memberi power pada arduino dan komponen lainnya.

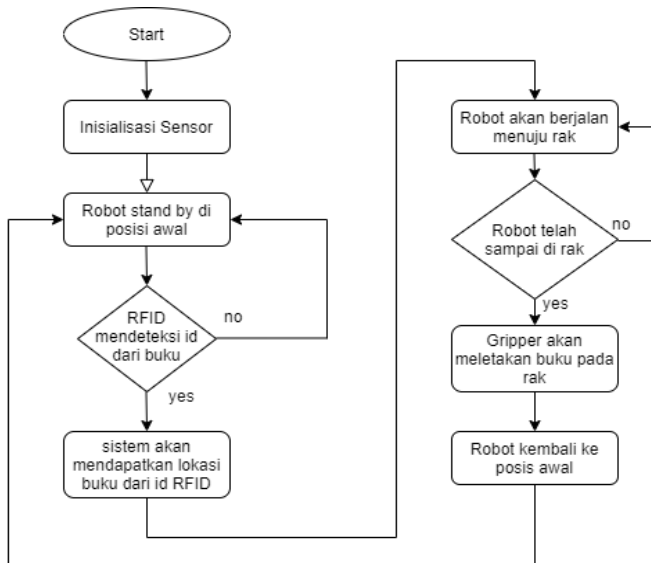
### **Mekanik Atas**

1. ESP32 yang merupakan mikrokontroller utama yang akan membaca data dari rfid, lalu menggerakkan stepper motor, dc motor pendorong, servo, dan juga melakukan request data pada server untuk mendapatkan lokasi rak buku sesuai id pada buku.
2. Stepper Motor sebagai akuator yang akan menggerakkan mekanik pembawa buku untuk bisa bergerak naik dan turun.
3. A4988 driver stepper moto sebagai driver penggerak stepper motor yang dapat menggerakkan stepper motor dengan arus maksimal 2 A.
4. DC Motor yang digunakan sebagai akuator pendorong buku yang dapat bergerak maju dan mundur untuk menggerakkan buku.
5. L298D driver dc motor yang menjadi driver untuk pergerakan mekanik pendorong yang dihubungkan dengan dc motor.
6. RFID reader sebagai sistem yang akan membaca id rfid dari buku yang di tempelkan pada robot. Yang nantinya dari data ini akan di dikirim ke server untuk didapatkan lokasi rak buku.
7. Limit switch sebagai pembatas pergerakan mekanik naik turun dan juga pembatas pergerakan mekanik pendorong.
8. Servo sebagai gripper atau pencepit pada mekanik pembawa buku supaya buku tetap pada posisi tegak pada saat robot berjalan.
6. Regulator 5v sebagai pembatas tegangan dari baterai yang digunakan untuk di turunkan ke 5v supaya bisa untuk memberi power pada arduino dan komponen lainnya.

### **3.2.1 Flowchart Keseluruhan Sistem**

Pada Gambar 3.2 merupakan flowchart sistem sebagai gambaran dari tahapan kerja sistem yang dirancang. Flowchart disini merupakan urutan sistem berjalan mulai dari awal atau saat mobile robot mendeteksi buku dan sampai dengan robot mengantarkan buku sampai ke depan rak buku.

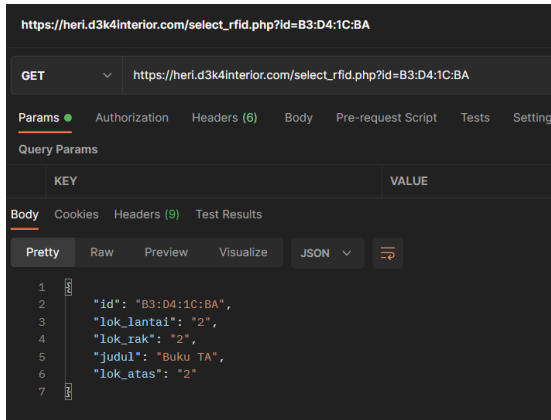




**Gambar 3. 3** Flowchart Sistem

Sumber : Penulis

Sebagaimana yang telah digambarkan pada gambar 3.2, sistem ini memiliki cara kerja yaitu robot akan *stand by* di tempat terdekat dengan petugas perpustakaan atau tempat lainya yang di tentukan, sensor rfid akan *stand by* menunggu data yang masuk. Lalu apabila ada seseorang yang meletakkan buku pada robot maka rfid akan mendeteksi dan mendapatkan id dari buku tersebut, setelah itu ESP32 akan melakukan *request* ke *server* untuk mendapatkan lokasi rak buku dari data – data lokasi rak buku yang telah tersimpan pada database. Berikut adalah gambaran saat melakukan *request* ke *server* untuk mendapatkan lokasi rak buku.



**Gambar 3. 4** *Request id ke server*  
Sumber : Penulis

Setelah lokasi rak buku didapatkan maka robot akan mulai bergerak menuju lokasi rak buku dengan metode mengikuti garis, robot akan mengikuti garis sesuai dengan lokasi rak buku yang telah ditentukan dan akan berjalan sesuai dengan program yang ada. Gambaran saat robot berjalan mengikuti garis adalah sebagai berikut :



**Gambar 3. 5** Robot berjalan mengikuti garis  
Sumber : Penulis

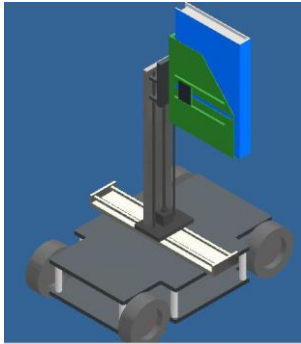
Dan apabila robot telah sampai pada rak buku yang telah ditentukan maka mekanik akan naik atau turun sesuai dengan lokasi tinggi rak yang telah didapatkan, lalu selanjutnya mekanik akan mendorong buku pada rak yang telah ditentukan. Dan mekanisme pengembalian buku telah selesai lalu robot akan kembali ke posisi awal untuk stand by di posisi awal. Gambaran saat robot mendorong buku pada rak adalah seperti berikut :



**Gambar 3. 6** Robot mendorong buku ke dalam rak  
Sumber : Penulis

### **3.3 Perancangan dan pembuatan mekanik**

Pada tahapan ini adalah perancangan mekanik sesuai dengan desain yang telah dibuat. Pada tahapan ini adalah pembuatan mekanik dari mobile robot, yang dibuat dari beberapa bahan utama yaitu akrilik sebagai kerangka utama robot, part hasil 3d print sebagai part melengkapi part utama, motor dan roda sebagai penggerak robot, dan juga mekanik untuk mendorong buku pada rak. Tingkat keberhasilan dari tahapan ini adalah mekanik dapat berjalan dan memiliki mekanik yang kuat.



**Gambar 3.7 Desain mekanik robot**

Sumber: penulis

### **3.3.1 Perancangan dan pembuatan mobile robot**

Pada tahapan ini adalah perancangan dan pembuatan mekanik untuk mobile robot, yang nantinya mobile robot ini akan bergerak menuju rak buku yang akan dituju. Disini digunakan alumunium holo sebagai body utama mobile robot dan digunakan paku rivet sebagai penyambung antara alumunium holo. Pada pembuatan mobile robot ini digunakan roda yang terbuat dari plastic dan busa. Pada mobile robot dipasang dua buah motor DC Mitsubishi OH1283 sebagai penggerak utama mobile robot dan satu buah free wheel untuk membantu manuver mobile robot.



**Gambar 3.8 Mobile robot**

Sumber: penulis

### 3.3.2 Perancangan dan pembuatan mekanik atas

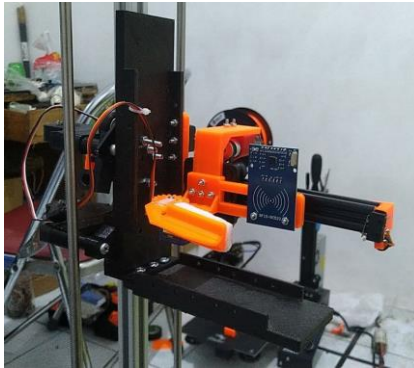
Pada bagian ini adalah pembuatan mekanik atas untuk sistem pengangkat dan pendorong buku, pada bagian ini digunakan satu buah alumunium profile dan dua buah alumunium tabung dengan panjang 1 meter sebagai penyangga utama mekanik atas, untuk sistem mekanik penggerak ke atas digunakan lead screw dengan panjang 1 meter. Untk penyambungan mekanik mekanik ini digunakan angle bracket dan part dari 3d print.



**Gambar 3.9 Mekanik atas**

Sumber: penulis

Pada sistem mekanik ini juga terdapat sistem pendorong yang dapat mendorong buku pada sesuai dengan rak nya. Pada mekanik ini terdiri dari alumunium profile, motor dc, dan limit switch. Sistem mekanik ini di control dengan mikrokontroller Arduino Nano.



**Gambar 3.10 Mekanik pendorong**

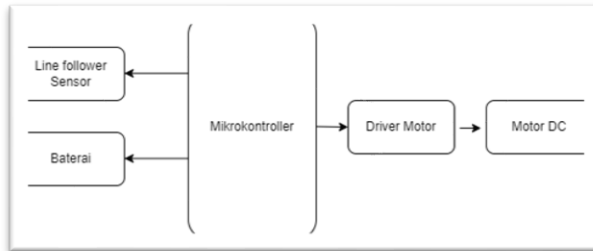
Sumber: penulis

### **3.4 Perancangan hardware**

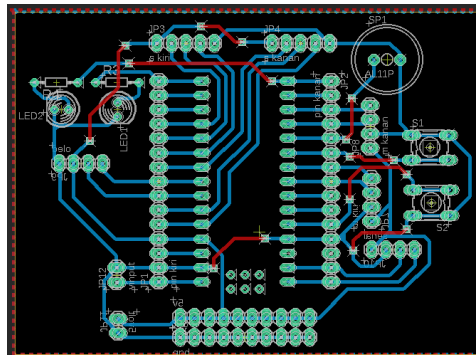
Pada tahapan ini adalah perancangan hardware untuk bisa menggerakkan mobile robot, komponen utama yang diperlukan adalah dc motor untuk berjalannya robot, infrared sensor sebagai input pergerakan robot untuk mengikuti garis, driver dan mikrokontroller untuk mengatur pergerakan robot, dan rfid sebagai pembaca id buku. Tingkat keberhasilan dari tahapan ini adalah mobile robot dapat bergerak otomatis menuju titik yang ditentukan dan menata buku pada rak.

#### **3.4.1 Perancangan hardware mobile robot**

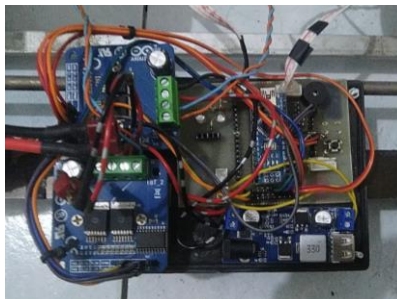
Pada hardware mobile robot digunakan metode line follower untuk pergerakan robot, pada mobile robot digunakan 6 buah sensor infrared sebagai input pembaca garis, lalu untuk control digunakan microcontroller Arduino nano, dan digunakan motor DC 20 volt serta driver motor BTS sebagai driver penggerak motor. Untuk diagram dari sistem ini adalah sebagai berikut :



**Gambar 3.11 Blok diagram hardware mobile robot**  
 Sumber: penulis



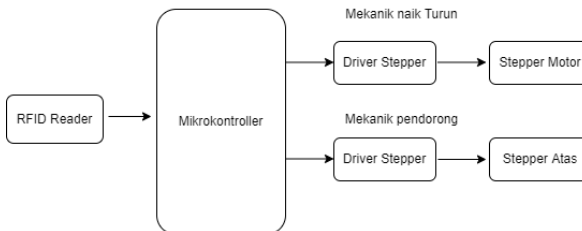
**Gambar 3.12 Desain PCB board**



**Gambar 3.13 Hasil akhir hardware**  
 Sumber: penulis

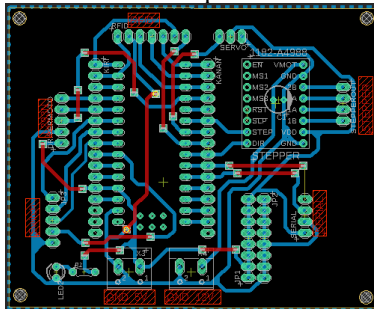
### 3.4.2 Perancangan hardware mekanik atas

Pada rangkaian ini terdapat beberapa hardware yang digunakan seperti stepper motor, driver stepper motor, RFID reader, dan Arduino nan sebagai controller utama. Rangkaian ini yang pertama berfungsi sebagai penggerak mekanik untuk naik dan turun dan juga untuk mekanik pendorong. Lalu yang kedua adalah rangkaian untuk membaca id buku melalui RFID.



**Gambar 3.14 Blok diagram hardware mekanik atas**

Sumber: penulis



**Gambar 3.15 Desain PCB mekanik atas**

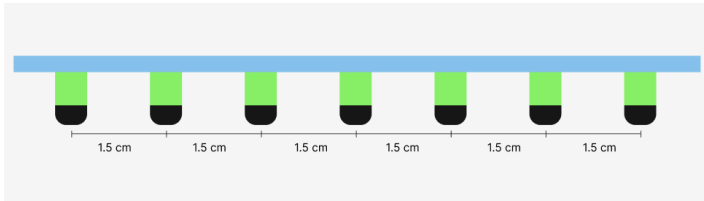
Sumber: penulis

### 3.4.3 Perancangan Sensor Garis

Pada perancangan sistem pembuatan desain sensor garis disini digunakan tujuh buah sensor infrared dengan tiga buah di kanan sebagai acuan error bernilai positif, tiga buah di kiri sebagai acuan error negatif, dan satu buah di tengah sebagai acuan nilai error nol. Ketujuh sensor yang disusun memiliki jarak antar sensor 1.5 cm. dengan acuan jarak seperti ini



maka dapat ditentukan untuk kenaikan nilai error adalah bernilai +10 atau -10. Untuk gambaran dari desain perancangan sensor garis adalah sebagai berikut :



**Gambar 3.16 Desain Sensor Garis**

Sumber: penulis

Dapat dilihat dari gambar diatas bahwa jarak antar sensor adalah linear yaitu berjarak 1.5 cm. dengan adanya penentuan jarak yang linear ini maka terdapat syarat maksiman dan minimal lebar garis yang dapat dilewati oleh desain sensor ini. Untuk rumus dari penentuan lebar garis yang dilewati oleh desain sensor ini adalah :

$$l > w > \frac{l}{2}$$

Yang dimana nilai  $w$  adalah lebar garis dan nilai  $l$  adalah jarak antar sensor, jadi sesuai rumus diatas maka minimal dari lebar garis adalah 0.75 cm dan maksimal lebar dari garis adalah 1.5 cm. untuk hasil desain dari sensor garis yang telah dibuat adalah sebagai berikut :



**Gambar 3.17 Hasil Perancangan Sensor**

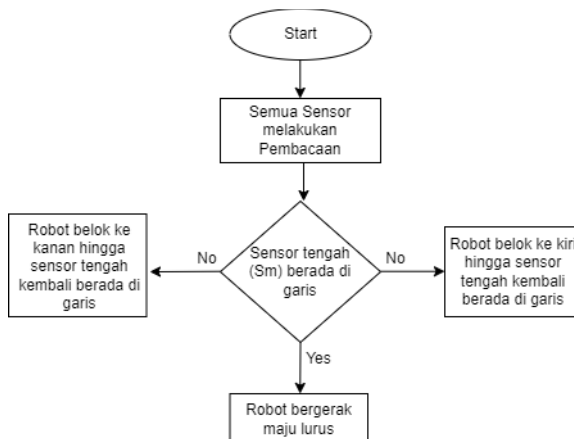
Sumber: penulis

### 3.5 Perancangan perangkat lunak

Perancangan perangkat lunak pada sistem ini berisi tentang algoritma pemrograman dari sistem otomatis yang dijalankan pada sistem ini. Pembuatan perangkat lunak atau program dalam mikrokontroler menggunakan Arduino IDE untuk mobile robot dan mekanik atas. Lalu juga digunakan backend yang dibuat dengan program php untuk membuat sebuah REST API yang nantinya akan bisa di ambil untuk mendapatkan lokasi rak buku. Untuk yang terakhir adalah perancangan aplikasi android yang berfungsi sebagai *interface* yang dapat digunakan dengan berbagai manfaat seperti pendaftaran id buku dan lokasi rak, list judul buku dan lokasi rak, dan juga untuk monitoring robot secara realtime.

#### 3.5.1 Perancangan perangkat lunak line follower

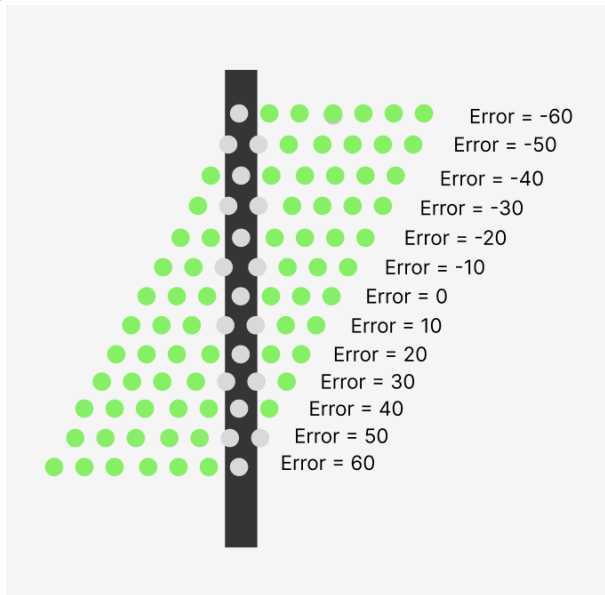
Pada sistem pergerakan robot dengan line follower adalah controller akan membaca terus menerus dari input nilai infrared, lalu dari nilai infrared tersebut akan didapatkan nilai error, dari nilai error tersebut akan diolah oleh sistem perhitungan PID, dalam sistem ini terdapat satu sensor di tengah (Sm), tiga sensor di kanan (Sl) dan tiga sensor di kiri (Sr). untuk flowchart dari sistem pembacaan sensor dan pengolahan pada PID adalah sebagai berikut :



**Gambar 3. 16:** Flowchart Line Follower dengan PID

Sumber : Penulis

Untuk system perancangan perangkat lunak pada line follower disini digunakan Bahasa C++ dan dengan platform pemrograman Visual Studio Code. Untuk Langkah pertama disini adalah program pembacaan nilai sensor untuk diambil nilai error, nilai error disini dihitung dari posisi tengah ke kanan untuk nilai positif dan tengah ke kiri untuk nilai negatif. Semua data dari sensor tersebut akan diubah menjadi nilai biner dan diambil nilai error, dan nantinya nilai error ini akan diolah dengan PID dan didapatkan hasil output motor yang stabil dan memiliki respon yang baik untuk hasil dari mendapatkan nilai error dari sensor adalah sebagai berikut:



**Gambar 3. 17:** Nilai error berdasarkan posisi sensor pada garis  
Sumber : Penulis

Untuk nilai error diatas dapat di definisikan dalam bentuk nilai biner untuk memudahkan dalam mendapatkan nilai error dan memasukan pada sistem PID. Untuk tabel dari nilai error diatas adalah :

Case Pembacaan Sensor	Nilai Error
0000001	-60
0000011	-50
0000010	-40
0000110	-30
0000100	-20
0001100	-10
0001000	0
0011000	10
0010000	20
0110000	30
0100000	40
1100000	50
1000000	60

**Tabel 3. 1:** Nilai error dari sensor  
Sumber : Penulis

Setelah didapatkan nilai error seperti diatas maka semua data tersebut akan diolah dengan sistem PID yang telah dibuat pada program, setelah di tentukan tuning nilai kp dan kd maka akan didapatkan hasil output pid yang akan masuk sebagai output pwm motor. Setelah didapatkan hasil akhir pid disini maka robot akan mengikuti garis dengan stabil dan respon yang baik. Secara garis besar untuk program pid tersebut adalah :

```

void linefollower(int Skiri, int Skanan)
{
    float errorB, error, lasterror = 0, sumerror = 0;
    float KP = 0.038, KI = 0.0, KD = 3.00;
    float kp = kp, ki = ki, kd = kd;
    float P, I, D, out;
    float speedKa, speedKi;

    while (true)
    {
        error = errorlineF();
        if (error != 0)
            sumerror += error;
        else
            sumerror = 0;
        P = error * KP;
        D = (error - lasterror) * KD;
        I = sumerror * KI;
        out = P + I + D;
        error = lasterror;
        speedKi = Skiri + out; //speed motor kiri
        speedKa = Skanan - out; //speed motor kanan
        if (speedKi >= 255) speedKi = 255;
        if (speedKa >= 255) speedKa = 255;
        if (speedKi <= -255) speedKi = -255;
        if (speedKa <= -255) speedKa = -255;
    }
}

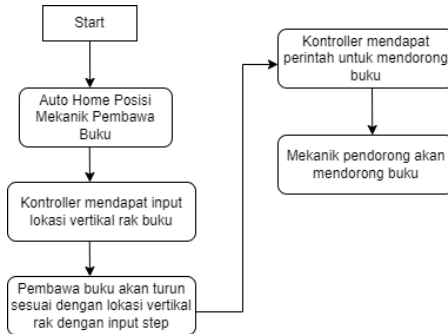
```

**Gambar 3. 18:** Program logika *line follower* dengan PID  
Sumber : Penulis

Dapat dilihat pada gambar 3.18 setelah nilai input data masuk pada program akan dicari nilai error dari input dibandingkan dengan setpoint yang diatur. Kemudian dari nilai error ini bisa didapatkan nilai jumlah error (sum error) dan perbedaan error dengan sebelumnya (delta error). Semua nilai error yang dihitung ini digunakan untuk menghitung nilai parameter P, I, dan D. Hasil output PID kemudian akan digunakan sebagai acuan input PWM pengatur kecepatan motor. Proses ini dilakukan terus menerus dengan nilai input yang baru dan menghitung nilai error yang baru.

### 3.5.2 Perancangan perangkat lunak mekanik atas

Gambar di bawah ini menunjukkan flowchart dari sistem pergerakan mekanik atas untuk bisa naik turun dan mendorong buku.



**Gambar 3. 19:** Flowchart Mekanik Atas

Sumber : Penulis

Pada system perancangan perangkat lunak disini adalah sistem untuk menggerakkan mekanik naik turun dan mekanik pendorong. Untuk yang pertama adalah program untuk sistem mekanik naik turun dan mekanik pendorong yang digunakan program untuk menggerakkan stepper motor dengan diberi input pulse step yang di inginkan untuk menggerakkan stepper motor, dan juga terdapat sebuah program limit switch untuk membatasi pergerakan motor stepper secara garis besar dari program untuk pergerakan motor stepper adalah sebagai berikut :

```
digitalWrite(dirPin, LOW);  
for (int x = 0; x < stepsPerRevolution1; x++)  
{  
    digitalWrite(stepPin, HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(stepPin, LOW);  
    delayMicroseconds(1000);  
}
```

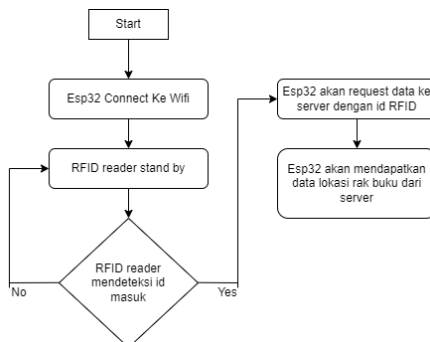
Selain program untuk pergerakan stepper motor naik dan turun juga terdapat program untuk pergerakan dc motor untuk mendorong buku

ke dalam rak, pada program ini diberikan dua buah limit switch sebagai pembatas untuk pergerakan dc motor untuk maju dan mundur. Secara garis besar dari program ini adalah sebagai berikut :

```
bool sw1 = false; int baca = 0;
while (!sw1)
{
    baca = digitalRead(pin);
    analogWrite(pwm, 255);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    if (baca == 0)
    {
        sw1 = true;
    }
}
```

### 3.5.3 Perancangan perangkat lunak *request server*

Gambar di bawah ini menunjukkan flowchart dari sistem request server untuk mendapatkan lokasi rak buku dari server.



**Gambar 3. 20:** Flowchart Request Server  
Sumber : Penulis

Perangkat lunak program untuk sistem request server ini adalah saat rfid reader mendeteksi atau membaca id, maka sistem akan melakukan request data ke server untuk mendapatkan data lokasi rak buku

yang telah tersimpan pada database. Setelah data dari database ini diambil maka akan dikirimkan ke sistem robot untuk segera dilakukan pengembalian buku sesuai dengan lokasi rak. Pada sistem ini digunakan http get untuk mendapatkan data lokasi buku, untuk programnya adalah sebagai berikut :

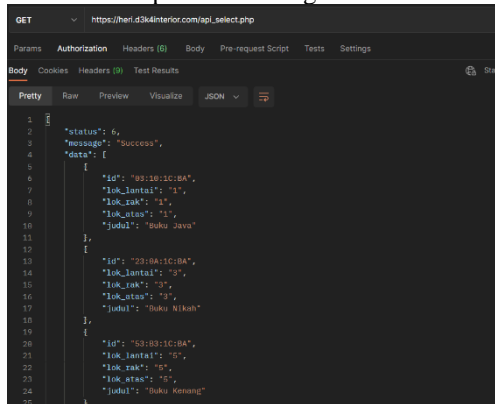
```

HTTPClient http;
String serverB = "http://heri.d3k4interior.com/select_rfid.php";
String serverPath = serverB + "?id=" + strID;
http.begin(serverPath.c_str());
int httpResponseCode = http.GET();
if (httpResponseCode > 0)
{
    payload = http.getString();
    Serial.println(payload);
}
http.end();

```

### 3.5.4 Perancangan *REST API*

Pada sistem ini adalah perancangan sebuah sistem rest api yang datanya diambil dari database yang telah dibuat sebelumnya. Semua data ini akan dibuat dalam bentuk rest api supaya dapat dengan mudah diambil oleh kontroller saat melakukan request data berdasarkan id rfid. Untuk hasil dari pembuatan rest api adalah sebagai berikut :



**Gambar 3. 21:** Hasil dari *rest api*

Sumber : Penulis

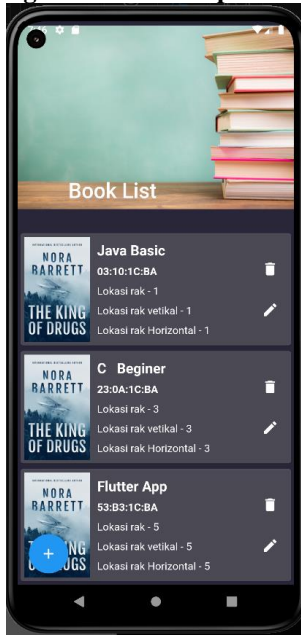


### **3.5.5 Perancangan pembuatan Aplikasi Android *book management***

Pada bagian ini adalah perancangan aplikasi android untuk sistem manajemen buku yang terdaftar pada *database*. Pada aplikasi ini dapat beberapa fitur yang dapat digunakan untuk mendukung sistem robot ini. Aplikasi ini dapat digunakan untuk melihat list buku, menambahkan list buku baru, mengedit lokasi ataupun judul buku dan lainnya. Jadi pada sistem ini sudah terintegrasi penuh dengan sistem *internet of things* jadi petugas perpustakaan nantinya dapat mengatur sistem lokasi dan judul buku secara otomatis melalui aplikasi Android. Pada perancangan aplikasi android ini digunakan *framework* flutter dengan Bahasa pemrograman dart.

Beberapa fitur dari aplikasi android ini adalah antara lain sebagai berikut :

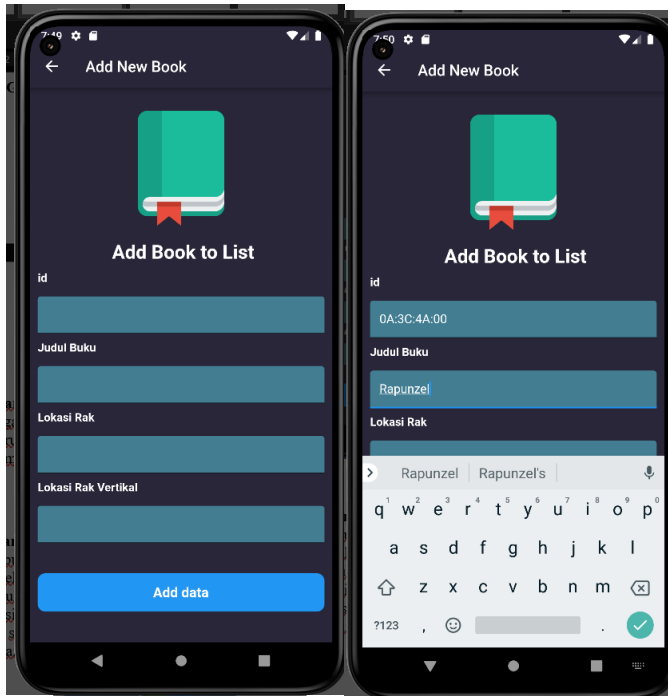
**Melihat List buku yang telah terdaftar pada *database***



**Gambar 3. 22:** List Buku di aplikasi android  
Sumber : Penulis

### **Menambahkan daftar buku baru dalam list**

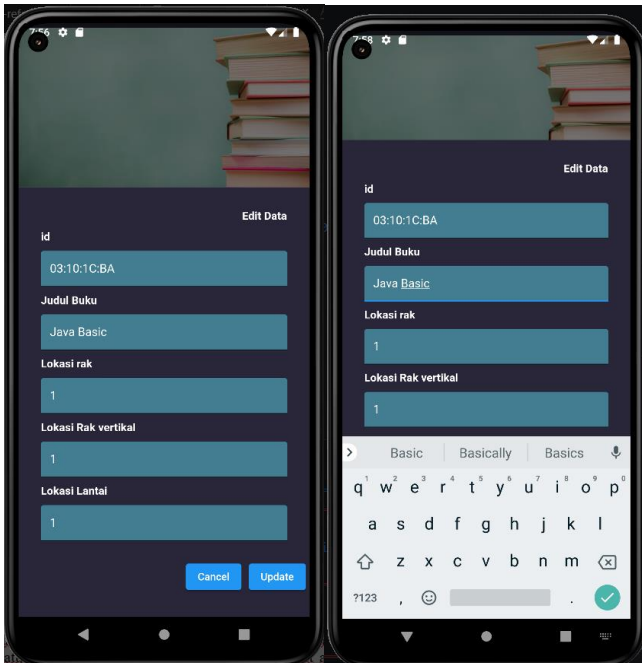
Petugas perpustakaan bisa langsung menambahkan list buku baru tanpa harus melakukan upload ulang program pada robot.pada fitur ini dapat digambarkan seperti berikut :



**Gambar 3. 23:** Menambahkan list buku baru pada *database*  
Sumber : Penulis

### **Update daftar buku pada list**

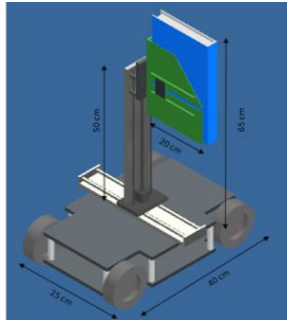
Pada sistem aplikasi android ini juga dapat melakukan update data yang telah ada. Jadi jika apabila terjadi perubahan lokasi buku maka petugas perpustakaan dapat melakukan perubahan lokasi buku melalui aplikasi android. Gambaran dari sistem update data ini adalah seperti berikut :



**Gambar 3. 24:** Melakukan update buku pada *database*  
Sumber : Penulis

### 3.5 Pembuatan Alat

Pembuatan alat merupakan proses perakitan meliputi perakitan komponen elektronika, pembuatan mekanik untuk mobile robot dan mekanik untuk naik turun, dan pembuatan program yang akan diimplementasikan pada alat. Kemudian ketiga hal tersebut disatukan agar menjadi satu alat yang utuh dan dapat dijalankan. Berikut adalah gambaran desain alat yang akan diterapkan.



**Gambar 3. 25** : Desain mekanik awal

Sumber : Penulis

### **3.6 Pengujian Alat**

Pengujian alat dilakukan untuk mengetahui error apa yang terjadi pada alat yang dibuat, dan kemudian akan dilakukan pembenahan pada blok sistem yang mengalami error. Agar alat yang dibuat dapat berjalan sesuai keinginan peneliti dan untuk mempermudah troubleshooting pada alat maka pengujian alat ini dibagi menjadi 2 tahap yaitu pengujian pada tiap blok system dan pengujian keseluruhan alat.

#### **3.6.1. Pengujian blok system**

Pengujian ini dilakukan pada setiap blok system mulai dari blok analog pembangkit gelombang sampai ke blok AC to DC, juga pengujian blok pengukur tinggi badan dan juga berat badan. Dan kemudian dari blok digital dilakukan pengujian dan pengecekan program yang ditanamkan pada mikrokontroler. Jika semua sudah sesuai dengan yang diharapkan maka tiap blok system tadi akan dirangkai menjadi satu alat yang utuh. Tujuan pengujian ini adalah meminimalisir kesalahan yang akan terjadi ketika di implementasikan ke manusia.

#### **3.6.2. Pengujian keseluruhan**

Pengujian ini dilakukan setelah keseluruhan alat sudah dirangkai menjadi satu alat yang utuh, kemudian mengambil sampel minimal 30 orang. Tujuan dari pengujian ini adalah untuk mengetahui hasil pengukuran dari alat ini.

### **3.7. Analisis dan Kesimpulan**

Setelah melakukan pengujian alat secara keseluruhan Kemudian ke 30 sampel tersebut melakukan tes lagi menggunakan alat pendeteksi dehidrasi berbasis warna urin. Dan hasilnya dilakukan perbandingan dan analisis data untuk mengetahui karakteristik dan tingkat keberhasilan dari system yang telah diterapkan dan kemudian hasil dari analisis data tersebut akan disimpulkan. Dan untuk alat ini dikatakan berhasil apabila 90% atau lebih pengukuran hasilnya sama dengan pengukuran alat laboratorium.

Penyusunan laporan dilakukan berdasarkan data yang telah diperoleh dan telah diproses menjadi sebuah informasi yang dapat dimanfaatkan untuk penelitian selanjutnya.



# **BAB IV**

## **PENGUJIAN DAN ANALISA**

## **BAB IV**

### **PENGUJIAN DAN ANALISA**

Pada bab ini akan dilakukan pengujian dan analisa terhadap perangkat keras dan program yang telah dibuat. Adapun beberapa pengujian yang telah dilakukan adalah sebagai berikut:

1. Pengujian nilai ADC sensor infrared
2. Pengujian motor penggerak
3. Pengujian motor stepper
4. Pengujian RFID
5. Pengujian keakuratan motor stepper
6. Pengujian motor stepper dengan beban
7. Pengujian tuning PID
8. Pengujian *line follower* mendeteksi simpangan.
9. Pengujian Pendorong Buku

#### **4.1 Pengujian nilai ADC sensor infrared**

##### **4.1.1 Tujuan**

Tujuan pengujian nilai adc dari sensor line infrared adalah untuk melihat pembacaan sensor infrared saat mendeteksi warna hitam dan bukan warna hitam, yang nantinya pembacaan nilai adc ini dapat menjadi patokan dari nilai kalibrasi sensor infrared. Jadi nilai ini akan digunakan sebagai kalibrasi nilai sensor secara otomatis.

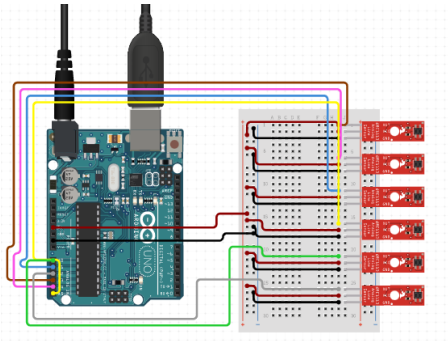
##### **4.1.2 Setting Pengujian**

###### **A. Peralatan Yang Digunakan**

1. Arduino Uno
2. 6 Sensor infrared

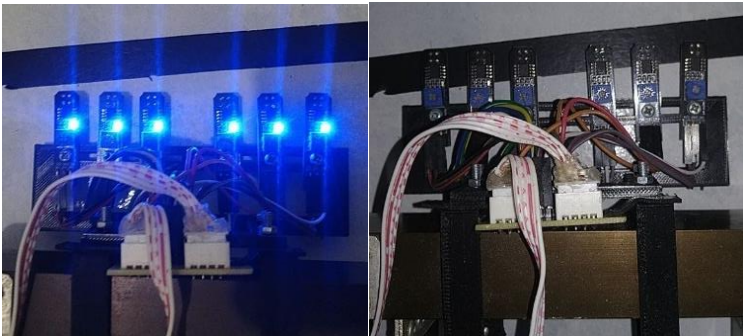


## B. Blok Diagram



**Gambar 4. 1 :** Rangkaian Sensor Infrared

## C. Dokumentasi Pengujian



**Gambar 4. 2 :** Pengujian pembacaan sensor infrared

### 4.1.3 Prosedur Pengujian

Untuk mendapatkan data adc dari sensor infrared maka dilakukan pengujian di tempat dengan tiga pencahayaan yang berbeda, nilai dari adc saat mendeteksi warna hitam dan bukan warna hitam tersebut selanjutnya akan disimpan dalam EEPROM sebagai nilai set untuk kalibrasi pembacaan sensor infrared.

#### 4.1.4 Hasil dan Analisa

**Tabel 4. 1** : Data hasil pembacaan sensor infrared

Keterangan	Hasil ADC (Sensor 1 – Sensor 6)
<b>Cahaya Terang</b>	
Warna hitam	[401,405,448,437,433,427]
Bukan warna hitam	[42,25,43,29,42,37]
<b>Cahaya Cukup Gelap</b>	
Warna hitam	[495, 490, 537, 537, 528, 519]
Bukan warna hitam	[44, 36, 38, 33, 37, 31]
<b>Cahaya Gelap</b>	
Warna hitam	[438,451,497,479,479,465]
Bukan warna hitam	[44, 27, 27, 28, 30, 39]

## 4.2 Pengujian Motor Penggerak

### 4.2.1 Tujuan

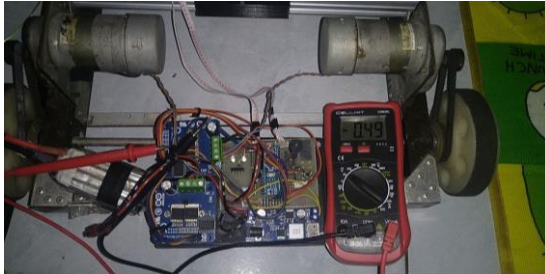
Tujuan pengujian ini adalah untuk mengetahui performa motor DC dan karakteristik tegangan motor terhadap RPM sebagai penggerak utama mobile robot, selain itu juga untuk mengetahui perkiraan jarak daya tahan sumber baterai 12 V yang digunakan.

### 4.2.2 Setting Pengujian

#### A. Peralatan Yang Digunakan

1. Motor DC
2. Multimeter
3. Tachometer
4. Driver motor BTS

## B. Dokumentasi Pengujian



**Gambar 4. 3** : Dokumentasi Pengujian motor

### 4.2.3 Prosedur Pengujian

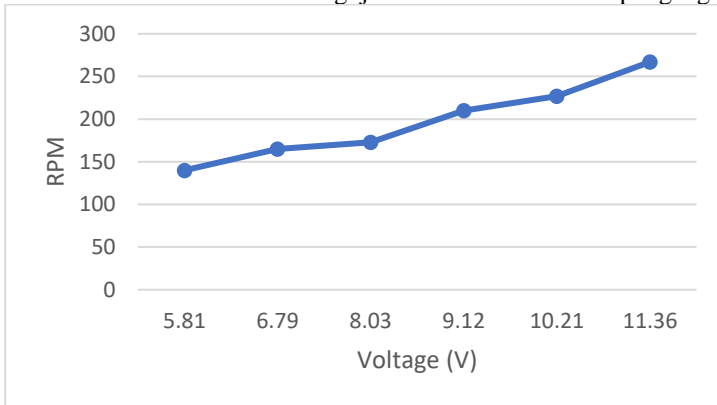
Pada pengujian ini dilakukan dengan mengukur daya motor saat berjalan dengan RPM tertentu, yang nantinya data ini dapat diolah dan menjadi prediksi atau perkiraan daya tahan baterai 12 V. diambil data berupa RPM motor, arus, dan tegangan pada motor. Pengujian ini dilakukan dengan memberi sinyal pwm pada motor dengan range maksimal 255.

### 4.2.4 Hasil dan Analisa

**Tabel 4. 2** : Data Pengujian Motor

RPM	Speed ( m/menit)	Arus motor (mA)	Tegangan motor (Volt)	Daya motor (mW)	Daya tahan baterai (jam)
267	68,3024	343	11.36	3896	1.25
227	56.45	311	10.21	3175	1.32
210	46.32	294	9.12	2681	1.51
173	42.11	273	8.03	2192	1.79
165	40.6	260	6.79	1749	2.12
140	36.51	245	5.81	1423	2.9

**Gambar 4. 4** : Grafik Pengujian RPM motor terhadap tegangan



Berdasarkan tabel 4.2 adalah hasil pengambilan data performa motor dengan sumber motor yaitu baterai 12 volt dan dijlankan tanpa beban. Dari hasil tabel dapat dilihat bahwa pada saat tegangan penuh motor mampu berputar pada kecepatan 267 RPM. Dan pada saat dibuat grafik hubunga performa dari RPM dan tegangan maka didapat hasil grafik yang linear seperti grafik 4.2 diatas.

### **4.3 Pengujian Motor Stepper**

#### **4.3.1 Tujuan**

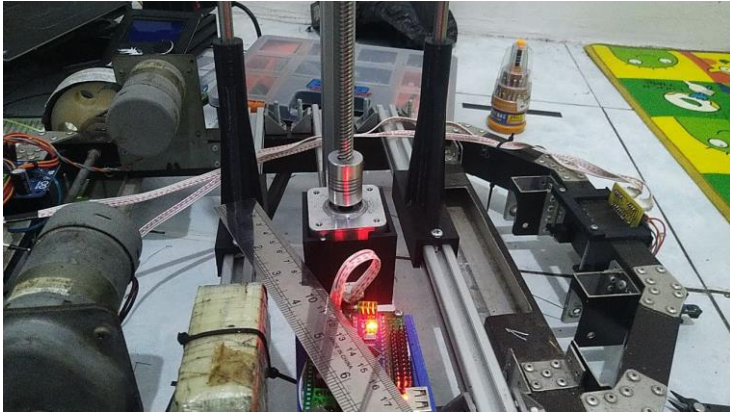
Tujuan pengujian ini adalah untuk melihat berapa besar sudut yang dihasilkan oleh motor stepper, pengujian ini dilakukan dengan memeberikan input berupa step pada motor stepper. Selain melihat sudut yang dihasilkan disini juga dapat dilihat berapa mm perpindahan yang dihasilkan oleh motor stepper.

#### **4.3.2 Setting Pengujian**

##### **A. Peralatan Yang Digunakan**

1. Motor Stepper
2. Driver stepper
3. Penggaris
4. Mekanik dengan lead screw
5. Busur derajat

## B. Dokumentasi Pengujian



**Gambar 4. 4** : Dokumentasi Pengujian motor stepper

### 4.2.3 Prosedur Pengujian

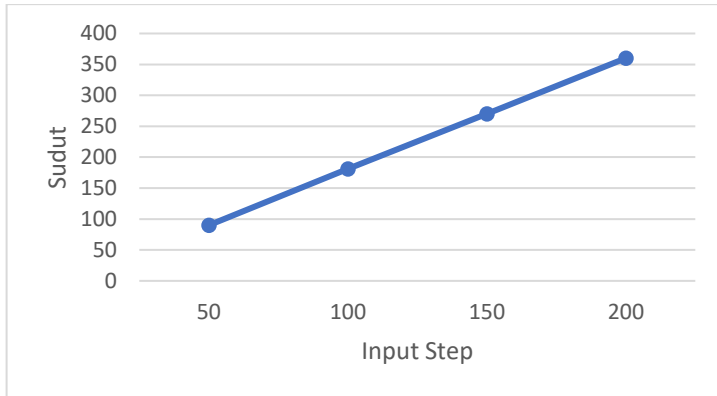
Pada pengujian ini dilakukan memberikan input step mulai dari 50 – 200 step, dari datasheet motor stepper Nema17 didapatkan bahwa motor stepper ini membutuhkan 200 step untuk mendapatkan satu putaran penuh dengan sudut 360 derajat. Pengujian ini dilakukan untuk mendapatkan nilai derajat dan juga perpindahan dengan satuan mm

### 4.3.4 Hasil dan Analisa

Tabel 4. 3 : Data Pengujian Motor Stepper

Input Step	Sudut
50	90
100	181
150	270
200	360

**Gambar 4. 6** : Grafik Perbandingan antara input step dan sudut yang dihasilkan



Berdasarkan tabel 4.3 diatas didapatkan bahwa output sudut derajat yang didapatkan dari motor stepper adalah 1.8 derajat per step. Output sudut ini sudah sesuai dengan datasheet dari motor stepper. Dan juga didapatkan perpindahan dengan satuan mm yaitu sebesar 1.7 mm per 50 step atau 0.0034 mm per 1 step. Data ini nantinya akan menjadi data patokan atau kalibrasi untuk pergerakan mekanik naik turun.

## **4.4 Pengujian RFID**

### **4.4.1 Tujuan**

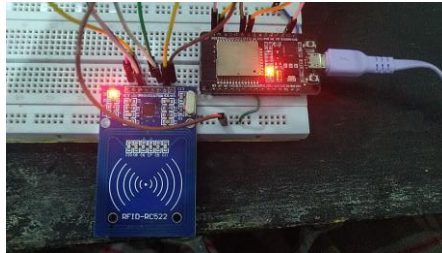
Tujuan pengujian ini adalah untuk mendapatkan id RFID yang nantinya akan menjadi input lokasi dari rak buku yang telah terdaftar yang nantinya akan mendapat feedback dari robot.

### **4. 4. 2 Setting Pengujian**

#### **A. Peralatan Yang Digunakan**

1. RFID Reader
2. RFID Card
3. Esp32

#### **B. Dokumentasi Pengujian**



**Gambar 4. 5** : Dokumentasi Pengujian RFID

#### 4.4.3 Prosedur Pengujian

Pada pengujian ini dilakukan menempelkan RFID Card atau RFID Sticket pada RFID reader. Setelah data bisa terbaca maka selanjutnya data tersebut di inputkan dalam tabel yang selanjutnya akan di convert dalam data JSON untuk bisa dikirim kepada robot

#### 4.4.4 Hasil dan Analisa

**Tabel 4. 4** : Data RFID dan detail buku

ID RFID	Judul Buku	Lokasi Rak	Nomor Rak Vertical	Nomor Rak Horizontal
C5:98:B0:1A	Elektronika	1	1	3
86:6E:02:1F	PID Basic	2	1	2
40:1C:5B:D3	Sejarah	5	2	3

Dari hasil data tabel diatas maka akan di convert dalam bentuk data JSON supaya lebih mudah unutm di parsing oleh robot. Contoh data JSON nya adalah sebagai berikut :

```
{
  "robot": "ready",
  "Jumlahbuku": "3",
  "details": [
    {
      "id": "C5:98:B0:1A",
      "judul": "Elektronika",
      "lokasirak": "1",
```

```
        "rakvertikal": "1",  
        "rakhorizontal": "3",  
    },  
    ]  
}
```

Dari data id yang telah didapatkan maka akan ditambahkan lokasi rak yang selanjutnya semua data tersebut akan masuk ke dalam tabel dan di convert menjadi tipe data JSON. Data JSON akan dikirim ke robot untuk menjadi data utama list buku dan lokasi.



## **4.5 Pengujian Keakuratan Pergerakan Motor Stepper**

### **4.5.1 Tujuan**

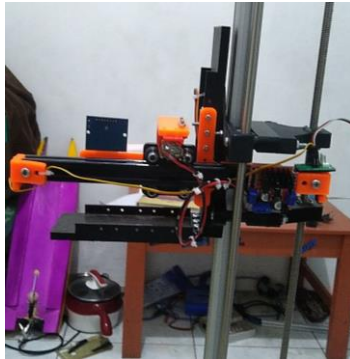
Tujuan pengujian ini adalah untuk melihat tingkat keakuratan pergerakan naik dan turun mekanik pembawa buku. Pada pengujian ini dilakukan dengan memberikan input step dengan output mm.

### **4.5.2 Setting Pengujian**

#### **A. Peralatan Yang Digunakan**

1. Motor Stepper
2. Lead Screw
3. Microcontroller

#### **B. Dokumentasi Pengujian**



**Gambar 4.6 :** Dokumentasi Pengujian Keakuratan Motor Stepper

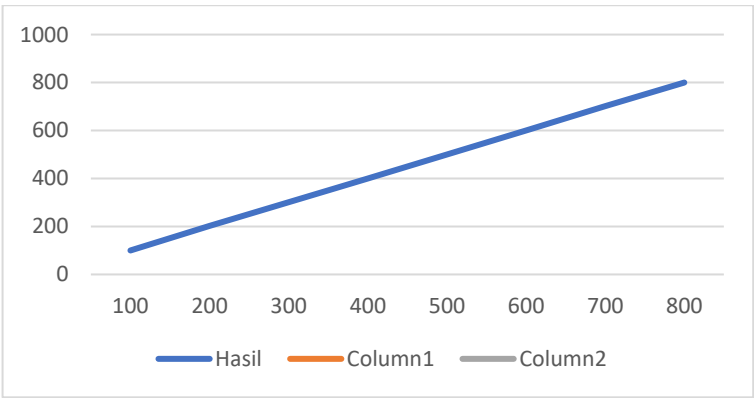
### **4.5.3 Prosedur Pengujian**

Pada pengujian ini dilakukan dengan memberikan input step yang akan mengeluarkan output mekanik yang naik per mm. Pada percobaan ini diberi target output per mm yang dimulai dari titik nol atau titik terendah dari posisi lead screw.

4.5.4 Hasil dan Analisa

No	Tinggi yang ditentukan (mm)	Hasil (mm)	Step
1	100	100	2500
2	200	200	5000
3	300	300	7500
4	400	400	10000
5	500	500	12500
6	600	600	15000
7	700	700	17500
8	800	800	20000

Tabel 4. 5 : Tabel hasil perbandingan antara tinggi yang ditentukan dan hasil



Grafik 4. 5.4 : Grafik hasil perbandingan antara tinggi yang ditentukan dan hasil

Pada pengujian keakuratan motor stepper ini menghasilkan data yang sangat bagus, dikarenakan spesifikasi motor stepper yang tepat dan driver stepper yang tepat, dan hampir memiliki error sebesar 0 %

#### **4.6 Pengujian Motor Stepper dengan beban**

##### **4.6.1 Tujuan**

Tujuan pengujian ini adalah untuk melihat tingkat keakuratan dan kekuatan maksimal pergerakan naik dan turun mekanik pembawa buku dengan tambahan beban bervariasi dengan maksimal beban 1 kg.

##### **4. 6. 2 Setting Pengujian**

###### **A. Peralatan Yang Digunakan**

1. Motor Stepper
2. Beban
3. Timbangan

###### **B. Dokumentasi Pengujian**



**Gambar 4. 7 : Dokumentasi Pengujian Motor Stepper dengan Beban**

#### 4.6.3 Prosedur Pengujian

Pada pengujian ini dilakukan dengan memberikan input step yang akan mengeluarkan output mekanik yang naik per mm. dengan diberi beban buku yang sesuai dengan spesifikasi beratnya.

#### 4.6.4 Hasil dan Analisa

No	Tinggi yang ditentukan (mm)	Beban (gram)	Hasil (mm)
1	500 mm	100	500
2	500 mm	500	500
3	500 mm	800	500
4	500 mm	1000	500

**Tabel 4. 6 :** Tabel hasil pengujian motor stepper dengan beban

Pada hasil pengujian ini didapatkan hasil yang bagus, yaitu motor stepper dapat mengangkat beban sampai dengan 1 kg. pada pengujian ini digunakan motor stepper Nema dengan beban maksimal pengangkatan hingga 2 kg.

### 4.7 Pengujian Tuning PID

#### 4.7.1 Tujuan

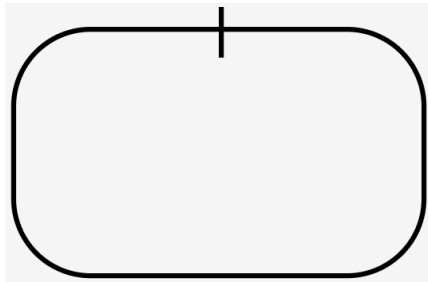
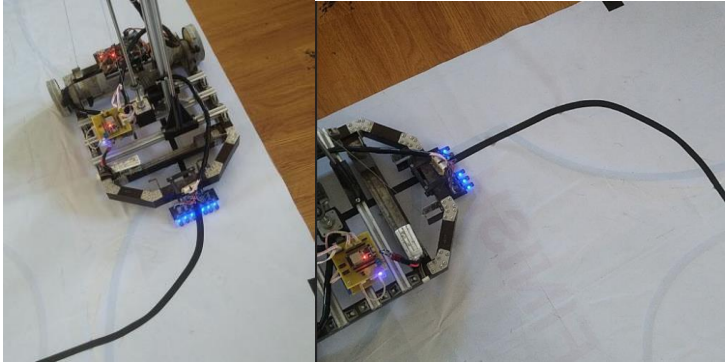
Tujuan pengujian ini adalah untuk melakukan tuning nilai pid yang tepat dengan kecepatan yang di tentukan. Dengan pengujian ini diharapkan robot dapat bergerak dengan baik saat mengikuti garis dengan nilai PID yang tepat.

#### 4.7.2 Setting Pengujian

##### A. Peralatan Yang Digunakan

1. Mobile Robot
2. Jalur *line follower*

## B. Dokumentasi Pengujian



**Gambar 4. 10** : Dokumentasi pengujian

### 4.7.3 Prosedur Pengujian

Sistem tuning PID yang dilakukan adalah trial and error untuk mendapatkan kondisi stabil robot sehingga dapat mencapai titik finish dengan cepat. Tuning PID menghasilkan waktu mencapai finish tercepat dengan pergerakan robot yang halus. Dalam pengujian juga dilakukan perubahan kecepatan maksimum putaran motor DC dengan variasi PWM, hal ini dilakukan untuk mengetahui apakah perubahan kecepatan mempengaruhi terhadap kestabilan manuver robot dan tuning PID yang telah dilakukan.

#### 4.7.4 Hasil dan Analisa

Pengujian hasil perbandingan tuning nilai PID dan maksimum kecepatan terhadap waktu tempuh robot line follower seperti yang ditunjukkan pada Tabel dibawah :

Kecepatan PWM 100			
KP	KI	KD	Hasil
0.2	0.01	0.1	5.0 s
0.18	0.01	0.8	4.5 s
0.16	0.02	0.5	4.2 s
0.15	0.00	0.3	3.9 s
0.15	0.00	0.1	3.5 s

**Tabel 4. 7** : Tabel hasil pengujian tuning PID

Dapat dilihat dari hasil table diatas dapat dilihat bahwa nilai tuning yang tepat adalah terdapat pada kolom terakhir yaitu dengan nilai  $KP = 0.15$ ,  $KI = 0.00$ , dan  $KD = 0.1$ . dan didapatkan waktu dari *start* sampai *finish* adalah 3.5 s. dan untuk line follower ini tidak menggunakan nilai KI karena respon yang terlalu cepat saat diberi nilai KI.

#### 4.8 Pengujian *Line Follower* mendeteksi simpangan.

##### 4.8.1 Tujuan

Tujuan pengujian ini adalah untuk melihat seberapa cepat dan akurat sensor dan sistem yang telah dibuat dalam mendeteksi sebuah simpangan dan melakukan aksi belok ataupun lanjut lurus.

##### 4.8.2 Setting Pengujian

###### A. Peralatan Yang Digunakan

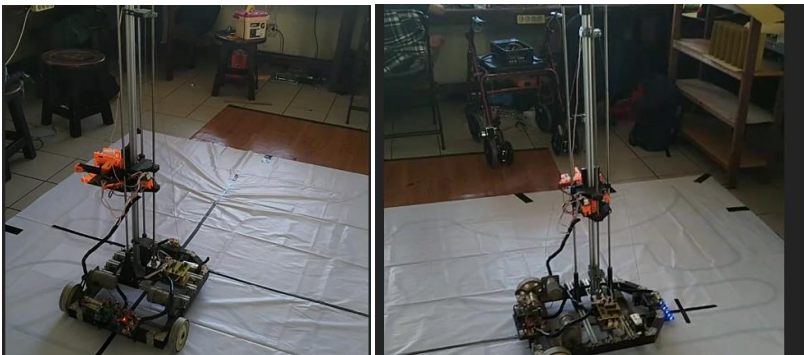
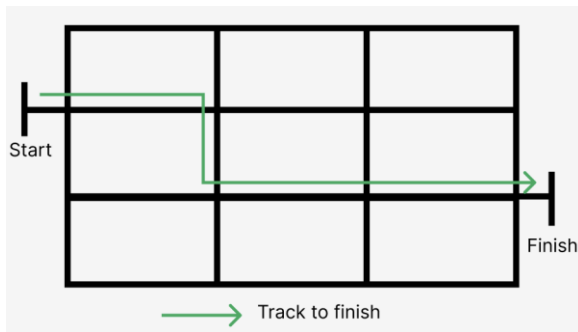
1. Mobile Robot
2. Jalur *line follower*

###### B. Blok Diagram

Blok diagram atau logika yang digunakan untuk mengikuti garis dan menemukan simpangan adalah seperti berikut :

```
bool isFindCross = false;  
while(!isFindCross)  
{  
    linefollow(speed, speed);  
    if (detectedCross == true)  
        isFindCross = true;  
}
```

## B. Dokumentasi Pengujian



**Gambar 4. 11 :** Dokumentasi Pengujian

#### 4.8.3 Prosedur Pengujian

Pada pengujian mendeteksi simpangan ini merupakan pengujian *mobile robot* ketika bergerak dengan logika line follower untuk mengikuti suatu garis dan menemukan persimpangan, tujuan dari pengujian ini adalah sebagai acuan bagi *mobile robot* untuk berbelok ke kiri, kanan, atau maju. Untuk prosedur pengujian ini *mobile robot* akan bergerak maju mengikuti garis sampai semua sensor *infrared* menemukan garis persimpangan (semua sensor membaca nilai *high*), sehingga terdeteksi persimpangan.

#### 4.8.4 Hasil dan Analisa

Line Follower Detect Cross Test		
Speed (pwm)	Time Finish	Result
50	20 s	Finish
70	17 s	Finish
90	15 s	Finish
100	13 s	Finish
120	Not Finish	Not Finish
150	Not Finish	Not Finish

**Tabel 4. 8** : Tabel hasil pengujian

Dalam pengujian ini *mobile robot* bergerak dengan kecepatan antara 50 - 150, dari hasil tabel di bawah ini kita ketahui robot dapat finish dengan baik pada kecepatan 50 - 100, untuk kecepatan lebih dari 100 robot gagal mencapai finish atau error untuk mendeteksi tanda silang .





# **BAB V**

## **PENUTUP**

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah melakukan tahap perancangan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan sebagai berikut :

1. Hasil pengujian daya motor pada RPM 267 memiliki daya tahan baterai sekitar 1.25 jam. Dan kecepatan motor rata rata yang digunakan ini adalah 140 RPM dengan rata rata daya tahan baterai adalah 2 jam
2. Hasil pengujian stepper untuk mendapatkan hasil derajat didapatkan hasil yang akurat sebesar 98%. Dengan diberi input step sesuai dengan spesifikasi motor stepper yaitu 50 step untuk 90 derajat
3. Hasil pengujian motor stepper untuk perpindahan naik dan turun sebesar mm didapatkan hasil akurat sebesar 97% dengan hasil 1.7 mm per 50 step atau 0.0034 mm per 1 step.
4. Hasil pengujian untuk tuning PID didapatkan hasil nilai  $K_P = 0.15$ ,  $K_I = 0.00$ , dan  $K_D = 0.1$ . dan didapatkan waktu dari *start* sampai *finish* adalah 3.5 s. dengan kecepatan PWM 100.
5. Dari hasil akhir keseluruhan sistem yang dijalankan didapatkan hasil bahwa semua sistem dapat berjalan mulai dari deteksi RFID dari id buku sampai dengan mobile robot mengantarkan buku sampai ke dalam rak buku.

#### **5.2 Saran**

Dari hasil PA ini masih ada beberapa kekurangan dan dimungkinkan untuk pengembangan lebih lanjut oleh karenanya terdapat beberapa saran sebagai berikut :

1. Penyempurnaan lagi dalam sistem perancangan mekanik yang lebih kuat dan bisa berjalan lebih stabil dalam membawa buku.
2. Penyempurnaan tuning PID supaya dapat berjalan mengikuti garis dengan respon yang lebih baik.
3. Penyempurnaan sistem program secara keseluruhan supaya controller dapat menjalankan program dengan baik dengan penggunaan RAM lebih sedikit
4. Pengujian yang lebih banyak dan dilakukan di tempat yang cukup luas seperti perpustakaan umum yang luas.

## DAFTAR PUSTAKA

- [1] B. Jiang, G.-y. Yang, S. Shen and J. Ma, "Design of books arranging robot control system in library," 6th International Conference on Information Engineering for Mechanics and Materials (ICIMM 2016), vol. I, pp. 748-754, 2016.
- [2] Chaudhary, A. (2015). Library Management System: An Open alternative. Conference cum Workshop on DSpace Gandhinagar.
- [3] "otomasi.sv.ugm.ac.id," 09 October 2018. [Online]. Available: <https://otomasi.sv.ugm.ac.id/mobile-robot/>. [Accessed 07 July 2021].
- [4] B. Jiang, G.-y. Yang, S. Shen and J. Ma, "Design of books arranging robot control system in library," 6th International Conference on Information Engineering for Mechanics and Materials (ICIMM 2016), vol. I, pp. 748-754, 2016.
- [5] C. S, S. Kumari and P. Kumari, "AUTOMATED SMART ROBOT LIBRARY MANAGEMENT SYSTEM," International Research Journal of Modernization in Engineering Technology and Science, vol. 02, no. 07/July-2020, pp. 747-750, 2020.
- [6] H. R. Oktaviani, S. and R. E. Puspita, "Kualitas Layanan sebagai Strategi Peningkatan Kepuasan," Journal of Library and Information Science, vol. 3, pp. 159-174, 2019.
- [7] M. Pathak, M. Pandit, S. Kanthed and R. Sanvaliya, "Library Management Robot," International Journal of Engineering Research & Technology (IJERT), vol. 5, no. 10, October-2016, pp. 356-354, 2016.

- [8] Kaustubh Ghadge, Saurabh More, Pravin Gaikwad And Shrenik Chillal (2018), Robotic Arm for Pick and Place Application. IJMET 9 (1). Pp: 125 – 133. Kaustubh Ghadge, Saurabh More, Pravin Gaikwad And Shrenik Chillal (2018), Robotic Arm for Pick and Place Application. IJMET 9 (1). Pp: 125 – 133.
- [9] TAN, LEW, TAN, GOH, LEE & KHOR. (2010) A New Automated Food Delivery System Using Autonomous Track Guided Centre-Wheel Drive Robot .In IEEE Conference on Sustainable Utilization and Development in Engineering and Technology. Malaysia and 20th -21st November 2010. pp. 32-36.
- [10] [Francisco Rubio, Francisco Valero and Carlos Llopis-Albert (2019), A review of mobile robots: Concepts, methods, theoretical framework, and applications, International Journal of Advanced Robotic Systems. doi : 10.1177/1729881419839596.
- [11] Liam Lynch, Thomas Newe, John Clifford , Joseph Coleman, Joseph Walsh and Daniel Toal (2018). Automated Ground Vehicle (AGV) and Sensor Technologies- A Review. Twelfth International Conference on Sensing Technology (ICST).

## LAMPIRAN

### A.1 Program Mobile Robot

//Program header

```
#ifndef LIBRARYROBOT_H_
#define LIBRARYROBOT_H_

#include <Arduino.h>
// pin output motor
#define pwm1 5
#define pwm2 6
#define pwm3 9
#define pwm4 10
// pin sensor
#define s0 A6
#define s1 A1
#define s2 A0
#define s3 A3
#define s4 A2
#define s5 A7
#define st A4

const uint8_t sPin[7] = {s0, s1, s2, st, s3, s4, s5};
#define initlibraryrobot initlibraryrobot

#define ff true
#define bb false

// led
#define buzzer 8
#define led1 12
#define led2 13

void remotesetup();
void initlibraryrobot();
void manualcontrol();
void autocontrol();
void readsensor();
void lampusensor();
```

```

void checkComplete();

void motorjalankan(int8_t speed);
void motorjalankiri(int8_t speed);
void testmotorkanan(int8_t speed);
void testmotorkiri(int8_t speed);
void testkeduamotor(int8_t speed_kiri, int8_t speed_kanan);
void belokiri(int8_t speed);
void belokanan(int8_t speed);
void belokirimaju(int8_t speed);
void belokananmaju(int8_t speed);
void majutimer(int8_t speed, int delay_);
void linefindkanan(int8_t speed, int sensor);
void linefindkiri(int8_t speed, int sensor);
void lf_crossfind(int8_t speed);
void lf_delay(int8_t speed, int delay_);
void linefollower(int8_t speed);
void lf_crsfindskip(int8_t speed, bool arah, int skip);
void buzzeron(int jumlah);
void lf_crossfindArah(int8_t speed, bool arah);
int caseSensor();
int samplingSensor();
void lf_newcross(int speed_);
void pidvalue(float kp_, float ki_, float kd_);

int detectcross();
float sampling();
int satusensor(int pin);
void kirimdata(int data);
void ifReadcond(int data);

#endif

```

### Program .cpp

```
#include "library_robot.h"

void setup()
{
    // Setup pin out dan lain lain pada void setup arduino
    Serial.begin(9600);
    pinMode(pwm1, OUTPUT);
    pinMode(pwm2, OUTPUT);
    pinMode(pwm3, OUTPUT);
    pinMode(pwm4, OUTPUT);

    pinMode(sPin[0], INPUT);
    pinMode(sPin[1], INPUT);
    pinMode(sPin[2], INPUT);
    pinMode(sPin[3], INPUT);
    pinMode(sPin[4], INPUT);
    pinMode(sPin[5], INPUT);
    pinMode(sPin[6], INPUT);

    pinMode(buzzer, OUTPUT);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);

    // Void setup saat di Arduino IDE
    initlibraryrobot();
}

// read new sensor
int caseSensor()
{
    int sensor[7];
    int casesensor = 0;

    sensor[0] = analogRead(sPin[0]);
```

```

    sensor[0] = (sensor[0] > 220) ? 1 : 0;
    sensor[1] = analogRead(sPin[1]);
    sensor[1] = (sensor[1] > 220) ? 1 : 0;
    sensor[2] = analogRead(sPin[2]);
    sensor[2] = (sensor[2] > 220) ? 1 : 0;
    sensor[3] = analogRead(sPin[3]);
    sensor[3] = (sensor[3] > 220) ? 1 : 0;
    sensor[4] = analogRead(sPin[4]);
    sensor[4] = (sensor[4] > 220) ? 1 : 0;
    sensor[5] = analogRead(sPin[5]);
    sensor[5] = (sensor[5] > 220) ? 1 : 0;
    sensor[6] = analogRead(sPin[6]);
    sensor[6] = (sensor[6] > 220) ? 1 : 0;

    casesensor = (sensor[6] * 1) + (sensor[5] * 2) + (sensor[4] * 4) +
    (sensor[3] * 8) + (sensor[2] * 16) + (sensor[1] * 32) + (sensor[0] * 64);
    return casesensor;
}

// sampling data sensor
int samplingSensor()
{
    int sampling = 0;
    sampling = caseSensor();
    int error = 0;
    switch (sampling)
    {
        case 0b00000000:
            error = 1;
            break;
        case 0b10000000:
            error = -60;
            break;
        case 0b11000000:

```



```
    error = -50;
    break;
case 0b0100000:
    error = -40;
    break;
case 0b0110000:
    error = -30;
    break;
case 0b0010000:
    error = -20;
    break;
case 0b0011000:
    error = -10;
    break;
case 0b0001000:
    error = 0;
    break;
case 0b0001100:
    error = 10;
    break;
case 0b0000100:
    error = 20;
    break;
case 0b0000110:
    error = 30;
    break;
case 0b0000010:
    error = 40;
    break;
case 0b0000011:
    error = 50;
    break;
case 0b0000001:
    error = 60;
```

```

        break;
    }
    return error;
}

// pid
float kp = 0.00;
float ki = 0.0;
float kd = 0.00;
void pidvalue(float kp_, float ki_, float kd_)
{
    kp = kp_;
    kd = kd_;
    ki = ki_;
}

// line follower baru
void lf_newcross(int speed_)
{
    float errorBf, error, lasterror = 0, sumerror = 0;
    float KP = kp, KI = ki, KD = kd;
    float P, I, D, out;
    float speedKa, speedKi;
    bool isFind = false;
    bool errorB = false;

    while (!isFind)
    {
        error = samplingSensor();
        errorB = detectcross();
        if (errorB == 1)
            isFind = true;
        if (error != 0)
            sumerror += error;
    }
}

```

```

else
    sumerror = 0;

P = error * KP;
D = (error - lasterror) * KD;
I = sumerror * KI;
out = P + I + D;

error = lasterror;

speedKi = speed_ + out;
speedKa = speed_ - out;

if (speedKi >= 255)
    speedKi = 255;
if (speedKa >= 255)
    speedKa = 255;
if (speedKi <= -255)
    speedKi = -255;
if (speedKa <= -255)
    speedKa = -255;

if (error > 2)
{
    belokirimaju(speed_ + out);
}
else if (error < -2)
{
    belokananmaju(speed_ + (out * -1));
}
else
{
    testkeduamotor(speed_, speed_);
}

```

```

    }
}

// Kirim Serial
void kirimdata(int data)
{
    Serial.print(data);
}

// check complete
void ifReadcond(int data)
{
    bool isComplete = false;
    String _str;
    char mystr[10];
    int _strint = 0;
    while (!isComplete)
    {
        testkeduamotor(0, 0);
        while (Serial.available())
        {
            _str = Serial.readString();
            _strint = _str.toInt();
            Serial.print(_str);
        }
        if (_strint == data)
            isComplete = true;
    }
    isComplete = false;
}

void checkComplete()
{
    bool isComplete = false;

```

```

String _str;
char mystr[10];
while (!isComplete)
{
    testkeduamotor(0, 0);
    while (Serial.available())
    {
        _str = Serial.readString();
        Serial.print(_str);
    }
    if (_str == "0")
        isComplete = true;
}
isComplete = false;
}

// Program membaca garis dengan salah satu sensor yang di pilih
dengan input nomor pin
int satusensor(int pin)
{
    int hasilbaca;
    hasilbaca = analogRead(sPin[pin]);
    if (hasilbaca > 350)
        hasilbaca = 1;
    return hasilbaca;
}

// Program untuk mendeteksi perempatan dengan membaca sensor
paling kanan dan kiri
int detectcross()
{
    int hasilbaca;
    if (analogRead(sPin[0]) > 350 && analogRead(sPin[5]) > 350)
    {

```

```

        hasilbaca = 1;
    }
    else
    {
        hasilbaca = 0;
    }
    return hasilbaca;
}

void motorjalankan(int8_t speed)
{
    if (speed >= 0)
    {
        analogWrite(pwm1, speed);
        analogWrite(pwm2, 0);
    }
    if (speed < 0)
    {
        analogWrite(pwm1, 0);
        analogWrite(pwm2, speed * -1);
    }
}

void motorjalankiri(int8_t speed)
{
    if (speed >= 0)
    {
        analogWrite(pwm3, speed);
        analogWrite(pwm4, 0);
    }
    if (speed < 0)
    {
        analogWrite(pwm3, 0);
        analogWrite(pwm4, speed * -1);
    }
}

```

```

    }
}

void testmotorkanan(int8_t speed)
{
    motorjalankanan(speed);
}

void testmotorkiri(int8_t speed)
{
    motorjalankiri(speed);
}

void testkeduamotor(int8_t speed_kiri, int8_t speed_kanan)
{
    motorjalankanan(speed_kanan);
    motorjalankiri(speed_kiri);
}

void belokiri(int8_t speed)
{
    motorjalankanan(speed);
    motorjalankiri(-speed);
}

void belokanan(int8_t speed)
{
    motorjalankanan(-speed);
    motorjalankiri(speed);
}

void belokirimaju(int8_t speed)
{
    motorjalankanan(speed);

```

```

    motorjalankiri(speed - (speed / 2));
}

void belokananmaju(int8_t speed)
{
    motorjalankanan(speed - (speed / 2));
    motorjalankiri(speed);
}

void majutimer(int8_t speed, int delay_)
{
    bool isTimeout = false;
    unsigned long timeStart = millis();

    while (!isTimeout)
    {
        testkeduamotor(speed, speed);
        if ((unsigned long)millis() - timeStart > delay_)
        {
            isTimeout = true;
            testkeduamotor(0, 0);
        }
    }
}

void linefindkanan(int8_t speed, int sensor)
{
    bool isFound = false;
    while (!isFound)
    {
        int pin = satusensor(sensor);
        belokanan(speed);
        if (pin == 1)
        {

```



```

        isFound = true;
    }
}
testkeduamotor(0, 0);
}

void linefindkiri(int8_t speed, int sensor)
{
    bool isFound = false;
    while (!isFound)
    {
        int pin = satusensor(sensor);
        belokiri(speed);
        if (pin == 1)
        {
            isFound = true;
        }
    }
    testkeduamotor(0, 0);
}

```

## **A.2 Program Mekanik Naik Turun**

```

const int dirPin = 4;
const int stepPin = 7;
const int stepsPerRevolution = 9300;
const int stepsPerRevolution3 = 8000;
const int stepsPerRevolution1 = 200;

#include "ServoTimer2.h"
ServoTimer2 servo1;

#define ls1 A2 //pendorong depan
#define ls2 A3 //pendorong belakang
#define ls3 A4 //pendorong atas
#define ls4 A5

```

```

#define pwm 9
#define in1 A0
#define in2 A1
#define speed_ 100
int _read[4] = {0, 0, 0, 0};

bool sw1 = false;
bool sw2 = false;
bool sw3 = false;

#include <SoftwareSerial.h>
SoftwareSerial seriala(3, 2);
String _str = "-";
char mystr[10];
int innt = 0;

void setup()
{
    servo1.attach(6);
    servo1.write(1500);
    delay(2000);
    Serial.begin(9600);
    while (!Serial) continue;
    seriala.begin(9600);
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);

    pinMode(ls1, INPUT_PULLUP);
    pinMode(ls2, INPUT_PULLUP);
    pinMode(ls3, INPUT_PULLUP);
    pinMode(ls4, INPUT_PULLUP);
    pinMode(pwm, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    analogWrite(pwm, 0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);

    Serial.println("start auto home");

```

```

int sts = 0;
delay(200);
while (!sw3)
{
    digitalWrite(dirPin, LOW);
    bacaswitch();
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
    sts++;
    if (_read[2] == 0)
    {
        sw3 = true;
    }
}
sts = 0;
digitalWrite(dirPin, HIGH);
for (int x = 0; x < 100; x++)
{
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
}
delay(200);
sw3 = false;
}

int flags = 0;
void loop()
{
    while (seriala.available())
    {
        _str = seriala.readString();
        innt = _str.toInt();
        Serial.println(innt);
        if (innt == 1 || innt == 2)
        {

```

```

servo1.write(2250);
delay(1000);
tahan(stepsPerRevolution);
seriala.print(0);
innt = 0;
}
else if (innt == 3)
{
    servo1.write(2250);
    delay(1000);
    tahan(stepsPerRevolution3);
    seriala.print(0);
    innt = 0;
}
else if (innt == 10)
{
    delay(200);
    dorong();
    seriala.print(0);
    innt = 0;
}
else if (innt == 9)
{
    delay(200);
    _home();
    seriala.print(0);
    innt = 0;
}
else
{
    analogWrite(pwm, 0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(stepPin, LOW);
    digitalWrite(stepPin, LOW);
}
}
analogWrite(pwm, 0);
digitalWrite(in1, LOW);

```

```

digitalWrite(in2, LOW);
digitalWrite(stepPin, LOW);
digitalWrite(stepPin, LOW);
sw1 = 0;
sw2 = 0;
sw3 = 0;
innt = 0;
for (int a = 0; a <= 3; a++)
{
  _read[a] = 0;
}
}

void _home()
{
  Serial.println("start auto home");
  int sts = 0;
  delay(200);
  while (!sw3)
  {
    digitalWrite(dirPin, LOW);
    bacaswitch();
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
    sts++;
    if (_read[2] == 0)
    {
      sw3 = true;
    }
  }
  sts = 0;
  digitalWrite(dirPin, HIGH);
  for (int x = 0; x < 100; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
  }
}

```

```

    delayMicroseconds(1000);
  }
  delay(200);
  Serial.println("end auto home");
}
void tahan(int step_)
{
  //turun
  digitalWrite(dirPin, HIGH);
  for (int x = 0; x < step_; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
}
void turun()
{
  //turun
  digitalWrite(dirPin, HIGH);
  for (int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
  delay(500); // Wait a second
  while (!sw1)
  {
    bacaswitch();
    analogWrite(pwm, speed_);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    if (_read[0] == 0)
    {
      sw1 = true;
    }
  }
}

```

```

    }
    analogWrite(pwm, 0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    sw1 = false;
    while (!sw2)
    {
        bacaswitch();
        analogWrite(pwm, speed_);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        if (_read[1] == 0)
        {
            sw2 = true;
        }
    }
    analogWrite(pwm, 0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    sw2 = false;
    delay(500);

    sw1 = false;
    sw2 = false;
}

void dorong()
{
    while (!sw1)
    {
        bacaswitch();
        analogWrite(pwm, 255);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        if (_read[0] == 0)
        {
            sw1 = true;
        }
    }
}

```

```

analogWrite(pwm, 0);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
sw1 = false;
servo1.write(1800);
delay(1000);
while (!sw2)
{
    bacaswitch();
    analogWrite(pwm, speed_);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    if (_read[1] == 0)
    {
        sw2 = true;
    }
}
analogWrite(pwm, 0);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
sw2 = false;
delay(500);
}

void naik()
{
    digitalWrite(dirPin, LOW);
    for (int x = 0; x < stepsPerRevolution1; x++)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(1000);
    }
    delay(500);
    while (!sw1)
    {
        bacaswitch();
        analogWrite(pwm, 255);
    }
}

```



```

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
if (_read[0] == 0)
{
    sw1 = true;
}
}
analogWrite(pwm, 0);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
sw1 = false;
//mundur
servo1.write(1800);
delay(1000);
while (!sw2)
{
    bacaswitch();
    analogWrite(pwm, speed_);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    if (_read[1] == 0)
    {
        sw2 = true;
    }
}
analogWrite(pwm, 0);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
sw2 = false;
delay(500);
}

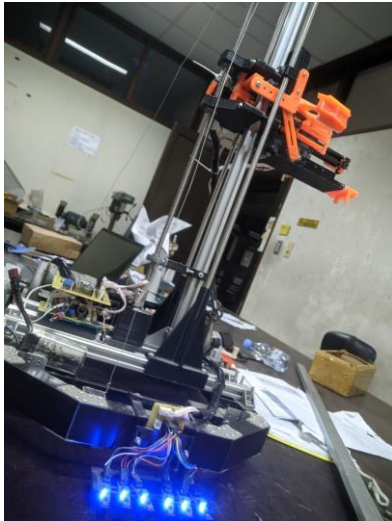
void bacaswitch()
{
    _read[0] = digitalRead(ls1);
    _read[1] = digitalRead(ls2);
    _read[2] = digitalRead(ls3);
    _read[3] = digitalRead(ls4);
}

```

## **B. Lampiran Gambar Hasil**



**Tampilan Keseluruhan Mobile Robot**



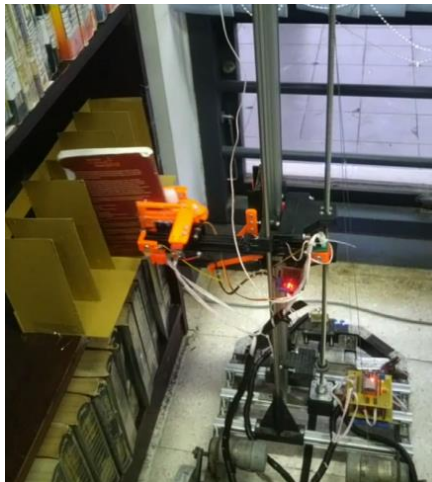
**Mobile Robot dengan Line Follower**



**Mekanik Pendorong**



**Pengujian di Perpustakaan D3**



**Pengujian di Perpustakaan D3**

### C. Lampiran Biodata Penulis



**Nama** : Heri Nur Alim  
**Tempat / Tanggal Lahir** : Ponorogo, 16 Oktober 1999  
**Alamat** : Jl. Sedap Malam no. 3, Desa Sekaran,  
Kecamatan Siman, Ponorogo.  
**Telp / HP** : 085606954140  
**Halaman Web** : [herrynur.github.io/eifra](https://herrynur.github.io/eifra)  
**Github** : [github.com/herrynur](https://github.com/herrynur)  
**Linkedin** : [linkedin.com/heri-nur-alim](https://linkedin.com/heri-nur-alim)

#### **Riwayat Pendidikan :**

- SDN 1 Sekaran
- SMPN 1 Ponorogo
- SMAN 1 Ponorogo
- D4 Teknik Elektronika Politeknik Elektronika Negeri Surabaya