

**LAPORAN AKHIR**  
**PRAKTIKUM ROBOTIKA LANJUT**



**Disusun Oleh:**

- 1. Mira Aprilia Safitri (23090620005)**
- 2. Bisjarah Annaya (23090620008)**
- 3. Bagas Danu Saputro (23090620019)**

**D4 TEKNIK ELEKTRONIKA**  
**DEPARTEMEN TEKNIK ELEKTRO DAN ELEKTRONIKA**  
**FAKULTAS VOKASI**  
**UNIVERSITAS NEGERI YOGYAKARTA**

**2025**

## **I. Rancangan Sistem**

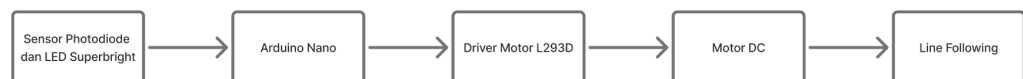
### **1.1 Deskripsi Umum Sistem**

Robot line follower merupakan robot yang dapat bergerak otomatis mengikuti lintasan berupa garis kontras yang tinggi, biasanya menggunakan garis hitam di atas permukaan berwarna lebih terang. Mekanisme utama robot adalah membaca posisi garis menggunakan sensor photodiode dan LED superbright, kemudian mikrokontroler memproses sensor dan mengatur kecepatan serta arah putaran motor melalui driver motor.

Pada praktikum ini robot dirancang agar mampu:

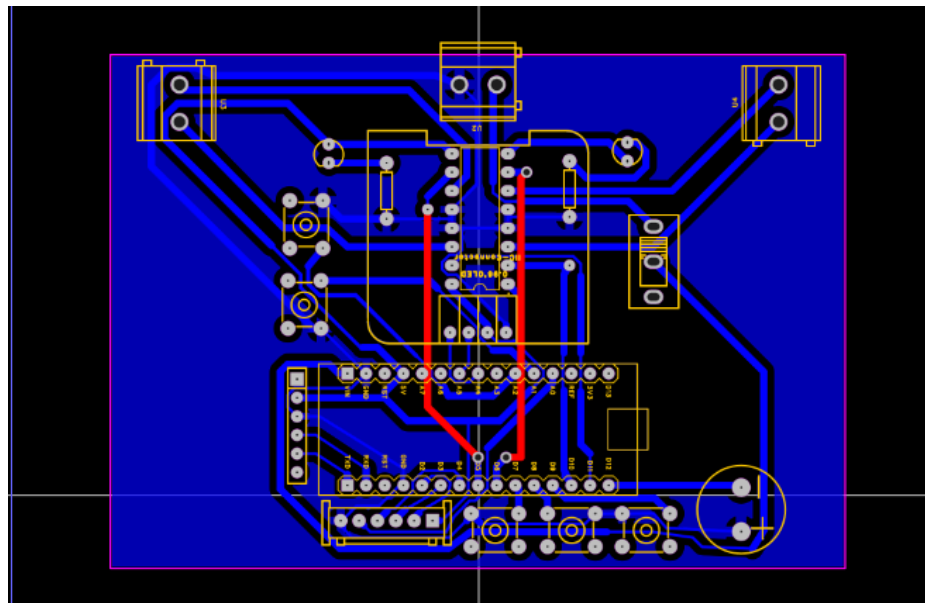
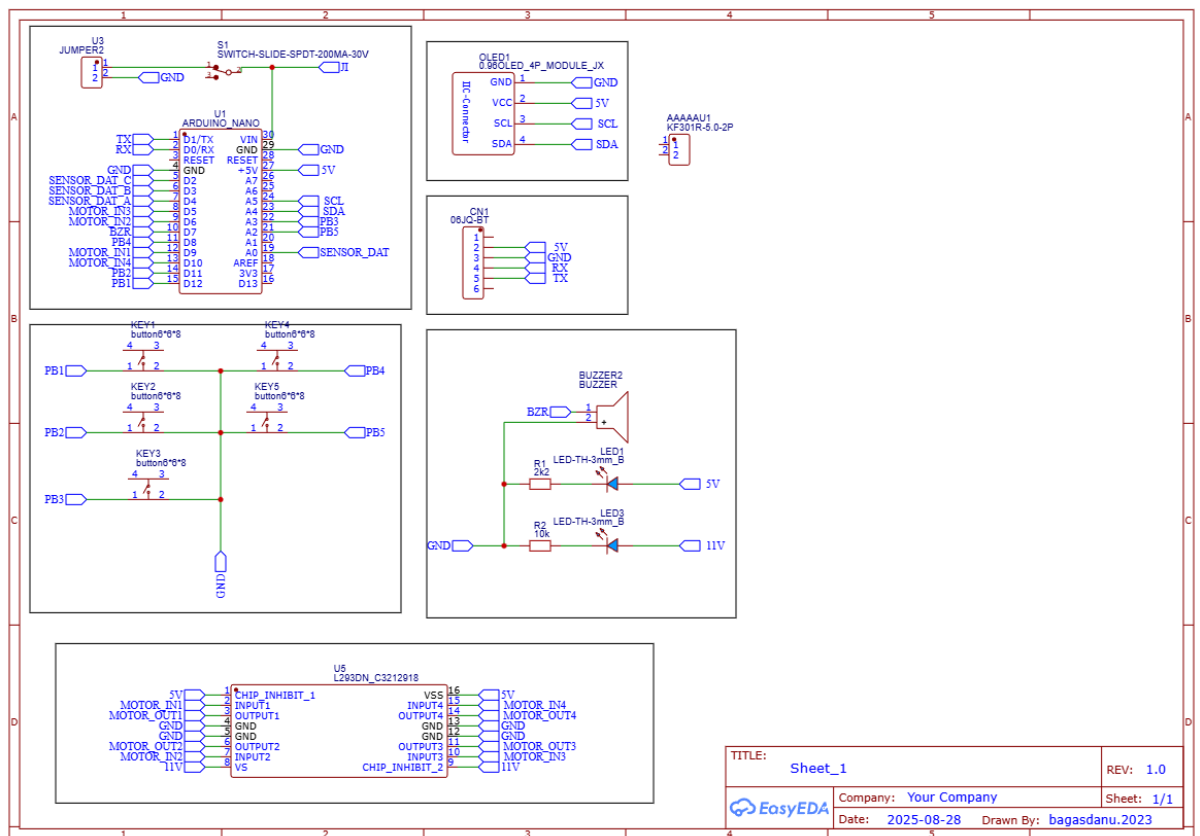
1. Mendeteksi garis dengan akurat menggunakan sensor photodiode dan LED superbright.
2. Menjaga posisi agar tetap berada di atas garis selama pergerakan.
3. Melakukan koreksi arah secara real-time ketika terjadi penyimpangan.
4. Mampu mengikuti lintasan lurus maupun belokan.

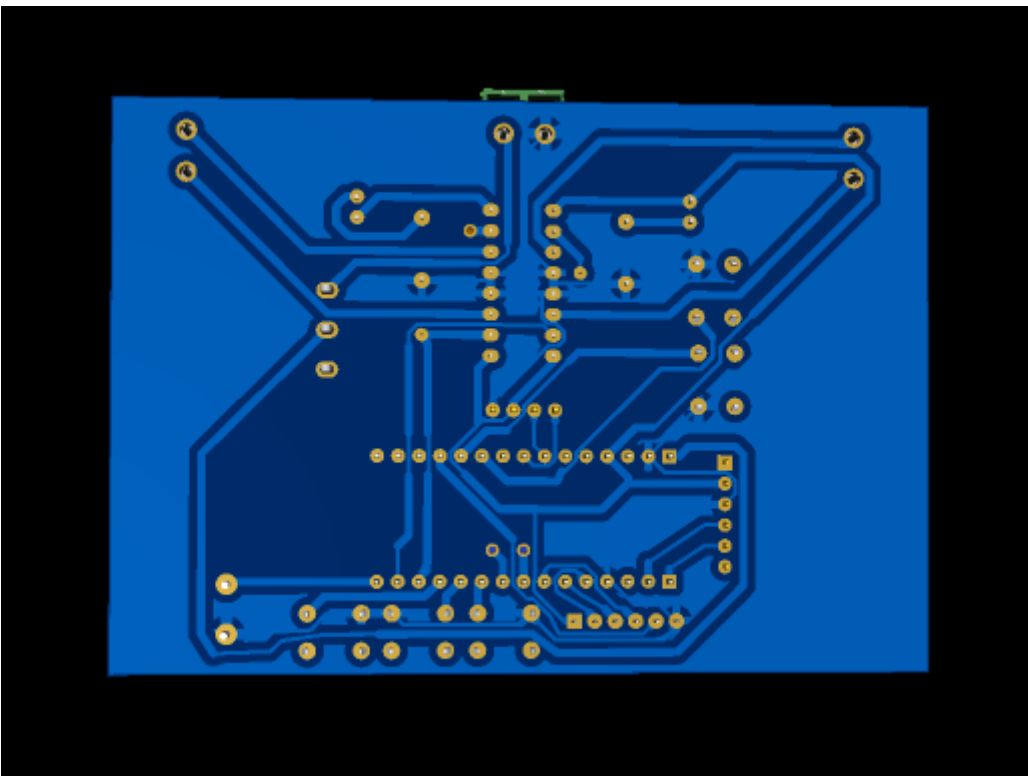
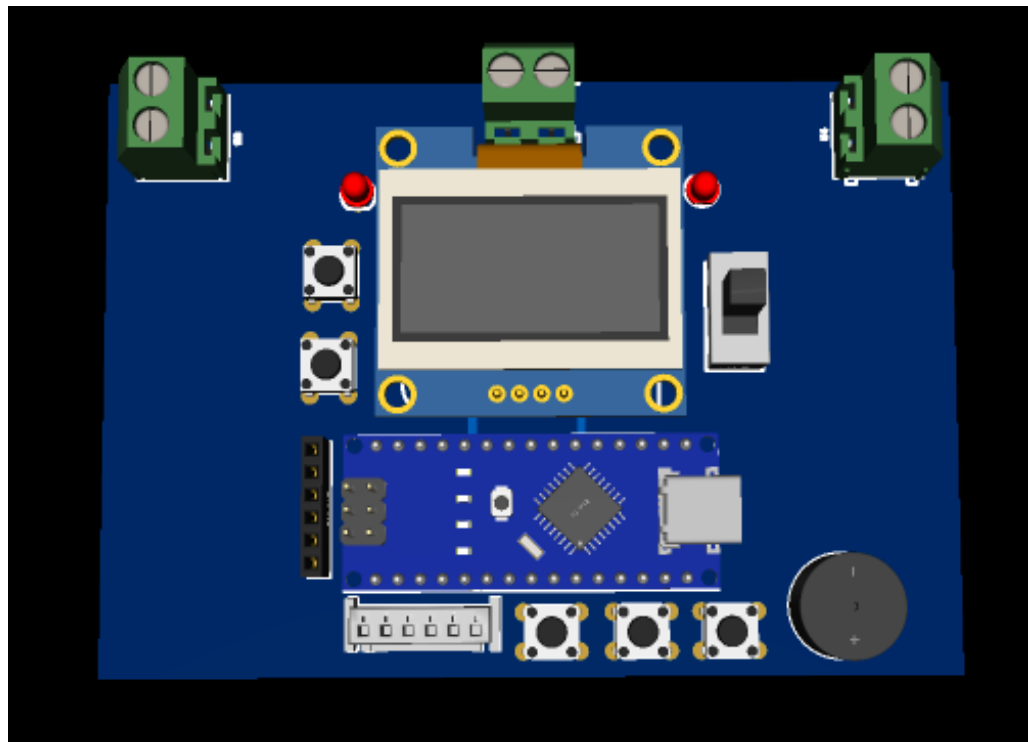
### **1.2 Diagram Blok Sistem**



## **1.3 Perancangan Hardware**

### **1.3.1 PCB Mainboard**

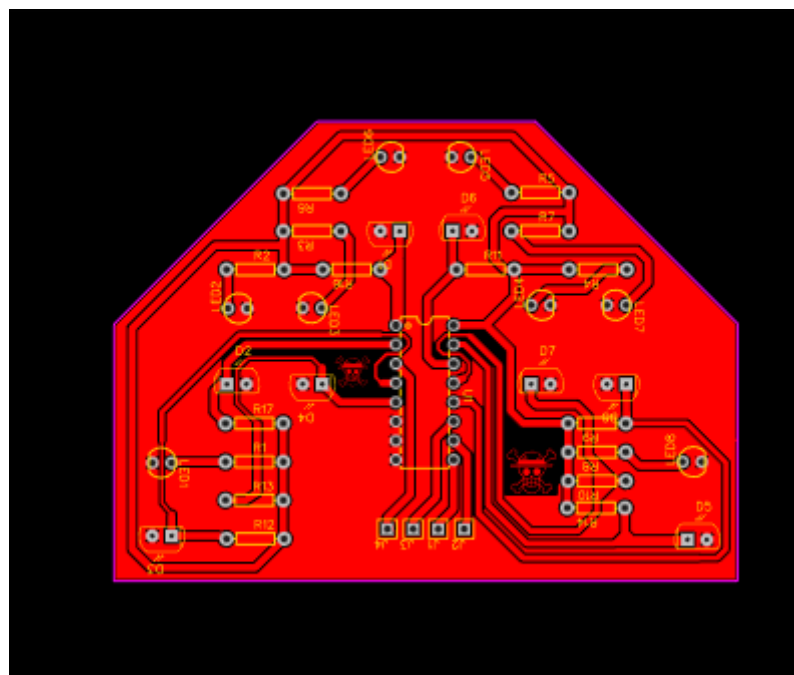
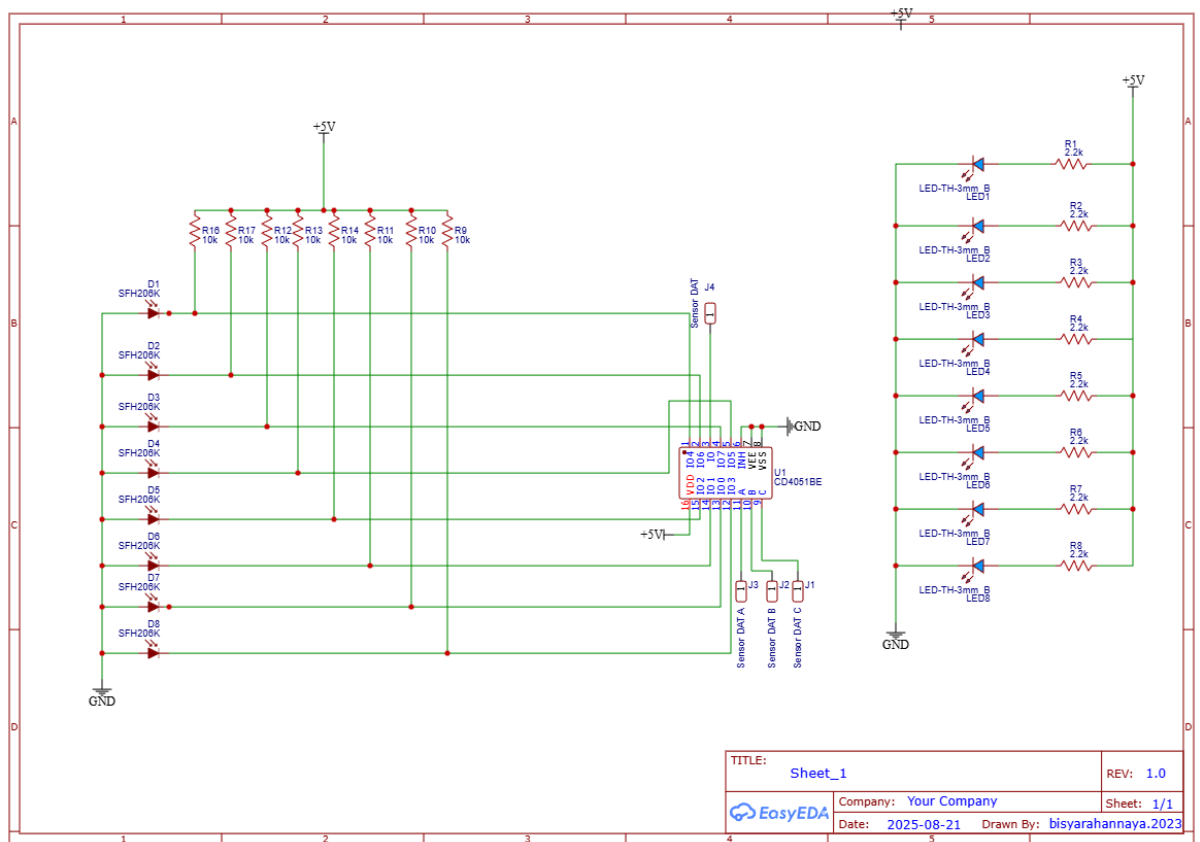


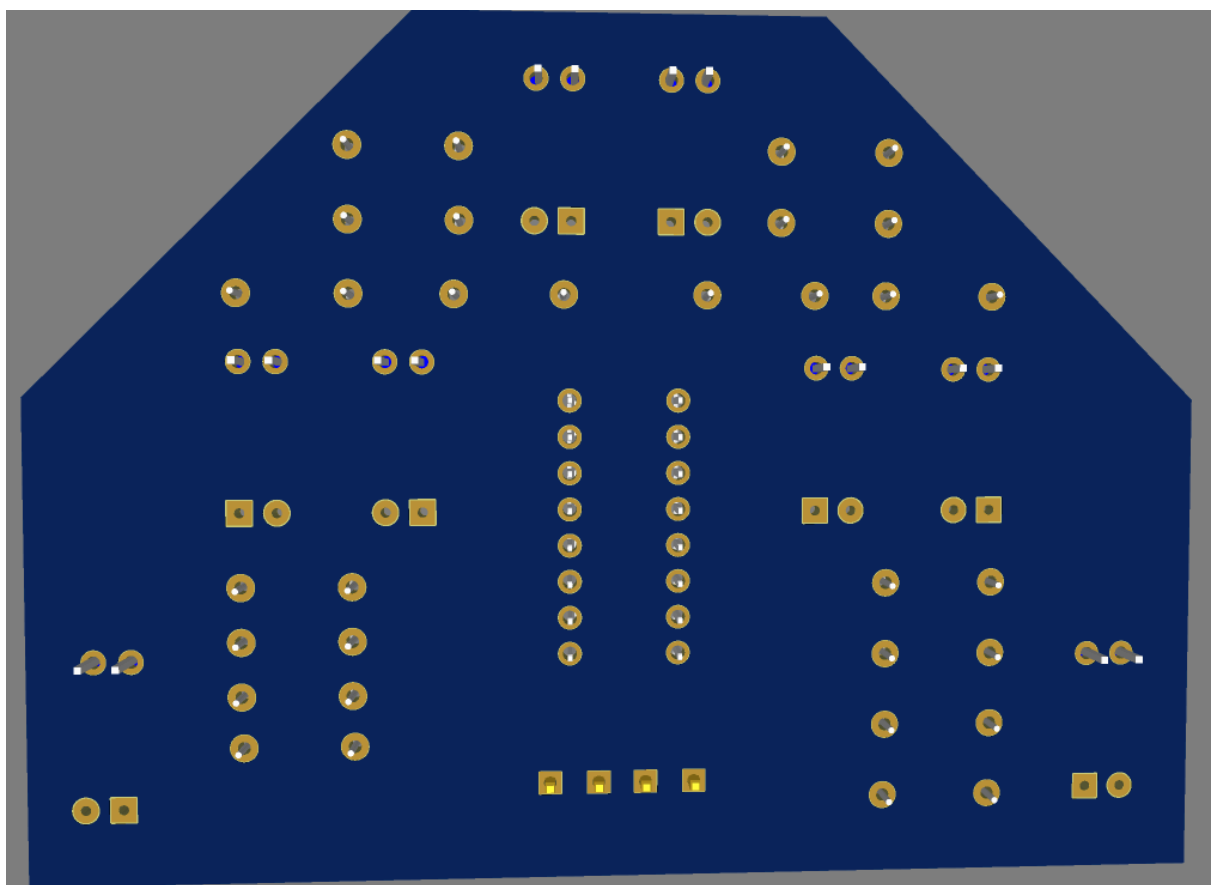
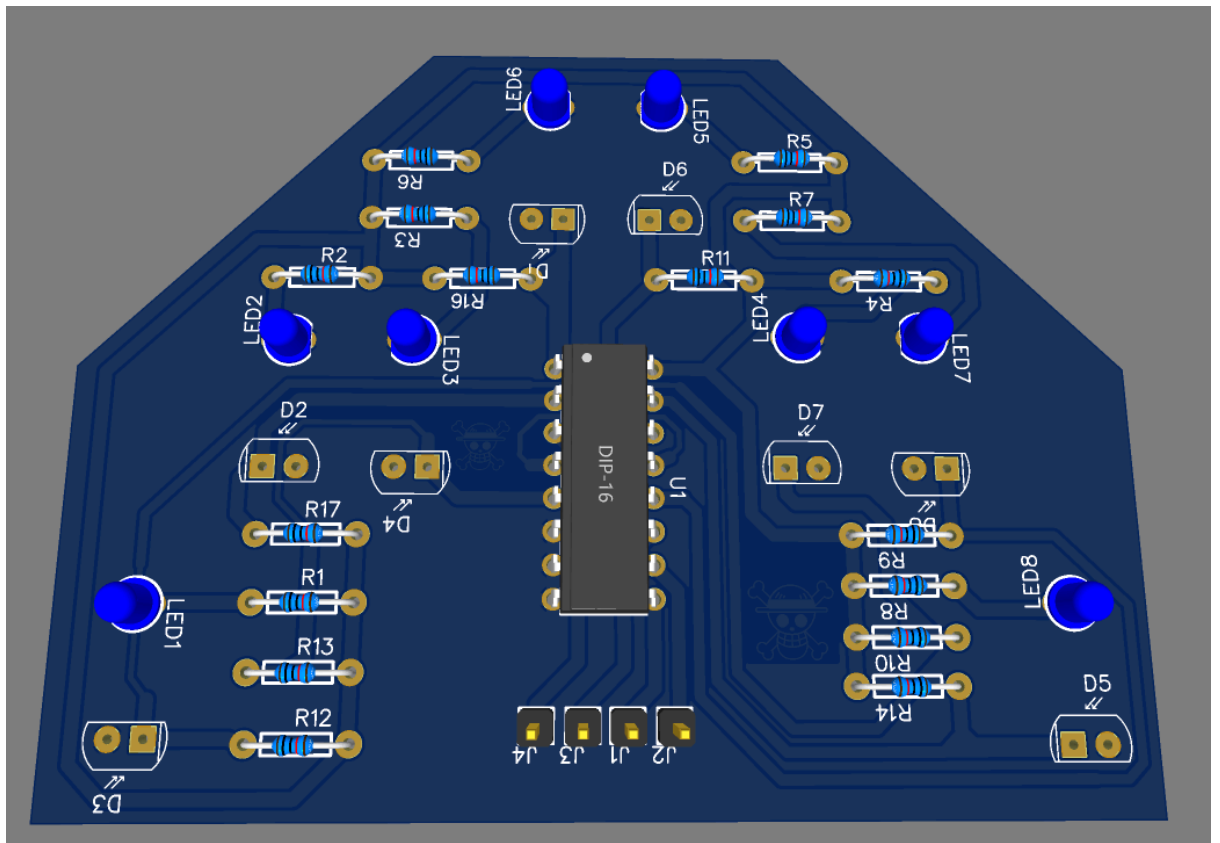


Pada sistem ini, mikrokontroler yang digunakan adalah Arduino Nano, yang berfungsi sebagai pusat pengendali untuk membaca data sensor line, memprosesnya menggunakan algoritma line following, dan menghasilkan sinyal kendali untuk aktuator. Driver motor L293D digunakan sebagai penguat sinyal agar motor DC tipe N20 dapat bergerak sesuai instruksi mikrokontroler. Selain itu, sistem juga dilengkapi dengan sebuah buzzer yang berfungsi sebagai indikator suara, misalnya untuk pemberitahuan saat robot dinyalakan atau ketika terjadi kondisi tertentu. Terdapat

lima buah saklar yang dipasang sebagai pengendali manual dan pemilih mode, sehingga memudahkan pengguna dalam mengoperasikan robot.

### 1.3.2 PCB Sensor



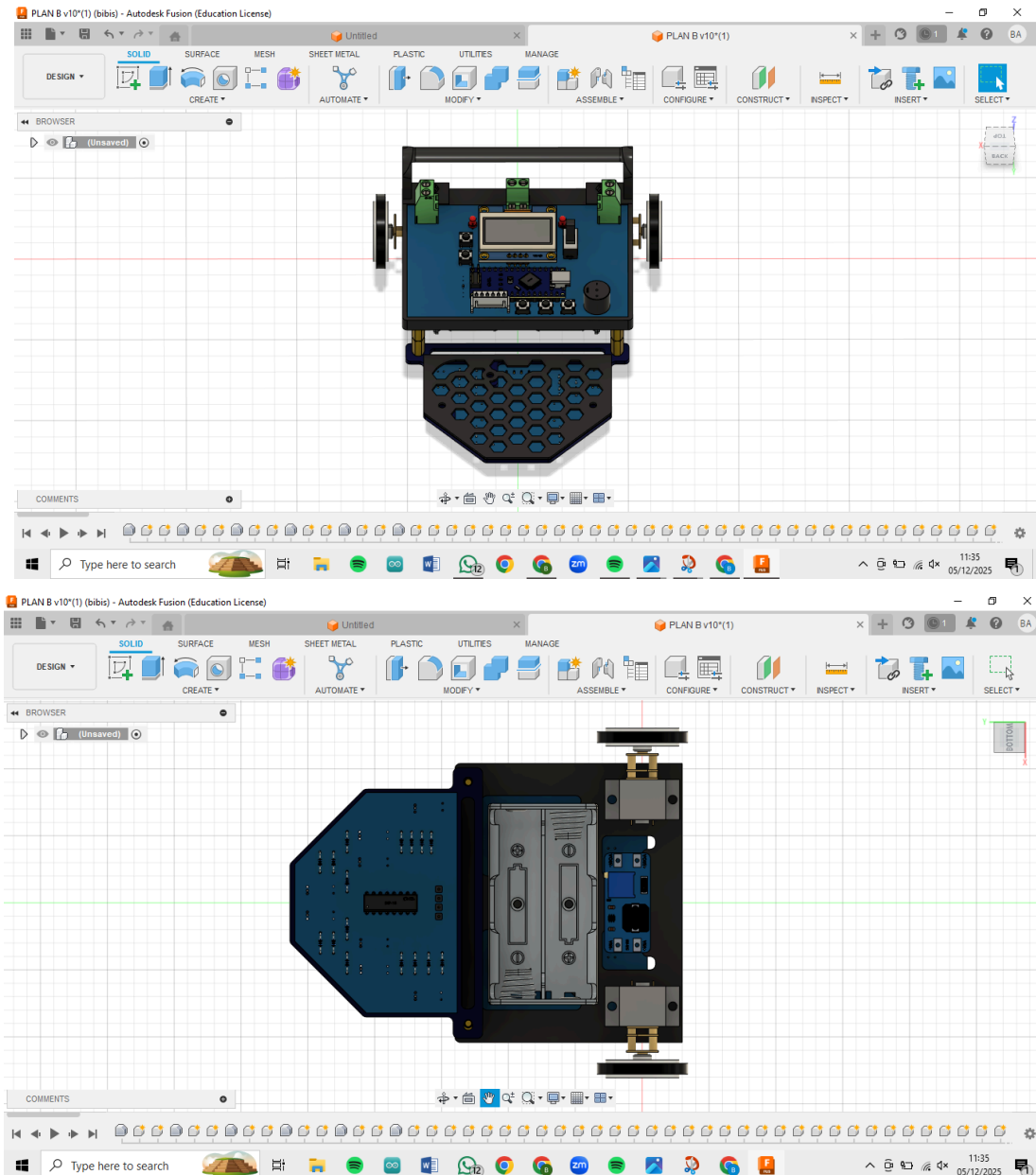


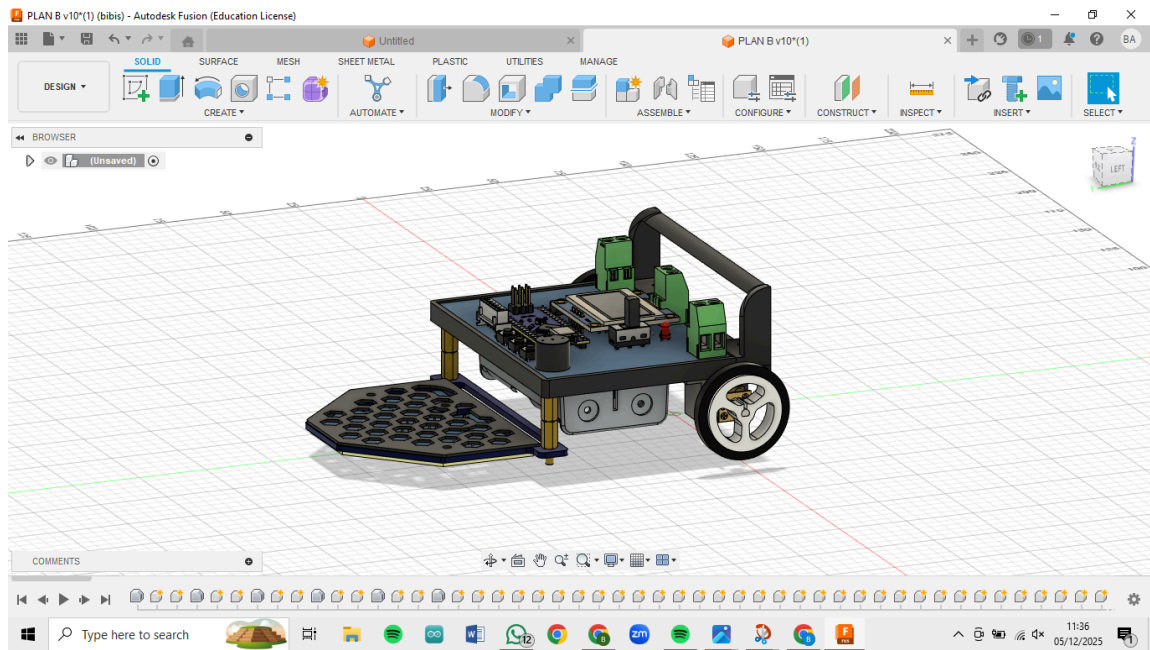
Sensor line photodiode dan LED superbright digunakan untuk mendeteksi garis hitam berdasarkan pantulan cahaya. Ketika sensor berada di atas area hitam, pantulan cahaya rendah sehingga output sensor bernilai kecil. Sebaliknya, area putih memberi pantulan lebih tinggi.

Konfigurasi:

1. Jumlah sensor : 3 sensor
2. Posisi : dipasang dibagian depan robot dengan jarak 1 cm antar sensor

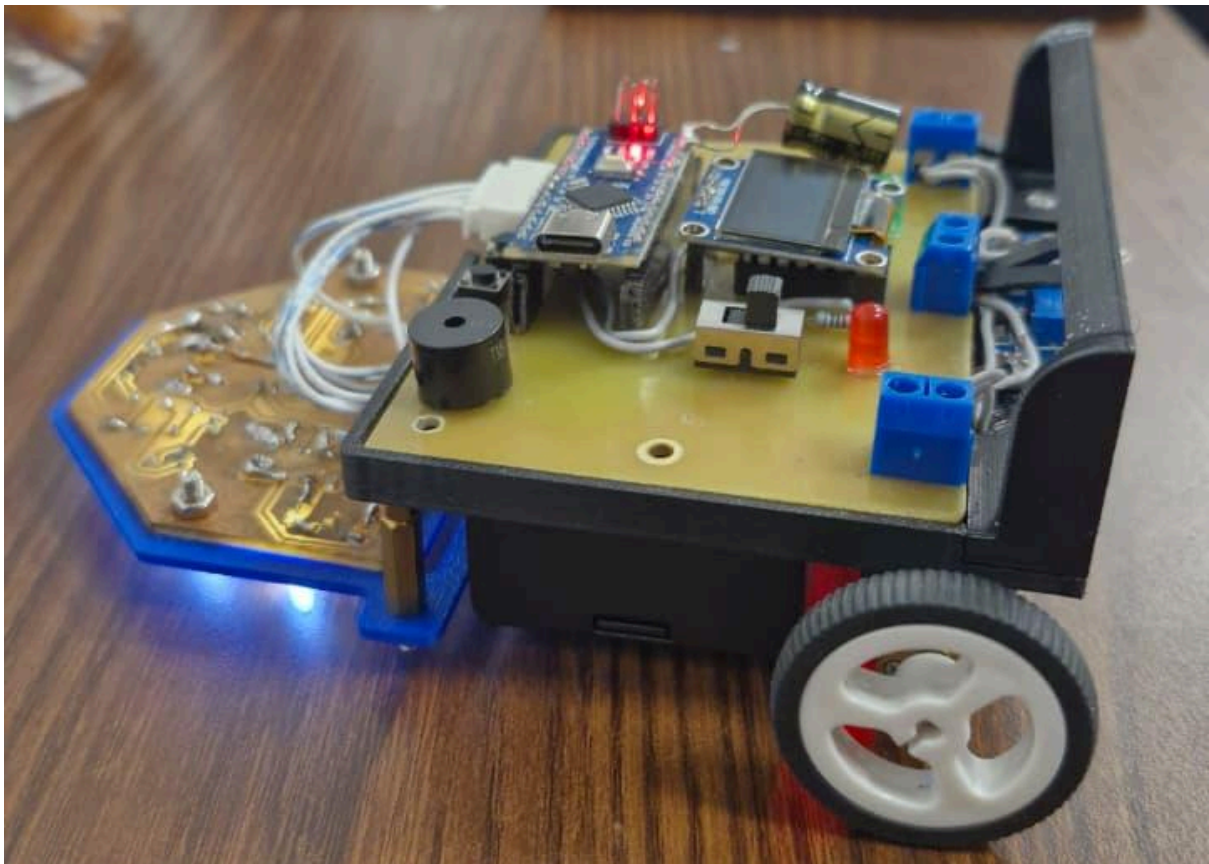
### 1.3.3 Body Robot



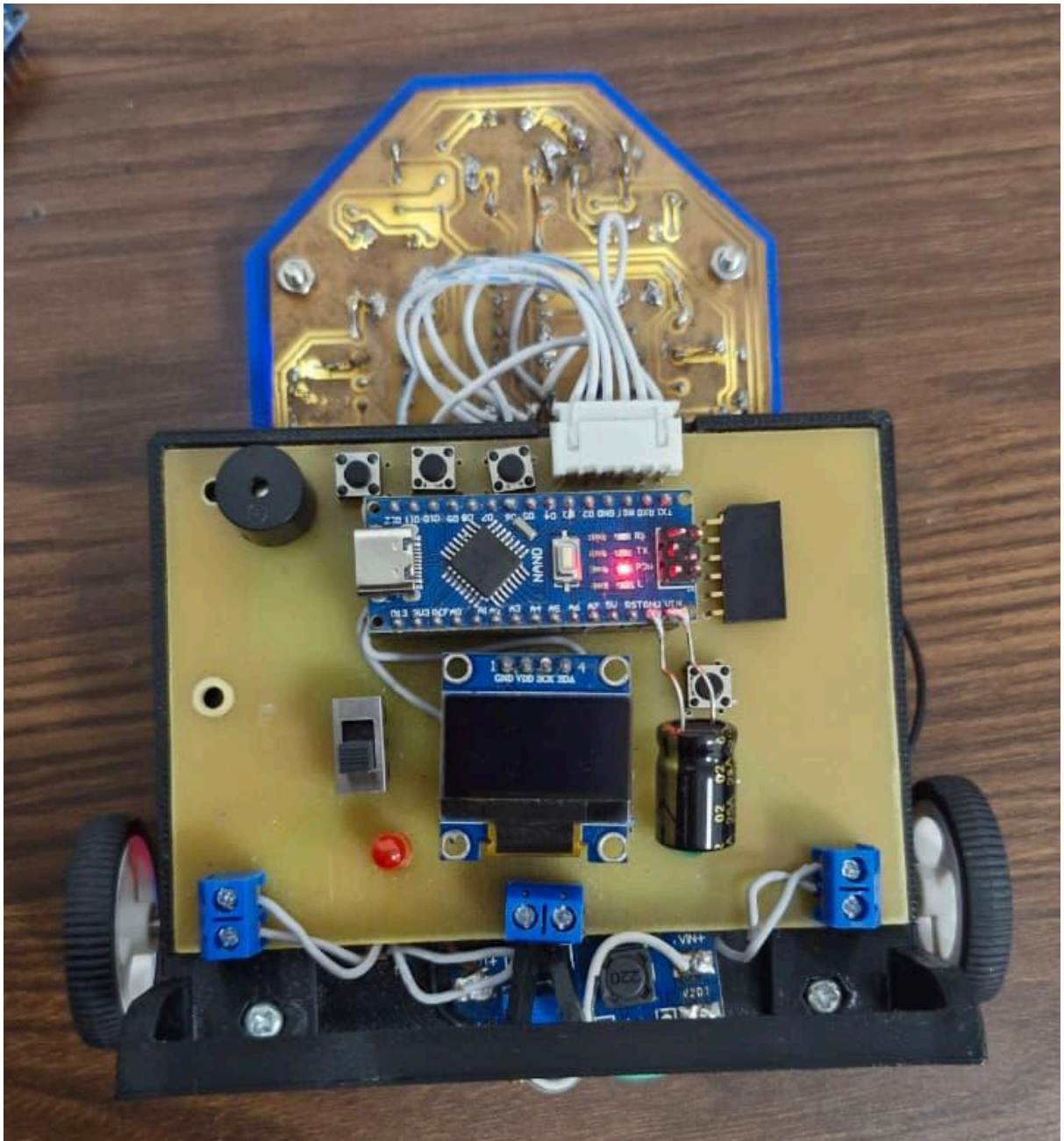


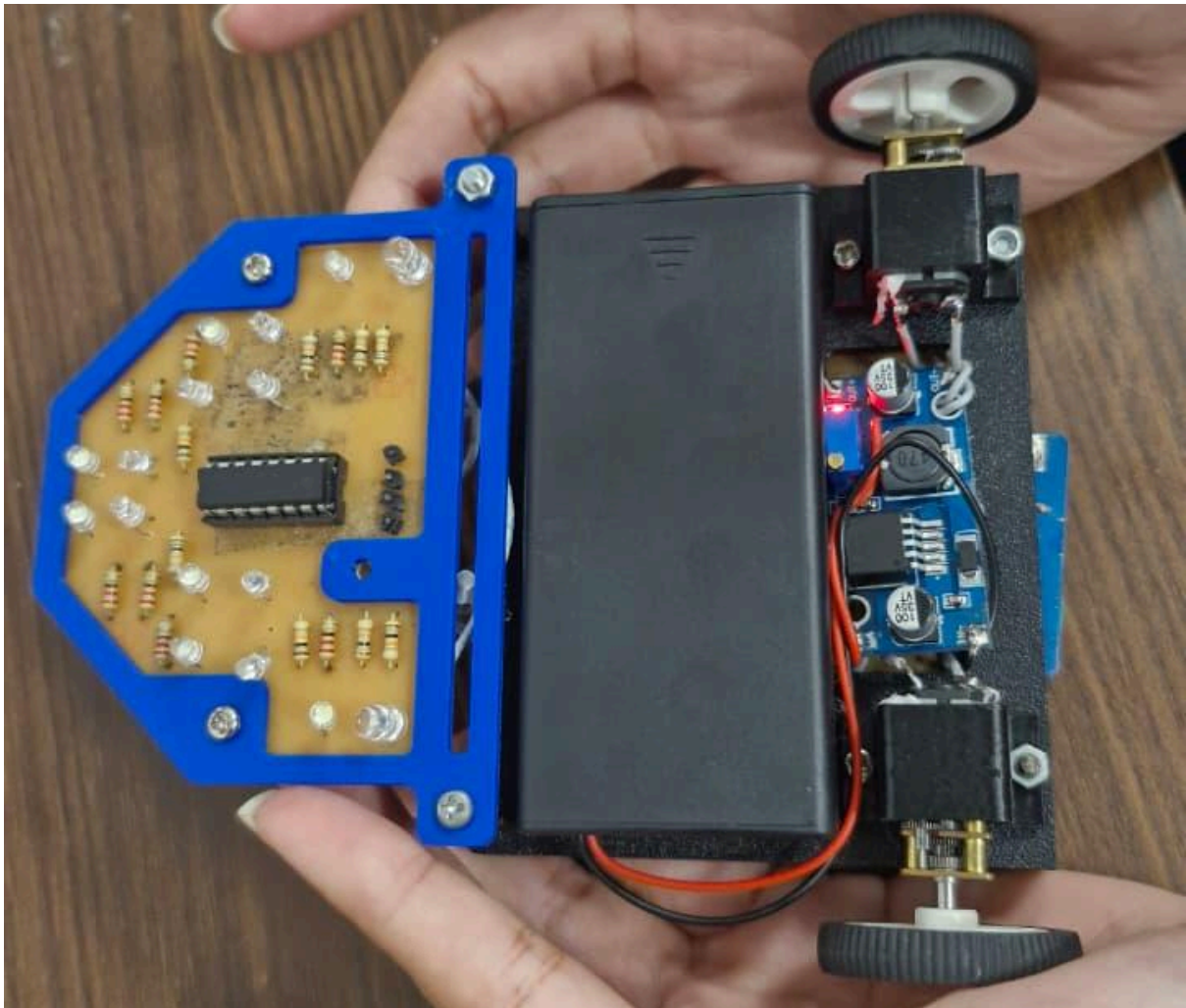
## II. Implementasi Sistem

### 2.1 Implementasi Hardware









## 2.2 Implementasi Program

```
#include <LineFollower.h>
LineFollower lf;

/*
Lakukan kalibrasi sensor terlebih dahulu.

Program ini akan menjalankan robot sesuai dengan index yang sudah
ditetapkan.
contoh track untuk latihan bisa di download di www.ichibot.id
*/

void setup() {
  /* Init Robot */
  lf.begin();

  /* Set PID (Kp, Ki, Kd) */
  lf.setPID(17, 0.4, 0.3);

  /* Logika untuk mengatur polaritas motor jika arah terbalik
  Cari kombinasi yang tepat agar logika arah putaran motor dc sesuai.
  pilih salah satu dibawah ini:
  ichibot.polaritasMotor(0, 0);
  ichibot.polaritasMotor(1, 0);
  */
}
```

```

ichibot.polaritasMotor(0, 1);
ichibot.polaritasMotor(1, 1);
*/
lf.polaritasMotor(1, 0);

/* Mengatur logika belok robot atau index */
lf.setIndex(0, LURUS, 100, S_SEMUA, 120, 300);
lf.setIndex(1, KANAN, 200, S_KANAN, 120, 150);
lf.setIndex(2, KANAN, 200, S_KANAN, 120, 350);
lf.setIndex(3, KIRI, 200, S_KIRI, 120, 900);
lf.setIndex(4, KANAN, 200, S_KANAN, 120, 100);
lf.setIndex(5, KANAN, 200, S_KANAN, 120, 900);
lf.setIndex(6, LURUS, 200, S_KIRI_KANAN, 0, 0);

/* Mengatur nomor index untuk robot berhenti */
lf.stopIndex(6);
}

void loop() {
  /* Program utama menjalankan robot, tidak perlu dirubah */
  lf.RobotLoop();
}

```

### 2.3 Kendala

Selama proses implementasi sistem, kami menghadapi beberapa kendala. Pertama, Arduino Nano mengalami kerusakan akibat menerima tegangan 11 volt. Untuk mengatasi hal tersebut, kami menambahkan modul step-down sehingga Arduino memperoleh suplai tegangan yang lebih aman, yaitu 8 volt. Selain itu, kami juga menambahkan kapasitor untuk menstabilkan arus dan tegangan karena sebelumnya robot menunjukkan perilaku yang kurang stabil saat berjalan.

Kendala berikutnya muncul saat pengujian pada lintasan yang lebih kompleks. Pada tahap ini, salah satu jalur sensor mengalami kerusakan (jalur putus), sehingga pembacaan sensor tidak optimal dan robot tidak dapat melakukan line following dengan baik. Selain itu, setelah melalui tikungan, robot sering kali tidak dapat kembali berada tepat di tengah garis dan cenderung bergerak ke arah kiri. Masalah ini kemungkinan disebabkan oleh tuning PID yang belum optimal, sehingga respons koreksi robot setelah belokan menjadi kurang seimbang. Untuk mengatasi jalur sensor yang bermasalah, kami melakukan penyolderan ulang agar pembacaan kembali normal.

Kendala terakhir berkaitan dengan penggunaan motor. Pada awalnya, motor DC N20 60 rpm dapat mengikuti garis dengan baik. Namun menjelang hari pelaksanaan, kami mencoba mengganti motor dengan kecepatan 1000 rpm. Kecepatan yang terlalu tinggi membuat robot menjadi agresif dan sulit dikendalikan, sehingga memerlukan proses tuning PID ulang. Pada akhirnya, motor diganti kembali ke N20 60 rpm, dan dengan konfigurasi tersebut robot dapat berjalan lebih stabil serta mampu mengikuti garis dengan baik tetapi masih perlu tuning PID karena robot jalannya masih belum mulus.

### III. Uji Coba

#### 3.1 Skenario Pengujian

Pengujian dilakukan dengan beberapa kondisi lintasan:

1. Lintasan lurus dan berbelok tanpa garis putus
2. Lintasan lurus dan berbelok dengan garis putus
3. Lintasan Kompleks (saat lomba)

#### 3.2 Parameter Pengujian

Parameter yang diuji:

1. Waktu tempuh lintasan
2. Jumlah keluar lintasan
3. Stabilitas pergerakan
4. Keberhasilan menyelesaikan lintasan

#### 3.3 Prosedur Pengujian

1. Robot ditempatkan di titik awal lintasan.
2. Robot dinyalakan dan dibiarkan mengikuti garis.
3. Hasil diamati dan dicatat yang meliputi waktu tempuh lintasan, jumlah kesalahan, serta apakah robot menyelesaikan lintasan.

### IV. Hasil dan Pembahasan

#### 4.1 Hasil Pengujian

Lintasan	Link Video	Waktu Tempuh	Keluar Garis	Keberhasilan
Lintasan lurus dan berbelok tanpa garis putus	<a href="#">Lintasan Lurus</a>	44 detik	Tidak	Berhasil
Lintasan lurus dan berbelok dengan garis putus	<a href="#">Lintasan Putus-Putus</a>	30 detik	Tidak	Berhasil
Lintasan Kompleks (lintasan lomba)	<a href="#">Lintasan Kompleks</a>	53 detik	Keluar garis saat ingin berbelok tajam setelah cp 2	Berhasil meskipun sempat keluar jalur

#### 4.2 Analisis

Pada saat melakukan pengujian dengan menggunakan lintasan lurus dan berbelok tanpa garis putus, robot mampu mengikuti garis secara stabil tanpa keluar dari jalur. Hal ini

dikarenakan lintasan masih cukup sederhana serta mampu memperlihatkan bahwa pembacaan sensor dan parameter PID cukup mampu dalam mengontrol gerakan robot.

Selanjutnya pada pengujian lintasan lurus dan berbelok dengan garis putus-putus, robot tetap dapat mengikuti garis dan membaca garis putus-putus tanpa keluar dari jalurnya. Hal ini membuktikan bahwa robot mampu mengatasi kondisi ketika garis menghilang (garis putus-putus-) serta menjaga arah gerak dengan memanfaatkan tuning PID sehingga robot tetap dapat mengikuti jalur lintasan.

Pada lintasan terakhir atau lintasan kompleks (lintasan lomba), robot kelompok kami pada saat menghadapi tikungan tajam sempat mengalami keluar jalur lintasan sebelum pada akhirnya kembali mengikuti jalur lintasan. Kondisi ini menunjukkan bahwa sistem kontrol masih belum optimal untuk menangani manuver ekstrem yang membutuhkan respon koreksi cepat dan presisi yang tinggi. Faktor lain yang mempengaruhi adalah kecepatan motor dan juga tuning PID yang masih belum sesuai untuk lintasan yang tajam.

## **V. Kesimpulan**

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa robot line follower kami mampu berfungsi dengan baik pada lintasan lurus, berbelok, dan lintasan garis putus-putus. Pada kedua lintasan tersebut, robot tidak keluar garis dan dapat menyelesaikan jalur lintasan dengan stabil, hal ini menandakan bahwa pembacaan sensor serta pengaturan daya sudah bekerja secara optimal setelah dilakukan perbaikan dengan menambahkan modul step-down dan kapasitor untuk menstabilkan tegangan.

Namun, pada lintasan kompleks robot kami masih mengalami kendala, terutama ketika menghadapi manuver ekstrem atau tikungan tajam. Robot sempat keluar dari jalur sebelum pada akhirnya kembali mengikuti jalur. Hal ini dipengaruhi oleh beberapa faktor yang terjadi selama proses implementasi, seperti kerusakan jalur sensor yang sempat mengganggu pembacaan sensor, serta tuning PID yang belum optimal untuk menangani manuver ekstrem. Selain itu, perubahan motor ke 1000 rpm juga sempat mengurangi kemampuan robot sehingga memerlukan penyesuaian ulang sebelum akhirnya kembali menggunakan motor dengan 60 rpm. Secara keseluruhan, meskipun terdapat kendala pada sistem daya, sensor, dan tuning PID, robot kelompok kami tetap dapat mencapai tujuan utama yaitu mengikuti garis pada berbagai lintasan.