

**PENERAPAN KONSEP ELEKTRONIKA DALAM  
PROJECT BASED LEARNING: MERANCANG DAN  
MEMBANGUN ROBOT LINE FOLLOWER**



**KELOMPOK 1**

ANGGOTA :

<b>NAMA</b>	<b>NIM</b>
Arfandi Ardiansyah	(223443052)
M.Hafizhan Ziyen Mailendra	(223443061)
Miftah Rizky Alamsyah	(223443062)
Rizki Triamadewa	(223443069)

**TEKNOLOGI REKAYASA INFORMATIKA INDUSTRI  
TEKNIK OTOMASI MANUFAKTUR DAN MEKATRONIKA  
POLITEKNIK MANUFAKTUR BANDUNG**

**2024**

## **KATA PENGANTAR**

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan laporan berjudul “Pengembangan Robot Line Follower Dalam Proyek Elektronika”. Laporan ini disusun sebagai salah satu tugas mata kuliah Elektronika yang diampu oleh Ibu Hilda Khoirunnisa, S.Tr.T., M.Sc.Eng. Selain itu, laporan ini bertujuan untuk menambah wawasan dan pengetahuan bagi pembaca mengenai Robot Line Follower yang dilengkapi dengan mode otomatis dan manual yang dapat diakses melalui aplikasi.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis menerima kritik dan saran yang membangun demi perbaikan laporan ini di masa depan. Semoga laporan ini bermanfaat bagi pembaca dan dapat menambah pengetahuan di bidang robotika dan elektronika.

Bandung, 09 Mei 2024

(Kelompok 1)

## DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>ii</b>
<b>DAFTAR ISI</b>	<b>iii</b>
<b>BAB I PENDAHULUAN</b>	<b>4</b>
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Alat dan Bahan	4
<b>BAB II LANDASAN TEORI</b>	<b>5</b>
2.1 Robot Line Follower	5
2.2 Komponen	5
2.2.1 Mikrokontroler : Arduino UNO	5
2.2.2 Motor Driver L219N	6
2.2.3 Sensor Infrared : TCRT5000 5 Channel	7
2.2.4 Bluetooth Module HC-05	8
2.2.5 DC Motor dan Gearbox	9
2.3 Software	10
2.3.1 Solidworks	10
2.3.2 Fritzing	12
2.3.3 Arduino IDE	13
2.3.4 MIT App Inventor	15
<b>BAB III PERANCANGAN DAN PRODUKSI</b>	<b>17</b>
3.1 Perencanaan	17
3.2 Alat dan Bahan	17
3.2.1 Daftar Alat	17
3.2.2 Daftar Komponen	18
3.2.3 Daftar Software	18
3.3 Perancangan	18
3.4 Tahap Produksi	19

3.4.1	Tahap 1 : Membuat robot tanpa adanya PCB .....	19
3.4.2	Tahap 2 : Membuat Case .....	20
3.4.3	Tahap 3 : Membuat layout Elektrikal Line Follower .....	25
3.4.4	Tahap 4 : Kalibrasi Sensor .....	30
3.4.5	Tahap 5 : Membuat Program pada Line Follower .....	31
3.4.6	Tahap 6 : Membuat Aplikasi .....	35
<b>BAB IV UJI COBA DAN ANALISA .....</b>		<b>40</b>
4.1	Mekanik .....	40
4.2	Elektrik .....	41
4.3	Program .....	42
4.4	Interface .....	43
<b>BAB V UJI COBA DAN ANALISA .....</b>		<b>44</b>
5.1	Kesimpulan .....	44
5.2	Saran .....	44
<b>DAFTAR PUSTAKA .....</b>		<b>45</b>
<b>DAFTAR LAMPIRAN .....</b>		<b>46</b>



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam pengembangan robot Line Follower, fokus utama adalah mengatasi sejumlah tantangan teknis yang kompleks. Hal ini meliputi perancangan arsitektur perangkat keras, pemilihan sistem kendali yang tepat, dan organisasi perangkat lunak untuk mengendalikan robot secara efisien. Implementasi robot Line Follower umumnya menggunakan sistem kontrol *closed loop*, di mana robot menerima umpan balik dari lingkungannya dan mengkoordinasikan aktuator untuk mencapai keadaan steady state. Programmer memainkan peran penting dalam merancang dan mengembangkan antarmuka pengguna yang memungkinkan kontrol robot secara manual maupun otomatis melalui aplikasi yang disesuaikan. Kemampuan untuk mengendalikan robot dengan fleksibilitas tinggi ini memberikan pengguna kemampuan untuk mengadaptasi robot sesuai kebutuhan spesifik mereka.

Dengan demikian, pengembangan robot Line Follower melibatkan pendekatan holistik yang mencakup berbagai aspek teknis dan desain. Dari perancangan sistem kendali hingga pembuatan antarmuka pengguna, setiap tahapan memerlukan perhatian yang cermat untuk memastikan robot dapat berfungsi secara optimal dan mudah digunakan oleh pengguna.

### **1.2 Rumusan Masalah**

1. Bagaimana merancang sebuah robot line follower berbasis Arduino UNO
2. Bagaimana mendesain elektrik pada PCB
3. Bagaimana mendesain case akrilik pada Solidwork untuk laser cutting
4. Bagaimana membuat Program pada robot Line Follower secara otomatis dan bisa diintegrasikan juga secara manual
5. Bagaimana mendesain Interface pada robot Line Follower untuk mendapatkan kondisi yang mendekati ideal ( $\text{error} \rightarrow 0$ )

### **1.3 Tujuan Penelitian**

1. Merancang sebuah robot line follower berbasis Arduino UNO
2. Mendesain elektrik pada Proteus untuk PCB
3. Mendesain case akrilik pada Solidwork untuk laser cutting
4. Membuat Program robot Line Follower secara otomatis dan bisa diintegrasikan secara manual
5. Mendesain Interface pada robot Line Follower untuk mendapatkan kondisi yang mendekati ideal ( $\text{error} \rightarrow 0$ )

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Robot Line Follower**

Robot line follower merupakan robot bergerak (*mobile robot*) yang memiliki kemampuan mengikuti jalur berupa garis yang telah ditentukan sebelumnya. Garis tersebut biasanya berupa kontras warna, seperti hitam pada permukaan putih atau sebaliknya. Robot Line follower memanfaatkan sensor untuk mendeteksi keberadaan garis dan membuat keputusan berdasarkan informasi tersebut. Sensor yang umum digunakan adalah sensor infrared yang mendeteksi permukaan reflektif dengan memancarkan sinar inframerah (IR) dan mendeteksi intensitas cahaya yang dipantulkan kembali. Sensor ini umumnya digunakan untuk mendeteksi garis hitam pada permukaan putih atau sebaliknya. Robot line follower didasarkan pada prinsip kontrol dan umpan balik (*feedback control*). Robot mengikuti jalur dengan cara mendeteksi posisi garis relatif terhadap pusat robot. Informasi ini kemudian diproses oleh *mikrokontroler* untuk menghasilkan sinyal kontrol yang dikirim ke motor penggerak. Sinyal kontrol ini akan mengatur kecepatan dan arah putaran motor, sehingga robot dapat menyesuaikan posisinya untuk tetap berada di jalur.

Pengembangan robot line follower juga melibatkan konsep robotika dasar seperti kinematika dan dinamika robot. Kinematika menentukan hubungan antara gerakan roda dan pergerakan keseluruhan robot, sedangkan dinamika memperhitungkan gaya dan momen yang mempengaruhi pergerakan robot. Pemahaman terhadap konsep-konsep ini membantu untuk mendesain robot yang lincah dan responsif dalam mengikuti jalur.

#### **2.2 Komponen**

##### **2.2.1 Mikrokontroler : Arduino UNO**

Mikrokontroler Arduino Uno merupakan platform pengembangan elektronika berbasis open-source yang populer digunakan dalam berbagai aplikasi seperti robotika, otomasi, dan Internet of Things. Arduino UNO didasarkan pada mikrokontroler ATmega328P 8-bit dari Atmel (Microchip), yang memiliki arsitektur Reduced Instruction Set Computer (RISC) untuk efisiensi dan konsumsi daya rendah. Arduino Uno menyediakan 14 pin input/output digital (dengan 6 diantaranya mendukung Pulse Width Modulation atau PWM), 6 input analog, resonator kristal 16 MHz untuk sinkronisasi clock, konektivitas USB untuk pemrograman dan catu daya, jack power eksternal, dan tombol reset manual.

Salah satu keunggulan utama Arduino Uno adalah penggunaan bahasa pemrograman berbasis C++ yang mudah dipelajari dan digunakan. Bahasa pemrograman ini memungkinkan pengguna untuk menulis program yang mengontrol pin input/output, membaca data dari sensor yang terhubung, melakukan perhitungan sederhana, serta berkomunikasi dengan perangkat lain melalui serial. Sifat open-source dari platform Arduino Uno turut mendukung pengembangan dan inovasi. Pengguna memiliki akses penuh terhadap skema perangkat keras dan kode sumber software, sehingga memungkinkan modifikasi dan pengembangan sesuai kebutuhan proyek.

Kombinasi antara perangkat keras yang terjangkau, bahasa pemrograman yang mudah dipelajari, dan komunitas pengguna yang besar dan aktif menjadikan Arduino Uno platform yang ideal untuk berbagai keperluan. Pengembang pemula maupun berpengalaman dapat memanfaatkan platform ini untuk membangun prototipe dengan cepat dan efisien, serta mewujudkan berbagai proyek inovatif di bidang elektronika dan robotika.

### 2.2.2 Motor Driver L298N

Motor driver L298N merupakan driver motor terintegrasi yang umum digunakan untuk mengendalikan motor DC dua arah (*bidirectional*). Driver ini memiliki kemampuan untuk mengontrol arah putaran motor (searah jarum jam dan berlawanan jarum jam) serta mengatur kecepatan putaran motor melalui modulasi pulse lebar (*Pulse Width Modulation* - PWM). L298N dilengkapi dengan empat terminal input untuk mengontrol dua motor DC, yaitu IN1, IN2, IN3, dan IN4. Mikrokontroler mengirimkan sinyal digital *HIGH* (5V) atau *LOW* (0V) ke terminal input ini untuk menentukan arah putaran motor. Kombinasi logika *HIGH/LOW* pada terminal input akan menentukan arah putaran dan kondisi diam (*stop*) motor.

Motor driver L298N juga memiliki dua terminal output untuk masing-masing motor (OUT1, OUT2 dan OUT3, OUT4). Terminal output ini terhubung ke motor DC dan mensuplai arus listrik yang lebih besar sesuai kebutuhan motor. Arus yang diberikan ke motor dikendalikan oleh sinyal PWM yang diberikan mikrokontroler pada terminal Enable (ENA dan ENB). Dengan mengatur *duty cycle* pada sinyal PWM, kecepatan putaran motor dapat dikendalikan secara proporsional. Semakin tinggi *duty cycle*, semakin besar arus yang disuplai ke motor sehingga menyebabkan motor berputar lebih cepat.

Penggunaan driver motor L298N menawarkan beberapa keuntungan dalam pengendalian motor DC. Keuntungan tersebut antara lain:

- Kemudahan Integrasi: L298N mengintegrasikan sirkuit pengendali motor dalam satu kemasan, sehingga memudahkan integrasi dengan mikrokontroler.
- Pengendalian Dua Arah: Driver ini memungkinkan untuk mengontrol arah putaran motor secara dua arah (bolak-balik).
- Kontrol Kecepatan: Melalui modulasi PWM, L298N memungkinkan untuk mengatur kecepatan putaran motor secara presisi.
- Arus Besar: Driver ini mampu mensuplai arus yang lebih besar dibandingkan dengan output langsung dari mikrokontroler, sehingga cocok untuk mengendalikan motor DC yang membutuhkan daya lebih besar.

Penggunaan driver motor ini memungkinkan pergerakan robot dikendalikan secara efisien dan presisi sesuai dengan kebutuhan manuver yang diperlukan dalam mengikuti lintasan.



### 2.2.3 Sensor Infrared : TCRT5000 5 Channel

Sensor TCRT5000 merupakan sensor inframerah reflektif yang dilengkapi dengan dioda inframerah (emitting diode - LED) sebagai pemancar dan fototransistor sebagai penerima. LED pada sensor memancarkan cahaya inframerah yang tidak tampak oleh mata manusia. Ketika cahaya inframerah ini mengenai permukaan objek, sebagian cahaya akan dipantulkan kembali. Fototransistor pada sensor berfungsi mendeteksi intensitas cahaya inframerah yang dipantulkan. Sensor TCRT5000 5 kanal memiliki susunan LED pemancar dan fototransistor penerima yang berjumlah lima, sehingga sensor ini dapat mendeteksi keberadaan objek dari lima arah atau garis secara simultan. Jarak deteksi optimal dari sensor ini umumnya berada di kisaran 2 cm hingga 4 cm. Semakin dekat objek dengan sensor, intensitas cahaya inframerah yang diterima oleh fototransistor semakin besar. Sebaliknya, semakin jauh objek, intensitas cahaya yang diterima semakin lemah.

Output dari sensor TCRT5000 berupa sinyal analog berupa tegangan listrik. Besarnya tegangan output bergantung pada intensitas cahaya inframerah yang diterima oleh fototransistor. Tegangan output yang tinggi menunjukkan objek berada dekat dengan sensor, sedangkan tegangan output yang rendah menunjukkan objek berada jauh dari sensor. Dalam rangkaian robot line follower, output analog dari sensor TCRT5000 5 kanal perlu dikonversi menjadi sinyal digital yang dapat dimengerti oleh mikrokontroler. Konversi ini biasanya dilakukan menggunakan ADC (Analog-to-Digital Converter). Dengan membaca nilai digital dari masing-masing kanal sensor, mikrokontroler dapat menentukan posisi robot terhadap garis hitam yang sedang diikuti. Informasi keberadaan garis tersebut kemudian digunakan oleh mikrokontroler untuk mengendalikan motor robot agar tetap berada pada jalurnya.

Penggunaan sensor TCRT5000 5 kanal menawarkan beberapa keuntungan dalam aplikasi line follower, yaitu:

- Deteksi Multi-Arah: Sensor ini memiliki kemampuan untuk mendeteksi keberadaan garis dari lima arah secara bersamaan, sehingga memberikan informasi yang lebih lengkap tentang lingkungan sekitar robot.
- Presisi Tinggi: Sensor ini mampu mendeteksi perubahan intensitas cahaya inframerah dengan baik, sehingga akurasi dalam mengikuti garis dapat ditingkatkan.
- Kemudahan Integrasi: Sensor TCRT5000 5 kanal memiliki bentuk yang ringkas dan mudah diintegrasikan dengan berbagai platform robot.

Sensor ini memungkinkan robot untuk secara efektif mendeteksi garis hitam dari berbagai arah dan membuat keputusan yang tepat untuk manuver koreksi lintasan.

#### 2.2.4 Bluetooth Module HC-05

Modul HC-05 merupakan perangkat elektronik kecil yang menggunakan teknologi Bluetooth versi 2.0+EDR untuk membangun koneksi nirkabel antara dua perangkat. Modul ini beroperasi pada frekuensi 2.4 GHz dan memiliki jangkauan komunikasi hingga 10 meter. Modul HC-05 memiliki enam pin konektor yang masing-masing memiliki fungsi spesifik:

- VCC (Voltage Supply): Pin ini menyediakan tegangan daya untuk modul, biasanya 3.3V atau 5V.
- GND (Ground): Pin ini merupakan ground atau referensi tegangan untuk modul.
- TXD (Transmitter Data): Pin ini mengeluarkan data serial dari modul ke perangkat lain.
- RXD (Receiver Data): Pin ini menerima data serial dari perangkat lain ke modul.
- EN (Enable): Pin ini mengaktifkan atau menonaktifkan fungsi komunikasi Bluetooth pada modul.
- STATE: Pin ini menunjukkan status koneksi Bluetooth, seperti terhubung, terputus.

Untuk membangun komunikasi menggunakan modul HC-05, diperlukan dua perangkat yang dilengkapi dengan modul tersebut. Salah satu perangkat akan bertindak sebagai master, yang menginisiasi koneksi dan mengontrol komunikasi, sedangkan perangkat lain bertindak sebagai slave, yang menerima koneksi dan merespons perintah dari master.

Komunikasi antara modul HC-05 dilakukan menggunakan protokol Bluetooth Serial Profile (SPP). Protokol ini memungkinkan transfer data serial antara dua perangkat Bluetooth. Data yang ditransfer dapat berupa teks, perintah, atau informasi sensor. Modul HC-05 dapat diprogram menggunakan perintah AT (Attention Command). Perintah AT memungkinkan pengguna untuk mengkonfigurasi pengaturan modul, seperti nama perangkat, baud rate, dan mode operasi. Pengaturan ini dapat dilakukan menggunakan terminal serial atau aplikasi Bluetooth di smartphone.

Penggunaan modul Bluetooth HC-05 menawarkan beberapa keuntungan dalam aplikasi robotika, yaitu:

- Komunikasi Nirkabel: Modul ini memungkinkan komunikasi antara robot dan perangkat kontrol tanpa perlu kabel, memberikan fleksibilitas dan kemudahan dalam pengoperasian.
- Jangkauan Luas: Jangkauan komunikasi hingga 10 meter memungkinkan robot dikendalikan dari jarak yang cukup jauh.

Modul Bluetooth HC-05 menjadi pilihan yang tepat untuk membangun sistem kendali robot line follower. Modul ini memungkinkan komunikasi nirkabel yang efektif antara robot dan smartphone, memberikan kontrol dan pemantauan robot yang lebih mudah dan fleksibel.

### 2.2.5 DC Motor dan Gearbox

Motor DC merupakan jenis motor listrik yang mengubah energi listrik searah (DC) menjadi energi gerak putar. Motor DC memiliki konstruksi yang sederhana, terdiri dari kumparan stator (bagian tetap) dan rotor (bagian berputar). Kumparan stator dialiri arus listrik DC, yang kemudian menghasilkan medan magnet. Rotor, yang biasanya berupa magnet permanen, berinteraksi dengan medan magnet stator, sehingga menimbulkan gaya putar.

Motor DC memiliki beberapa karakteristik penting, seperti:

- Tegangan Kerja: Motor DC beroperasi pada tegangan tertentu, biasanya antara 3V - 24V.
- Kecepatan: Kecepatan putaran motor DC dapat dikontrol dengan mengatur besarnya tegangan yang diberikan.
- Torsi: Torsi motor DC menunjukkan kemampuannya untuk menghasilkan gaya putar. Motor DC umumnya memiliki torsi yang tinggi pada putaran rendah.
- Arah Putaran: Arah putaran motor DC dapat diubah dengan membalik polaritas arus listrik yang diberikan.

Gearbox, atau reduktor, adalah komponen mekanis yang digunakan untuk mengubah karakteristik putaran motor, seperti kecepatan dan torsi. Gearbox terdiri dari serangkaian roda gigi dengan ukuran berbeda. Roda gigi yang lebih besar memiliki lebih sedikit gigi dibandingkan dengan roda gigi yang lebih kecil. Perbandingan ukuran roda gigi inilah yang menentukan rasio gear (gear ratio) dari gearbox. Rasio gear dihitung dengan membagi jumlah gigi pada roda gigi penggerak (driving gear) dengan jumlah gigi pada roda gigi yang digerakkan (driven gear).

Penggunaan gearbox dalam sistem penggerak robot menawarkan beberapa keuntungan:

- Peningkatan Torsi: Gearbox dengan rasio gear yang tinggi dapat meningkatkan torsi motor, sehingga robot dapat menghasilkan gaya putar yang lebih besar untuk menggerakkan beban yang berat.
- Penurunan Kecepatan: Gearbox dapat menurunkan kecepatan putaran motor menjadi nilai yang sesuai dengan kebutuhan robot. Kecepatan yang lebih rendah memungkinkan robot untuk bergerak dengan lebih presisi dan terkendali.
- Mempertahankan Arah Putaran: Gearbox dapat mempertahankan arah putaran motor meskipun terjadi perubahan arah putaran di dalam mekanisme gearbox itu sendiri.

Pemilihan motor DC dan gearbox yang tepat harus disesuaikan dengan kebutuhan robot. Faktor-faktor yang perlu dipertimbangkan dalam pemilihan tersebut meliputi:

- Beban yang akan digerakkan: Berat dan jenis beban yang akan digerakkan oleh robot akan menentukan kebutuhan torsi motor dan rasio gear gearbox.
- Kecepatan yang diinginkan: Kecepatan robot harus disesuaikan dengan fungsinya. Pemilihan motor DC dan gearbox yang tepat dapat menghasilkan kecepatan yang optimal.
- Ruang yang tersedia: Dimensi motor DC dan gearbox harus sesuai dengan ruang yang tersedia pada kerangka robot.

Pemilihan motor DC dan gearbox yang tepat akan memastikan robot memiliki kemampuan manuver dan kontrol kecepatan yang baik untuk mengikuti garis lintasan yang ditentukan.

## **2.3 Software**

### **2.3.1 Solidworks**

SolidWorks merupakan perangkat lunak Computer-Aided Design (CAD) 3D yang digunakan secara luas dalam bidang teknik mesin, desain produk, dan manufaktur. Skripsi ini menggunakan SolidWorks sebagai platform utama untuk melakukan perancangan design case robot line follower. SolidWorks menawarkan berbagai fitur untuk membantu proses perancangan 3D, diantaranya:

- Pembuatan Model Geometri 3D: SolidWorks menyediakan berbagai tool untuk membuat model geometri 3D, seperti fitur extrude, revolve, sweep, dan loft. Pengguna dapat membuat bentuk dasar hingga bentuk yang kompleks dengan presisi tinggi.
- Sketsa dan Fitur Parametris : SolidWorks menggunakan konsep sketsa dan fitur parametris untuk membangun model 3D. Sketsa merupakan gambaran 2D yang mendeskripsikan bentuk dasar objek. Dimensi dan hubungan antar elemen dalam sketsa dapat didefinisikan secara parametris, sehingga perubahan pada parameter tersebut dapat langsung mengubah keseluruhan model 3D secara otomatis.
- Assembly Modeling : SolidWorks memungkinkan pengguna untuk membangun model assembly, yaitu kumpulan komponen individual yang saling terhubung dan berfungsi bersama. Fitur ini berguna untuk memvisualisasikan keseluruhan struktur robot dan memastikan kompatibilitas antar komponen.
- Engineering Drawings : SolidWorks dapat digunakan untuk membuat gambar teknik (engineering drawings) yang detail dan komprehensif. Gambar teknik ini berisikan informasi dimensi, toleransi, dan detail lainnya yang diperlukan untuk proses manufaktur.

- Simulasi dan Analisa : SolidWorks menawarkan modul simulasi untuk menganalisa kinerja komponen dan assembly. Analisa ini dapat meliputi analisa gaya, analisa perpindahan panas, dan analisa lainnya, sehingga potensi permasalahan desain dapat diidentifikasi dan diatasi sebelum proses produksi.

Beberapa keuntungan menggunakan SolidWorks dalam perancangan design cas robot line follower, antara lain:

- Kemudahan Penggunaan : SolidWorks memiliki antarmuka yang intuitif dan user-friendly, sehingga mudah dipelajari dan digunakan bahkan bagi pengguna yang baru mengenal CAD 3D.
- Fitur Perancangan yang Lengkap : SolidWorks menyediakan berbagai fitur yang komprehensif untuk membuat model 3D yang detail dan akurat, serta memungkinkan pembuatan assembly dan gambar teknik yang dibutuhkan untuk proses manufaktur.
- Visualisasi 3D Realistik : SolidWorks mampu menghasilkan model 3D yang realistis dengan berbagai pilihan material, tekstur, dan pencahayaan. Visualisasi ini membantu perancang untuk memahami bentuk dan fungsi robot secara lebih baik.
- Simulasi dan Analisa : Fitur simulasi dan analisa yang ditawarkan SolidWorks dapat membantu mengidentifikasi potensi permasalahan desain sebelum proses produksi, sehingga menghemat waktu dan biaya.
- Standar Industri : SolidWorks merupakan software CAD 3D yang banyak digunakan di industri manufaktur. Penggunaan SolidWorks dalam perancangan robot line follower akan menghasilkan dokumentasi dan model 3D yang sesuai dengan standar industri.

Dengan berbagai kemudahan dan keuntungan yang ditawarkan, SolidWorks menjadi pilihan tepat untuk merancang komponen robot line follower dalam skripsi ini. Melalui SolidWorks, perancangan komponen robot dapat dilakukan secara efisien, akurat, dan sesuai dengan standar industri.

### 2.3.2 Fritzing

Fritzing merupakan software yang menawarkan perancangan design rangkaian listrik dalam bentuk PCB (Printed Circuit Board), dengan tampilan yang intuitif dan mudah digunakan, sehingga cocok bagi pengguna pemula maupun yang sudah berpengalaman.

Beberapa fitur utama Fritzing yang relevan untuk perancangan rangkaian elektronika meliputi:

- **Komponen Elektronik yang Beragam:** Fritzing menyediakan library komponen elektronika yang luas, meliputi komponen aktif seperti transistor, dioda, dan IC, serta komponen pasif seperti resistor, kapasitor, dan LED.
- **Breadboard Virtual:** Fritzing memiliki tampilan virtual breadboard yang menyerupai breadboard fisik yang biasa digunakan untuk membangun prototipe sirkuit. Komponen elektronik dapat diseret dan diletakkan pada breadboard virtual sesuai dengan skema rangkaian yang diinginkan.
- **Penghubung Rangkaian:** Fritzing menyediakan tool untuk membuat koneksi antar komponen secara visual pada breadboard virtual. Pengguna dapat dengan mudah menarik garis penghubung untuk merepresentasikan kabel yang menghubungkan kaki-kaki komponen dalam rangkaian sebenarnya.
- **Simulasi Sederhana:** Fritzing menawarkan fitur simulasi dasar untuk rangkaian elektronika sederhana. Fitur ini memungkinkan pengguna untuk melihat respon tegangan dan arus pada titik tertentu dalam rangkaian berdasarkan nilai komponen yang digunakan.
- **Dokumentasi Skema Rangkaian:** Fritzing dapat digunakan untuk membuat dokumentasi skema rangkaian secara profesional. Skema rangkaian yang dibuat di Fritzing dapat diekspor dalam berbagai format file gambar, seperti PNG atau SVG.

Beberapa keuntungan menggunakan Fritzing dalam perancangan rangkaian robot line follower, antara lain:

- **Mempercepat Proses Prototyping:** Fritzing memungkinkan pembuatan skema rangkaian dan layout breadboard secara virtual dengan cepat, sehingga proses iterasi dan perbaikan desain menjadi lebih efisien.
- **Meminimalisir Kesalahan:** Penggunaan komponen elektronik dan breadboard virtual dalam Fritzing dapat membantu mengurangi kesalahan yang mungkin terjadi saat membangun prototipe sirkuit secara manual di breadboard fisik.
- **Dokumentasi yang Rapi:** Fritzing menghasilkan skema rangkaian yang rapi dan profesional, sehingga memudahkan dokumentasi dan komunikasi dalam proses pengembangan robot line follower.

- Gratis dan Open-Source: Fritzing merupakan software open-source yang gratis untuk digunakan, sehingga menjadi pilihan yang hemat biaya dan mudah diakses.

Meskipun Fritzing menawarkan kemudahan dan fitur yang bermanfaat, perlu dicatat bahwa software ini memiliki keterbatasan dalam hal simulasi. Simulasi yang ditawarkan Fritzing bersifat sederhana dan tidak sedetail software simulasi elektronika professional. Oleh karena itu, Fritzing sebaiknya digunakan sebagai alat bantu untuk pembuatan skema rangkaian dan layout breadboard tahap awal. Untuk keperluan simulasi dan analisa rangkaian yang lebih kompleks, disarankan untuk menggunakan software simulasi elektronika yang lebih advance.

### **2.3.3 Arduino IDE**

Arduino IDE (Integrated Development Environment) merupakan perangkat lunak open-source yang dirancang khusus untuk memudahkan pengguna dalam menulis program untuk berbagai board mikrokontroler Arduino, termasuk Arduino UNO. Arduino IDE menawarkan berbagai kemampuan dan fitur yang bermanfaat bagi pengguna, diantaranya:

- Antarmuka yang Sederhana dan Intuitif : Arduino IDE memiliki antarmuka yang dirancang secara intuitif, sehingga mudah dipelajari dan digunakan bahkan bagi pemula yang tidak memiliki latar belakang pemrograman yang kuat.
- Bahasa Pemrograman Arduino : Arduino IDE menggunakan bahasa pemrograman Arduino, yaitu bahasa berbasis C/C++ yang telah disederhanakan. Sintaks program Arduino lebih mudah dipahami dibandingkan dengan bahasa C/C++ standar, sehingga pengguna dapat dengan cepat belajar dan menulis program untuk mengendalikan board Arduino.
- Library dan Fungsi Built-in: Arduino IDE menyediakan library dan fungsi built-in yang luas untuk berbagai komponen elektronika dan sensor yang umum digunakan. Library dan fungsi ini membantu pengguna berinteraksi dengan komponen tersebut tanpa perlu menulis kode program yang rumit.
- Manajemen Board dan Library: Arduino IDE menyediakan tools untuk manajemen board dan library. Pengguna dapat dengan mudah memilih board yang digunakan (dalam hal ini Arduino UNO) dan menginstal library tambahan yang dibutuhkan untuk project tertentu.
- Kompilasi dan Upload Program: Arduino IDE menyediakan fitur untuk mengkompilasi program yang ditulis ke dalam bahasa mesin yang dapat dimengerti oleh mikrokontroler dan mengunggah program tersebut ke board Arduino UNO melalui koneksi USB.

- Monitor Serial: Arduino IDE menyediakan fitur monitor serial yang memungkinkan pengguna untuk melihat data yang dikirim atau diterima oleh program yang sedang berjalan pada board Arduino UNO. Fitur ini berguna untuk debugging program dan memverifikasi fungsinya.

#### Kelebihan Menggunakan Arduino IDE

- Beberapa kelebihan menggunakan Arduino IDE untuk pemrograman mikrokontroler Arduino UNO dalam skripsi ini, antara lain:
- Kemudahan Penggunaan: Seperti yang telah disebutkan sebelumnya, Arduino IDE memiliki antarmuka yang intuitif dan bahasa pemrograman yang mudah dipelajari. Hal ini memungkinkan peneliti untuk fokus pada pengembangan logika program dan fungsionalitas robot line follower tanpa harus terpaku pada detail bahasa pemrograman tingkat rendah.
- Open-Source dan Gratis: Arduino IDE merupakan software open-source dan gratis untuk digunakan. Hal ini menjadikan Arduino sebagai platform yang hemat biaya dan mudah diakses, terutama untuk keperluan penelitian dan pengembangan.
- Komunitas Besar dan Dukungan Online: Arduino memiliki komunitas pengguna yang besar dan aktif. Terdapat banyak dokumentasi, tutorial, dan forum online yang dapat membantu pengguna dalam belajar dan memecahkan masalah yang dihadapi saat menggunakan Arduino IDE.
- Kesesuaian dengan Board Arduino UNO: Arduino IDE dirancang khusus untuk board mikrokontroler Arduino, termasuk Arduino UNO. Hal ini memastikan kompatibilitas antara software dan hardware yang digunakan, sehingga proses pengembangan program menjadi lebih mudah dan efisien.

Dengan berbagai kemampuan, fitur, dan kelebihan yang ditawarkan, Arduino IDE merupakan platform yang memberikan berbagai kemudahan dalam penggunaan, dukungan komunitas yang besar, dan kesesuaian dengan board Arduino UNO menjadikan Arduino IDE sebagai pilihan yang ideal untuk para peneliti yang ingin berkarya menggunakan platform mikrokontroler Arduino.



#### 2.3.4 MIT APP Inventor

MIT App Inventor merupakan platform berbasis web yang memungkinkan pengguna untuk membuat aplikasi mobile untuk perangkat Android secara visual, tanpa memerlukan penulisan kode program secara manual. MIT App Inventor dikhususkan untuk pengguna yang tidak memiliki latar belakang pemrograman yang kuat. Pengembangan aplikasi di App Inventor dilakukan secara visual, menggunakan antarmuka drag-and-drop. Komponen antarmuka pengguna (user interface) seperti tombol, label, dan image, serta komponen logika program seperti prosedur dan event handler, tersedia sebagai block visual yang dapat disusun dan dihubungkan sesuai dengan kebutuhan fungsional aplikasi.

Beberapa keuntungan menggunakan MIT App Inventor untuk pengembangan aplikasi mobile robot line follower dalam skripsi ini, antara lain:

- Kemudahan Penggunaan: Konsep pengembangan visual dengan block coding membuat App Inventor mudah dipelajari dan digunakan, bahkan bagi pengguna yang awam dengan pemrograman. Peneliti dapat fokus pada logika dan fungsionalitas aplikasi tanpa perlu terpaku pada sintaks kode program yang rumit.
- Antarmuka yang Intuitif: App Inventor memiliki antarmuka yang intuitif dan user-friendly. Pengguna dapat dengan mudah memahami fungsi dari setiap block visual dan bagaimana menghubungkannya untuk membangun logika program.
- Kompatibilitas dengan Android: Aplikasi yang dikembangkan di App Inventor secara native kompatibel dengan perangkat Android. Hal ini penting karena robot line follower yang akan dikendalikan menggunakan smartphone atau tablet berbasis Android.
- Integrasi dengan Bluetooth: App Inventor menyediakan tools untuk pengembangan aplikasi yang dapat berkomunikasi dengan perangkat lain melalui Bluetooth. Fitur ini penting untuk memungkinkan komunikasi antara aplikasi di smartphone dengan modul Bluetooth yang terpasang pada robot line follower.
- Open-Source dan Gratis: MIT App Inventor merupakan platform open-source dan gratis untuk digunakan. Hal ini menjadikan App Inventor sebagai pilihan yang hemat biaya dan mudah diakses untuk keperluan pengembangan aplikasi mobile robot line follower.

Meskipun menawarkan berbagai kemudahan, MIT App Inventor juga memiliki beberapa keterbatasan, antara lain:

- Kurang Fleksibel untuk Aplikasi Kompleks: App Inventor kurang fleksibel dibandingkan dengan pengembangan aplikasi menggunakan bahasa pemrograman native. Untuk aplikasi yang kompleks dengan fungsionalitas yang rumit, pengembangan menggunakan bahasa pemrograman seperti Java mungkin akan lebih efisien.
- Terbatasnya Library dan Fungsi Built-in: App Inventor menyediakan library dan fungsi built-in, namun jumlahnya mungkin lebih terbatas dibandingkan dengan platform pengembangan aplikasi lainnya.
- Keterbatasan Kustomisasi Antarmuka: Meskipun menawarkan berbagai komponen antarmuka pengguna, App Inventor mungkin memiliki keterbatasan dalam hal kustomisasi tampilan aplikasi secara detail.

MIT App Inventor merupakan platform yang tepat untuk pengembangan berbagai aplikasi mobile karena kemudahan penggunaan dan kompatibilitas dengan perangkat Android. Namun, keterbatasan yang dimiliki seperti kurangnya fleksibilitas untuk aplikasi kompleks perlu dipertimbangkan.

## **BAB III**

### **PERANCANGAN DAN PRODUKSI**

#### **3.1 Perencanaan**

Perencanaan dimulai dengan menentukan apa komponen komponen yang diperlukan untuk membuat robot line follower yang memadukan hardware dan software. Robot ini akan menggunakan komponen mikrokontroller Arduino UNO, modul Bluetooth HC-05, sensor infrared 5 channel, 2 DC motor dan gearbox, dan motor driver L291N yang akan dihubungkan dalam sebuah PCB. Robot ini akan disertai dengan 2 fitur utama yaitu sebagai robot line follower yang akan mengikuti garis dan juga bisa dikontrol pergerakannya melalui aplikasi yang telah kami buat.

#### **3.2 Alat dan Bahan**

##### **3.2.1 Daftar Alat**

<b>Nama Alat/Bahan</b>	<b>Jumlah</b>
Laptop	1 unit
Multimeter Digital	1 unit
Solder	1 unit
Tool - Kit	1 unit
Charger Baterai	1 unit
Penggaris	1 unit
Printer	1 unit
Cutter	1 unit
Gunting	1 unit
Lasser Cutting	1 unit
Timah Solder	1 unit
Kertas	5 lembar
Bubuk Feri Clorida	500g
Autan	2 pcs
Double Tape Foam	1 pcs
Acrylic	2 pcs
Flux Solder	1 pcs

### 3.2.2 Daftar komponen

Nama Komponen	Jumlah
Sensor IR 5 Channel	1 pcs
Jumper Male to Female	5 pcs
Jumper Female to Female	6 pcs
Mur 3mm	10 pcs
Spacer	8 pcs
Baut 3mm	8 pcs
Baut 30mm	4pcs
Arduino R3	1 pcs
Bluetooth HC-05	1 pcs
Motor Driver L298N	1 pcs
Roda	2 pcs
Gearbox + Motor DC	2 pcs
Baterai 18650	2 pcs

### 3.2.3 Daftar software

Nama Perangkat	Jenis
Solidworks 2022	Software
RDWorksV8	Software
MIT App Inventor	Software Browser
Arduino IDE	Software
Fritzing	Software Browser

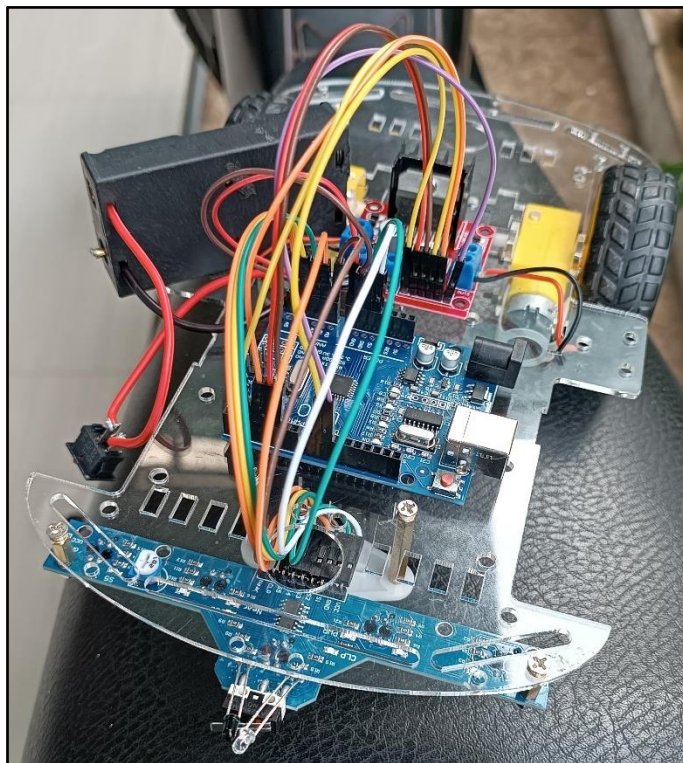
## 3.3 Perancangan

Perancangan dimulai dengan membuat sebuah robot line follower yang masih menggunakan banyak kabel jumper dan tanpa adanya PCB, hal ini penting karena dapat memberikan kami gambaran struktur secara sistematis sehingga kami dapat menentukan rangkaian yang akan digunakan, setelah menentukan struktur rangkaian selanjutnya adalah membuat program untuk bisa mengkonfigurasi kedua fitur yang direncanakan ke dalam mikrokontroler arduino UNO. Terakhir adalah pembuatan aplikasi yang dapat mengkoneksikan arduino UNO dengan menggunakan modul bluetooth, sehingga antara mekanik, elektrik, program, interface, akan menjadi sebuah kesatuan dalam pembuatan robot line follower.

### 3.4 Tahap Produksi

#### 3.4.1 Tahap 1 : Membuat robot tanpa adanya PCB

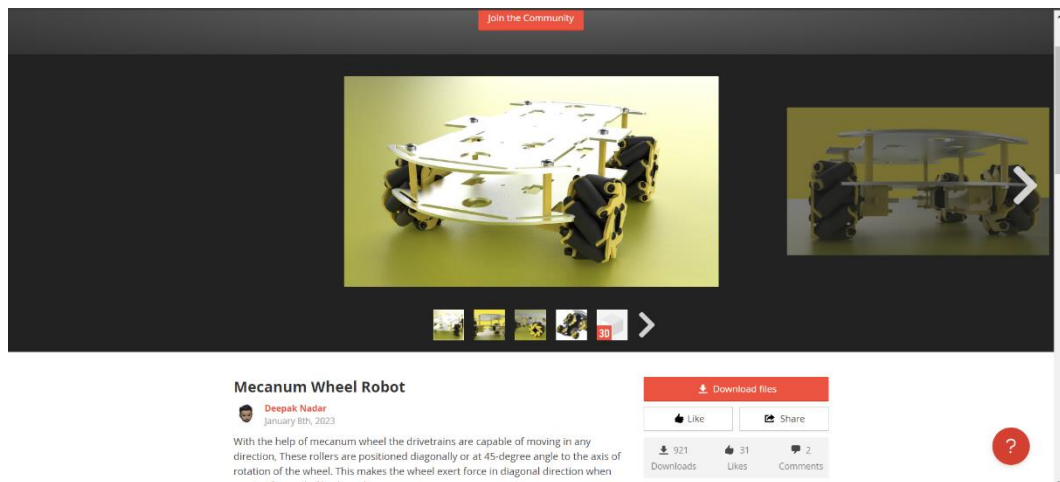
Langkah awal untuk membuat robot line follower yaitu dengan merangkai komponen elektrik ( Arduino Uno, Motor Driver L298N, Bluetooth HC-05, Battery, Sensor Infrared dan Dinamo N20 ) sesuai dengan wiring yang telah ditentukan, dengan memanfaatkan kabel jumper untuk menghubungkan dari satu komponen ke komponen lainnya. Tujuannya yaitu agar menghindari kesalahan pemasangan pin output/input ketika pcb sudah dibuat, dengan ini metode penggunaan kabel jumper pada wiring robot line follower yang hanya sementara dan masih bisa dirubah pemasangan pin input/output nya merupakan Langkah yang tepat.



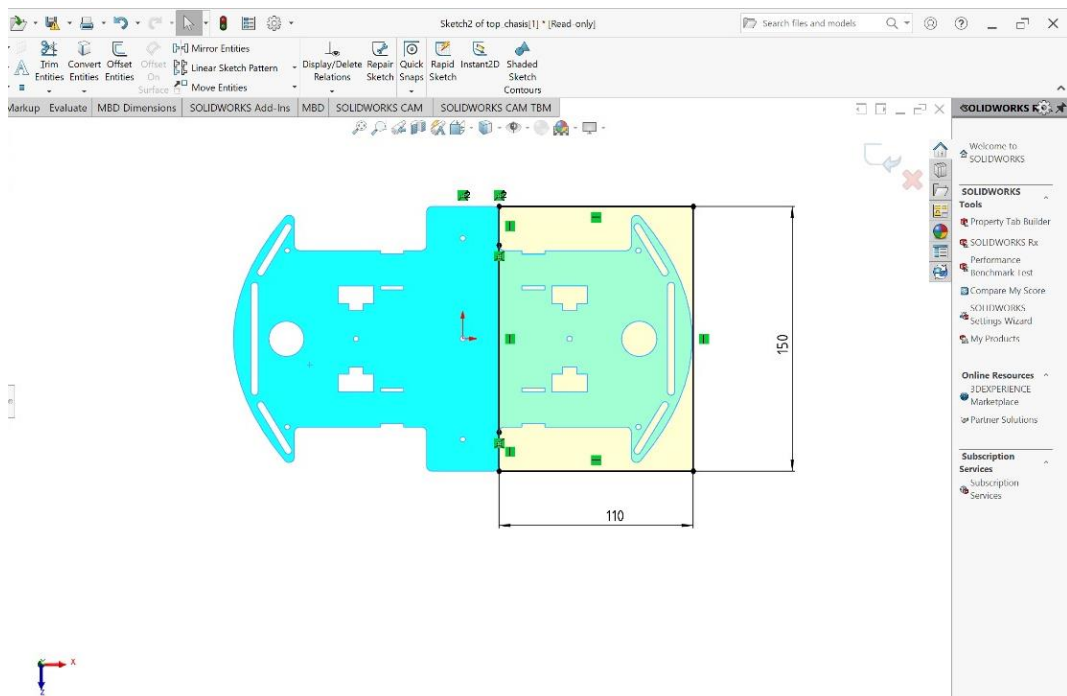
Kemudian jika dirasa susunan wiring elektrik ini sudah tepat dan sesuai dengan kebutuhan system dari robot line follower ini serta komponen sudah aktif dengan baik ketika diberikan tegangan input maka tahap ini akan diteruskan ke tahap berikutnya.

### 3.4.2 Tahap 2 : Membuat CASE

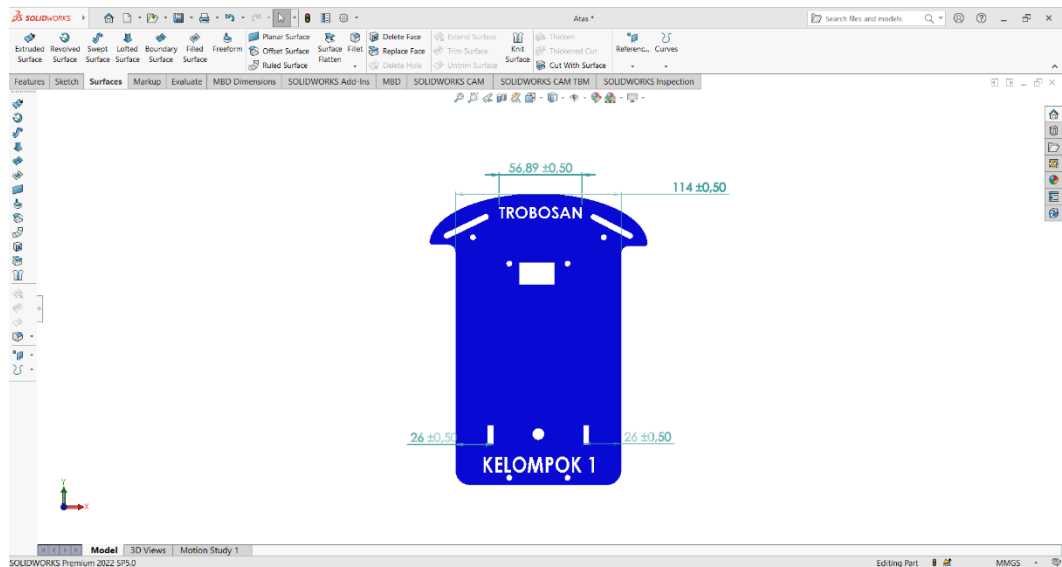
- Tahap awal untuk membuat case adalah dengan mencari file solidwork case rc di internet



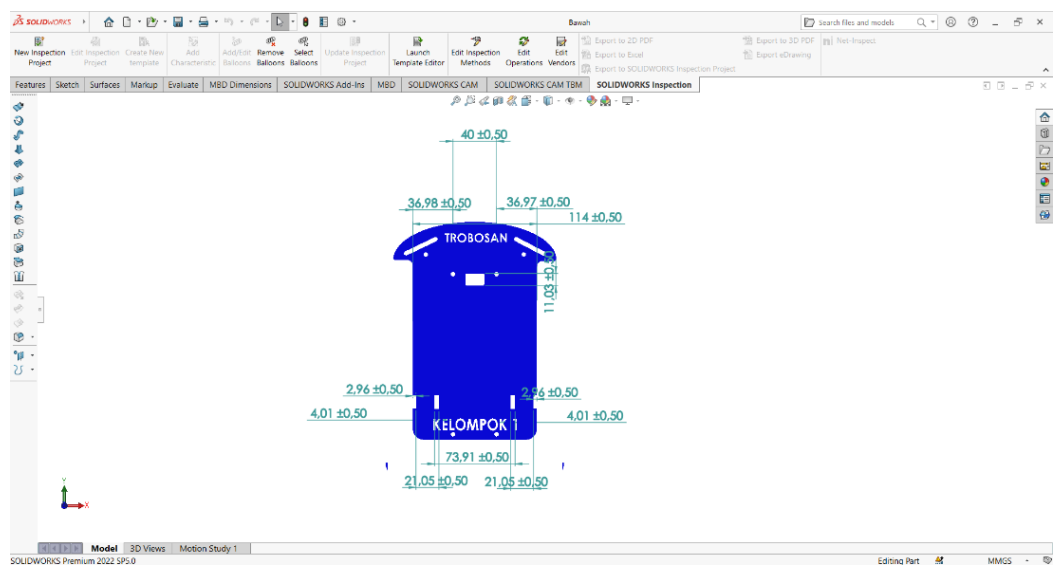
- Setelah didapatkan file dengan model case yang diinginkan, kemudian file tersebut diimport ke Solid Work.



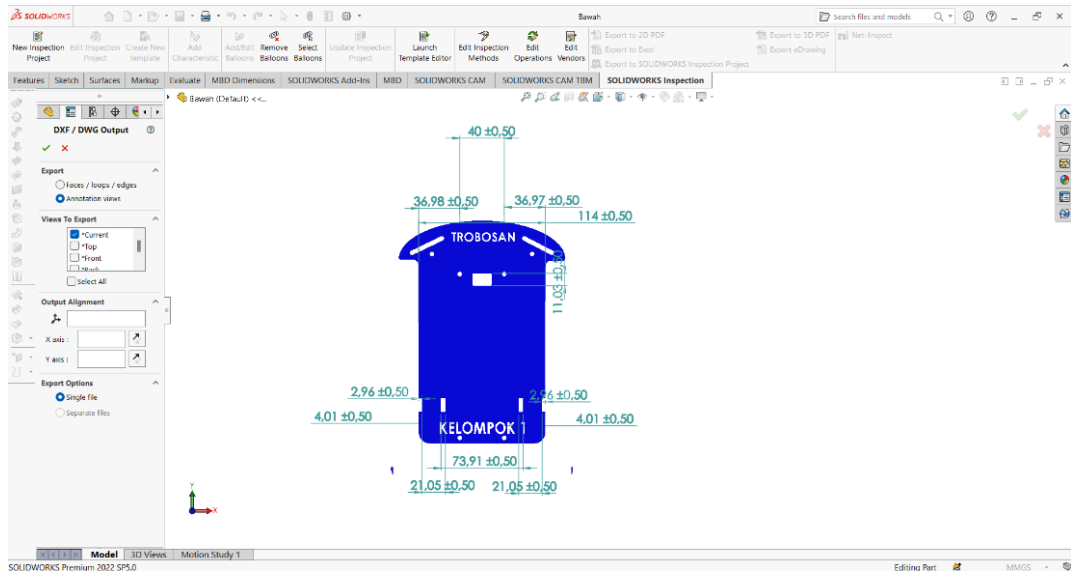
- Model case yang sudah ada kemudian dimodifikasi sesuai dengan yang dibutuhkan. Pertama dengan menghilangkan 1/3 dari model awal.



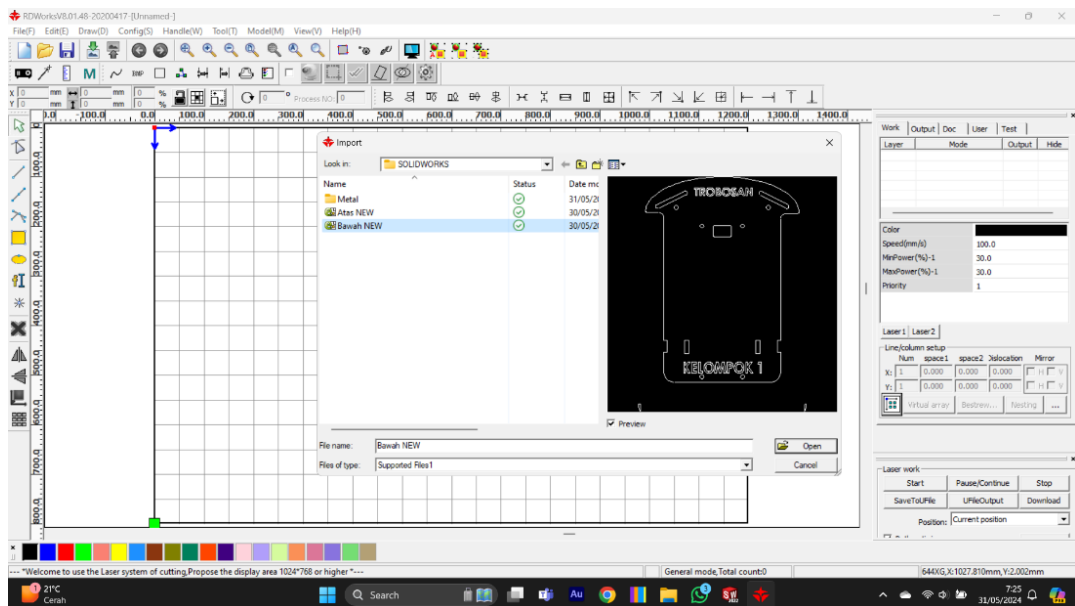
- Lalu menutup lubang-lubang yang tidak dibutuhkan dan menambahkan beberapa modifikasi, seperti lubang untuk spacer, lubang untuk kabel jumper menuju sensor dan tombol on/off, lubang untuk gearbox, penambahan tulisan “TROBOSAN” dan “KELOMPOK 1”, serta menumpulkan sudut-sudut pada model case tersebut. Kemudian file design tersebut di beri nama “ATAS” yang tanpa lubang untuk gearbox, dan “BAWAH” untuk yang dengan lubang untuk gearbox yang mana file tersebut masih berformat ‘.sldprt’.



- Design awal case tersebut kemudian harus dikonversi yang awalnya berbentuk ‘.sldprt’ menjadi ‘.dxf’ agar bisa diimport ke MetalCut.

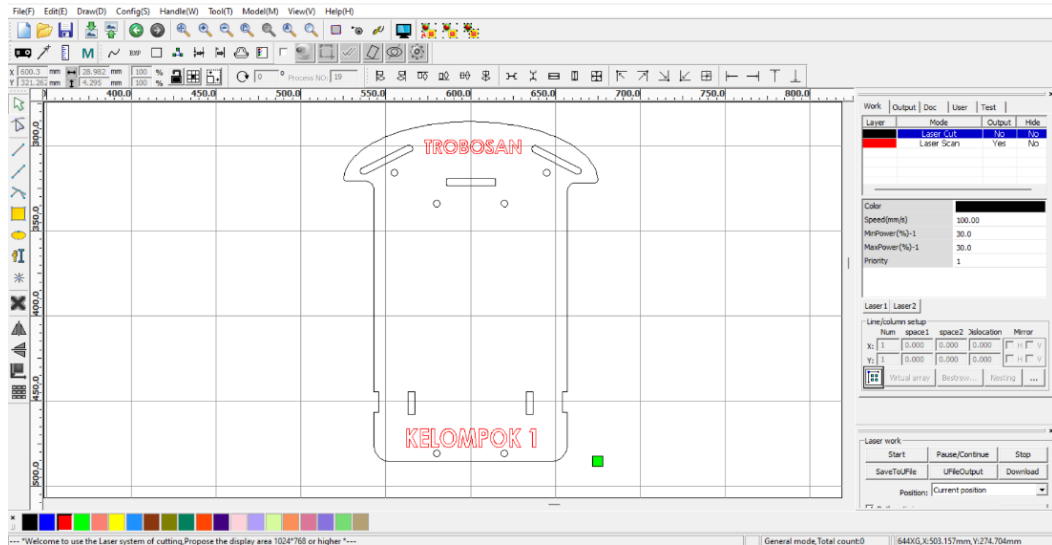


- Import file di MetalCut untuk design yang ingin dilakukan pemotongan

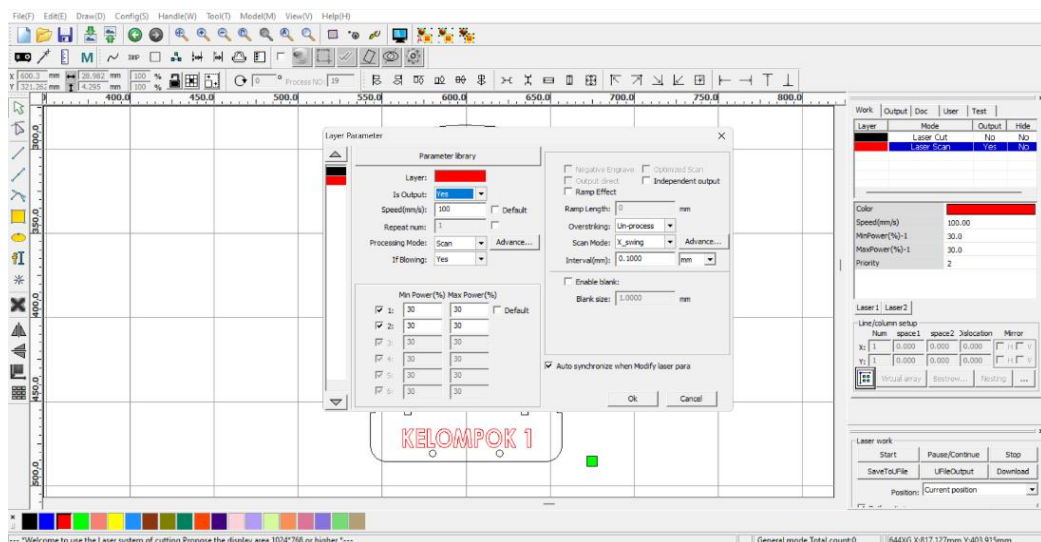




- Mengkonfigurasi beberapa bagian di design Solid Work sebelumnya dikarenakan tidak sesuai dengan yang diinginkan. Maka dilakukan beberapa Kembali modifikasi design model di Metal cut. Setelah design dianggap sesuai, design dibagi menjadi dua layer.



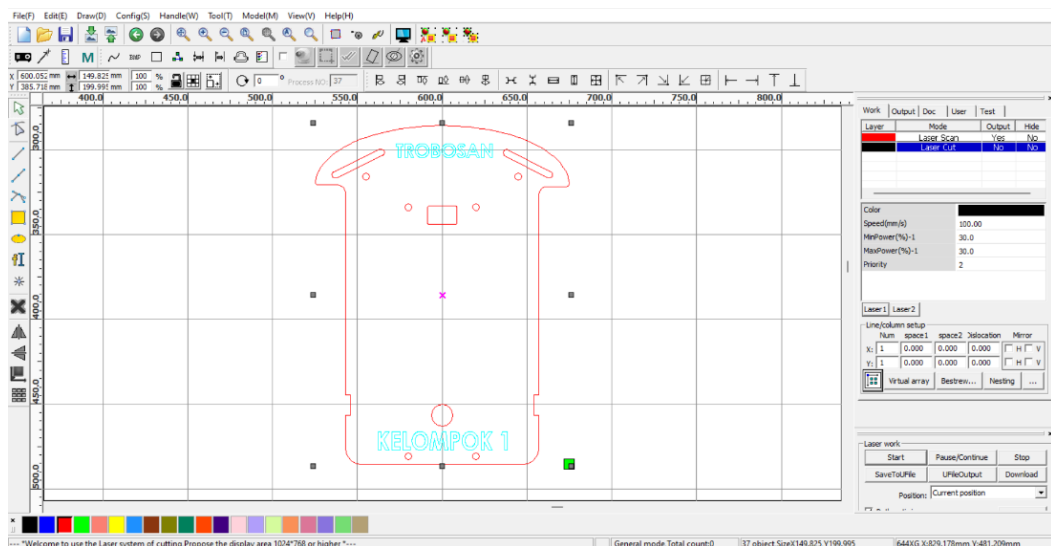
- Layer berwarna hitam dengan mode 'Laser Cut' yang bertujuan untuk memotong dan untuk bagian tulisan "TROBOSAN" dan "KELOMPOK 1" di ubah menjadi layer berwarna merah dan mode diubah ke 'Laser Scan' yang bertujuan untuk membuat engraving pada tulisan.



- Sebelum melakukan cutting dengan mesin, terlebih dahulu harus melakukan simulasi di software dengan tujuan meminimalisir kesalahan saat pemotongan di mesin.



- Setelah simulasi cutting dianggap tidak terjadi kesalahan, maka proses cutting bisa dilakukan pada mesin Laser Cutting dengan menekan tombol start pada software.

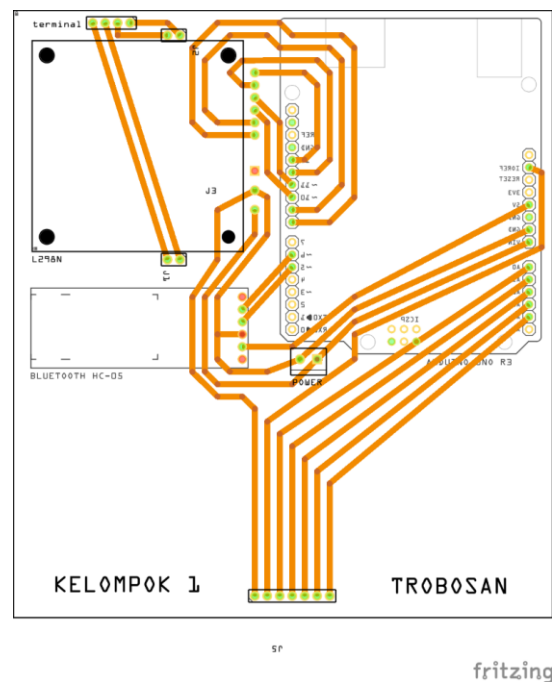


### 3.4.3 Tahap 2 : Membuat PCB Layout Elektrikal Line Follower

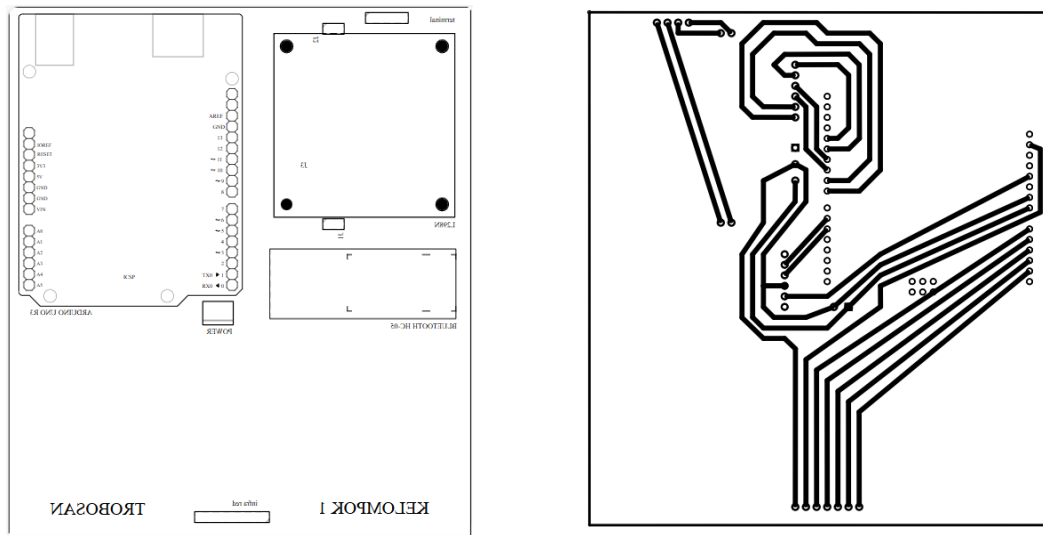
Tujuan dibuat nya piranted circuit board (PCB) berfungsi sebagai tempat menyusun komponen - komponen elektronik sehingga terpasang lebih rapih dan terorganisir, dan dapat menjadi pengganti kabel untuk menyambung berbagai komponen yaitu menghubungkan kaki komponen satu dengan komponen yang lainnya baik kaki komponen aktif maupun pasif. Berikut adalah tahapan untuk membuat pcb :

- **Membuat Disain PCB**

Langkah awal untuk membuat PCB adalah dengan membuat rancangan elektrikal yaitu berupa disain yang nantinya disain itu akan ditransfer ke permukaan papan pcb. Dibawah ini merupakan disain yang telah dibuat dari susunan komponen yang akan digunakan untuk system elektrikal robot line follower.



Software yang digunakan untuk mendesain sebuah pcb layout yaitu **Fritzing**. Dengan mengimport bahan disain komponen yang akan digunakan dan pembuatan jalur secara manual dengan sesederhana dan serapih mungkin dari referensi wiring elektrik yang sudah ditentukan. Maka akan menghasilkan 2 disain yaitu disain permukaan atas pcb board dan disain permukaan bawah pcb board. Dibawah ini merupakan disain terpisah dari top layer dan bottom layer :



Disebelah kiri merupakan disain top layer dan disebelah kanan merupakan disain bottom layer.

- **Transfer Disain PCB Pada Kedua Layer PCB Board (Etching)**

Setelah melalui tahap pembuatan disain maka disain itu akan diaplikasikan kepada pcb board yang akan digunakan untuk system robot line follower. Pertama siapkan PCB board single layer dengan ukuran yang sudah ditentukan.



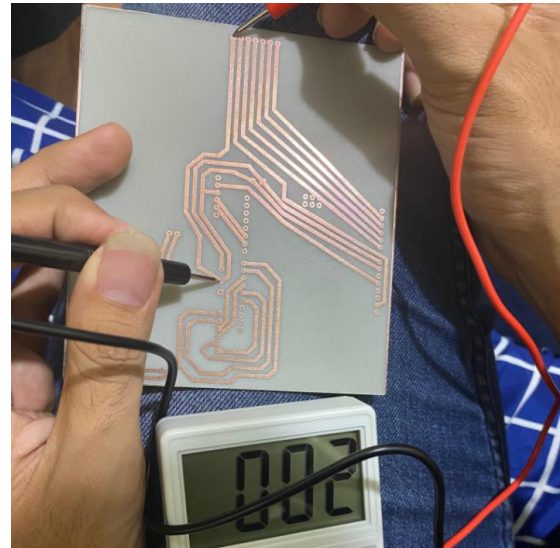
Kemudian print out disain yang telah dibuat memakai kertas hvs setelah itu fotocopy hasil print out karena nantinya disain yang digunakan untuk etching yaitu hasil fotocopy dari print out karena tinta hasil fotocopy lebih baik untuk digunakan untuk proses etching.

Selanjutnya tempelkan disain yang sudah difotocopy pada bottom layer pcb yang memiliki permukaan tembaga dan lumuri kertas disain dengan lotion anti nyamuk hingga merata, kemudian tunggu beberapa saat agar disain menempel pada bottom layer pcb dengan merata, lakukan tahap ini dengan sama saat mentransfer disain pada bagian top layer.



- **Peleburan Tembaga Pada PCB Menggunakan Bubuk Ferric Chloride**

Setelah desain menempel dengan baik pada kedua layer pcb board maka tahap selanjutnya pcb board akan direndam kedalam larutan ferric chloride guna meleburkan bagian tembaga pada bottom layer pcb yang tidak digunakan sehingga pcb board dapat menyisakan bagian wiring tembaga dari disain yang telah menempel pada layer tersebut. Jika sudah berhasil meleburkan tembaga yang tidak terpakai selanjutnya melakukan checking connectivitas pada setiap wire pada board pcb guna memeriksa apakah ada wire yang menempel atau tidak terhubung.



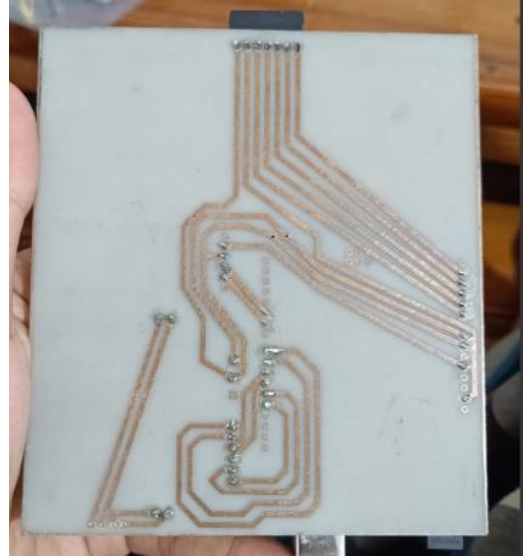
- **Drilling Board PCB**



Tahap selanjutnya ketika pcb board telah selesai proses etching selanjutnya board pcb akan di bor pada bagian pin – pin yang sudah tersedia sebagai tempat kaki komponen menggunakan mesin bor dengan diameter mata bor 0.1mm

- **Pemasangan Komponen Pada PCB Board**

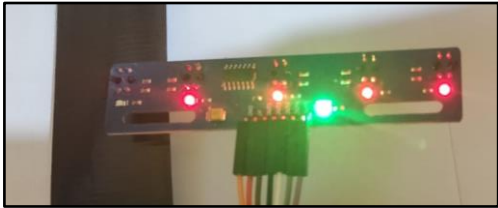
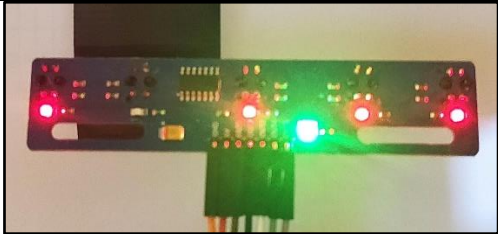
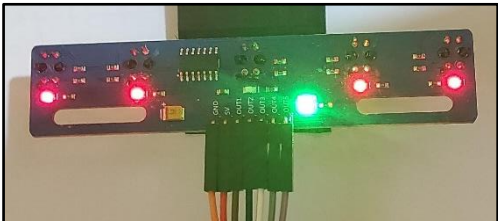
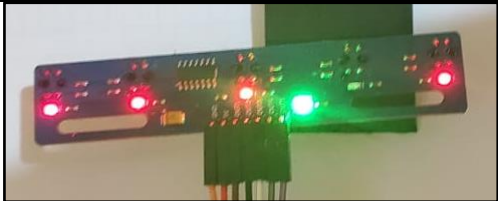
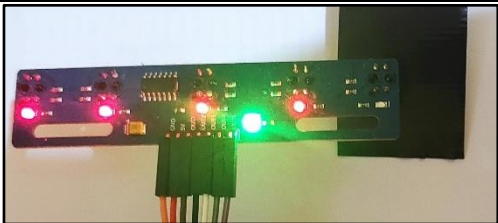
Pada tahapan ini komponen elektrik akan dipasang sesuai lubang yang telah disediakan dan terhubung pada wire pcb board dengan cara melakukan proses soldering agar kaki – kaki komponen elektrik pada wire pcb board dapat terhubung dengan baik.





### 3.4.4 Tahap 3 : Kalibrasi Sensor

Tahap ketiga adalah proses untuk mengkalibrasikan sebuah sensor sekaligus dapat menggambarkan situasi pembacaan sensor sehingga dapat mempermudah pada saat membuat program line followernya, berikut adalah prosesnya.

Kondisi Sensor	Serial Monitor	Penjelasan
	<div>Output Serial Monitor ✕</div> <div>Message (Enter to send mes:</div> <pre> 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 </pre>	Kondisi ketika sensor paling kiri mendeteksi garis hitam
	<div>Output Serial Monitor ✕</div> <div>Message (Enter to send messa</div> <pre> 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 </pre>	Kondisi ketika sensor kiri mendeteksi garis hitam
	<div>Output Serial Monitor ✕</div> <div>Message (Enter to send mes</div> <pre> 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 </pre>	Kondisi ketika sensor tengah mendeteksi garis hitam
	<div>Output Serial Monitor ✕</div> <div>Message (Enter to send mess</div> <pre> 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 </pre>	Kondisi ketika sensor kanan mendeteksi garis hitam
	<div>Output Serial Monitor ✕</div> <div>Message (Enter to send mes</div> <pre> 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 </pre>	Kondisi ketika sensor paling kanan mendeteksi garis hitam



### 3.4.5 Tahap 4 : Membuat Program pada robot line follower

Tahap selanjutnya merupakan tahap pembuatan program yang akan digunakan langsung pada robot line follower yang bisa digerakan secara manual maupun otomatis, berikut ini adalah penjelasan program yang digunakan.

- Import Library untuk mengkoneksikan pin RX, TX pada Bluetooth

```
1  #include <SoftwareSerial.h>
2  SoftwareSerial BT_Serial(5, 6); // RX, TX
```

- Inisialisasi pin-pin yang digunakan pada rangkaian

```
4  //Definisi pin Motor Driver
5  #define e1 8 //Pin EA
6  #define m1 9 //Motor MA1
7  #define m2 10 //Motor MA2
8  #define m3 11 //Motor MB1
9  #define m4 12 //Motor MB2
10 #define e2 13 //Pin EB
11
12 //Definisi pin sensor
13 #define ir1 A4
14 #define ir2 A3
15 #define ir3 A2
16 #define ir4 A1
17 #define ir5 A0
18
19 //inisialisasi
20 int bt_data; // variable untuk menerima data
21 int Speed = 255;
22 int mode = 0;
```

- Deklarasi pin yang sudah di inisialisasi menjadi sebuah fungsi

```
24 void setup() {
25 //Fungsi sensor
26   pinMode(ir1, INPUT);
27   pinMode(ir2, INPUT);
28   pinMode(ir3, INPUT);
29   pinMode(ir4, INPUT);
30   pinMode(ir5, INPUT);
31
32 //Fungsi motordriver
33   pinMode(m1, OUTPUT);
34   pinMode(m2, OUTPUT);
35   pinMode(m3, OUTPUT);
36   pinMode(m4, OUTPUT);
37   pinMode(e1, OUTPUT);
38   pinMode(e2, OUTPUT);
39
40   Serial.begin(9600); /// memulai komunikasi serial di 9600bps
41   BT_Serial.begin(9600);
42   delay(500);
43 }
```

- Memberikan perintah pada fungsi perulangan, yang akan terus di eksekusi secara berulang ketika robot dijalankan

```

45 void loop() {
46   if (BT_Serial.available() > 0) { //Jika data terkirim, maka akan memanggil fungsi baca
47     bt_data = BT_Serial.read();
48     if (bt_data > 20) { Speed = bt_data; }
49   }
50
51   if (bt_data == 8) { mode = 1; Speed = 255; } //Auto Line Follower
52   else if (bt_data == 9) { mode = 0; Stop(); } //Manual Kontrol
53
54   analogWrite(e1, Speed); // Menulis pin A sebagai speed motor 1
55   analogWrite(e2, Speed); // Menulis pin B sebagai speed untuk motor 2

```

- Fungsi agar robot bisa digerakan secara manual

```

57   if (mode == 0) {
58     /*
59     =          Kontrol Manual          =
60     =====*/
61     if (bt_data == 1) { maju(); }
62     else if (bt_data == 2) { mundur(); }
63     else if (bt_data == 3) { kiri(); }
64     else if (bt_data == 4) { kanan(); }
65     else if (bt_data == 5) { berhenti(); }
66   }

```

- Fungsi agar robot bisa bergerak otomatis mendeteksi garis

```

69   } else {
70     /*
71     |          Kontrol Line Follower          |
72     =====*/
73     int s1 = digitalRead(ir1); // Sensor paling kiri
74     int s2 = digitalRead(ir2); // Sensor kiri
75     int s3 = digitalRead(ir3); // Sensor tengah
76     int s4 = digitalRead(ir4); // Sensor kanan
77     int s5 = digitalRead(ir5); // Sensor paling kanan

```

- Pembacaan sensor pada fungsi otomatis

```

77     //hanya sensor3 mendeteksi garis hitam akan maju
78     if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 1 && s5 == 1) {
79       analogWrite(e1, 255);
80       analogWrite(e2, 255);
81       digitalWrite(m1, HIGH);
82       digitalWrite(m2, LOW);
83       digitalWrite(m3, HIGH);
84       digitalWrite(m4, LOW);
85     }
86     //hanya sensor2 mendeteksi garis hitam akan ke kiri
87     else if (s1 == 1 && s2 == 0 && s3 == 1 && s4 == 1 && s5 == 1) {
88       analogWrite(e1, 255);
89       analogWrite(e2, 255);
90       digitalWrite(m1, HIGH);
91       digitalWrite(m2, LOW);
92       digitalWrite(m3, LOW);
93       digitalWrite(m4, LOW);
94     }
95     //hanya sensor1 mendeteksi garis hitam akan ke kiri
96     else if (s1 == 0 && s2 == 1 && s3 == 1 && s4 == 1 && s5 == 1) {
97       analogWrite(e1, 255);
98       analogWrite(e2, 255);
99       digitalWrite(m1, HIGH);
100      digitalWrite(m2, LOW);
101      digitalWrite(m3, LOW);
102      digitalWrite(m4, HIGH);
103     }

```

```

104 //hanya sensor4 mendeteksi garis hitam akan ke kanan
105 else if (s1 == 1 && s2 == 1 && s3 == 1 && s4 == 0 && s5 == 1) {
106     analogWrite(e1, 255);
107     analogWrite(e2, 255);
108     digitalWrite(m1, LOW);
109     digitalWrite(m2, LOW);
110     digitalWrite(m3, HIGH);
111     digitalWrite(m4, LOW);
112 }
113 //hanya sensor5 mendeteksi garis hitam akan ke kanan
114 else if (s1 == 1 && s2 == 1 && s3 == 1 && s4 == 1 && s5 == 0) {
115     analogWrite(e1, 255);
116     analogWrite(e2, 255);
117     digitalWrite(m1, LOW);
118     digitalWrite(m2, HIGH);
119     digitalWrite(m3, HIGH);
120     digitalWrite(m4, LOW);
121 }
122 //sensor3 dan sensor4 mendeteksi garis hitam akan ke kanan
123 else if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 0 && s5 == 1) {
124     analogWrite(e1, 255);
125     analogWrite(e2, 255);
126     digitalWrite(m1, LOW);
127     digitalWrite(m2, LOW);
128     digitalWrite(m3, HIGH);
129     digitalWrite(m4, LOW);
130 }
}

131 //sensor2 dan sensor3 mendeteksi garis hitam akan ke kiri
132 else if (s1 == 1 && s2 == 0 && s3 == 0 && s4 == 1 && s5 == 1) {
133     analogWrite(e1, 255);
134     analogWrite(e2, 255);
135     digitalWrite(m1, HIGH);
136     digitalWrite(m2, LOW);
137     digitalWrite(m3, LOW);
138     digitalWrite(m4, LOW);
139 }
140 //sensor1, sensor2 dan sensor3 mendeteksi garis hitam akan ke kiri
141 else if (s1 == 0 && s2 == 0 && s3 == 0 && s4 == 1 && s5 == 1) {
142     analogWrite(e1, 255);
143     analogWrite(e2, 255);
144     digitalWrite(m1, HIGH);
145     digitalWrite(m2, LOW);
146     digitalWrite(m3, LOW);
147     digitalWrite(m4, LOW);
148 }
149 //sensor3, sensor4 dan sensor5 mendeteksi garis hitam akan ke kanan
150 else if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 0 && s5 == 0) {
151     analogWrite(e1, 255);
152     analogWrite(e2, 255);
153     digitalWrite(m1, LOW);
154     digitalWrite(m2, LOW);
155     digitalWrite(m3, HIGH);
156     digitalWrite(m4, LOW);
157 }
}

158 //semua sensor mendeteksi garis hitam akan ke kiri akan berhenti
159 else if (s1 == 0 && s2 == 0 && s3 == 0 && s4 == 0 && s5 == 0) {
160     digitalWrite(m1, LOW);
161     digitalWrite(m2, LOW);
162     digitalWrite(m3, LOW);
163     digitalWrite(m4, LOW);
164 }
165 }
166 delay(10);
167 }

```

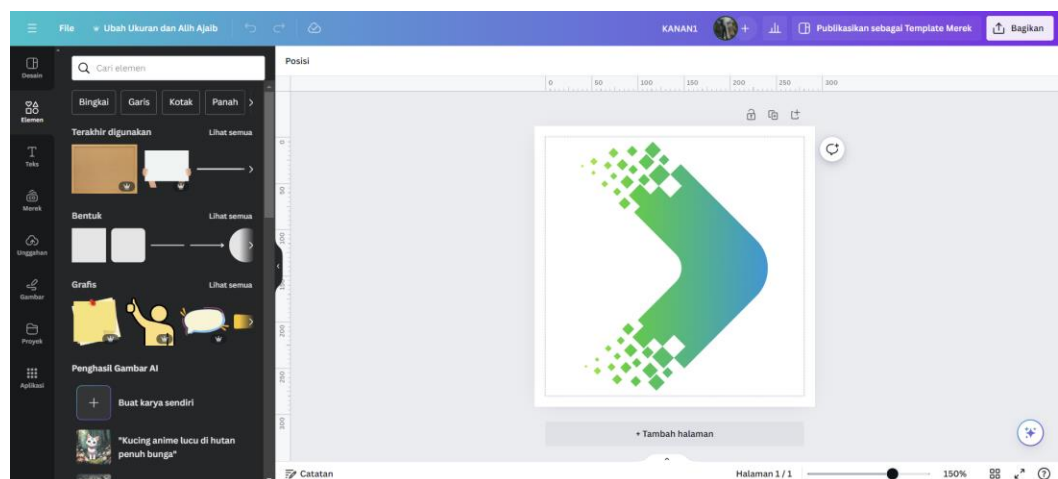
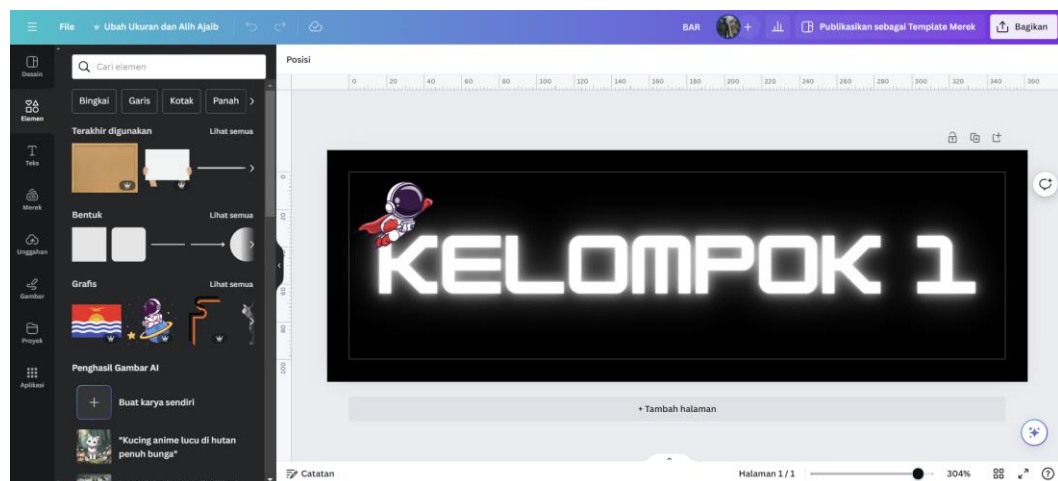
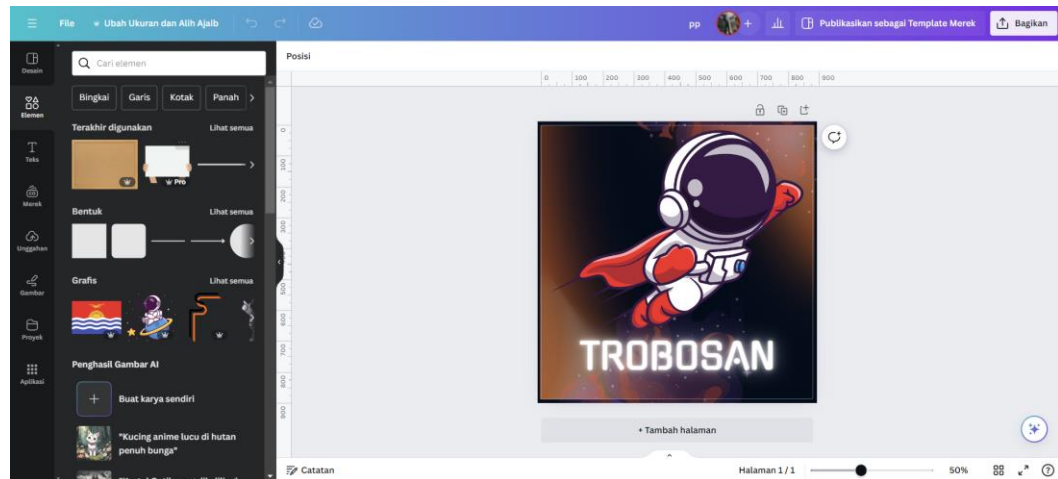
- Fungsi ini adalah untuk mengatur pergerakan DC Motor, dimana in12 merupakan DC motor 1, dan in34 merupakan DC Motor 2

```
169 void maju() {
170     digitalWrite(m1, HIGH);
171     digitalWrite(m2, LOW);
172     digitalWrite(m3, HIGH);
173     digitalWrite(m4, LOW);
174 }
175
176 void mundur() {
177     digitalWrite(m1, LOW);
178     digitalWrite(m2, HIGH);
179     digitalWrite(m3, LOW);
180     digitalWrite(m4, HIGH);
181 }
182
183 void kanan() {
184     digitalWrite(m1, LOW);
185     digitalWrite(m2, HIGH);
186     digitalWrite(m3, HIGH);
187     digitalWrite(m4, LOW);
188 }
189
190 void kiri() {
191     digitalWrite(m1, HIGH);
192     digitalWrite(m2, LOW);
193     digitalWrite(m3, LOW);
194     digitalWrite(m4, HIGH);
195 }
196
197 void berhenti() {
198     digitalWrite(m1, LOW);
199     digitalWrite(m2, LOW);
200     digitalWrite(m3, LOW);
201     digitalWrite(m4, LOW);
202 }
```

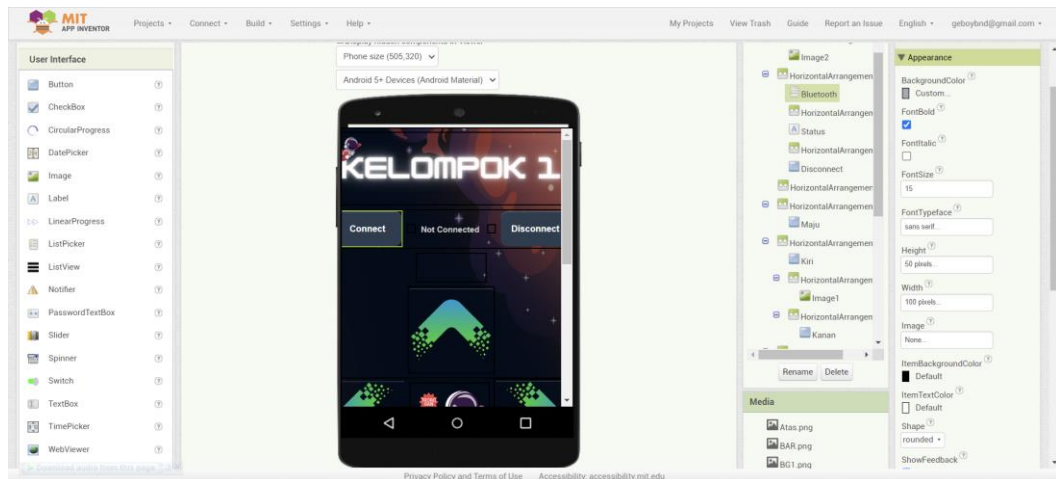
### 3.4.6 Tahap 5 : Membuat Aplikasi

Tahap kelima adalah proses membuat aplikasi interface dimana pada aplikasi ini dibuat untuk mengkonfigurasi kode program otomatis menjadi manual yang bisa dikendalikan melalui aplikasi yang terinstall pada smartphone. Berikut ini adalah penjelasan mengenai isi dari sebuah diagram block pada Aplikasi Interface.

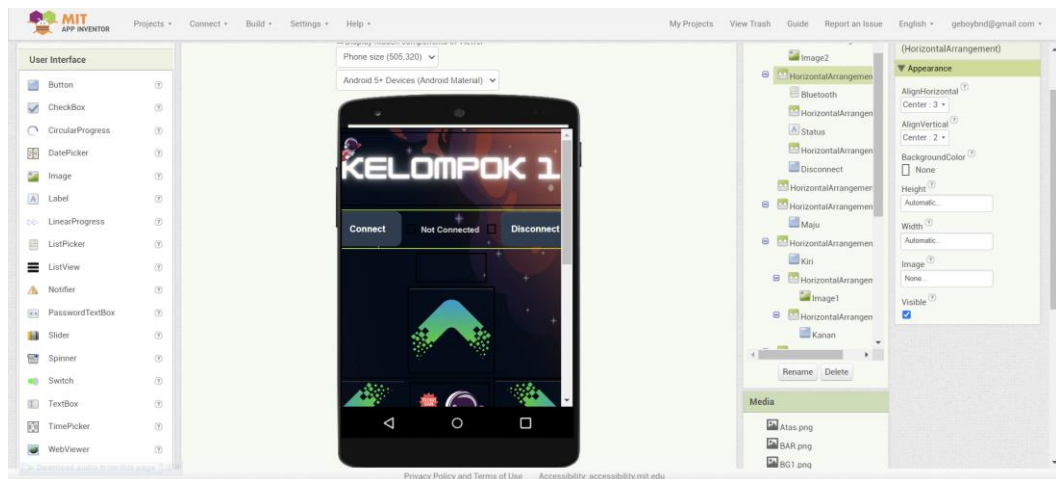
- Merancang design untuk tampilan aplikasi interface



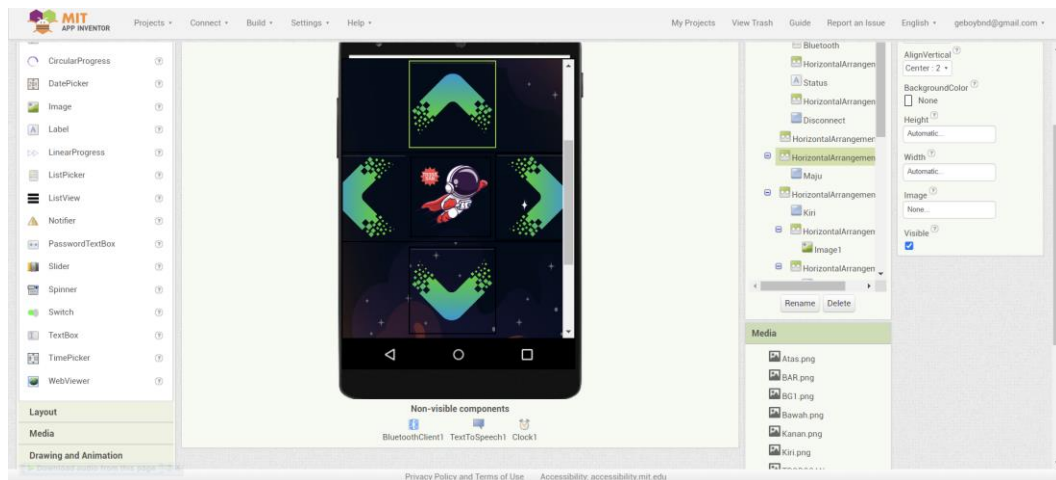
- Memasukkan fitur bluetooth dan button pada 'Home' sebagai penunjang utama aplikasi



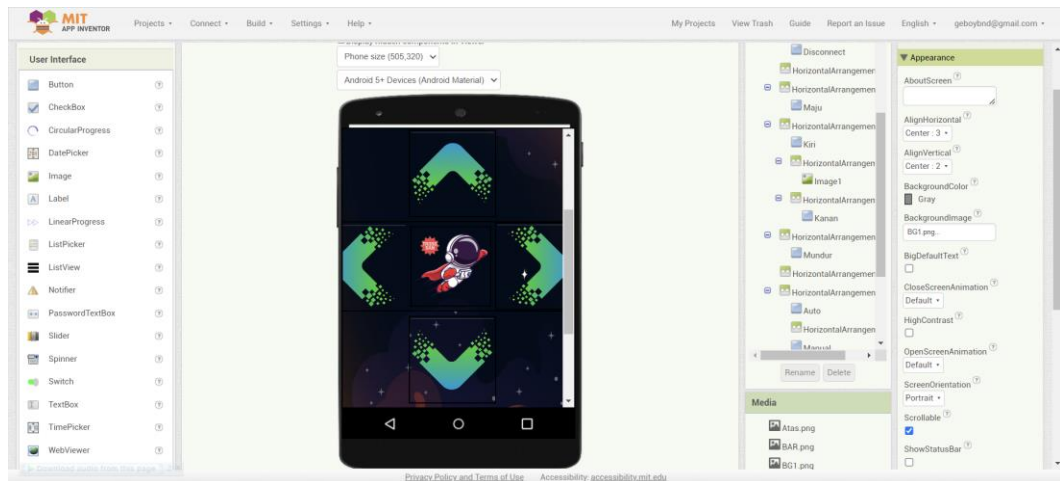
- Pada fitur 'Bluetooth' display menggunakan Horizontal Arrangement



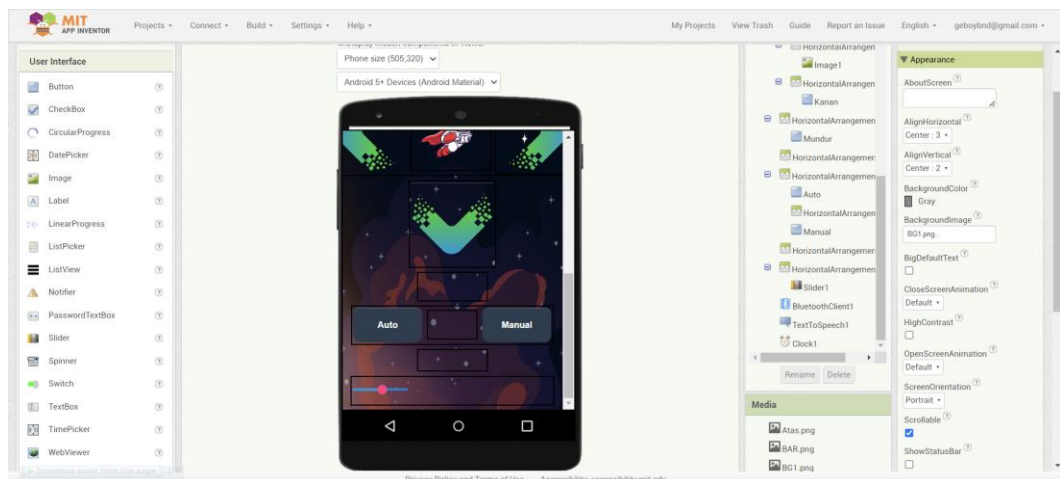
- Pada fitur 'Arrow' display menggunakan Horizontal Arrangement



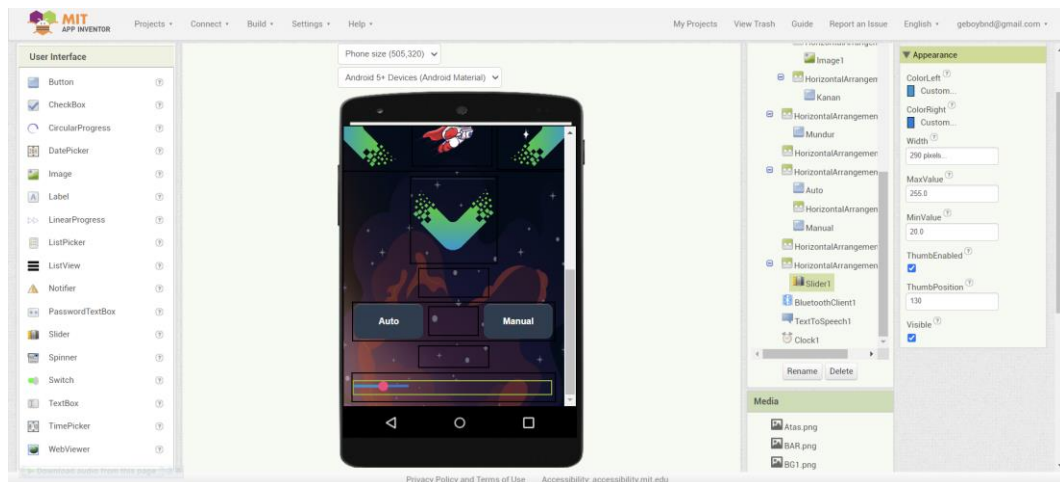
- Pada fitur ‘Arrow’ terdapat 4 button sebagai penunjang



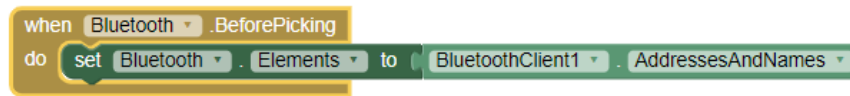
- Pada fitur ‘Connection’ terdapat 2 button kondisi auto dan manual



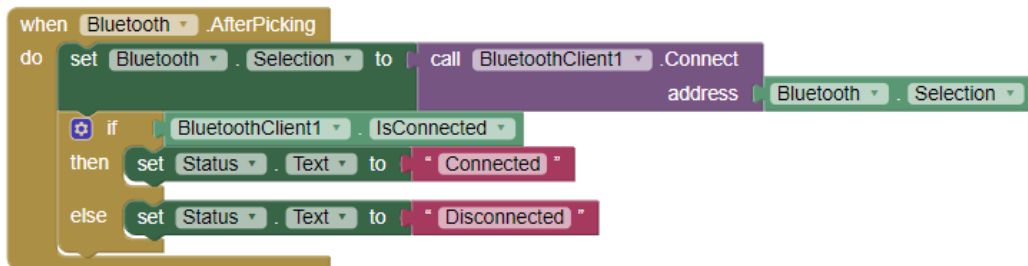
- Pada fitur ‘Speed’ terdapat 1 slider sebagai pengatur kecepatan



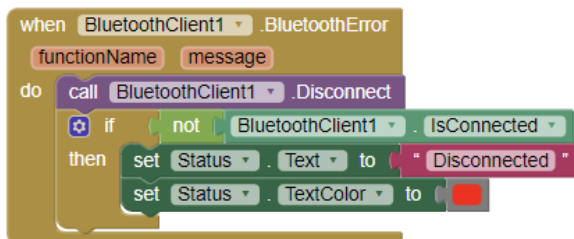
- Pada tahap ini merancang sebuah block untuk konfigurasi 'Bluetooth' menggunakan control untuk melakukan sebuah logic bluetooth untuk mendapatkan sebuah 'Adress And Name' dari sebuah bluetooth



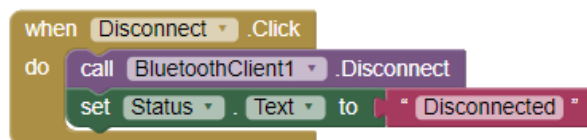
- Dalam tahap ini ketika user menekan button 'Connect' maka logic dari 'Bluetooth' akan memproses dan memanggil 'Bluetooth Client' untuk mencari ketersediaan bluetooth pada area sekitar



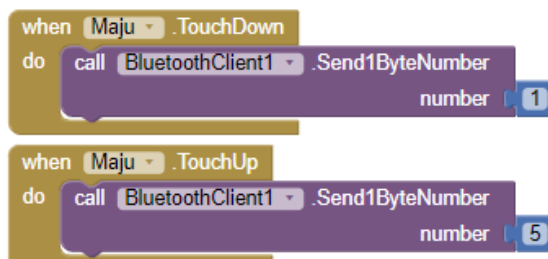
- Untuk mengetahui status bluetooth dapat memanggil sebuah control untuk memberikan text berupa 'Connected' atau 'Disconnected'



- Pada saat user menekan 'Disconnected' status bluetooth akan memutuskan jaringan bluetooth dari mobil ke aplikasi dengan pop up alert "Disconnected"

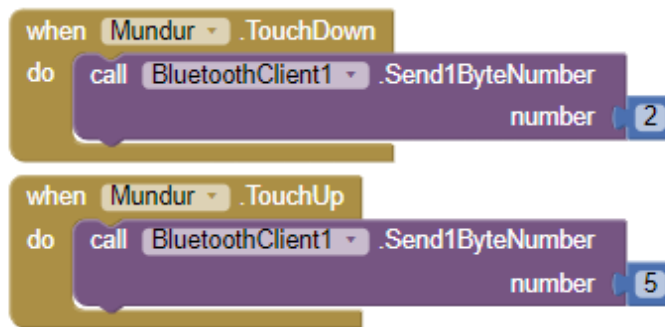


- Deklarasi 'Arrow' Maju

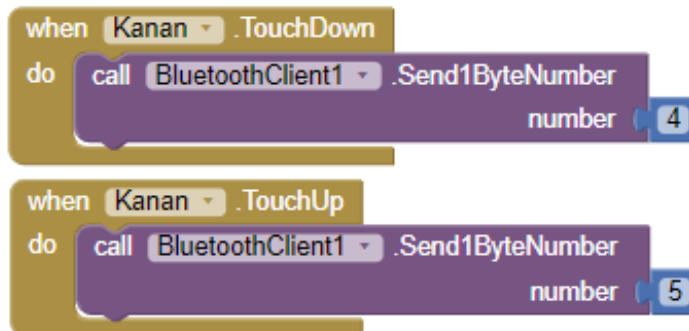




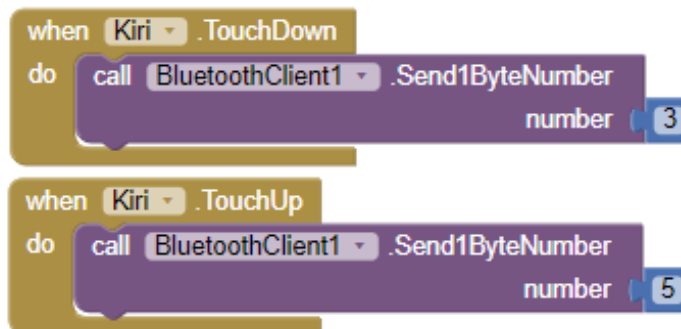
- Deklarasi 'Arrow' Mundur



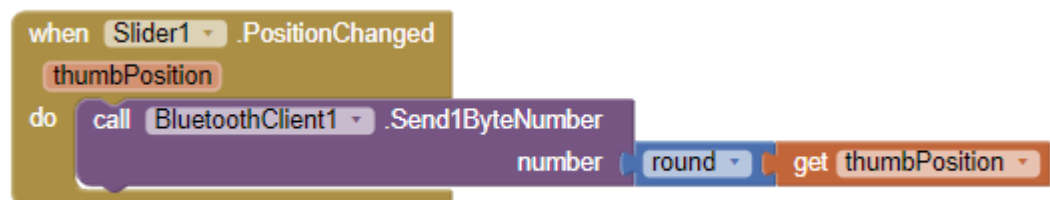
- Deklarasi 'Arrow' Kanan



- Deklarasi 'Arrow' Kiri



- Pengatur kecepatan pada fitu 'Slider'



## BAB IV

### UJI COBA DAN ANALISA

#### 4.1 Mekanik

**Uji coba pergerakan kondisi ban pada 10 kali putaran pada permukaan keramik**

**Tujuan :** Untuk mengetahui pengaruh sebuah bahan pada ban pada medan yang dilalui

**Metode :** Mengelilingi lintasan selama 10 kali putaran

Percobaan	Waktu
1	13,51
2	10,26
3	08,87
4	11,66
5	11,64
6	09,15
7	09,92
8	08,90
9	10,00
10	10,34

- **Grafik :**



**Analisa :**

Dari hasil percobaan didapat dengan menggunakan ban dengan bahan karet dipermukaan keramik, waktu tempuh 1 putaran dengan panjang lintasan 2,4 meter memiliki waktu berkisar pada 8 - 14 detik. Ini berkesimpulan bahwa ban berbahan karet tidak dapat bergerak secara efektif dikarenakan kurangnya gaya gesek ban pada keramik yang memiliki permukaan yang licin.

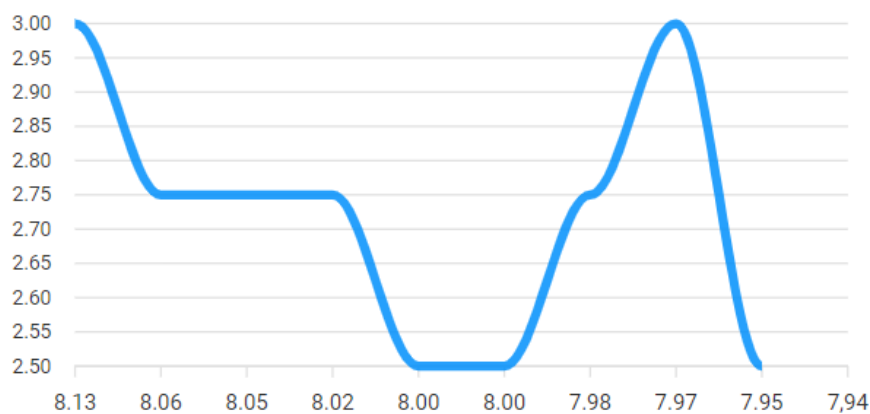
## 4.2 Elektrik

**Uji coba perubahan tegangan pada baterai per 30 detik.**

**Tujuan :** Untuk mengetahui perubahan tegangan dari baterai ketika digunakan

**Metode :** Menghitung laps, dan nilai tegangan setiap 30 detik

Voltage	Laps
8,13	3
8,06	2,75
8,05	2,75
8,06	2,75
8,00	2,50
8,00	2,50
7,98	2,75
7,97	3
7,95	2,50
7,94	2,25



### Analisa

Dari hasil percobaan yang telah dilakukan menghasilkan Kesimpulan bahwa ketika rangkaian aktif dan robot memutar line dengan kondisi otomatis dalam 30 detik terlihat perubahan nilai voltage yang signifikan pada percobaan pertama yaitu dari 8,13 volt menjadi 8,06 volt dan perubahan jumlah laps terlihat juga karena akibat dari berkurang nya daya input dari battery, namun konsumsi daya battery tidak stabil dan konsisten terlihat pada kurva yang menunjukkan bertambahnya kecepatan saat battery sudah mencapai 7,97 volt jumlah laps yang didapat bertambah. Hal ini terjadi karena beberapa factor yang mempengaruhi pergerakan robot dari sensitivitas sensor yang tidak stabil dan medan line yang licin membuat roda robot selip.

### 4.3 Program

Uji coba kestabilan robot pada pembacaan sensor ketika diberi delay yang berbeda dalam kondisi otomatis.

**Tujuan :** Untuk mengetahui pengaruh perubahan jumlah delay pada sensor

**Metode :** Mengubah jumlah delay dalam 10 kali percobaan

Percobaan	Jumlah Delay	Stabilitas	
		Jalan lurus	Jalan berbelok
1	10	✓	✓
2	30	✓	✓
3	50	✓	✓
4	70	✓	✗
5	90	✓	✗
6	110	✗	✗
7	130	✗	✗
8	150	✗	✗
9	170	✗	✗
10	190	✗	✗

#### Analisa :

Selama 10 kali percobaan dengan menggunakan delay yang berbeda beda dengan kelipatan 20, hasil yang didapat pada nilai delay kisaran 10 – 50, pembacaan sensor berjalan dengan baik, ketika menemui jalan lurus maupun berbelok sensor masih bisa membaca dengan baik sehingga stabilitas pada robot masih baik, ketika nilai delay berkisar 70 – 90 pembacaan sensor mulai tidak stabil hal itu ditandai dengan stabilitas robot yang kurang baik saat menemui jalan berbelok, sehingga seringkali keluar jalur namun untuk jalan lurus sensor masih bisa membaca dengan stabil, ketika nilai berkisar 110 – 190 pembacaan sensor sangat tidak stabil, hal ini ditandai dengan kurangnya stabilitas robot ketika melewati jalan lurus sehingga mengakibatkan robot sering kali keluar jalur pada jalan lurus maupun berbelok, kesimpulan yang bisa diambil adalah nilai delay yang berkisar 10 – 50 cocok untuk digunakan untuk pembacaan sebuah sensor infrared pada sebuah robot line follower.

#### 4.4 Interface

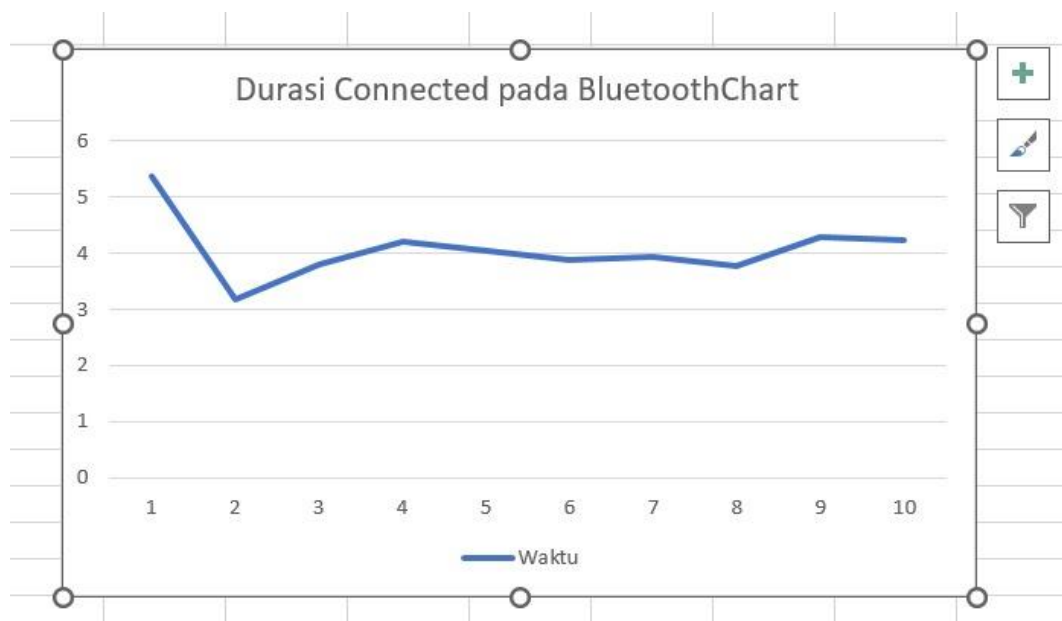
Uji coba durasi konektifitas bluetooth pada kondisi disconnect menuju connect.

**Tujuan :** Untuk mengetahui durasi yang diperlukan untuk mengoneksikan bluetooth

**Metode :** Menguji konektifitas selama 10 kali percobaan

Percobaan	Waktu
1	5,38
2	3,18
3	3,80
4	4,20
5	4,05
6	3,89
7	3,93
8	3,77
9	4,28
10	4,23

- **Grafik**



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dalam lomba Ujian Akhir Semester elektronika yang menggunakan robot line follower, terdapat beberapa temuan penting. Robot line follower adalah robot yang didesain untuk mengikuti jalur atau garis tertentu yang biasanya berwarna kontras dengan permukaan arena. Lomba ini bertujuan untuk menguji kemampuan robot dalam mengenali dan mengikuti jalur dengan akurat dan cepat. Meskipun konsep dan implementasi robot line follower umumnya berhasil, terdapat beberapa kendala yang muncul selama pelaksanaan lomba. Salah satu kendala utama adalah ukuran arena yang terlalu kecil, yang mengakibatkan ruang gerak robot menjadi terbatas dan sulit untuk bermanuver. Selain itu, perubahan bentuk arena yang terlalu mendadak juga menyulitkan robot untuk menyesuaikan diri dengan cepat, sering kali menyebabkan robot keluar jalur atau berhenti.

#### **5.2 Saran**

1. Peningkatan Ukuran Arena:

Untuk memberikan lebih banyak ruang bagi robot untuk bermanuver, disarankan untuk memperbesar ukuran arena lomba. Dengan ukuran arena yang lebih besar, robot akan memiliki lebih banyak ruang untuk menyesuaikan diri dengan perubahan jalur dan mengurangi risiko tabrakan atau keluar jalur.

2. Perencanaan Bentuk Arena yang Lebih Bertahap:

Bentuk arena sebaiknya direncanakan dengan lebih bertahap dan tidak terlalu mendadak. Transisi yang lebih halus antara tikungan dan jalur lurus akan membantu robot menyesuaikan kecepatan dan arah dengan lebih baik, sehingga meningkatkan kinerja keseluruhan robot dalam mengikuti jalur.

3. Pemberitahuan Desain Arena Sebelum Lomba:

Pemberitahuan bentuk arena kepada peserta sebelum lomba dimulai sangat penting. Dengan memberikan gambaran umum tentang desain arena sebelumnya, peserta dapat melakukan kalibrasi dan penyesuaian pada robot mereka agar lebih siap menghadapi tantangan yang ada.

## DAFTAR PUSTAKA

Ridarmin, R., Fauzansyah, F., Elisawati, E., & Prasetyo, E. (2019). Prototype robot line follower Arduino Uno menggunakan 4 sensor TCRT5000. *Informatika*, 11(2), 17-23.

Pomalingo, M. F., Suarjana, I. W. G., & Parhusip, B. R. (2022). Pedoman Dasar Penggunaan Solidworks.

Suharto, A. (2021). Tutorial Mudah Membuat Aplikasi Android Dengan MIT APP INVENTOR (AI2). Penerbit Adab.

Kadir, A. (2017). Pemrograman Arduino & Android Menggunakan App Inventor. Elex Media Komputindo.

Fahmizal, S. T., Mayub, A., Kom, M., Arrofiq, I. M., & Ruciyanti, F. (2022). Mudah Belajar Arduino dengan Pendekatan berbasis Fritzing, Tinkercad dan Proteus. Deepublish.

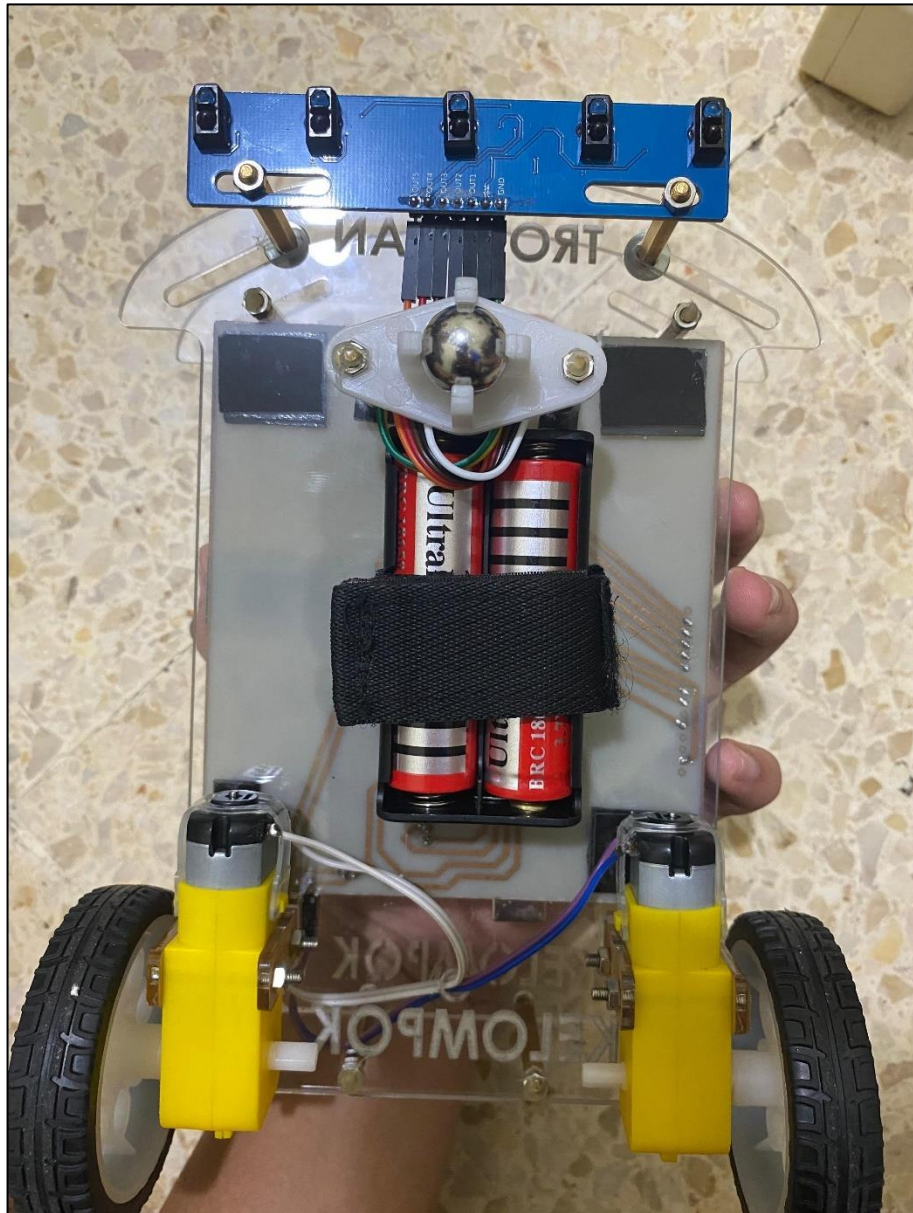
## LAMPIRAN

- Mekanik
  - Case bagian atas

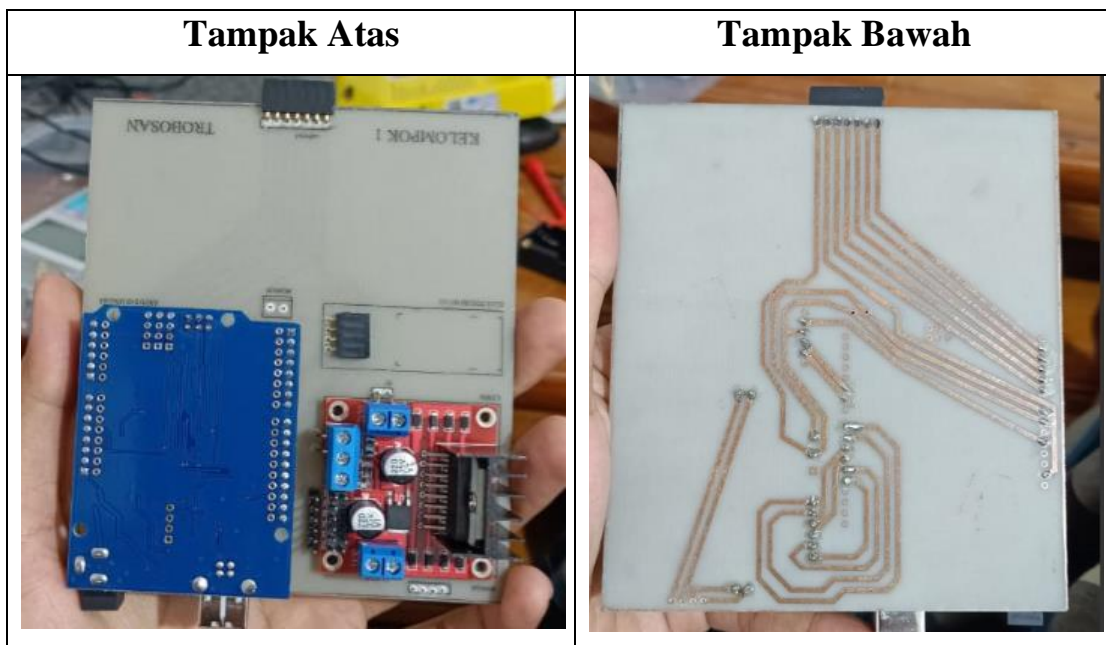




- Case bagian bawah



- **Elektrik**



- **Program**

```
#include <SoftwareSerial.h>

SoftwareSerial BT_Serial(5, 6); // RX, TX

//Definisi pin Motor Driver
#define e1 8 //Pin EA
#define m1 9 //Motor MA1
#define m2 10 //Motor MA2
#define m3 11 //Motor MB1
#define m4 12 //Motor MB2
#define e2 13 //Pin EB

//Definisi pin sensor
#define ir1 A4
#define ir2 A3
#define ir3 A2
#define ir4 A1
#define ir5 A0

//inisialisasi
int bt_data; // variable untuk menerima data
int Speed = 255;
int mode = 0;
```

```

void setup() {
//Fungsi sensor
  pinMode(ir1, INPUT);
  pinMode(ir2, INPUT);
  pinMode(ir3, INPUT);
  pinMode(ir4, INPUT);
  pinMode(ir5, INPUT);

//Fungsi motordriver
  pinMode(m1, OUTPUT);
  pinMode(m2, OUTPUT);
  pinMode(m3, OUTPUT);
  pinMode(m4, OUTPUT);
  pinMode(e1, OUTPUT);
  pinMode(e2, OUTPUT);

  Serial.begin(9600); /// memulai komunikasi serial di 9600bps
  BT_Serial.begin(9600);
  delay(500);
}

void loop() {
  if (BT_Serial.available() > 0) { //Jika data terkirim, maka akan
memanggil fungsi baca
    bt_data = BT_Serial.read();
    if (bt_data > 20) { Speed = bt_data; }
  }

  if (bt_data == 8) { mode = 1; Speed = 255; } //Auto Line Follower
  else if (bt_data == 9) { mode = 0; berhenti(); } //Manual Kontrol

  analogWrite(e1, Speed); // Menulis pin A sebagai speed motor 1
  analogWrite(e2, Speed); // Menulis pin B sebagai speed untuk motor 2

  if (mode == 0) {
/*=====
=                      Kontrol Manual                      =
=====*/
    if (bt_data == 1) { maju(); }
    else if (bt_data == 2) { mundur(); }
    else if (bt_data == 3) { kiri(); }
    else if (bt_data == 4) { kanan(); }
    else if (bt_data == 5) { berhenti(); }
  }
  else {

```

```

/*=====
=           Kontrol Line Follower           =
//=====*/

int s1 = digitalRead(ir1); // Sensor paling kiri
int s2 = digitalRead(ir2); // Sensor kiri
int s3 = digitalRead(ir3); // Sensor tengah
int s4 = digitalRead(ir4); // Sensor kanan
int s5 = digitalRead(ir5); // Sensor paling kanan

//hanya sensor3 mendeteksi garis hitam akan maju
if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 1 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}
//hanya sensor2 mendeteksi garis hitam akan ke kiri
else if (s1 == 1 && s2 == 0 && s3 == 1 && s4 == 1 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}
//hanya sensor1 mendeteksi garis hitam akan ke kiri
else if (s1 == 0 && s2 == 1 && s3 == 1 && s4 == 1 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, HIGH);
}
//hanya sensor4 mendeteksi garis hitam akan ke kanan
else if (s1 == 1 && s2 == 1 && s3 == 1 && s4 == 0 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

```

```

//hanya sensor5 mendeteksi garis hitam akan ke kanan
else if (s1 == 1 && s2 == 1 && s3 == 1 && s4 == 1 && s5 == 0) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, LOW);
    digitalWrite(m2, HIGH);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}
//sensor3 dan sensor4 mendeteksi garis hitam akan ke kanan
else if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 0 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}
//sensor2 dan sensor3 mendeteksi garis hitam akan ke kiri
else if (s1 == 1 && s2 == 0 && s3 == 0 && s4 == 1 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}
//sensor1, sensor2 dan sensor3 mendeteksi garis hitam akan ke kiri
else if (s1 == 0 && s2 == 0 && s3 == 0 && s4 == 1 && s5 == 1) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}
//sensor3, sensor4 dan sensor5 mendeteksi garis hitam akan ke kanan
else if (s1 == 1 && s2 == 1 && s3 == 0 && s4 == 0 && s5 == 0) {
    analogWrite(e1, 255);
    analogWrite(e2, 255);
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

```

```

//semua sensor mendeteksi garis hitam akan ke kiri akan berhenti
else if (s1 == 0 && s2 == 0 && s3 == 0 && s4 == 0 && s5 == 0) {
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}
}
delay(10);
}

void maju() {
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

void mundur() {
    digitalWrite(m1, LOW);
    digitalWrite(m2, HIGH);
    digitalWrite(m3, LOW);
    digitalWrite(m4, HIGH);
}

void kanan() {
    digitalWrite(m1, LOW);
    digitalWrite(m2, HIGH);
    digitalWrite(m3, HIGH);
    digitalWrite(m4, LOW);
}

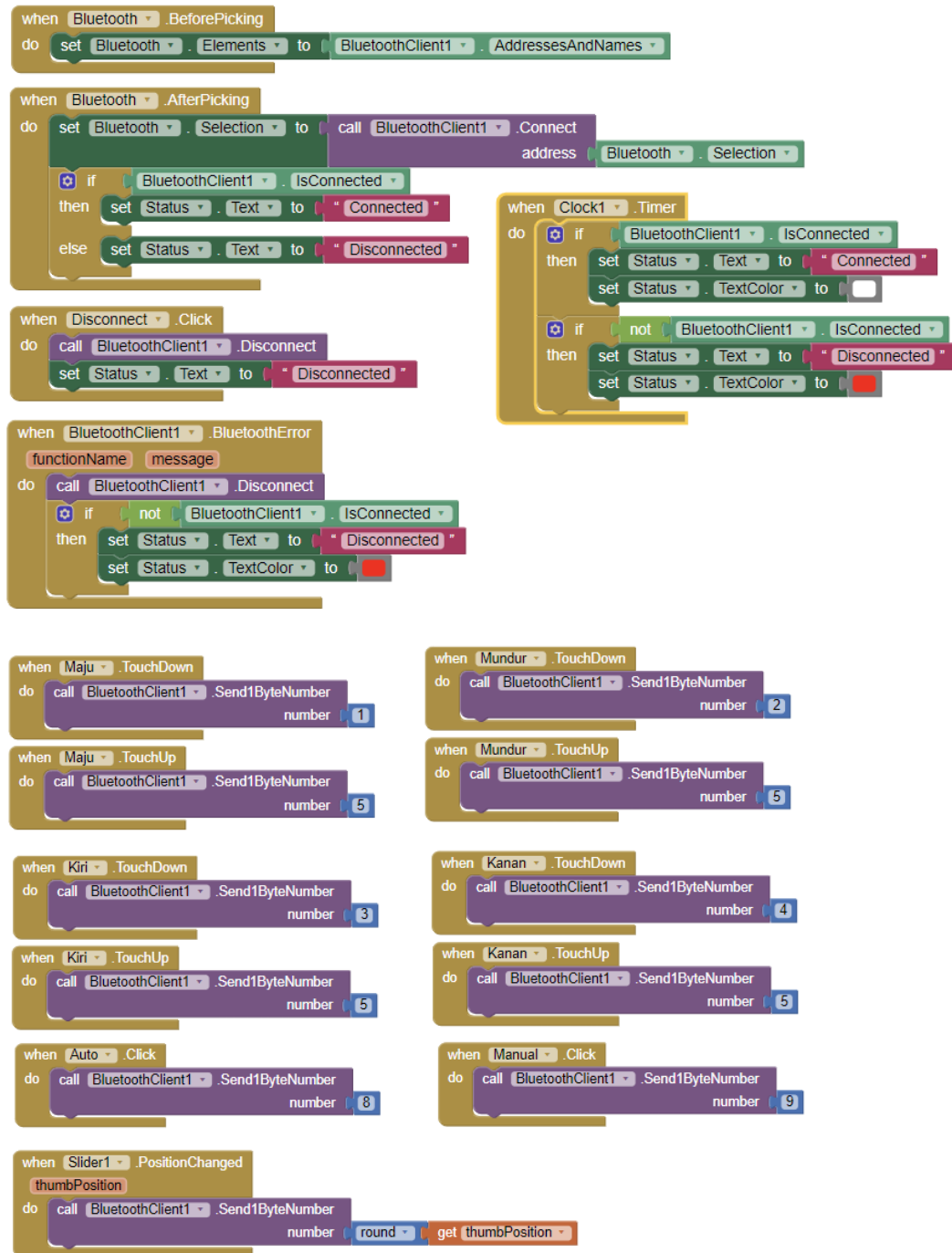
void kiri() {
    digitalWrite(m1, HIGH);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, HIGH);
}

void berhenti() {
    digitalWrite(m1, LOW);
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
}

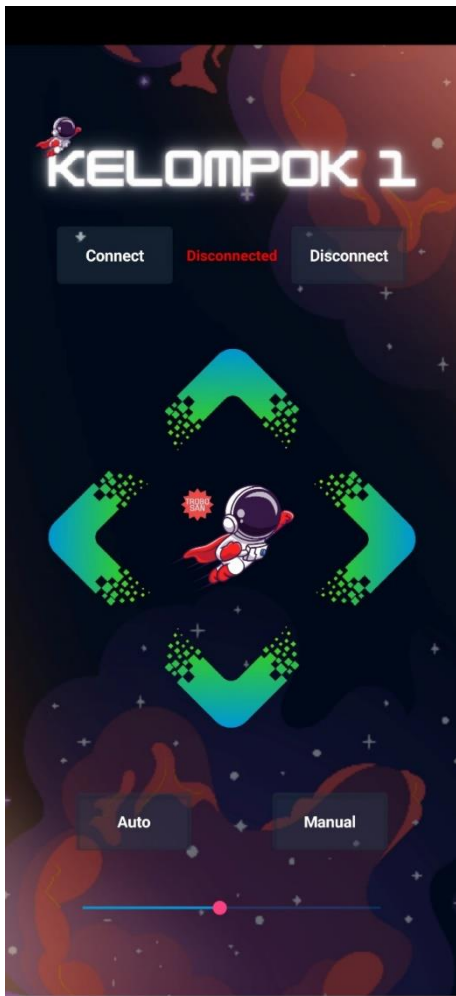
```

- **Interface**

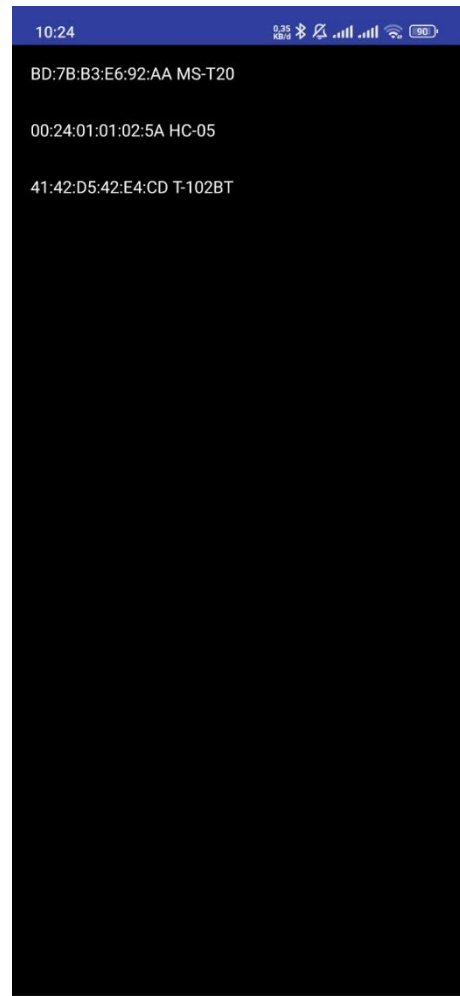
- **Code Block**



- Tampilan aplikasi



(Tampilan Home)



(Tampilan Bluetooth)