

LAPORAN TUGAS BESAR

Grass Cutter Robot with Obstacle Avoider

Laporan

Diajukan untuk memenuhi tugas besar mata kuliah mekatronika.

Disusun oleh :

Gede Haris Widiarta

(NIM 1102174038)

Rizqy Seto Atmaji

(NIM 1102174158)



PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

BANDUNG

2020

DAFTAR ISI

BAB I.....	4
1.1 Pendahuluan	4
1.2 Desain Mekanik Robot.....	4
1.2.1 Dimensi	7
1.2.2 Bahan Material	7
1.2.3 Posisi Komponen	7
BAB II	8
2.1 Definisi	8
2.2 Penerapan <i>Level Control</i>	10
BAB III.....	14
3.1 Sensor.....	14
3.2 Mikrokontroller	15
3.3 Aktuator	16
3.4 <i>Driver</i> Pendukung.....	17
3.5 <i>Power Supply</i>	19
3.6 <i>Wiring Komponen</i>.....	19
BAB IV	21
4.1 <i>Flowchart</i>	21
4.2 <i>Source Code</i>	21
BAB V.....	29
5.1 Analisa	29
5.2 Kesimpulan	32
BAB VI.....	33
6.1 Foto	33
6.2 Video	35

DAFTAR GAMBAR

Gambar 1 : Desain 3D Grass Cutter with Obstacle Avider	5
Gambar 2 : Referensi Desain 3D Robot.....	6
Gambar 3 : Level Control System.....	10
Gambar 4 : FSM Grass Cutter Robot.....	12
Gambar 5: Sensor Ultrasonic HCSR-04 dan Cara Kerjanya	14
Gambar 6 : Arduino Uno.....	15
Gambar 7 : Motor DC 3V dengan Roda	16
Gambar 8 : Motor Servo SG 90	17
Gambar 9 : Motor Driver L298N (menyatu dengan Arduino Uno).....	18
Gambar 10 : Baterai Li-Ion 3.7V	19
Gambar 11 : Wiring Komponen Grass Cutter Robot (Part 1).....	19
Gambar 12 : Wiring Komponen Grass Cutter Robot (Part 2).....	20
Gambar 13 : Flowchart Grass Cutter Robot.....	21
Gambar 14 : Hasil Pengukuran Jarak Sensor Ultrasonic	29
Gambar 15 : Pisau Pemotong Rumput	30
Gambar 16 : Ilustrasi Robot Saat Bekerja.....	31
Gambar 17 : Awal Pengerjaan 1	33
Gambar 18 : Awal Pengerjaan 2	33
Gambar 19 : Hasil Jadi 1 (Tampak Samping & Atas)	34
Gambar 20 : Hasil Jadi 2 (Tampak Depan).....	34
Gambar 21 : Hasil Jadi 3 (Tampak Bawah)	35

BAB I

Robot Platform Mechanic Design

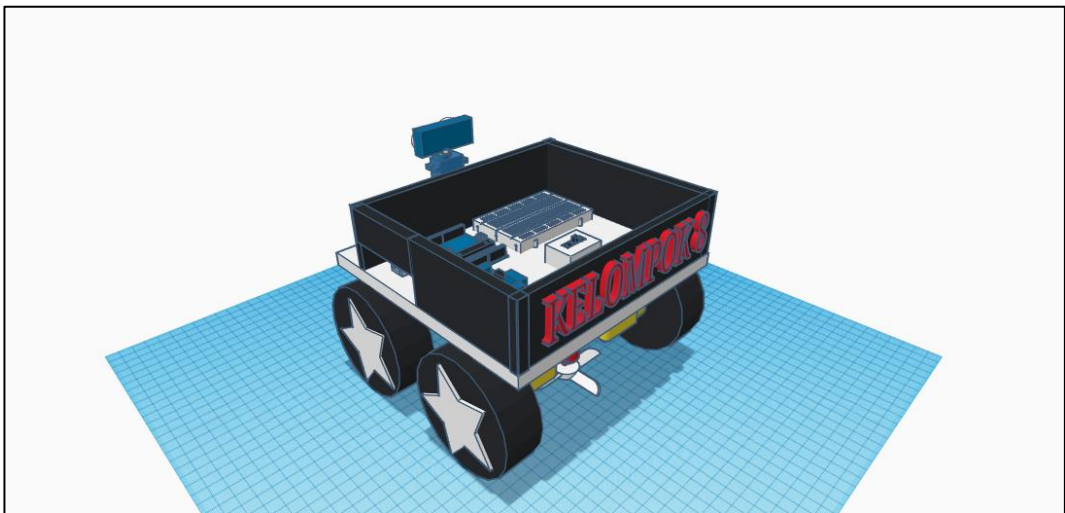
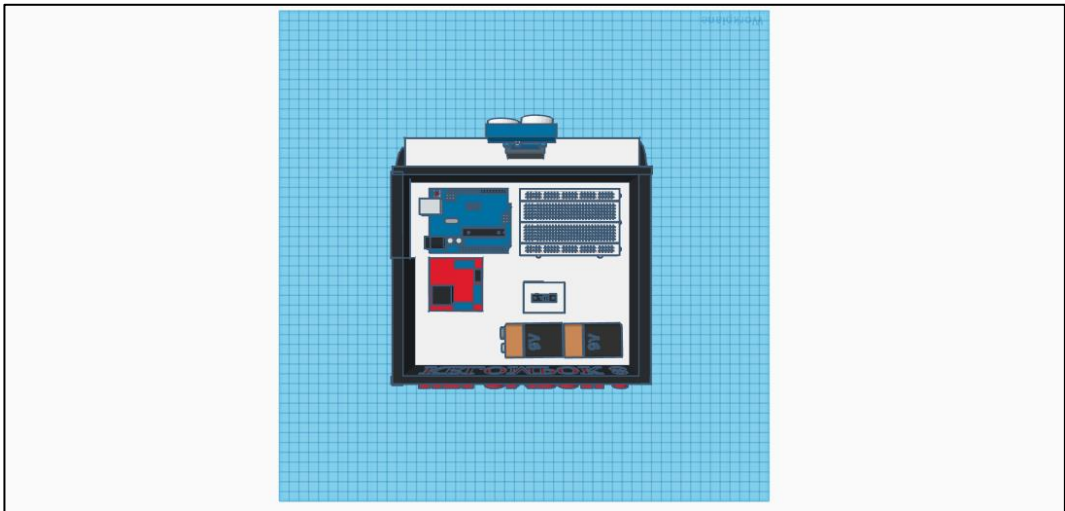
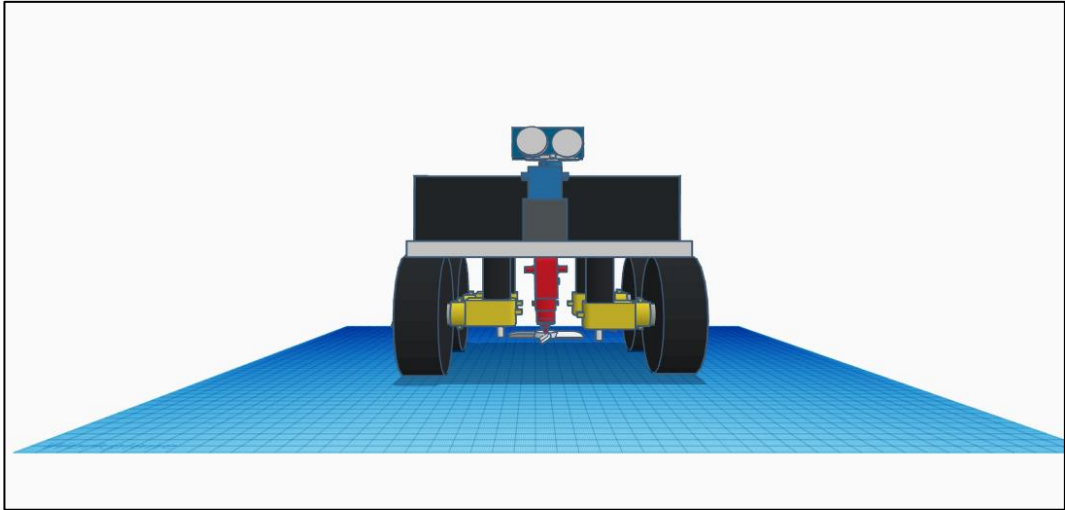
1.1 Pendahuluan

Mobile robot adalah robot dengan kemampuan bergerak di sekitar lingkungan dan tidak menetap pada satu lokasi fisik. Menghindari tabrakan (*obstacle avoidance*) adalah salah satu isu utama pada *mobile robot*, dalam aplikasi untuk perindustrian, pendukung aktivitas manusia, dan lain-lain. Pada pengaplikasiannya *mobile robot* baik digunakan di perindustrian, agar robot dapat memiliki lebih banyak fungsi dan efisien. *Mobile robot* adalah robot yang ciri konstruksinya menggunakan roda atau kaki sebagai aktuatornya untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat berpindah dari satu tempat ke tempat lain dengan sistem navigasi yang berasal dari sensor. Roda sebagai daya penggerak sudah sangat populer digunakan oleh banyak orang karena mudah dan tidak perlu memikirkan dan melakukan perhitungan dari sisi keseimbangan dimana itu sangat rumit. Kinerja dari *mobile robot* bergantung pada komponen-komponen dasar seperti hardware dari robot (body, motor, sensor, dan lain-lain), kondisi lingkungan tempat robot bekerja, dan programming.

Dengan berlandaskan konsep *mobile robot*, maka robot yang kami buat yaitu *Grass Cutter Robot with Obstacle Avoider*. Dalam pengaplikasiannya diharapkan robot ini dapat berfungsi sesuai dengan tujuannya yaitu membantu manusia dalam pekerjaan memotong rumput di suatu lahan atau kebun sehingga dapat terbantu dan terselesaikan dengan cepat. Peran *obstacle avoider* di robot ini yaitu untuk menghindari adanya tabrakan antara robot dengan objek disekitarnya (batu, tembok, dan lainnya) sehingga robot dapat bekerja dengan baik.

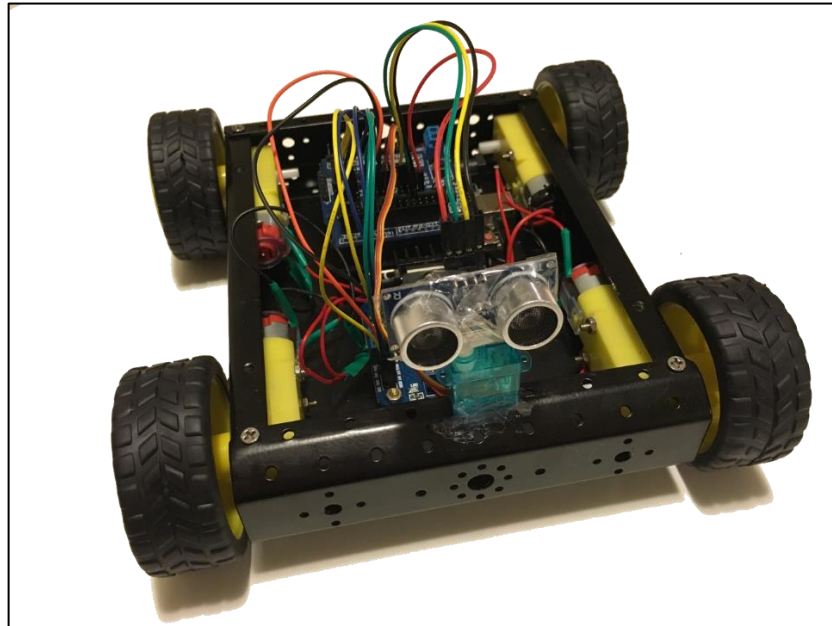
1.2 Desain Mekanik Robot

Dalam pembuatan robot *Grass Cutter Robot with Obstacle Avoider* ini diperlukan desain dasar sebagai acuan dalam pengerjaan robot ini. Berikut desain dasar sederhana yang dapat kami buat.



Gambar 1 : Desain 3D Grass Cutter with Obstacle Avider

Untuk membuat desain 3D tersebut kami mempunyai desain referensi yaitu sebagai berikut.



Gambar 2 : Referensi Desain 3D Robot

Dalam desain tersebut terlihat komponen utama apa saja yang digunakan dalam robot yang kami buat. Untuk rangka atau *chasis* robot kami menggunakan papan kayu triplek dengan ukuran 15 x 25 x 0.5 cm, alasan kami menggunakan material ini karena daya tahan papan kayu yang cukup bagus dan mumpuni sebagai rangka robot serta lebih mudah didapat jika dibandingkan dengan akrilik melihat juga kondisi yang tidak memungkinkan sekarang ini. Untuk aktuatornya kami menggunakan 5 buah motor DC yang 4 diantaranya disambungkan ke roda sebagai penggerak robot dan 1 buah disambungkan ke mata pisau untuk memutar mata pisau sehingga dapat memotong rumput. Untuk mengatur pergerakan motor maka digunakanlah *Motor Driver Shield L293D*. Aktuator lainnya yaitu motor servo untuk menggerakkan sensor ultrasonic sesuai dengan sudut yang diperintahkan, motor servo diletakkan dibagian depan robot. Sensor yang kami gunakan pada robot ini yaitu Sensor Ultrasonic HCR – 04 sebagai pemandu dan *feedback* posisi robot, sensor tersebut diletakkan di bagian atas motor servo dengan menggunakan *bracket* sensor ultrasonic. Kemudian, untuk kontroler dari robot ini, kami menggunakan *microcontroller Arduino Uno* dan untuk *power supply* digunakan empat buah baterai AA atau *Lithium-ion 3.7 V*.

Untuk lebih jelasnya, kami membuat rinciannya sebagai berikut.

1.2.1 Dimensi

- Panjang : 15 cm
- Lebar : 25 cm
- Tinggi : 0.5 cm

1.2.2 Bahan Material

- *Chasis Robot* : 1 buah (berbahan kayu triplek atau papan kayu)
- *Arduino Uno* : 1 buah
- *Driver Motor Shield L293D* : 1 buah
- Motor DC dan *Gearbox* : 5 buah (4 untuk penggerak robot ; 1 untuk pemutar pisau pemotong)
- Sensor *Ultrasonic HC-SR04* : 1 buah
- Pisau Pemotong : 1 buah
- Roda Besar : 4 buah
- *Power Supply* : 4 buah Baterai AA 1.5V
- Tempat Baterai : 1 buah
- Switch : 1 buah (2 pin)
- Kabel *Jumper Male Female dan Male Male*
- *Bracket Sensor Ultrasonic* : 1 buah
- Motor *Servo* : 1 Buah

1.2.3 Posisi Komponen

- *Arduino Uno* terletak di atas *chasis*, tepatnya ditengah – tengah.
- *Driver Motor Shield L293D* terletak di atas *chasis* dan menjadi satu dengan *Arduino Uno* (*attachment*).
- Motor DC dan *Gearbox* sudah menjadi satu, 2 diletakkan di samping kiri beserta roda besar dan 2 lagi diletakkan di samping kanan beserta roda besar, serta 1 lagi di bagian depan dengan pisau pemotong tepatnya di bawah *chasis* dan motor servo.
- Sensor *Ultrasonic HC-SR04* diletakkan di atas motor servo dengan memanfaatkan fungsi *bracket sensor*.
- Tempat Baterai dan Switch diletakkan di atas *chasis* robot.

BAB II

Level Control

2.1 Definisi

Menurut skema – skema sistem control, yaitu control PID, RMRC dan RMAC/RAC dapat dikategorikan sebagai *low-level control*. Pengertian ini didasarkan pada cara instalasi sensor dan cara membaca datanya, yakni dengan mengukur langsung pada bagian tubuh/sendai/sumbu-putar dari struktur (robot). Sedangkan *high-level control* adalah kontrol yang bekerja berdasarkan data-data sensor yang merupakan informasi tentang lingkungan dimana robot itu bekerja. Misalkan kontrol gerak mobile robot untuk menghindari halangan dan mengejar obyek. Data-data sensor yang diperoleh dari sensor jarak (*range finder sensor*) seperti ultrasonic, TX-RX inframerah, dan sebagainya, adalah mengandung informasi lingkungan (eksternal). *high-level control* bertugas membentuk input atau trajektori referensi bagi *low-level control* pada sistem mobile robot *high-level control* berkaitan dengan berbagai hal yang berhubungan dengan pemetaan medan, perencanaan jelajah, metoda penghindaran halangan, koordinasi antar robot, dan sebagainya. Beberapa skema *high-level control* yang umum digunakan antara lain:

- Pendekatan *Model-Plan-Act (MPA)*

Prinsip kerjanya yaitu robot melakukan pembacaan data-data sensor. Dari data ini kontroler melakukan pemodelan medan kerja robot (*the world*). Jika model telah dikenal secara pasti maka kontroler dapat melakukan perencanaan jelajah. Jika rencana “telah matang”, robot melakukan aksi dengan bergerak menuju target. Sesampai di posisi target, robot melakukan pembacaan sensor lagi. Dari sini langkah pemodelan dilakukan lagi. Demikian seterusnya sehingga setiap kali robot telah mencapai target yang baru ia akan menjalankan prosedur yang berulang hingga target terakhir dicapai.

Kelebihan :

- Dari pengetahuan (knowledge) secara global hasil dari tiap testing (penjelajahan) memungkinkan untuk dilakukan langkah-langkah optimasi (jalur terpendek) dengan cukup baik.

- Perencanaan jelajah yang terpendek hasil dari optimasi dapat dijamin keakuratannya.

Kelemahan :

- Pemrogramannya rumit dan berbelit-belit (computationally intensive).
- Membutuhkan sensor-sensor yang berkualitas sangat bagus untuk memperoleh model lingkungan yang akurat.
- Model lingkungan medan yang diperoleh hanyalah sekedar perkiraan (aproksimasi) saja.
- Bila diterapkan dalam medan kerja yang dinamis, misalnya terdapat obyek yang selalu bergerak, metoda ini tidak dapat bekerja dengan baik.

- Pendekatan *Behavior-Based (BB)*

Robot yang menggunakan algoritma high-level control berbasis behavior (tabiat, kelakuan) biasa disebut sebagai behavior-based robot. Algoritma ini adalah salah satu metoda dalam high-level control yang relatif paling banyak diteliti dan diuji. Metoda BB memiliki kelebihan disbanding dengan metoda MPA, yaitu dapat bekerja dengan baik dalam lingkungan yang dinamis. Algoritma BB diturunkan dari sifat-sifat alami makhluk hidup, yaitu bertingkah laku sesuai dengan keadaan lingkungannya. Berdasarkan informasi dari panca indera tertentu, syaraf yang berhubungan dengan otot gerakan anggota badan terkait akan mendapat stimuli dari otak sehingga memberikan suatu respon yang khas.

Kelebihan :

- Dalam pemrograman, behavior dinyatakan sebagai suatu fungsi rutin yang dapat dengan mudah diaktifkan atau tidak. Dengan demikian disain sistem kontrolnya menjadi lebih transparan.
- Sistem modul yang digunakan dalam pemrogramannya juga memberikan kemudahan dalam eksperimen karena setiap modul dapat dites satu persatu sebelum digabung.
- Metode control BB ini sangat proaktif dan reaktif terhadap perubahan dinamik lingkungan sehingga sangat sesuai digunakan

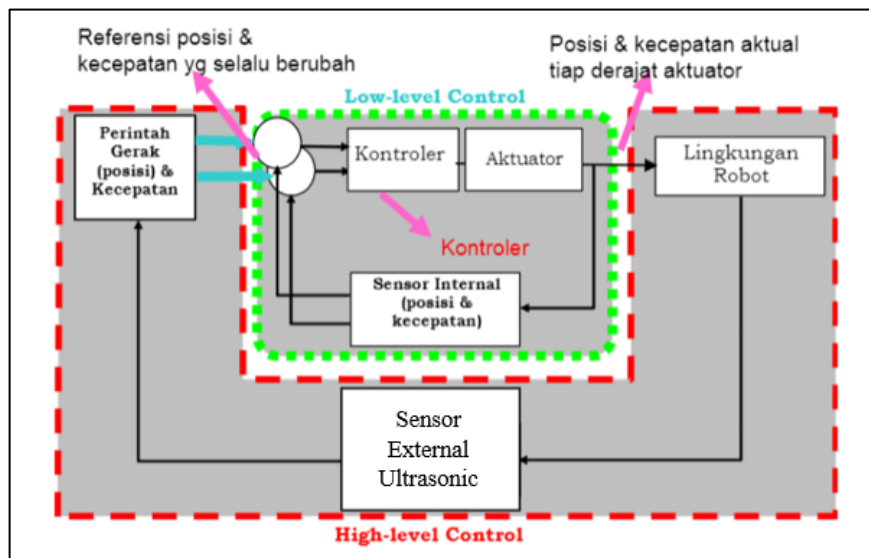
sebagai basis robot autonomous yang mampu belajar sendiri secara online dan terus-menerus.

Kelemahan :

- Dari sudut pandang programmer: tidak dapat dipastikan apa sebenarnya yang sedang dan akan dilakukan oleh robot ketika beraksi dilapangan. Susah untuk dievaluasi dan ditentukan letak kesalahannya jika robot mampu menyelesaikan tugas.
 - Tidak bisa dijamin robot selalu mampu menyelesaikan tugasnya
 - Maintenance dan kalibrasi program juga amat sulit.
- Pendekatan *Finite State Machine (FSM)*

Finite State Machine adalah suatu mekanisme untuk menentukan suatu solusi berdasarkan perubahan-perubahan keadaan (state) waktu demi waktu. Setiap solusi yang diperoleh untuk satu perubahan state pada dasarnya adalah identik dengan sebuah behavior seperti dalam algoritma BB. Dalam teknik FSM, perubahan suatu state terjadi berdasarkan informasi data umpan balik dari sensor-sensor. Jadi dari segi waktu, FSM dapat bersifat realtime karena state slalu diperbaharui setiap kali lingkungan berubah.

2.2 Penerapan *Level Control*



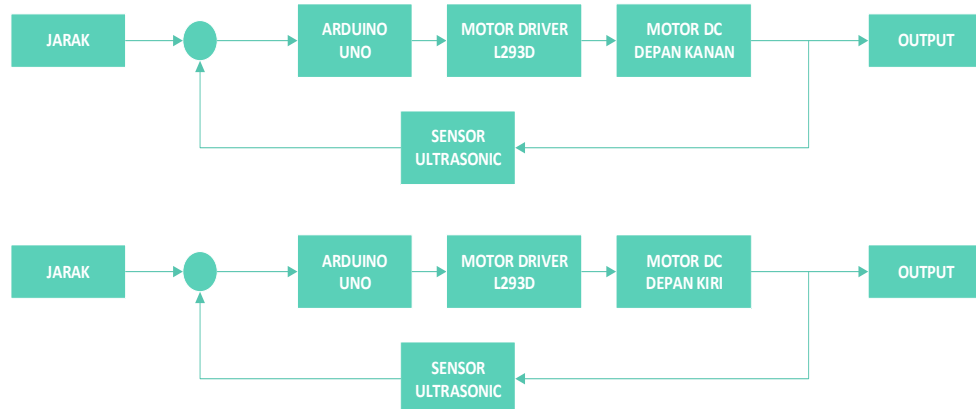
Gambar 3 : Level Control System

Berdasarkan teori *level control* yang sudah dijelaskan, pemodelan sistem *level control* yang kami gunakan disini yaitu *High Level Control* dan *Low Level*

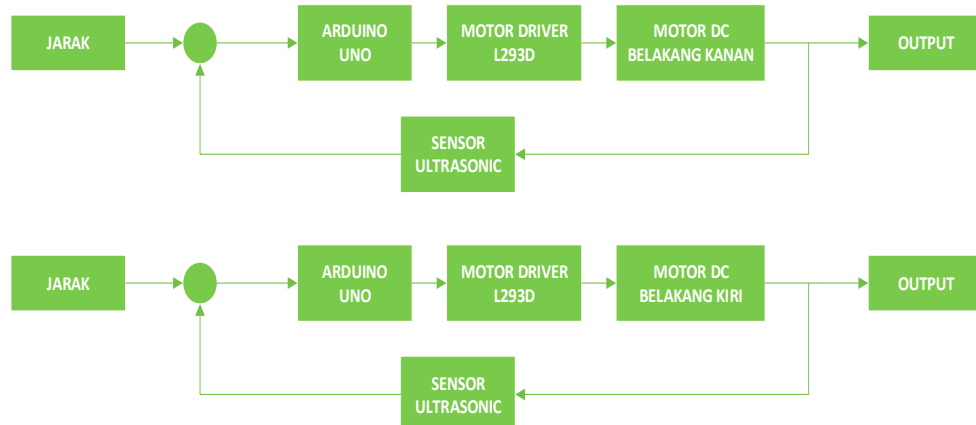
Control. Alasan kami menggunakan *High Level Control* karena sistem kontrol ini bekerja berdasarkan data-data sensor yang merupakan informasi tentang lingkungan dimana robot itu bekerja. *High Level Control* juga banyak digunakan di *mobile robot* untuk menghindari halangan dan mengejar obyek. Penggunaan pendekatan Finite State Machine (FSM) pada robot yang kami buat dikarenakan robot tersebut harus bisa menentukan atau menemukan solusi berdasarkan perubahan keadaan secara real time yang terjadi berdasarkan informasi data feedback dari sensor yang ada.

Low Level Control pada robot ini yaitu pada bagian kendali yang terdapat pada robot, maka digunakan data sensor yang diperoleh dari sensor jarak (range finder sensor) seperti sensor ultrasonic.

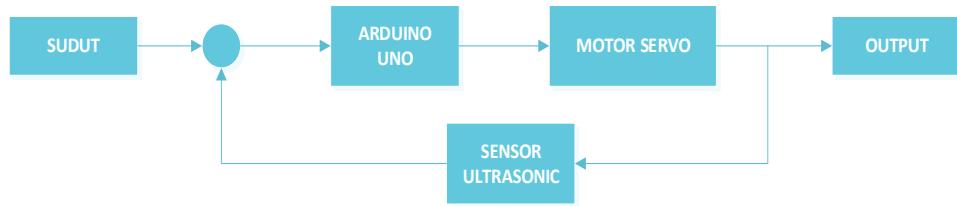
- Roda Penggerak Depan



- Roda Penggerak Belakang

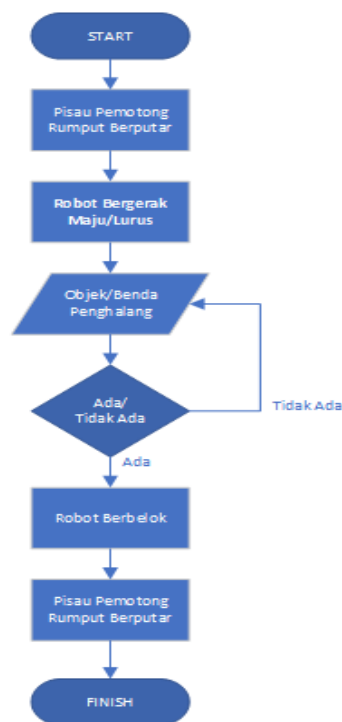


- Motor Servo

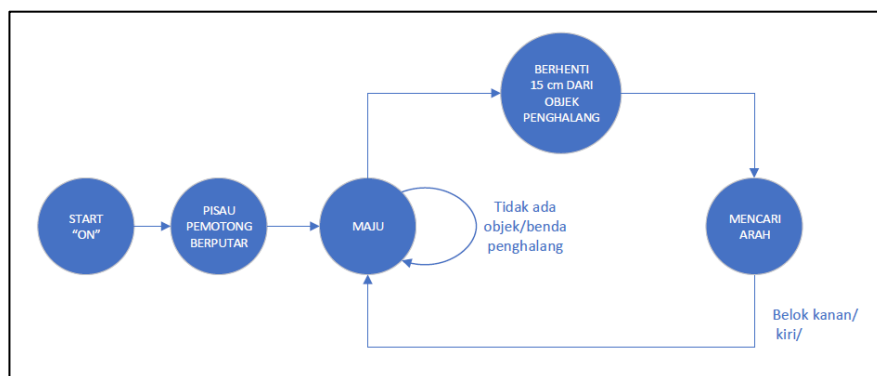


High Level Control pada robot ini direalisasikan dalam bentuk *flowchart* dan *FSM*.

- Pemodelan *Flowchart* untuk Grass Cutter with Obstacle Avoider.



- Pemodelan *Finite State Machine (FSM)* Grass Cutter with Obstacle Avoider yaitu sebagai berikut.



Gambar 4 : FSM Grass Cutter Robot

Dimulai dengan *initial state* robot yaitu diam (OFF) kemudian di *switch* ke ON maka motor dc pada pisau pemotong akan berputar, kemudian robot berjalan maju apabila tidak ada objek penghalang maka robot akan terus maju mengikuti jalur awalnya, jika ada objek penghalang maka robot berhenti 15 cm dari objek tersebut dan mencari arah alternatif melalui sensor ultrasonic dan selanjutnya robot akan bergerak ke kanan atau ke kiri lalu berjalan maju kembali.

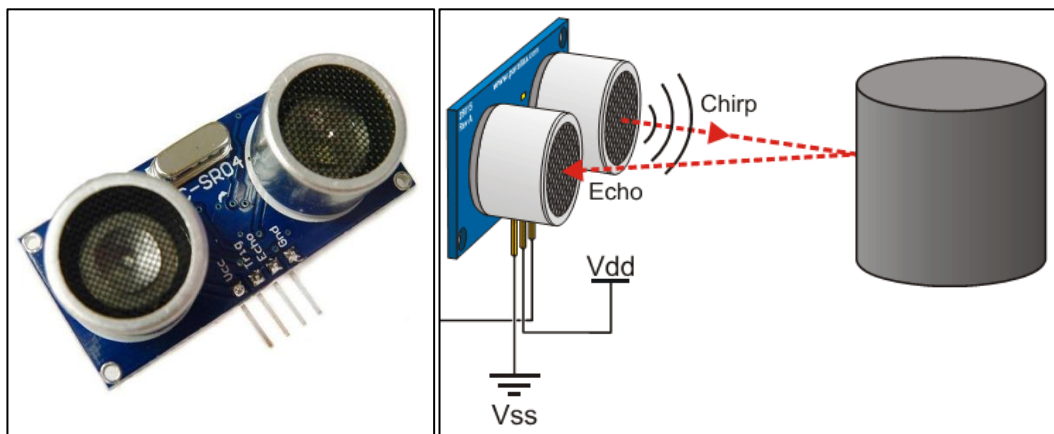
BAB III

Components and Wiring

3.1 Sensor

Sensor adalah perangkat yang digunakan untuk mendeteksi perubahan besaran fisik seperti tekanan, gaya, besaran listrik, cahaya, gerakan, kelembaban, suhu, kecepatan dan fenomena-fenomena lingkungan lainnya. Sensor pada dasarnya dapat digolong sebagai Transduser Input karena dapat mengubah energi fisik seperti cahaya, tekanan, gerakan, suhu atau energi fisik lainnya menjadi sinyal listrik ataupun resistansi (yang kemudian dikonversikan lagi ke tegangan atau sinyal listrik).

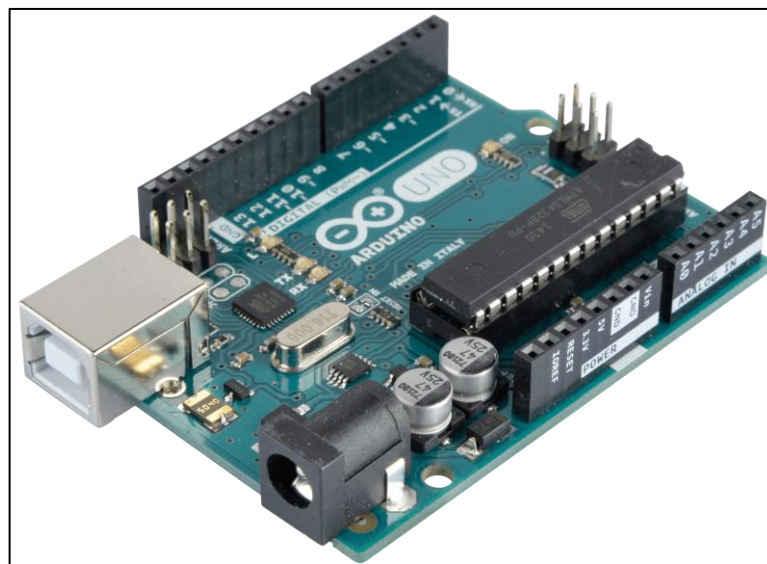
Jenis sensor yang digunakan pada robot ini yaitu Sensor Ultrasonic HCSR-04. Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik). Sensor ini memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.



Gambar 5: Sensor Ultrasonic HCSR-04 dan Cara Kerjanya

3.2 Mikrokontroller

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board *Arduino Uno* ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke adaptor-DC atau baterai untuk menjalankannya.



Gambar 6 : *Arduino Uno*

Spesifikasi *Arduino Uno* sebagai berikut :

- Microcontroller ATmega328
- Operasi dengan daya 5V Voltage
- Input Tegangan (disarankan) 7-12V
- Input Tegangan (batas) 6-20V
- Digital I / O Pins 14 (dimana 6 memberikan output PWM)
- Analog Input Pin 6
- DC Lancar per I / O Pin 40 mA
- Saat 3.3V Pin 50 mA DC
- Flash Memory 32 KB (ATmega328) yang 0,5 KB digunakan oleh bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

3.3 Aktuator

Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan lengan mekanis yang biasanya digerakkan oleh motor listrik, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroler. Aktuator adalah elemen yang mengkonversikan besaran listrik analog menjadi besaran lainnya misalnya kecepatan putaran dan merupakan perangkat elektromagnetik yang menghasilkan daya gerakan sehingga dapat menghasilkan gerakan pada robot.

- Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Motor Listrik DC atau DC Motor ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (revolutions per minute) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalik. Terdapat dua bagian utama pada sebuah Motor Listrik DC, yaitu Stator dan Rotor. Stator adalah bagian motor yang tidak berputar, bagian yang statis ini terdiri dari rangka dan kumparan medan. Sedangkan Rotor adalah bagian yang berputar, bagian Rotor ini terdiri dari kumparan Jangkar. Dua bagian utama ini dapat dibagi lagi menjadi beberapa komponen penting yaitu diantaranya adalah *Yoke* (kerangka magnet), *Poles* (kutub motor), *Field Winding* (kumparan medan magnet), *Armature Winding* (kumparan jangkar), *Commutator* (komutator) dan *Brushes* (kuas/sikat arang).



Gambar 7 : Motor DC 3V dengan Roda

- Motor Servo adalah sebuah motor DC dengan sistem umpan balik tertutup dimana posisi rotornya akan diinformasikan kembali ke rangkaian control yang ada di dalam motor servo. Motor servo terdiri dari tiga bagian utama, yaitu: motor, sistem kontrol dan potensiometer/encoder yang terhubung dengan satu set roda gigi ke poros output. Potentiometer atau encoder ini lah yang berfungsi sebagai sensor yang memberikan sinyal umpan balik (feedback) ke sistem kontrol apakah posisi targetnya sudah benar atau belum. Potentiometer ini terdiri dari tiga kabel dengan 2 kabel untuk power dan 1 kabel untuk kabel sinyal.



Gambar 8 : Motor Servo SG 90

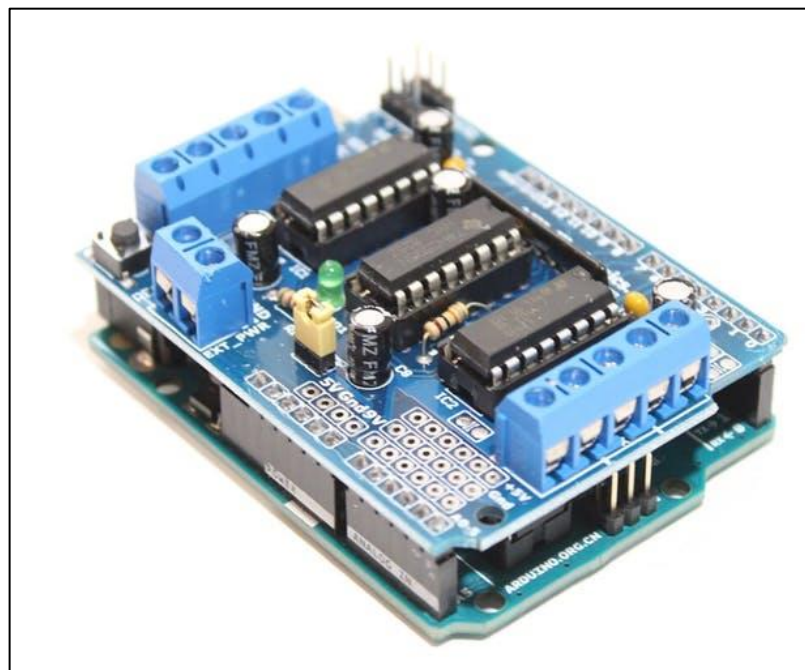
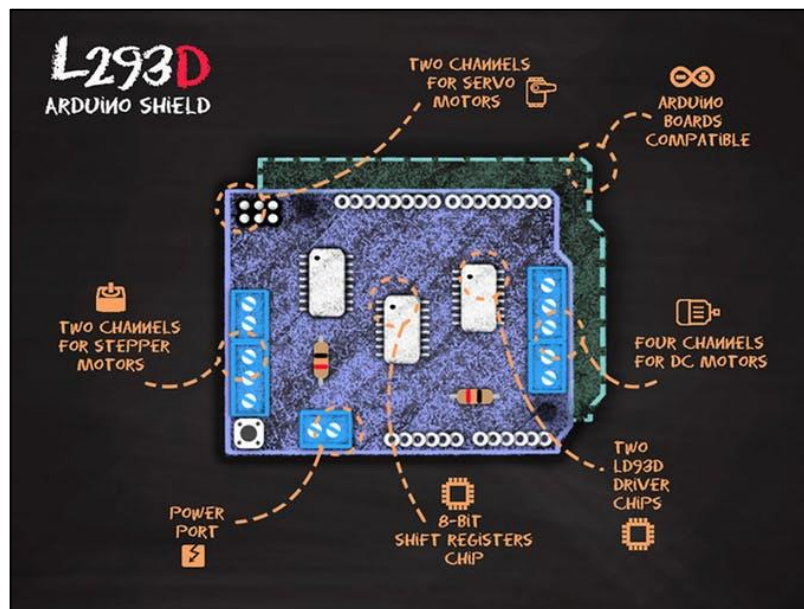
3.4 Driver Pendukung

Dual L293D motor shield, merupakan shield arduino yang mudah penggunaannya untuk pembuatan aplikasi robot beroda. Karena shield ini dapat menjalankan 4 buah motor dan dua buah servo sekaligus. Shield ini adalah produk buatan dari adafruit.

Spesifikasi lengkap shield ini adalah sebagai berikut.

- Konektor untuk 5V Servo.
- Dapat menjalankan 4 motor DC atau 2 stepper motor atau 2 Servo.
- Dapat menjalankan 4 motor bi-directional DC dengan kecepatan pemilihan 8-bit.

- Menjalankan 2 stepper motor (unipolar atau bipolar) dengan single coil atau double coil.
- 4 H-Bridges: per bridge menyediakan 0.6A (1.2A saat puncak) dengan perlindungan termal, dapat menjalankan motor 4.5V sampai 36V DC.
- Tombol reset.
- 2 konektor daya eksternal.
- Kompatibel untuk Uno, Mega Diecimila & Duemilanove.



Gambar 9 : Motor Driver L298N (menyatu dengan Arduino Uno)

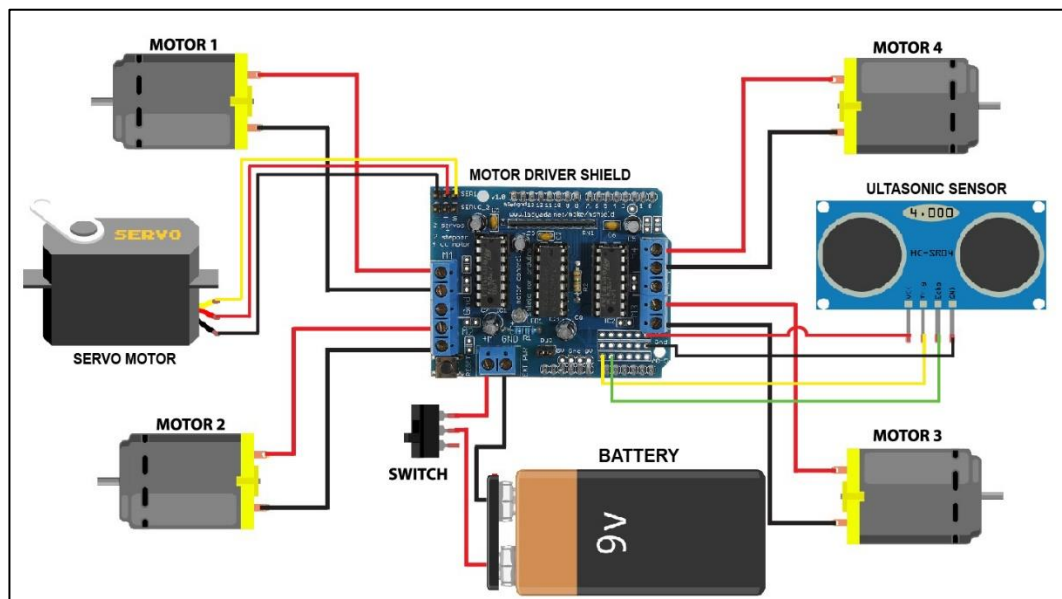
3.5 Power Supply

Baterai ion litium (biasa disebut Baterai Li-ion atau LIB) adalah salah satu anggota keluarga baterai isi ulang (rechargeable battery). Di dalam baterai ini, ion litium bergerak dari elektroda negatif ke elektroda positif saat dilepaskan, dan kembali saat diisi ulang. Baterai Li-ion memakai senyawa litium interkalasi sebagai bahan elektrodanya, berbeda dengan litium metalik yang dipakai di baterai litium non-isi ulang.

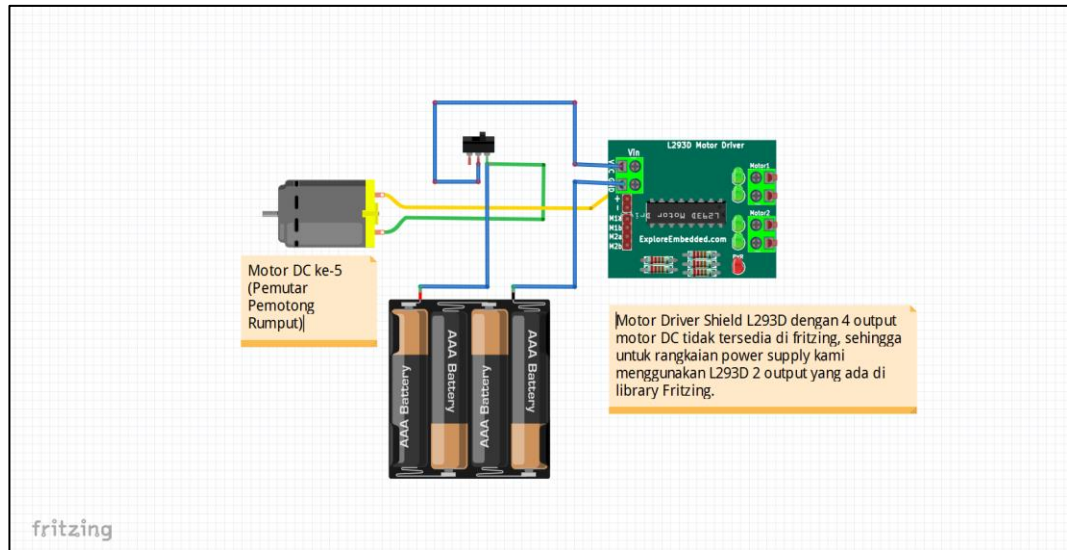


Gambar 10 : Baterai Li-Ion 3.7V

3.6 Wiring Komponen



Gambar 11 : Wiring Komponen Grass Cutter Robot (Part 1)

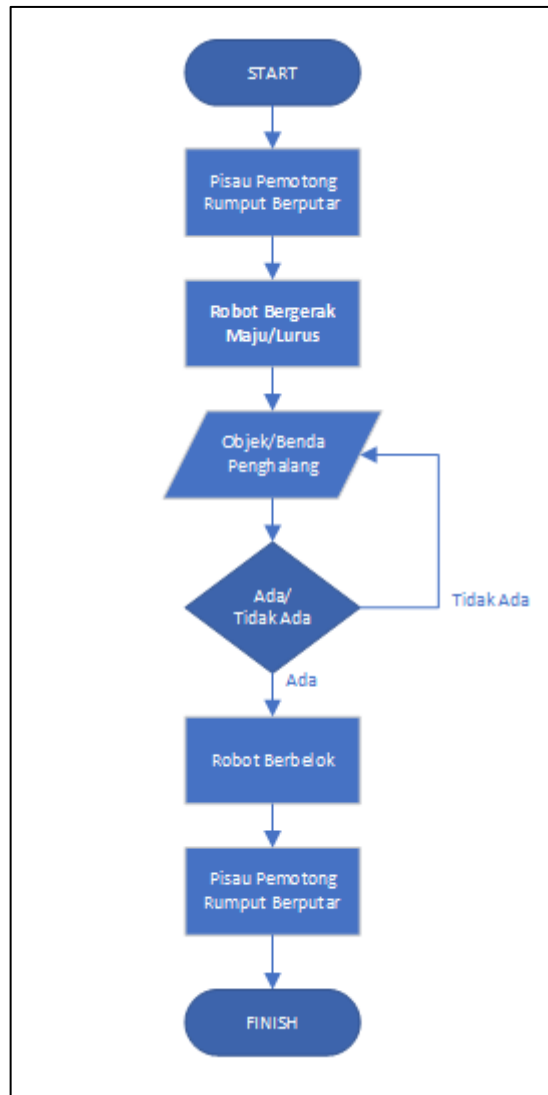


Gambar 12 : Wiring Komponen Grass Cutter Robot (Part 2)

BAB IV

Robot System

4.1 Flowchart



Gambar 13 : Flowchart Grass Cutter Robot

4.2 Source Code

```
#include <AFMotor.h>

#include <NewPing.h>

#include <Servo.h>
```

```

#define TRIG_PIN A0

#define ECHO_PIN A1

#define MAX_DISTANCE 200

#define MAX_SPEED 250 //max. speed motor DC

#define MAX_SPEED_OFFSET 20 //kenaikan kecepatan speed motor DC

NewPing sonar (TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

Servo motservo;

boolean goMaju=false; //keadaan awal sistem

int jarak = 100;

int kecepatan = 0;

void setup() {
  Serial.begin(115200);

  pinMode (TRIG_PIN, OUTPUT); //trigger sebagai output
  pinMode (ECHO_PIN, INPUT); //echo sebagai input

  motservo.attach(10);

  motservo.write(115);

  delay(2000);

```

```

    jarak = readPing();
    delay(100);
    jarak = readPing();
    delay(100);
    jarak = readPing();
    delay(100);
    jarak = readPing();
    delay(100);
}

```

```

void loop() {
    int jarakR = 0;
    int jarakL = 0;
    delay(40);

    if(jarak<=15)
    {
        stop();
        delay(100);
        gerakMundur();
        delay(300);
        stop();
        delay(200);
        jarakR = hadapKanan();
        Serial.print("Jarak R : ");
    }
}

```

```

Serial.println(jarakR);

float jarakR = pulseIn(ECHO_PIN, HIGH);

delay(200);

jarakL = hadapKiri();

Serial.print("Jarak L : ");

Serial.println(jarakL);

float jarakL = pulseIn(ECHO_PIN, HIGH);

delay(200);


if(jarakR>=jarakL)
{
    belokKanan();

    stop();
}else
{
    belokKiri();

    stop();
}
}else
{
    gerakMaju();
}

jarak = readPing();
}

```



```
int hadapKanan()
{
    motservo.write(50);
    delay(500);
    int jarak = readPing();
    delay(100);
    motservo.write(115);
    return jarak;
}
```

```
int hadapKiri()
{
    motservo.write(170);
    delay(500);
    int jarak = readPing();
    delay(100);
    motservo.write(115);
    return jarak;
    delay(100);
}
```

```
int readPing() {
    delay(70);
    int cm = sonar.ping_cm();
    if(cm==0)
```

```

{
    cm = 250;
}

return cm;
}

```

```

void stop() {
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
}

```

```

void gerakMaju() {

```

```

    if(goMaju)

```

```

    {
        goMaju=true;
        motor1.run(FORWARD);
        motor2.run(FORWARD);
        motor3.run(FORWARD);
        motor4.run(FORWARD);

```

```

        for (kecepatan = 0; kecepatan < MAX_SPEED; kecepatan +=2)

```

```

        {
            motor1.setSpeed(kecepatan);

```

```

    motor2.setSpeed(kecepatan);
    motor3.setSpeed(kecepatan);
    motor4.setSpeed(kecepatan);
    delay(5);
}
}
}

```

```

void gerakMundur() {
    goMaju=false;
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
    motor3.run(BACKWARD);
    motor4.run(BACKWARD);
    for (kecepatan = 0; kecepatan < MAX_SPEED; kecepatan +=3)
    {
        motor1.setSpeed(kecepatan);
        motor2.setSpeed(kecepatan);
        motor3.setSpeed(kecepatan);
        motor4.setSpeed(kecepatan);
        delay(5);
    }
}

```

```

void belokKanan() {

```

```
motor1.run(FORWARD);  
motor2.run(FORWARD);  
motor3.run(BACKWARD);  
motor4.run(BACKWARD);  
  
delay(500);  
  
motor1.run(FORWARD);  
motor2.run(FORWARD);  
motor3.run(FORWARD);  
motor4.run(FORWARD);  
  
}
```

```
void belokKiri() {  
  
    motor1.run(BACKWARD);  
    motor2.run(BACKWARD);  
    motor3.run(FORWARD);  
    motor4.run(FORWARD);  
  
    delay(500);  
  
    motor1.run(FORWARD);  
    motor2.run(FORWARD);  
    motor3.run(FORWARD);  
    motor4.run(FORWARD);  
  
}
```

BAB V

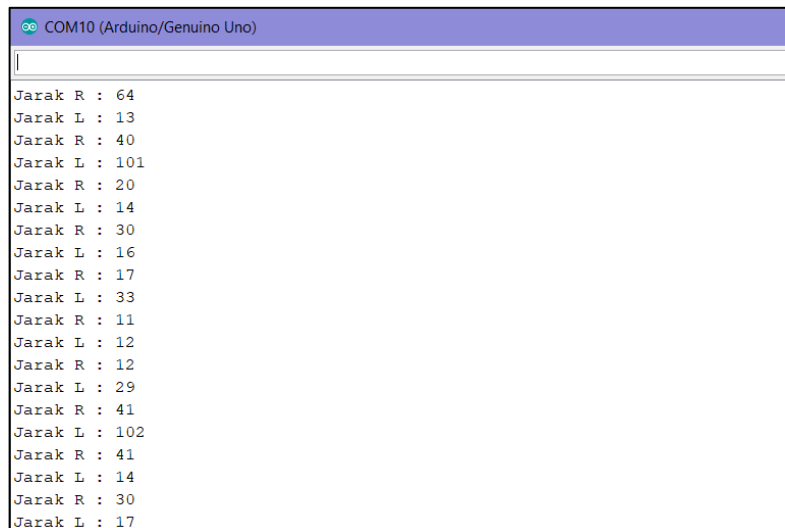
Analysis and Conclusion

5.1 Analisa

Analisa pada robot *Grass Cutter with Obstacle Avoider*, kami bagi kedalam sub-analisis per bagiannya sebagai berikut.

- Analisa Sensor Ultrasonic dan Motor Servo

Sensor yang kami gunakan yaitu sensor ultrasonic HC-SR04. Pada robot sensor sudah bekerja dengan baik, dapat menentukan jalur mana yang akan ditempuh robot dengan mengukur jarak menggunakan pantulan dari gelombang ultrasonic terhadap objek penghalang dan kemudian hasil pengukurannya dijadikan *feedback* untuk pemilihan jalur robot. Kekurangan sensor yang kami gunakan ini yaitu harus adanya objek penghalang yang dapat memantulkan gelombang ultrasonic dan letaknya tidak melebihi range dari sensor tersebut sehingga agak sulit digunakan di lingkungan terbuka yang memiliki sedikit objek. Apabila robot dijalankan di situasi yang memiliki objek sedikit dan letaknya berjauhan, saat robot berhenti karena ada objek penghalang maka robot akan memilih jalur yang memiliki jarak lebih pendek dari yang lain karena jarak yang lebih pendek tersebut dapat diukur oleh sensor ultrasonic dan jarak yang lebih jauh tidak dapat diukur oleh sensor karena diluar range pantulan gelombang ultrasonic yang dipancarkan.



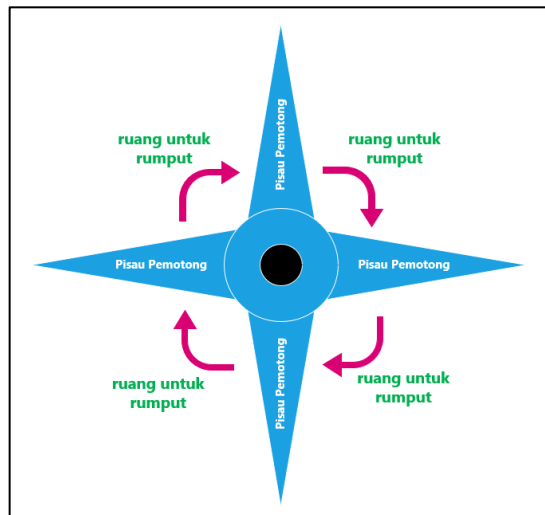
```
COM10 (Arduino/Genuino Uno)
Jarak R : 64
Jarak L : 13
Jarak R : 40
Jarak L : 101
Jarak R : 20
Jarak L : 14
Jarak R : 30
Jarak L : 16
Jarak R : 17
Jarak L : 33
Jarak R : 11
Jarak L : 12
Jarak R : 12
Jarak L : 29
Jarak R : 41
Jarak L : 102
Jarak R : 41
Jarak L : 14
Jarak R : 30
Jarak L : 17
```

Gambar 14 : Hasil Pengukuran Jarak Sensor Ultrasonic

Pada gambar diatas terdapat hasil pengukuran jarak dari sensor yang digunakan. Jarak tersebut dibagi 2 yaitu Jarak R : Jarak di kanan dan Jarak L : Jarak di kiri. Perbedaan jarak ini tak lepas dari peran motor servo yang berfungsi menghadapkan sensor ke kiri dan ke kanan sehingga didapat dua jarak tersebut (R dan L).

- Analisa Pisau Pemotong Rumput

Untuk pisau pemotong rumput kami menggunakan dua mata pisau *cutter* yang disusun secara bersilangan sehingga memberi ruang pada rumput untuk dipotong oleh pisau. Hasil pemotongannya lumayan bagus walaupun tidak menutup kemungkinan adanya sebagian rumput yang tidak terpotong karena putaran pemotong yang kurang cepat.



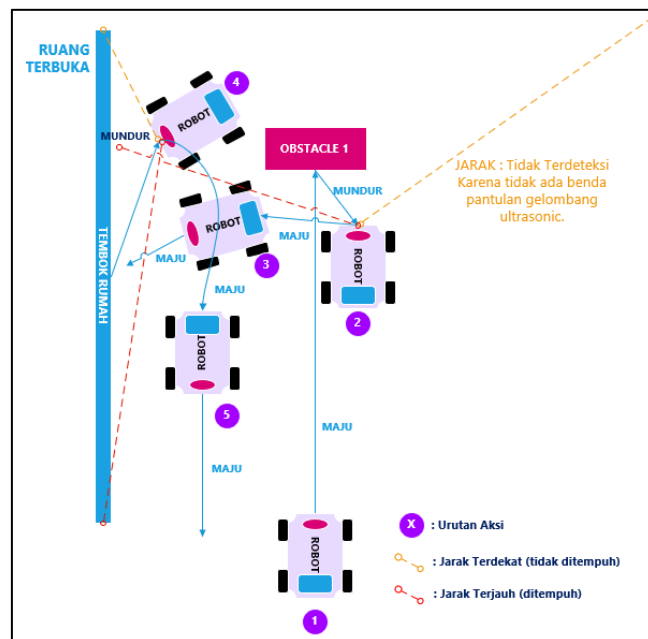
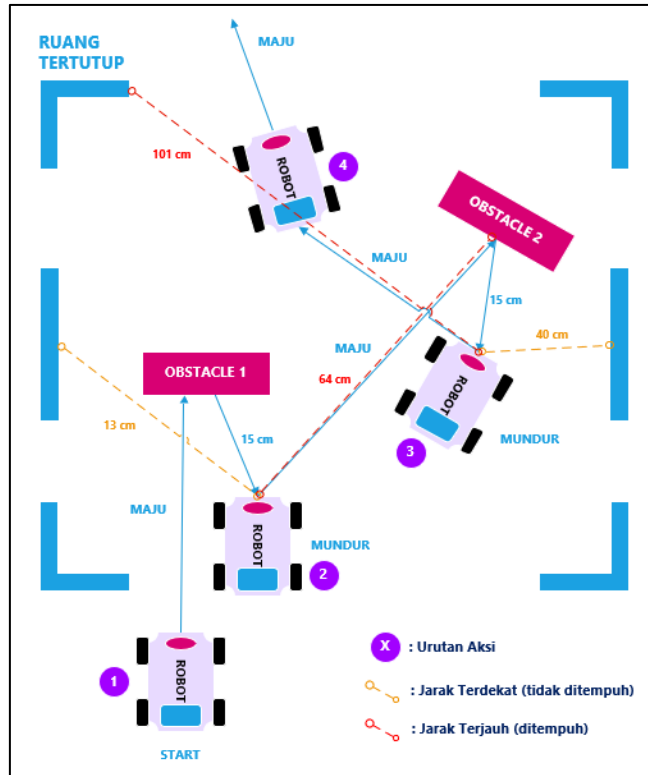
Gambar 15 : Pisau Pemotong Rumput

- Analisa *Power Supply*

Power supply yang kami gunakan awalnya 4 baterai AA 1.5V tetapi daya tersebut untuk mengoperasikan robot dengan baik, kemudian kami menggantinya dengan 3 buah baterai 9V dari *Energizer* yang kami alokasikan 2 buah untuk memutar 5 buah motor DC yang dihubungkan secara seri dan 1 buah untuk menyuplai daya ke *Arduino Uno* dan *Motor Driver L293D*. Hasilnya daya yang ada cukup untuk menyuplai kebutuhan daya robot tetapi dalam waktu yang singkat. Kendala utama kami di *power supply* yaitu susahnya mencari baterai yang cocok dan tidak bisa membeli baterai diluar pulau karena regulasi pengiriman.

- Analisa Robot Keseluruhan

Agar lebih mudah memahami cara robot bekerja, kami gambarkan ke dalam bentuk ilustrasi yang dibagi menjadi 2 yaitu di ruangan tertutup dan terbuka.



Gambar 16 : Ilustrasi Robot Saat Bekerja

Dalam ilustrasi diatas, kita dapat mengetahui alur kerja robot tersebut. Saat dinyalakan pisau pemotong akan berputar kemudian roda penggerak berjalan menyusuri jalur yang sudah ditentukan sampai nanti bertemu dengan objek penghalang, kemudian robot akan mundur dan melihat ke kiri dan kanan untuk menentukan jalur selanjutnya berdasarkan jarak yang diperoleh sensor. Perbedaan robot saat digunakan di ruangan terbuka dan tertutup dapat dilihat pada ilustrasi diatas. Untuk penggunaan di ruangan terbuka karena sesuai dengan fungsi robot ini yaitu pemotong rumput dibutuhkan lah objek penghalang yang letaknya tidak melebihi range pengukuran sensor, selain itu solusinya yaitu mengganti sensor yang ada dengan sensor yang memiliki range pantulan yang lebih jauh. Untuk jalur yang tidak rata, robot ini agak kesusahan untuk berbelok karena ban roda yang kami gunakan tidak terlalu bergerigi, solusinya bisa menggunakan roda jenis *belt* atau *chain* yang mampu berjalan dengan baik di lintasan tidak rata.

5.2 Kesimpulan

Kesimpulannya robot yang kami buat yaitu Grass Cutter with Obstacle Avoider dapat bekerja dengan baik di lintasan rata dan cukup baik di lintasan yang tidak rata. Selain itu fungsi utama dari robot ini yaitu pemotong rumput sudah berfungsi dengan baik karena dapat memotong rumput bahkan dedaunan yang memiliki tulang daun yang lebih kuat dari rumput. Kendala yang ada pada robot ini adalah *power supply* yang cepat sekali habis dayanya dan roda penggerak yang kesulitan berbelok di lintasan yang tidak rata. Solusinya yaitu untuk *power supply* diganti dengan baterai yang lebih awet dan *rechargeable* sedangkan untuk roda penggerak dapat diganti dengan roda tipe *belt* atau *chain* yang lebih cocok digunakan di lintasan tidak rata.

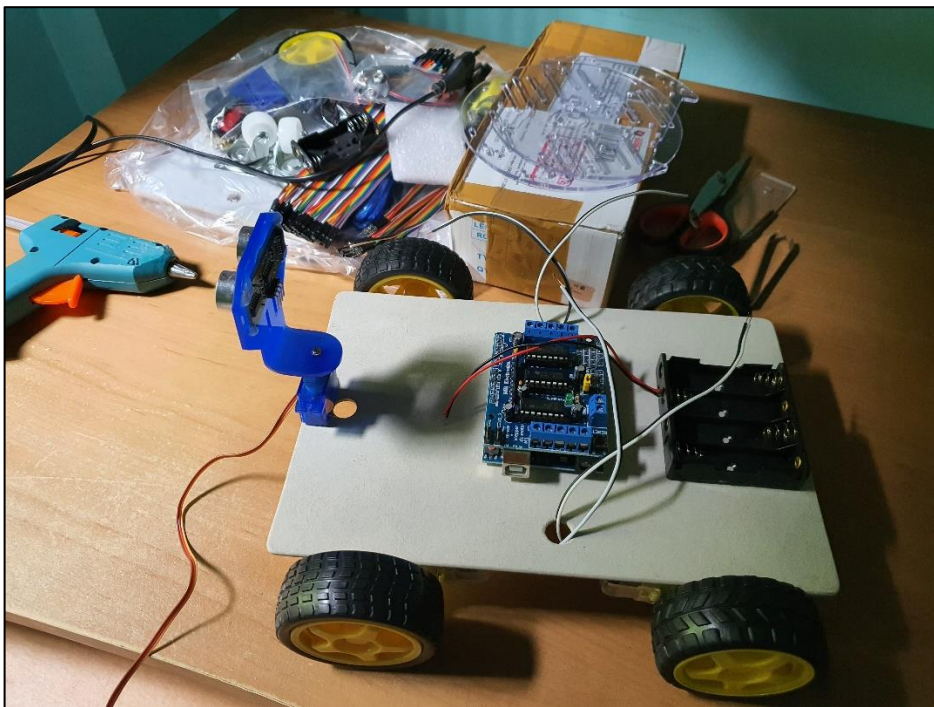
BAB VI

Documentation

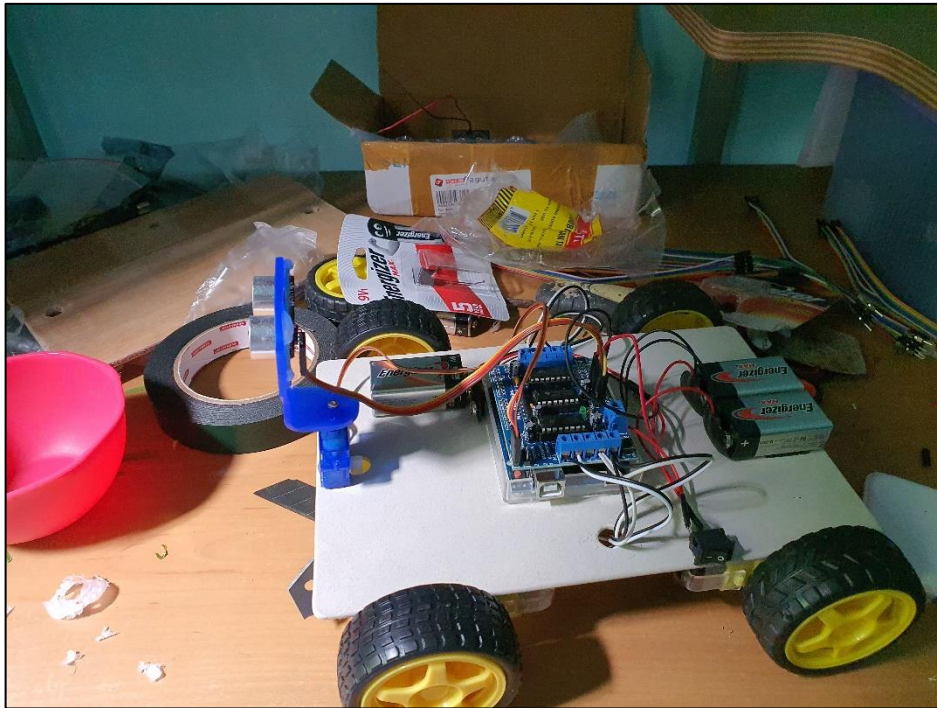
6.1 Foto



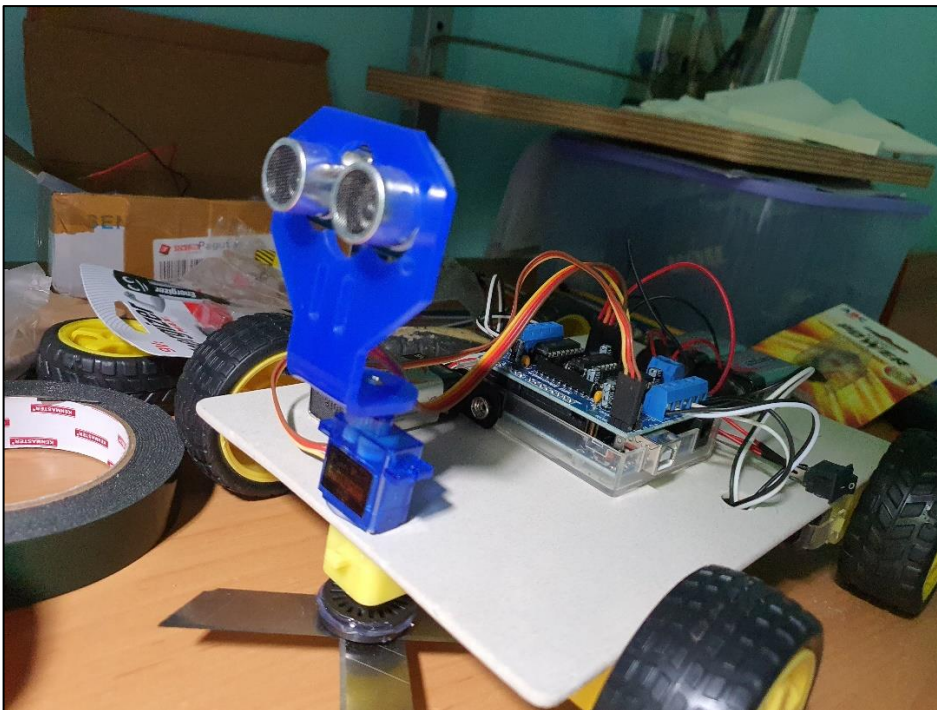
Gambar 17 : Awal Pengerjaan 1



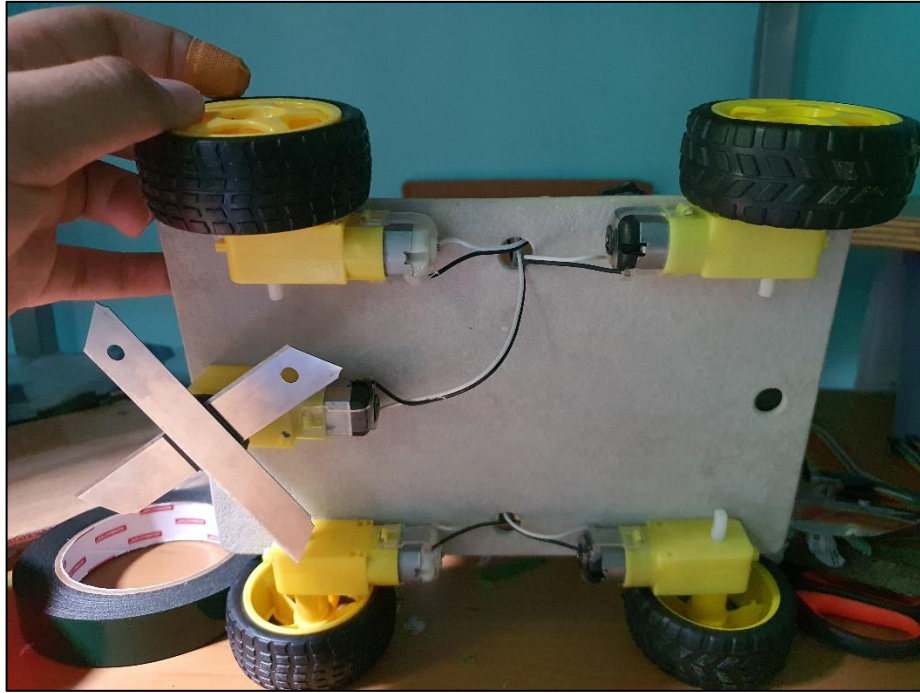
Gambar 18 : Awal Pengerjaan 2



Gambar 19 : Hasil Jadi 1 (Tampak Samping & Atas)



Gambar 20 : Hasil Jadi 2 (Tampak Depan)



Gambar 21 : Hasil Jadi 3 (Tampak Bawah)

6.2 Video

Link Video Demo :

https://drive.google.com/file/d/1DrPIWSCERZ7lhZGzlXV-xay4P6P7w_KV/view?usp=sharing