

## BAB III

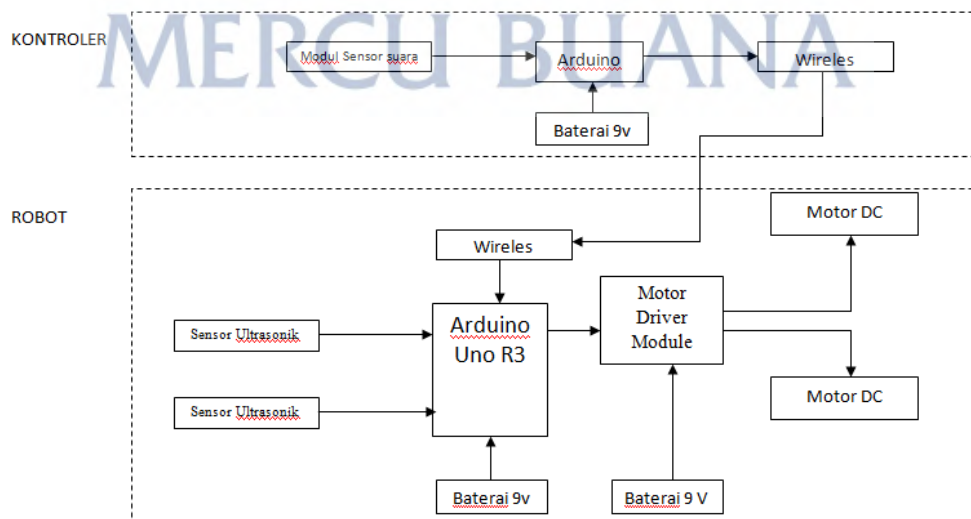
### PERANCANGAN DAN ANALISA SISTEM

Dalam bab 3 ini akan dibahas mengenai prinsip kerja dari rangkaian yang disusun untuk membuat Robot dengan sensor suara, dalam hal ini menggunakan suara sebagai komunikasi pada mobile robot dengan Arduino Uno juga Arduino Mini Pro Sebagai otaknya.

Sistem pada robot dibuat berdasarkan blok diagram yang telah di rencanakan. Pembahasan dari pembuatan robot mobil dengan membuat rancangan dari rangkaian mulai dari Motor Driver Module L298N /Dual Bridge,Arduino Mini Pro,Arduino Uno R3,Modul Sensor Suara dual output,Sensor Ultrasonik HC-SR04 yang di rangkai menjadi satu.

#### 3.1 Block Diagram Rangkaian.

Pada Pembuatan rangkaian diagram robot ini, Blok Diagram menggambarkan simulasi robot yang dikendalikan oleh suara dengan Arduino Pro mini sebagai prosessornya dan Arduino Uno sebagai prossesor dari modul ultrasonic yang mengamankan rangkaian dari benturan.



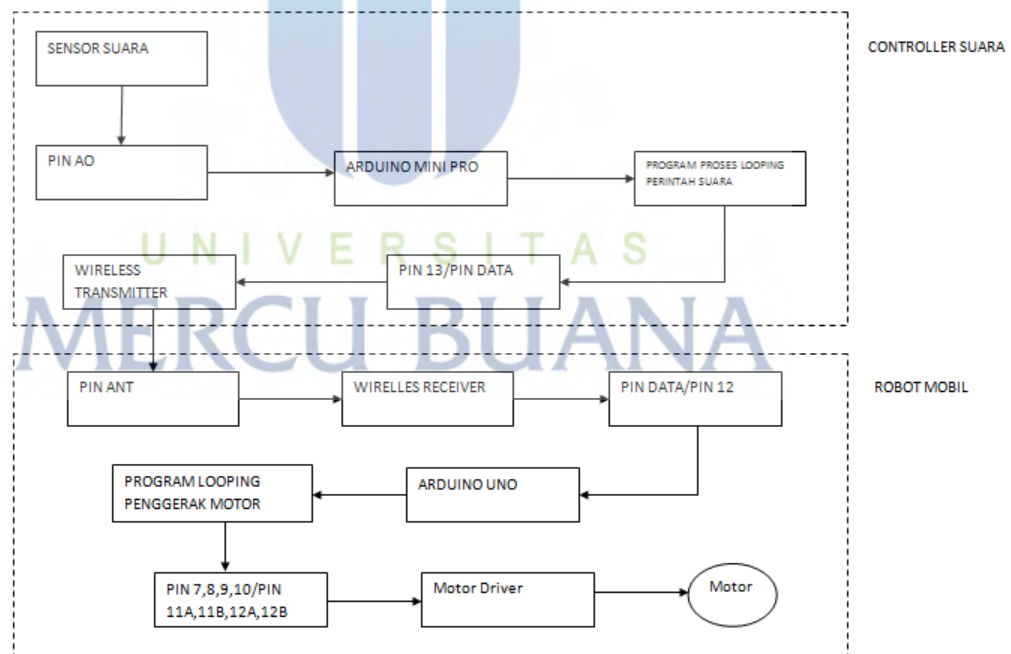
Gambar 3.1 Diagram Blok Rangkaian Kontrol Suara

Suara yang merupakan sinyal analog di ubah menjadi sinyal digital oleh sensor data tersebut di olah oleh arduino mini pro, setelah sinyal digital selesai di poses, data tersebut menuju wireless transmitter dan kemudian di kirimkan ke wireless receiver yang terpasang di robot mobil. Data yang diterima di teruskan ke arduino uno R3, data tersebut di proses dan hasilnya diteruskan ke motor driver yang digunakan untuk menggerkan motor DC. Untuk sensor ultrasonik kalkulasi jarak diubah menjadi data yang diteruskan ke arduino uno R3, data tersebut diolah di arduino R3 kemudian teruskan ke motor driver untuk menggerakan motor dc.

Fungsi dari diagram blok rangkaian kontrol *Suara* adalah :

- Arduino Uno R3  
Berfungsi untuk mengatur input dan output dari alat yang Ultrasonic modul. Arduino Uno R3 juga mengatur masukan power untuk Arduino Mini Pro dan juga memberikan perintah (menjembatani perintah arduino mini pro) ke Motor Driver L298N yang nantinya digunakan untuk menggerakan motor DC.
- Arduino Mini Pro  
Berfungsi untuk memproses sinyal digital yang telah di ubah sebelumnya dari sinyal analog oleh modul sensor suara. Mikrokontroler ini dihubungkan ke Arduino Uno R3 yang meneruskan ke Motor Driver L298N untuk menggerakan motor DC.
- Modul Sensor *Suara*  
Berfungsi sebagai penerima komunikasi dari Suara yang dikeluarkan oleh *User*
- Motor Driver L298N.  
Berfungsi sebagai pengatur tegangan dari arus yang masuk ke motor DC.

- Ultrasonic Sensor HC-SR04.  
Berfungsi sebagai sensor untuk pengaman terhadap benturan yang jaraknya ditentukan oleh Arduino Uno R3
- .Wireless Modul RF433 MHZ  
Merupakan modul wireless yang terdiri dari transmitter dan receiver untuk mengirimkan data.
- Baterai  
Baterai berfungsi sebagai sumber tegangan ke rangkaian Arduino Uno R3, Arduino Mini Pro dan Motor Driver L298N. Ketika tegangan 12 VDC baterai masuk ke Arduino Uno R3 maka tegangan akan terbagi menjadi tegangan *supply* 3.3 VDC untuk Arduino mini Pro dan 5 VDC untuk IC L298N.



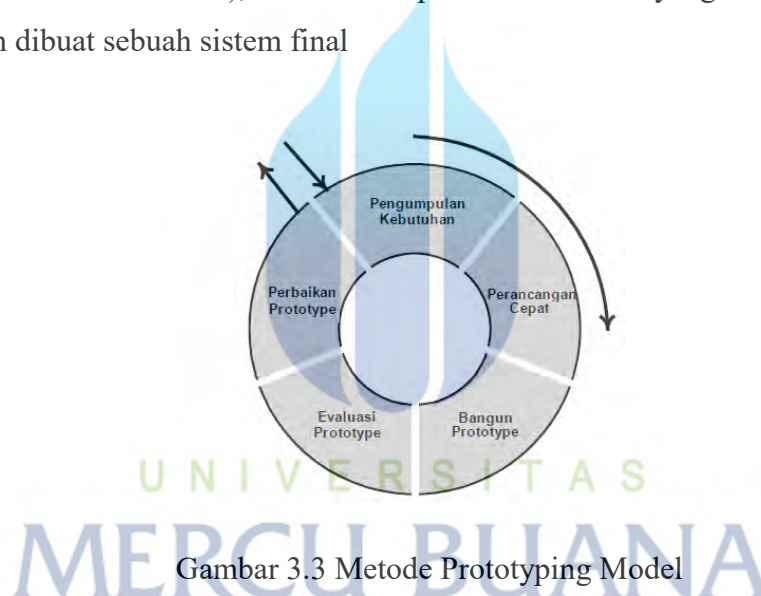
Gambar 3.2 Diagram Blok Perjalanan Data pada Komunikasi Robot dengan sensor suara

### 3.2 Perancangan Perangkat Keras

Pada perancangan robot ini menggunakan Arduino Uno R3 sebagai dasar utamanya, dan Arduino Pro Mini sebagai tambahan untuk memproses perintah suara.

Perancangan rangkaian ini dilakukan untuk menciptakan sebuah robot mobile yang dapat menangkap perintah suara yaitu maju,kanan,kiri,mundur dan stop juga dapat berhenti sebelum terjadi benturan.

Metode yang digunakan adalah metode prototyping,Membuat sebuah contoh prototipe untuk menunjukkan kebutuhan dan desain ke pemakai,Harus ada versi yang dapat dijalankan sebagai prototipe sebelum sistem dikembangkan (bisa berupa contoh sistem lain),Harus ada implementasi sistem yang dikembangkan sebelum dibuat sebuah sistem final

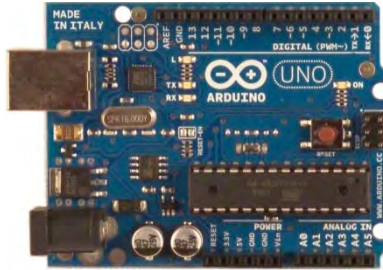


Gambar 3.3 Metode Prototyping Model

#### 3.2.1 Arduino Uno R3

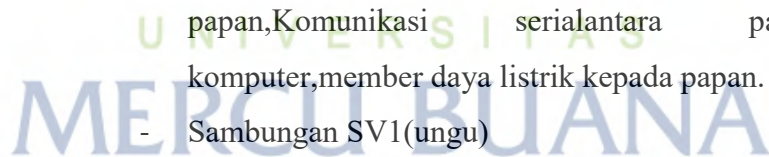
Komponen utama di dalam sebuah papan Arduino adalah sebuah mikrokontroler 8 bit dengan jenis ATmega yang dibuat oleh perusahaan Atmel corporation. Tiap jenis papan arduino memiliki tipe ATmega yang berbeda-beda tergantung dari spesifikasinya masing-masing. contoh

Arduino Uno menggunakan ATmega328 sedangkan Arduino Mega 2560 menggunakan ATmega2560



Gambar 3.4 Arduino Uno R3(Pengenalan Arduino,Feri Djuandi,2011)

Pada mikrokontroler ATmega328 memiliki Universal Asynchronous receiver/Transmitter (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti RS-232,RS-422 dan RS-485.Memiliki 2KB RAM pada memory kerja yang bersifat volatile (hilang saat daya dimatikan), digunakan oleh variabel-variabel didalam Program. Sedangkan 32KB flash memory bersifat non-volatile,digunakan untuk menyimpan program yang dimuat komputer.Selain Program,flash memory juga menyimpan bootloader.Bootloader adalah program inisiasi yang ukurannya kecil,dijalankan oleh CPU saat daya dihidupkan.Setelah bootloader selesai dijalankan,selanjutnya program di dalam RAM akan dieksekusi.1KB EEPROM bersifat non-volatile,digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan.Central Processing Unit (CPU),bagian dari mikrokontroller untuk menjalankan setiap intruksi dari program.Port Input /output,pin-pin untuk menerima data (input) digital atau analog dan mengeluarkan data (output) digital atau analog.



- Sambungan SV1(ungu)

- Sambungan SV1(ungu)

- Tombol Reset S1 (Biru)  
Untuk mereset papan sehingga program akan mulai dari awal.
- In-Circuit serial Programming (ICSP) (Hijau muda)  
Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, Tanpa melalui bootloader.
- IC1-Mikrokontroler ATmega  
Komponen utama dari papan Arduino, didalamnya terdapat CPU, ROM dan RAM.
- X1 – sumber daya eksternal (Merah)  
Jika akan di suplai dengan sumber daya eksternal, papan arduino dapat di berikan tegangan DC antara 9-12 V.
- 6 pin input analog (0-5)  
Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0-1023, dimana hal itu mewakili tegangan 0-5V.

### 3.2.2 Arduino Mini Pro

Arduino Pro Mini adalah papan pengembang purnarupa berbasis ATmega328P Mikrokontroler ini memiliki 14 digital I/O (di mana 6 pin di antaranya berkemampuan PWM / *Pulse Width Modulation*), 6 input analog, *on-board* resonator, dan dilengkapi dengan sebuah tombol reset. Pin header yang tersedia dapat dihubungkan dengan ISP *programmer* untuk mengunggah (*upload*) kode program dari komputer (Anda tetap dapat menggunakan Arduino IDE untuk menulis program).

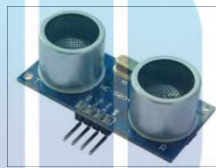


Gambar 3.6 Arduino Mini Pro

### 3.2.3 Ultrasonic Sensor.

Pembuatan mobil robot ini menggunakan ultrasonic sensor type HC-SR04 yang berfungsi sebagai modul sensor untuk identifikasi jarak yang akan diberikan kedalam mikrokontroler Arduino Mega 2560.

Fungsi utama pada Ultrasonic sensor HC SR04 adalah untuk mengetahui jarak benda pada Arduino Mega 2560, dengan batas jarak yang diberikan pada mikrokontroler 20 cm. Berikut adalah ultrasonic sensor HC-SR04 yang digunakan untuk Arduino Mega 2560 :



Gambar 3.7 Ultrasonic Sensor HC-SR04.

Ultrasonic modul ini bekerja dengan cara menghasilkan gelombang suara pada frekuensi tinggi, yang kemudian dipancarkan oleh bagian transmitter. Pantulan gelombang suara yang mengenai benda di depannya akan ditangkap oleh bagian receiver. Dengan mengetahui lamanya waktu antara dipancarkannya gelombang suara sampai ditangkap kembali, kita dapat menghitung jarak benda yang ada di depan modul tersebut. Kita mengetahui kecepatan suara adalah 340m/detik. Lamanya waktu tempuh gelombang suara dikalikan kecepatan suara, kemudian dibagi 2 akan menghasilkan jarak antara ultrasonic modul dengan benda didepannya.

### 3.2.4 Sound Sensor Module Dual Output.

Mikrokontroler Arduino Mini Pro menggunakan Sound Sensor Module sebagai input perintah yang digunakan untuk menggerakkan motor DC. Sound

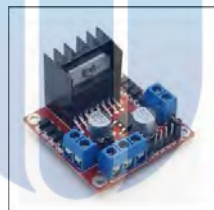
sensor module dual output merupakan module komunikasi nirkabel yang mengubah sinyal analog menjadi sinyal digital yang dapat diproses oleh mikrokontroler.



Gambar 3.8. Sound Sensor Module Dual Output

### 3.2.5 Motor Driver Shield.

Motor shield sebagai motor driver yang digunakan pada pembuatan mobil robot ini dengan type L298N adalah jenis IC driver motor yang dapat mengendalikan arah putaran robot mobil yaitu maju, mundur, kanan, kiri.

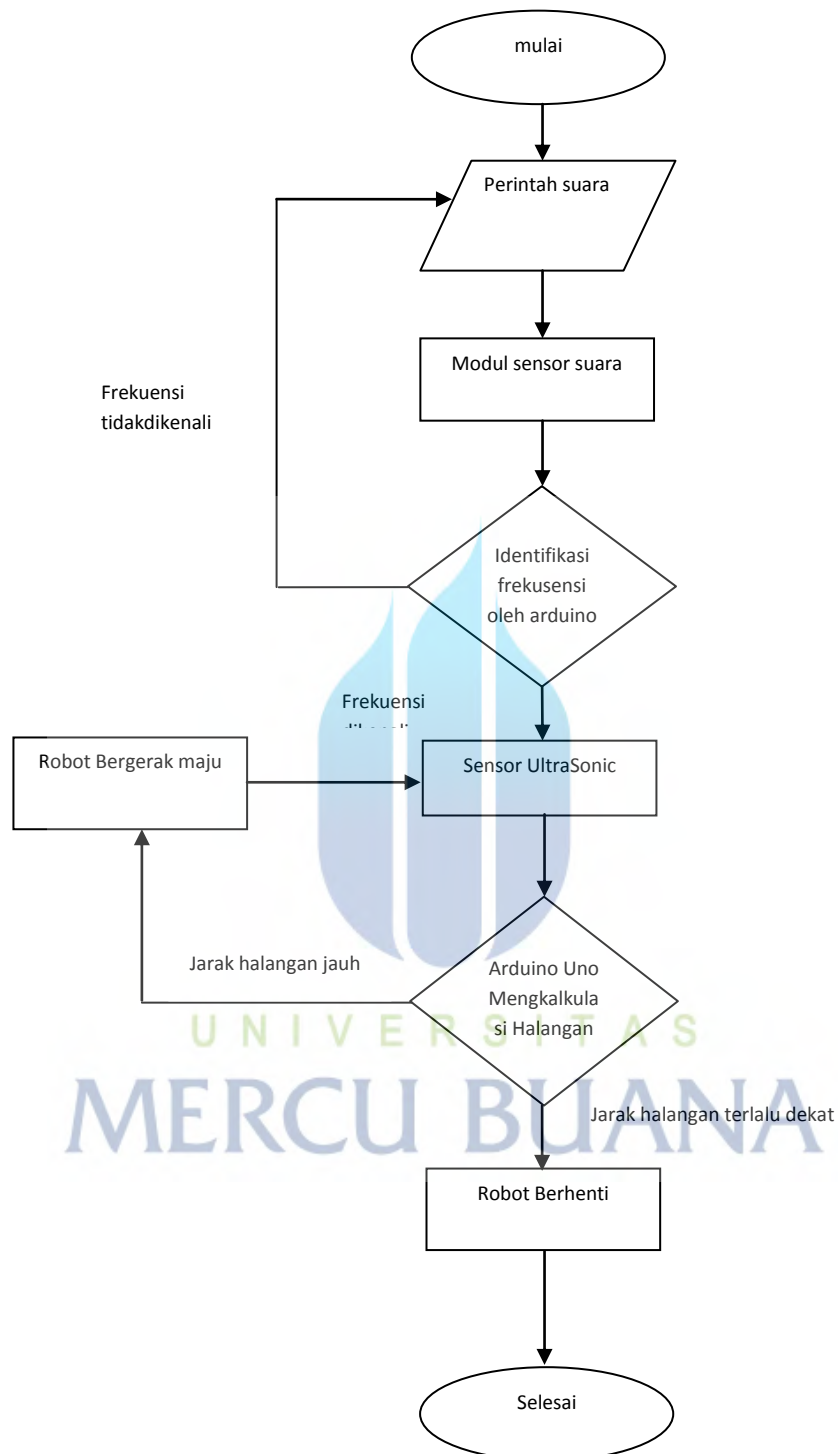


Gambar 3.9 Motor Shield L298N

## 3.3 Perancangan pembuatan Prototype Dengan Flowchart

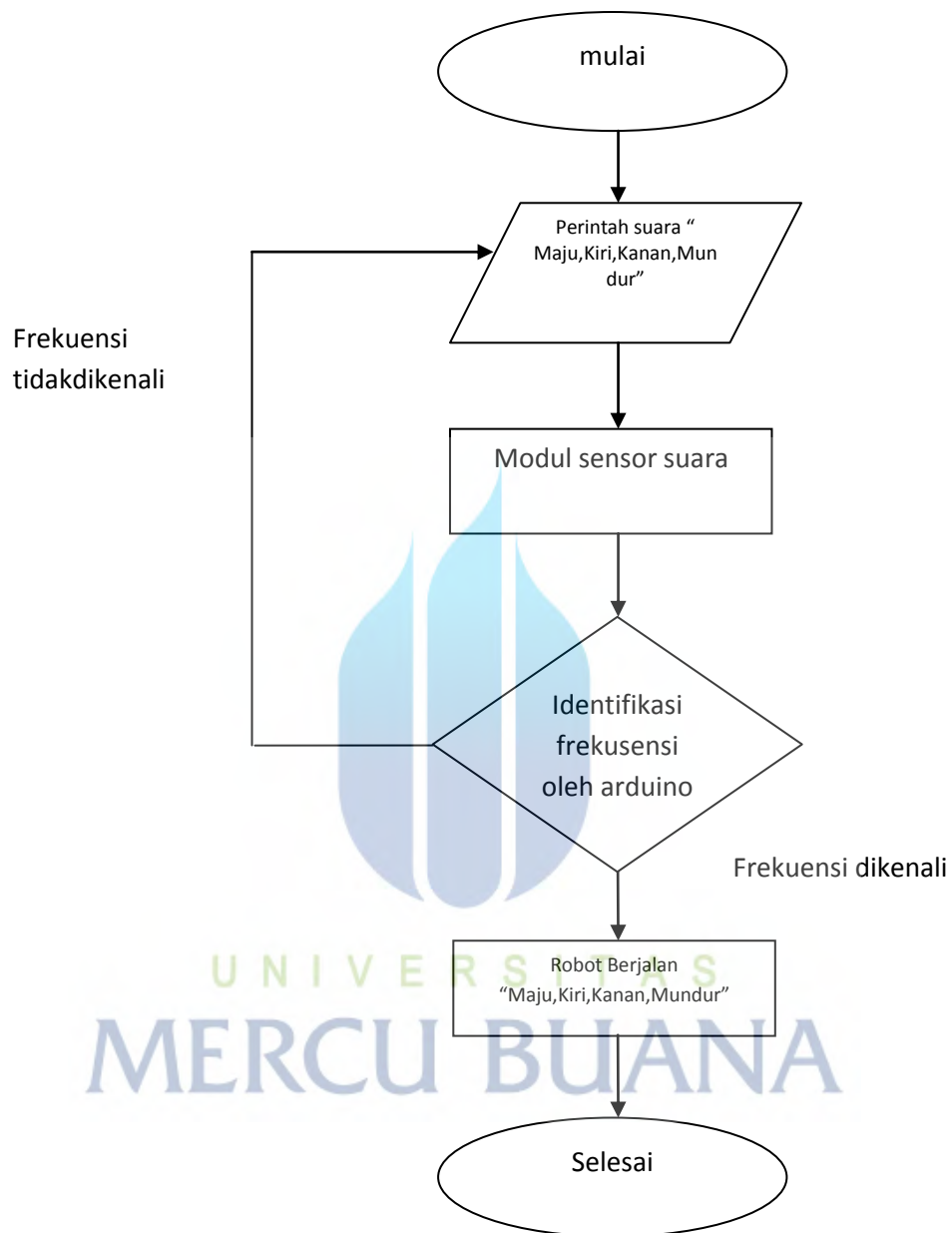
Pada perancangan ini dibuat beberapa flowchart yang menggambarkan pergerakan maju, mundur, kanan, kiri dan berhenti pada robot mobil yang digerakan oleh perintah suara

### 3.3.1 Pergerakan Secara Umum



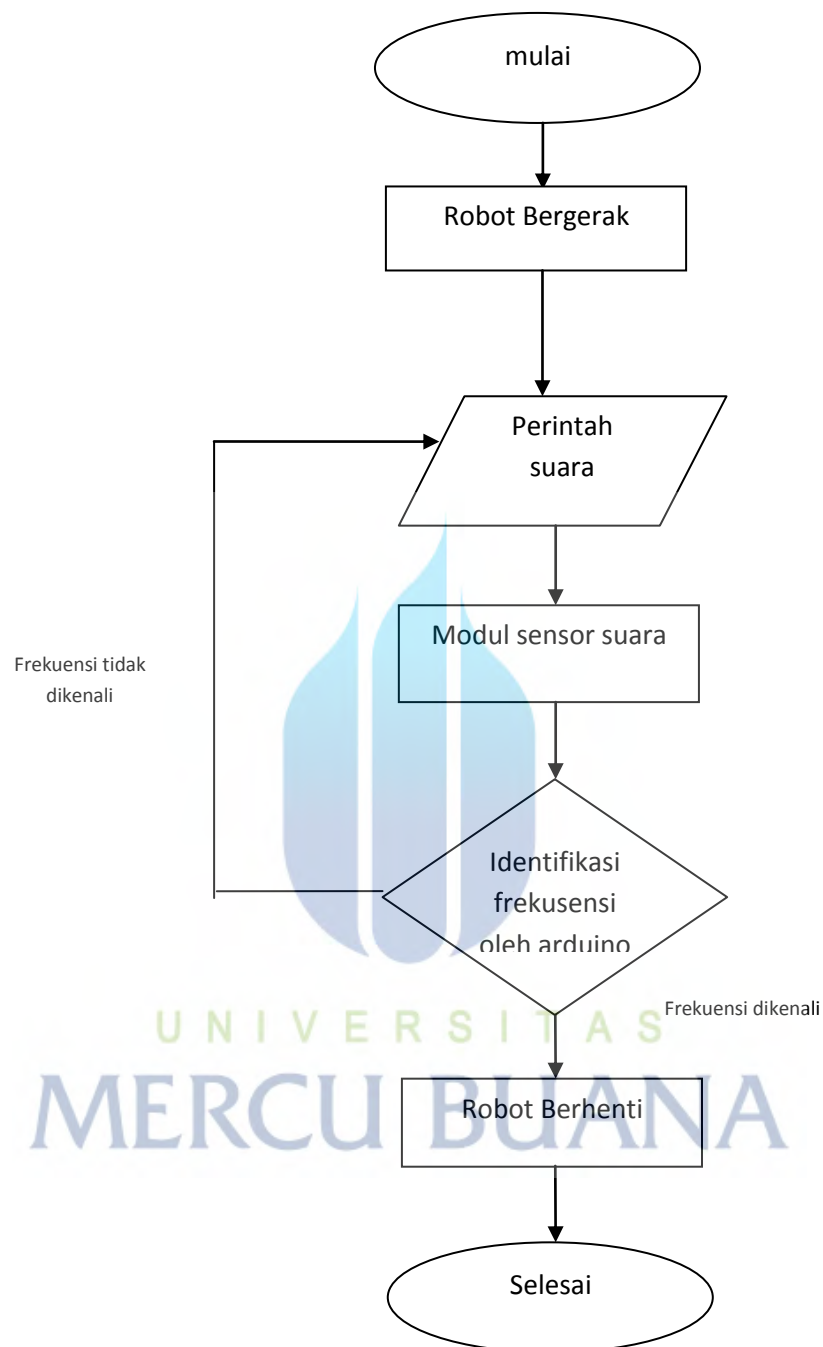
Gambar 3.10. Diagram Alur Pergerakan Robot Secara Umum

### 3.3.2 Pergerakan Maju,Kiri,Kanan,Mundur



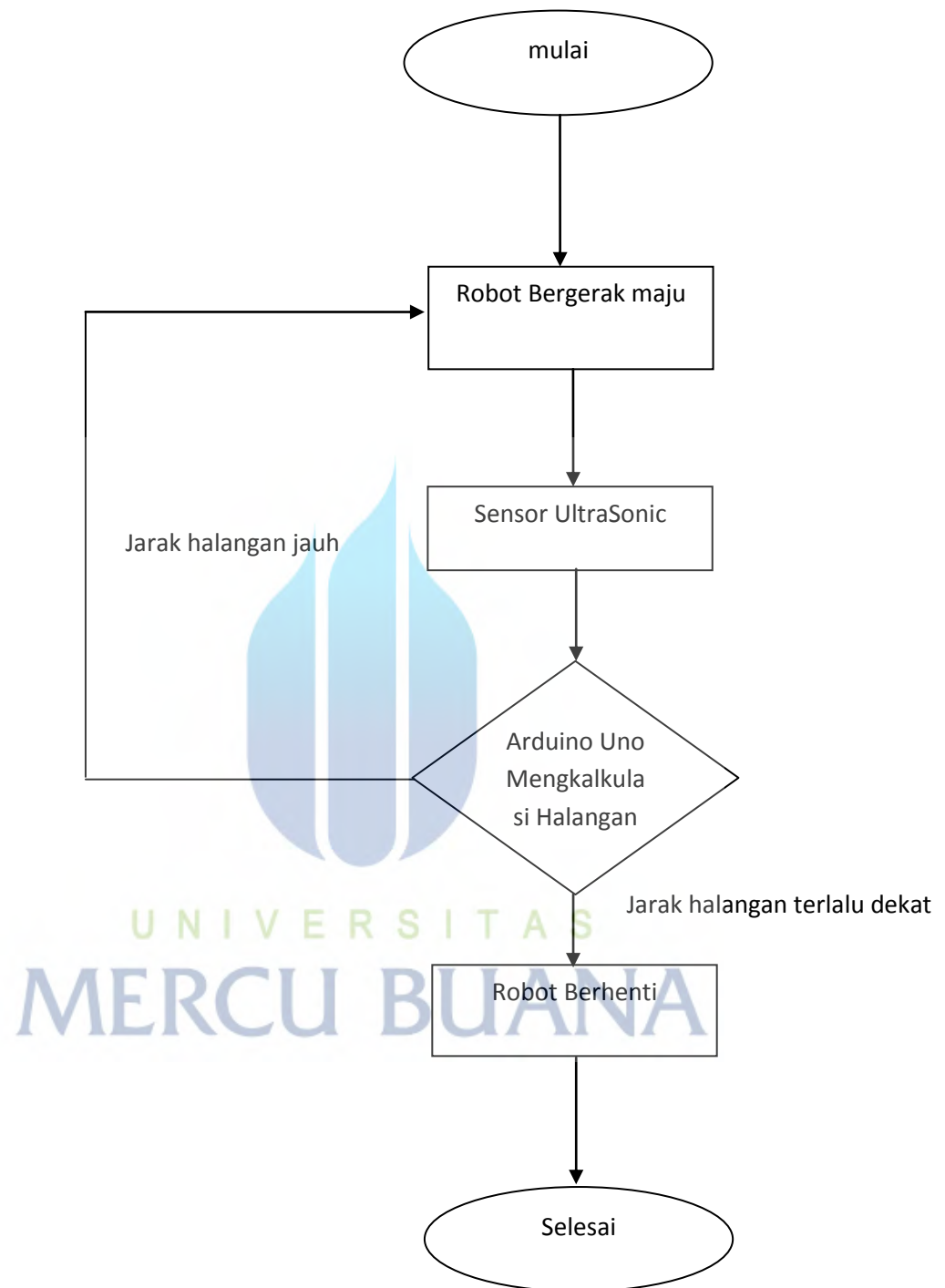
Gambar 3.11 Diagram Alur Pergerakan Robot untuk "Maju,Kiri,Kanan,Mundur"

## 3.3.3 Pergerakan Mundur



Gambar 3.12. Diagram Alur Pergerakan Robot untuk Berhenti

### 3.3.4 Robot Bereaksi dengan Halangan



Gambar 3.13. Diagram Alur Pergerakan Robot Dalam Mendeteksi Halangan.

### 3.3.5 Script Perintah Suara

Pada Arduino Mini telah di upload script untuk mengontrol robot dengan perintah suara. Script Tersebut Mengkalkulasi frekuensi setiap perintah yang akan di eksekusi oleh arduino mini

```
//433MHZ
#include <VirtualWire.h>
byte ledPin = 13;
char vbyte;

const int analogInPin = A0;
int sensorValue = 0;
int resultcounter;
int result;
int PinMaju = 2;
int PinMundur = 3;
int PinKiri = 4;
int PinKanan = 5;
int PinBerhenti = 6;
String Status;
char *controller;
int Counter;
void setup() {
  //433MHZ
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  vw_set_tx_pin(12);
  vw_setup(4000);
  pinMode(PinMaju, OUTPUT);
  pinMode(PinMundur, OUTPUT);
  pinMode(PinKiri, OUTPUT);
  pinMode(PinKanan, OUTPUT);
  pinMode(PinBerhenti, OUTPUT);
  Serial.begin(9600);
  Status = "Free";
  controller = "S";
  Counter = 0;
}

void loop() {
```

```

sensorValue = analogRead(analogInPin);
if (sensorValue > 250){ //sensitivitas

Serial.println("Count");
    resultcounter = 0;
    result += sensorValue;
}
resultcounter += 1;

if (resultcounter > 3000 and result > 0){
    Serial.print("sensor = " );
    Serial.println(result);

    if (result >= 10006 and result <= 15500) {
        controller = "F";
        Serial.println("Forward");
    }

    if (result >= 1550 and result <= 4990) {
        controller = "R";
        Serial.println("Right");
    }

    if (result >= 5550 and result <= 7932) {
        controller = "B";
        Serial.println("Back");
    }

    if (result >= 8120 and result <= 9947) {
        controller = "L";
        Serial.println("Left");
    }

    if (result >= 16000 and result <= 30000) {
        Serial.println("Stop");
        controller = "S";
    }

    result = 0;
    Serial.println("Free");
}

```

```

if(Counter > 10000){

    vw_send((uint8_t *)controller, strlen(controller));

    vw_wait_tx();

    Serial.println(controller);

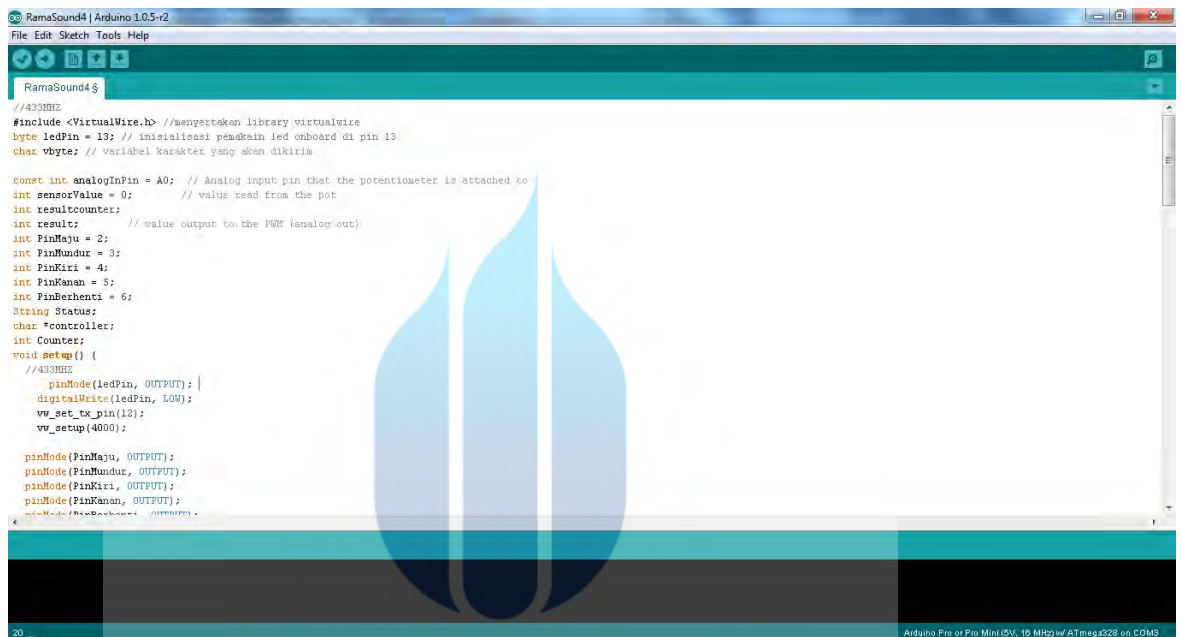
    Counter = 0;

}

Counter = Counter + 1;

}

```



Gambar 3.14. Arduino IDE Script Perintah Suara

### 3.3.6 Script Pengendali motor DC

Merupakan Script yang terdapat pada arduino uno dirobot mobil yang digunakan untuk mengendalikan motor dc yang nantinya akan mempengaruhi pergerakan robot mobil ini.

```

#define echoPin1 2

#define trigPin1 3

#define echoPin2 4

#define trigPin2 5

int maximumRange = 200;

int minimumRange = 0;

```

```
long duration1,distance1,duration2,distance2;
```

```
int BatasJarakMinimum;
```

```
String Status;
```

```
//motor driver L298N
```

```
int MotorKontrol1A = 7;
```

```
int MotorKontrol1B = 8;
```

```
int MotorKontrol2A = 9;
```

```
int MotorKontrol2B = 10;
```

```
//variabel umum
```

```
#include <VirtualWire.h>
```

```
void setup() {
```

```
    vw_set_rx_pin(12);
```

```
    vw_setup(4000);
```

```
    vw_rx_start();
```

```
//ultrasonic sensor
```

```
pinMode(trigPin1, OUTPUT);
```

```
pinMode(echoPin1, INPUT);
```

```
pinMode(trigPin2, OUTPUT);
```

```
pinMode(echoPin2, INPUT);
```

```
BatasJarakMinimum = 30;
```

```
pinMode(MotorKontrol1A, OUTPUT);
```

```
pinMode(MotorKontrol1B, OUTPUT);
```



```

pinMode(MotorKontrol2A, OUTPUT);

pinMode(MotorKontrol2B, OUTPUT);

//kontrol komunikasi

Serial.begin(9600);

Serial.println("Ready");

Status = "Stop";

}

void loop() {

  uint8_t buf[VW_MAX_MESSAGE_LEN]; //program bawaan 433mhz
  uint8_t buflen = VW_MAX_MESSAGE_LEN; //program bawaan 433mhz

  if (vw_get_message(buf, &buflen))
  {
    Serial.println(buf[0]);

    if(buf[0]=='F'){
      Serial.println(buf[0]);

      digitalWrite(MotorKontrol1A,LOW); // mengatur motor untuk maju
      digitalWrite(MotorKontrol1B,HIGH); // mengatur motor untuk maju
      digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk maju
      digitalWrite(MotorKontrol2B,HIGH); // mengatur motor untuk maju

      delay(500);

      Status = "Forward";

    }

    if(buf[0]=='R'){

      Serial.println(buf[0]);

      digitalWrite(MotorKontrol1A,LOW); // mengatur motor untuk belok kanan
      digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk belok kanan
      digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk belok kanan

```

```

digitalWrite(MotorKontrol2B,HIGH); // mengatur motor untuk belok kanan

delay(100);

}

if(buf[0]=='B'){

    Serial.println(buf[0]);

    digitalWrite(MotorKontrol1A,HIGH); // mengatur motor untuk maju mundur
    digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk maju mundur
    digitalWrite(MotorKontrol2A,HIGH); // mengatur motor untuk maju mundur
    digitalWrite(MotorKontrol2B,LOW); // mengatur motor untuk maju mundur
    delay(500); // menunda selama 1 detik untuk mengabaikan sensor
    Status = "Back";
}

if(buf[0]=='L'){
Serial.println(buf[0]);

    digitalWrite(MotorKontrol1A,HIGH); // mengatur motor untuk belok kiri
    digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk belok kiri
    digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk belok kiri
    digitalWrite(MotorKontrol2B,LOW); // mengatur motor untuk belok kiri
    delay(100);
}

if(buf[0]=='S'){
Serial.println(buf[0]);

    digitalWrite(MotorKontrol1A,LOW); // mengatur motor untuk berhenti semua
    digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk berhenti semua
    digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk berhenti semua
    digitalWrite(MotorKontrol2B,LOW); // mengatur motor untuk berhenti semua
    Status = "Stop";
}

```

```
}
```

```
if(Status == "Forward
```

```
digitalWrite(MotorKontrol1A,LOW); // mengatur motor untuk maju
```

```
digitalWrite(MotorKontrol1B,HIGH); // mengatur motor untuk maju
```

```
digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk maju
```

```
digitalWrite(MotorKontrol2B,HIGH); // mengatur motor untuk maju
```

```
}
```

```
if(Status == "Back"){
```

```
digitalWrite(MotorKontrol1A,HIGH); // mengatur motor untuk maju mundur
```

```
digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk maju mundur
```

```
digitalWrite(MotorKontrol2A,HIGH); // mengatur motor untuk maju mundur
```

```
digitalWrite(MotorKontrol2B,LOW); // mengatur motor untuk maju mundur
```

```
}
```

```
if(Status == "Stop
```

```
digitalWrite(MotorKontrol1A,LOW); // mengatur motor untuk berhenti semua
```

```
digitalWrite(MotorKontrol1B,LOW); // mengatur motor untuk berhenti semua
```

```
digitalWrite(MotorKontrol2A,LOW); // mengatur motor untuk berhenti semua
```

```
digitalWrite(MotorKontrol2B,LOW); // mengatur motor untuk berhenti semua
```

```
}
```

```
{
```

```
digitalWrite(trigPin1, LOW);
```

```
delayMicroseconds(2);
```

```
digitalWrite(trigPin1, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(trigPin1, LOW);
```

```
duration1 = pulseIn(echoPin1, HIGH);
```

```

distance1 = duration1/58.2;

if (distance1 >= maximumRange || distance1 <= minimumRange){
}

else {

if (distance1 < BatasJarakMinimum){ // jika jarak lebih kecil dari BatasJarakMinimum

digitalWrite(MotorKontrol1A,LOW); //mematikan kanal motor1

digitalWrite(MotorKontrol1B,LOW); //mematikan kanal motor1

digitalWrite(MotorKontrol2A,LOW); //mematikan kanal motor2

digitalWrite(MotorKontrol2B,LOW); //mematikan kanal motor2

Status = "Stop";

}

}

{

digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);

duration2 = pulseIn(echoPin2, HIGH);
distance2 = duration2/58.2;
if (distance2 >= maximumRange || distance2 <= minimumRange){
}

else {

if (distance2 < BatasJarakMinimum){ // jika jarak lebih kecil dari BatasJarakMinimum

digitalWrite(MotorKontrol1A,LOW); //mematikan kanal motor1

digitalWrite(MotorKontrol1B,LOW); //mematikan kanal motor1

digitalWrite(MotorKontrol2A,LOW); //mematikan kanal motor2

digitalWrite(MotorKontrol2B,LOW); //mematikan kanal motor2

Status = "Stop"; }

}

```