

Remote Control Car

1. Deskripsi

Remote Control Car adalah salah satu jenis robot beroda dengan kendali menggunakan sebuah remote. Transmisi data antara robot dan remote dapat menggunakan berbagai macam media baik melalui kabel, inframerah, Bluetooth, hingga wifi. Robot yang kita buat kali ini akan menggunakan Bluetooth sebagai media transmisi datanya.

2. Alat dan Bahan

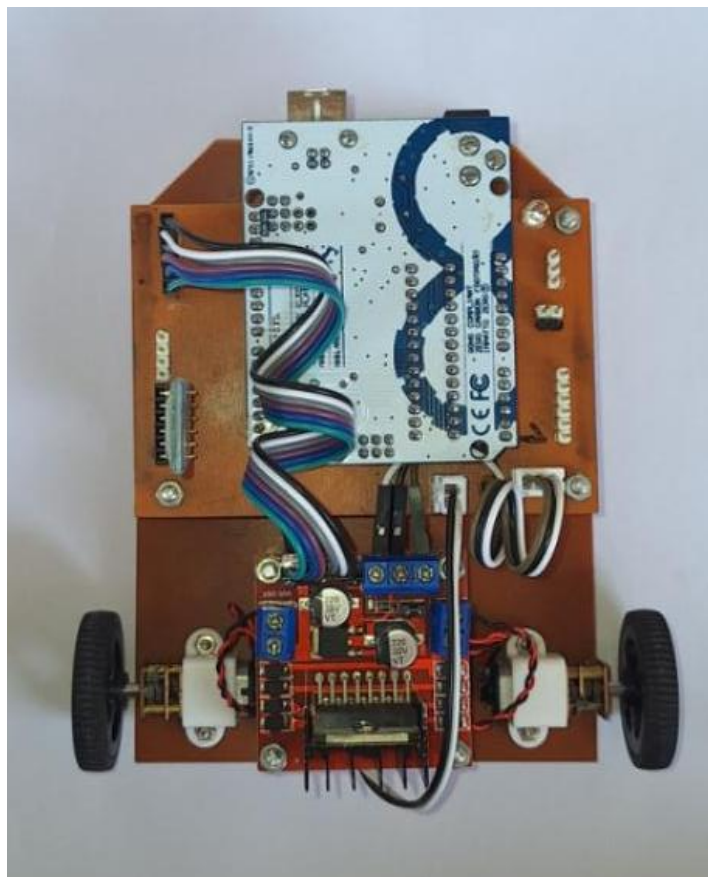
a. Remote Control

- i. Arduino
- ii. Modul Joystick
- iii. Modul Bluetooth HC-05
- iv. Jumper

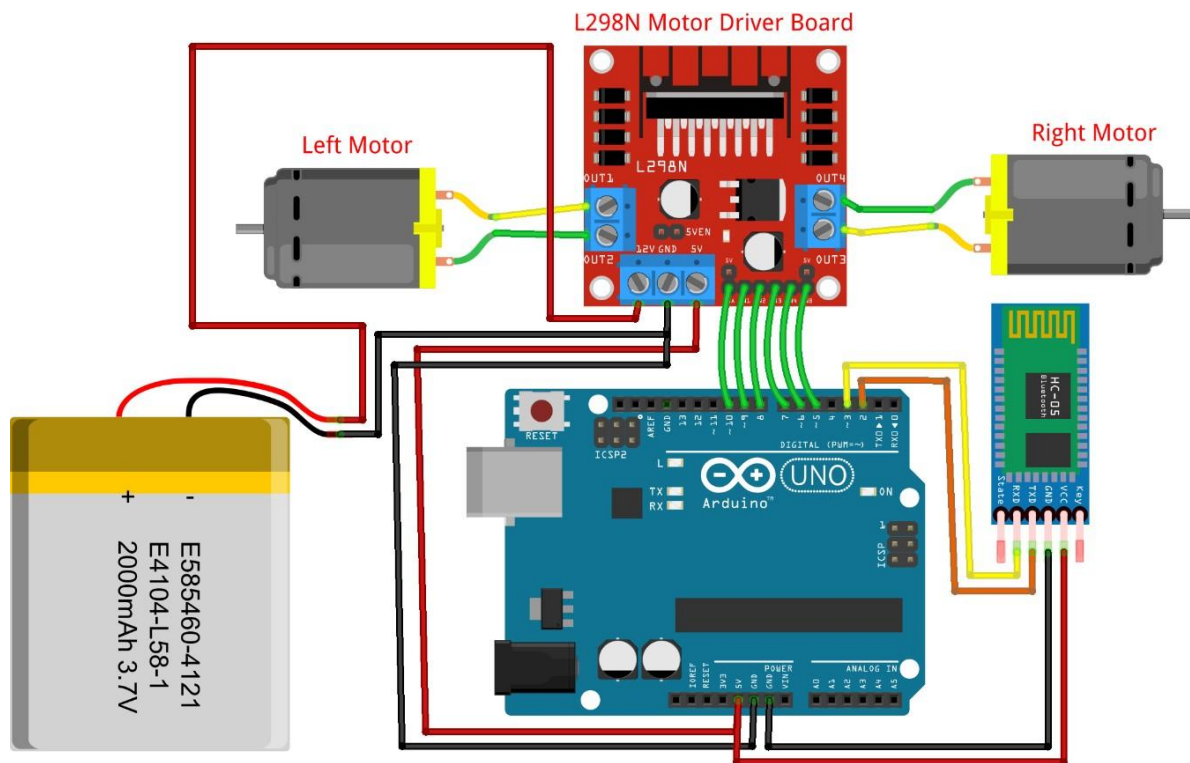
b. Robot RC/MikonkuBot

- i. Arduino
- ii. Modul Driver Motor L298N
- iii. Modul Bluetooth HC-05
- iv. Chassis RC Car
- v. Motor DC
- vi. Jumper

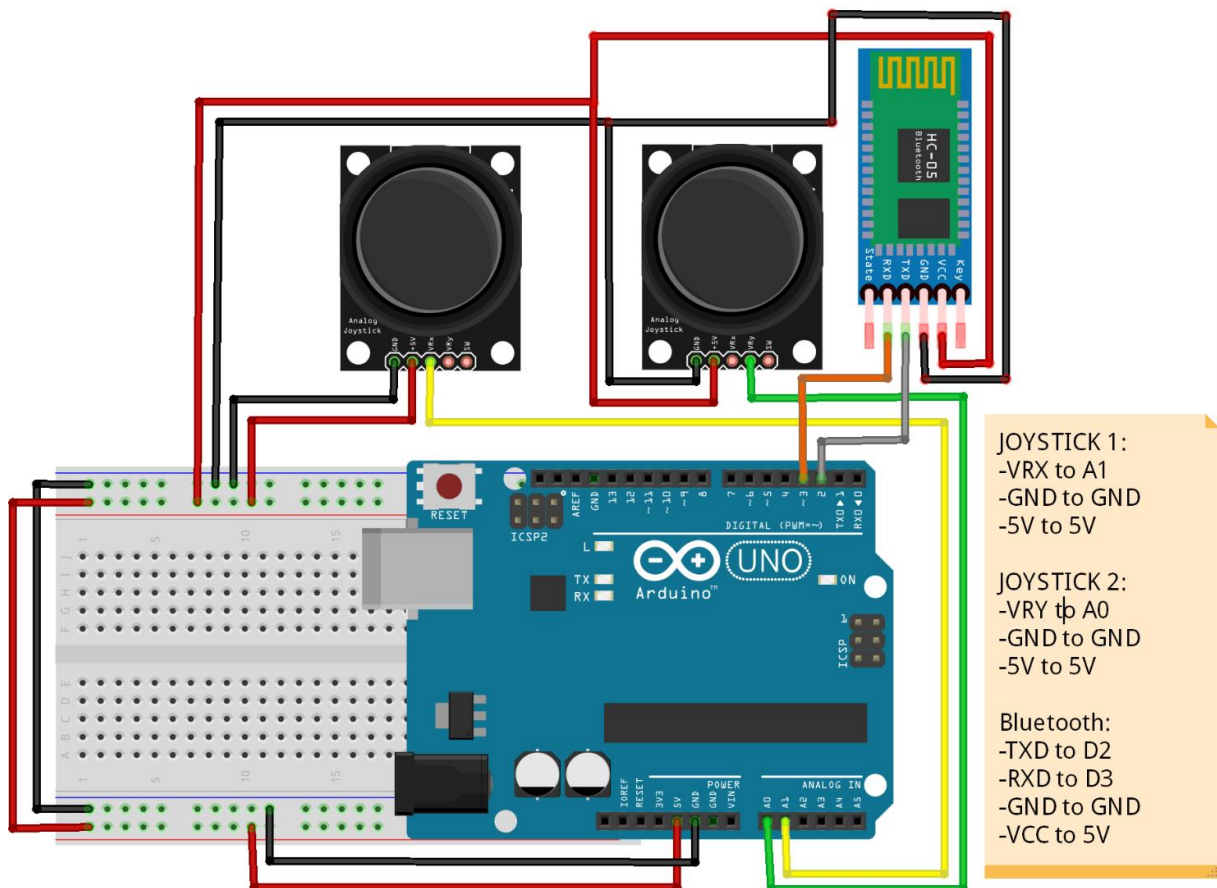
3. Rangkaian Elektronik



Gambar 1 MikonkuBot



Gambar 2 Skema elektronika untuk robot RC Car



4. Persiapan Pada Modul Bluetooth

Pertama-tama kita akan menentukan mana modul Bluetooth yang akan menjadi **Master** dan mana yang akan menjadi **Slave**. Modul Bluetooth yang akan dijadikan sebagai remote control kita akan set sebagai Master sedangkan modul Bluetooth pada robot akan menjadi Slave. Sebelum itu silahkan masukkan kodingan berikut kepada masing-masing Arduino terlebih dahulu:

****jika menggunakan mode master, maka Bluetooth akan mengirim pesan kesemua perangkat secara broadcast**

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(2,
3); char data;

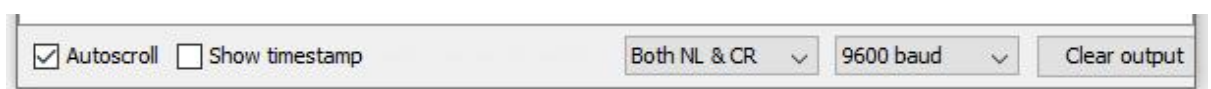
void setup()
{
    Serial.begin(9600);
    Serial.println("Enter AT
commands:");
    bluetooth.begin(38400);
}

void loop()
{
    // JIKA ADA PESAN DARI MODUL BLUETOOTH
    if (bluetooth.available()){
        // MAKA PESAN TERSEBUT AKAN DITAMPILKAN PADA SERIAL MONITOR
        Serial.print(bluetooth.readString());
    }

    // JIKA ADA COMMAND DARI USER
    if (Serial.available()){

        // MAKA COMMAND AKAN DIKIRIMKAN KE BLUETOOTH
        data =
        Serial.read();
        bluetooth.write(da
ta);
    }
}
```

Selanjutnya kita akan melakukan setting pada modul Bluetooth, silahkan buka serial monitor lalu gunakan mode **Both NL & CR**.



Tabel 1 AT Command Modul Bluetooth HC-05

No	Command	Response	Keterangan
1	AT	OK	Tes koneksi
2	AT+NAME?	+NAME:<name bluetooth>	Untuk melihat nama Bluetooth saat ini
3	AT+NAME=<name>	OK	<name> diisi dengan nama yang diinginkan
4	AT+ADDR?	+ADDR:<address>	Cek MAC Address, contoh 98D3:A1:F5BCE1
5	AT+CMODE?	+CMODE:<mode>	Cek mode koneksi ----- CMODE: 0 mode slave 1 mode master
6	AT+CMODE=<mode>	OK	SET mode koneksi
7	AT+BIND?	+BIND:<address>	Cek address bluetooth yang terkoneksi
8	AT+BIND= <address>	OK	Set address yang akan dikoneksikan contoh FCA8,9A,58D5

a. Setting Bluetooth As Slave

AT

AT+NAME=<isikan nama>

AT+ADDR?

AT+CMODE=0

b. Setting Bluetooth As Master

AT

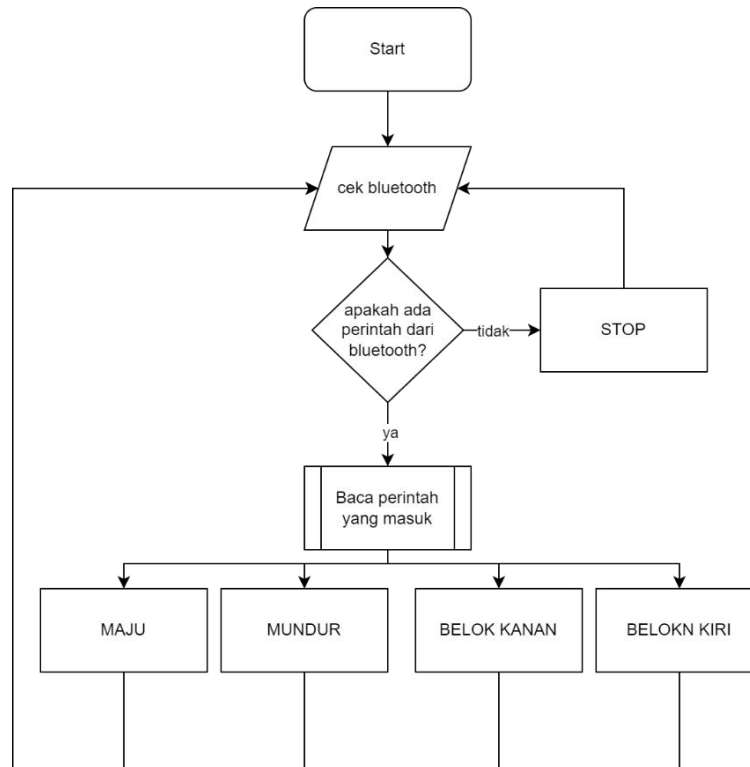
AT+NAME=<isikan

nama>AT+ADDR?

AT+CMODE=0

AT+BIND=<masukan mac address>

5. Robot Car



Gambar 3 Algoritma RC Car

File robot_rc.ino

```
#include <SoftwareSerial.h>
#include <MikonkuBot.h>

// MENENTUKAN INSTANCE UNTUK BLUETOOTH
SoftwareSerial bluetooth(2, 3);

// Define the pins for your MikonkuBot instance
const int ENA_PIN = 11;
const int IN1_PIN = 10;
const int IN2_PIN = 9;
const int IN3_PIN = 8;
const int IN4_PIN = 7;
const int ENB_PIN = 6;
const int LED_PIN = 13;

// VARIABLE UNTUK MENAMPUNG DATA DARI REMOTE CONTROL
char data;

// Create an instance of MikonkuBot
MikonkuBot myBot(ENA_PIN, IN1_PIN, IN2_PIN, IN3_PIN, IN4_PIN, ENB_PIN);

void setup() {
  bluetooth.begin(9600);
  Serial.begin(9600);
  Serial.println("Robot Started");

  // Initialize the MikonkuBot
  myBot.begin();
  myBot.setSpeed(200,200);
}

void loop() {

  // CEK APAKAH ADA SIGNAL YANG DIKIRIM OLEH TRANSMITTER/REMOTE
  if (bluetooth.available() > 0) {

    // JIKA ADA SIGNAL DARI TRANSMITTER/REMOTE BACA DATA YANG DIKIRIM
    data = bluetooth.read();

    // CETAK DATA YANG DITERIMA KE SERIAL MONITOR
    Serial.println(data);

    // KONTROL SETIAP DATA
    if (data == 'F') {
      myBot.light(LED_PIN,HIGH);
      myBot.goForward(50);
    } else if (data == 'B') {
      myBot.light(LED_PIN,HIGH);
```

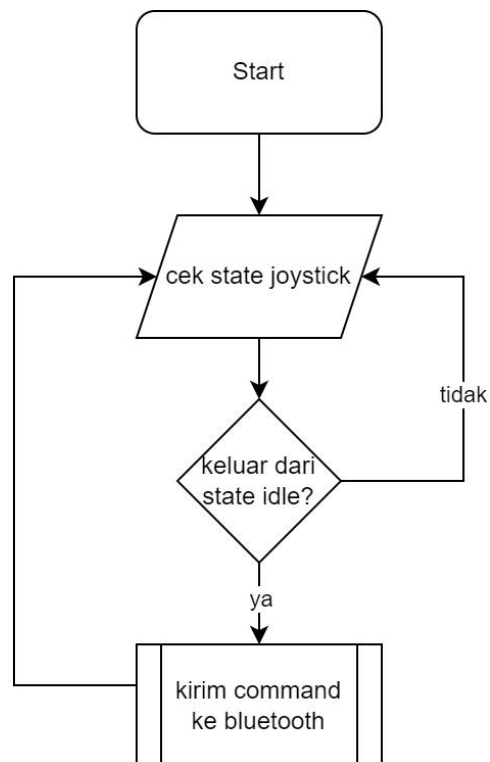
```

    myBot.goBackward(50);
  } else if (data == 'R') {
    myBot.light(LED_PIN,HIGH);
    myBot.goTurnRight(50);
  } else if (data == 'L') {
    myBot.light(LED_PIN,HIGH);
    myBot.goTurnLeft(50);
  } else {
    myBot.light(LED_PIN,LOW);
    myBot.stop();
  }
}

// MEMBERI JEDA 50 milisekon UNTUK ARDUINO
delay(50);
}

```

6. Remote Control



Gambar 4 Algoritma Remote Control



File remote_control.ino

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(2, 3);

// MENENTUKAN BATAS IDLE SUMBU X int BATAS_ATAS_X = 522 + 50;
int BATAS_BAWAH_X = 522 - 50;

// MENENTUKAN BATAS IDLE SUMBU Y int BATAS_ATAS_Y = 508 + 50;
int BATAS_BAWAH_Y = 508 - 50;

// UNTUK MENAMPUNG PERINTAH
char data;

void setup() {
  bluetooth.begin(9600); Serial.begin(9600);
}

void loop() {

  char data = command();

  // JIKA DATA SELAIN 'S'
  if (data != 'S') {

    // MAKA KIRIMKAN DATA KE BLUETOOTH
    bluetooth.write(data); Serial.println(data);
  }

  delay(50);
}
```

File command.ino

```
char command() {
  int Y = analogRead(A0); int X = analogRead(A1);

  if (Y < BATAS_BAWAH_Y) {
    return 'F';
  } else if (Y > BATAS_ATAS_Y) {
    return 'B';
  } else if (X < BATAS_BAWAH_X) {
    return 'L';
  } else if (X > BATAS_ATAS_X) {
    return 'R';
  } else {
    return 'S';
  }
}
```