

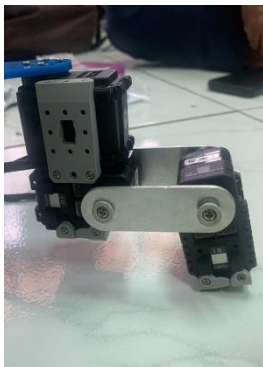
LAMPIRAN B.1 – DOKUMENTASI PROJEK LA
PERAKITAN MEKANIKAL DAN ELEKTRIKAL
Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



Menyiapkan frame dan toolkit



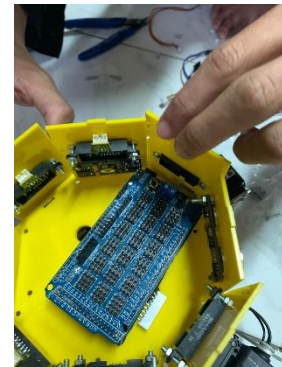
Menyambungkan Motor Servo



Merakit kaki pada frame robot



Merakit dan menyusun komponen



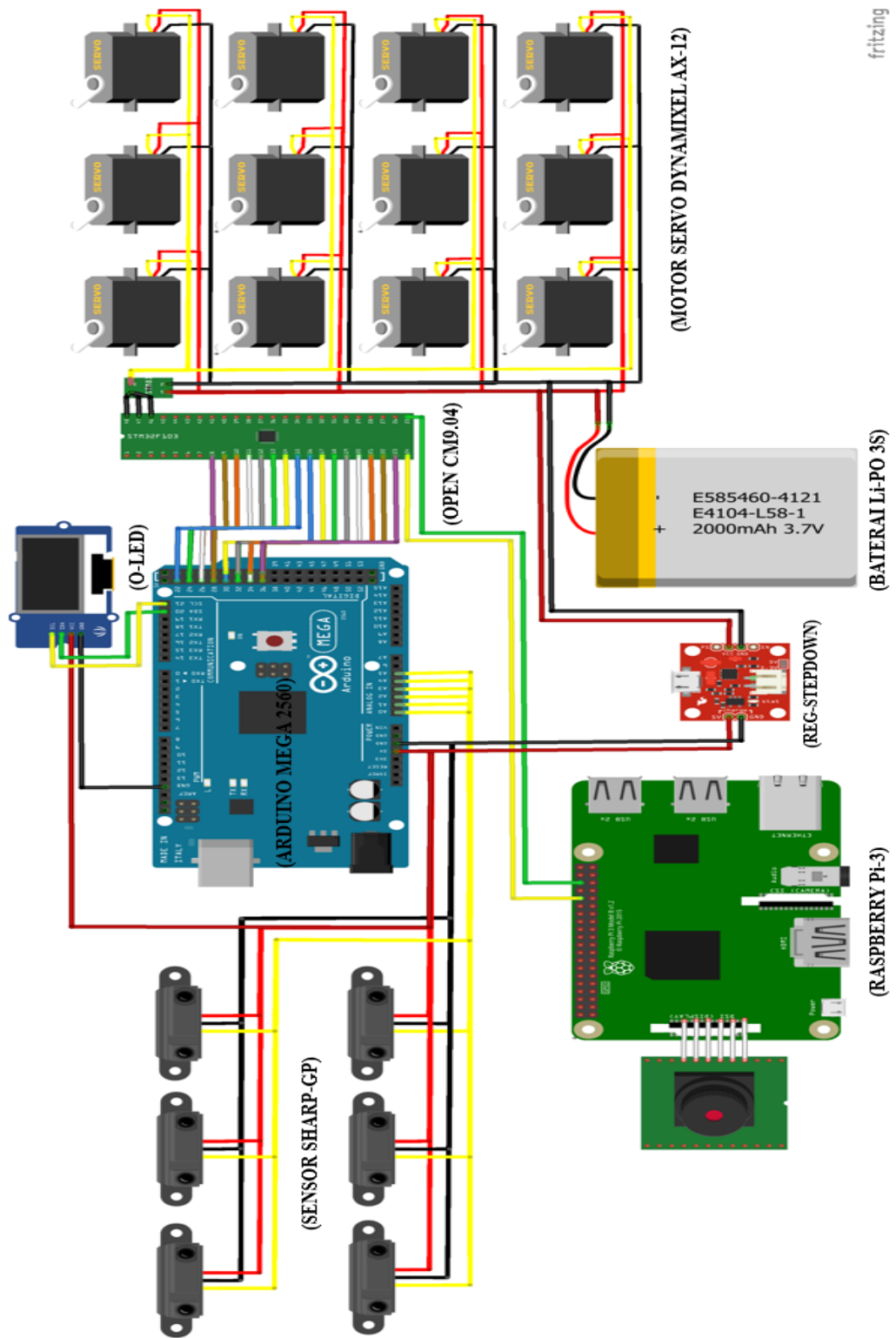
Konfigurasi mekanikal dan elektrikal



Tahapan perakitan mekanikal dan elektrikal merupakan tahapan lanjut setelah perancangan sistem dibuat;

Kegiatan ini dilakukan di
Laboratorium Mikrokontroler dan Mikroprosesor,
Gedung Bengkel/Lab Teknik Elektro,
Politeknik Negeri Sriwijaya, Februari – Juli 2022

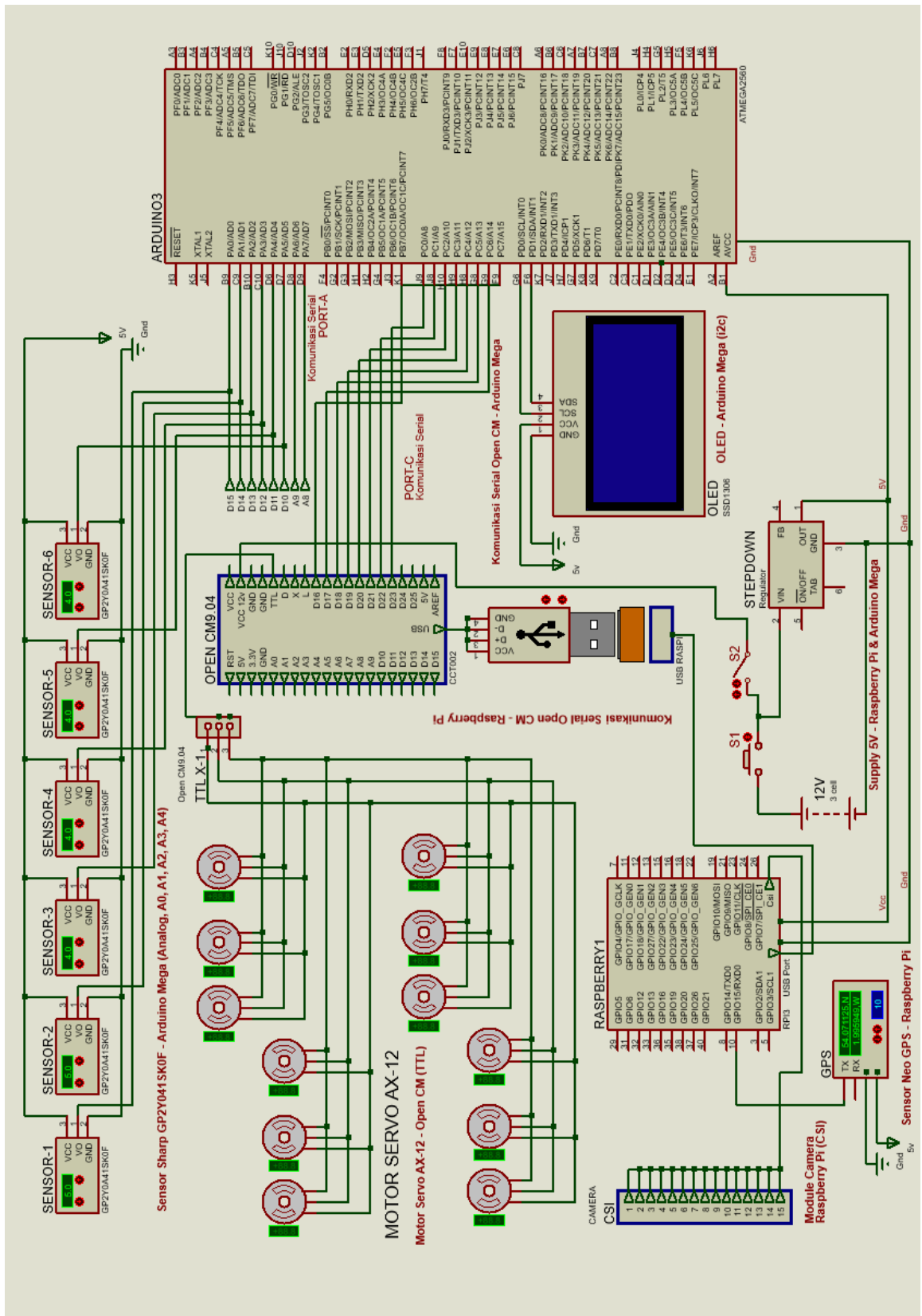
LAMPIRAN B.2 – DOKUMENTASI PROJEK LA
RANGKAIAN ELEKTRONIK ROBOT SAR QUADRUPED
Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



LAMPIRAN B.2 – DOKUMENTASI PROJEK LA

RANGKAIAN ELEKTRONIK ROBOT SAR QUADRUPED

Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



LAMPIRAN B.3 – DOKUMENTASI PROJEK LA
PENGUKURAN JARAK SENSOR INFRARED

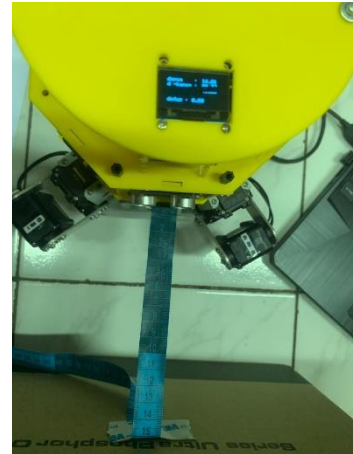
Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



Jarak 5 cm



Jarak 10 cm



Jarak 15 cm



Jarak 20 cm



Jarak 25 cm



Jarak 30 cm

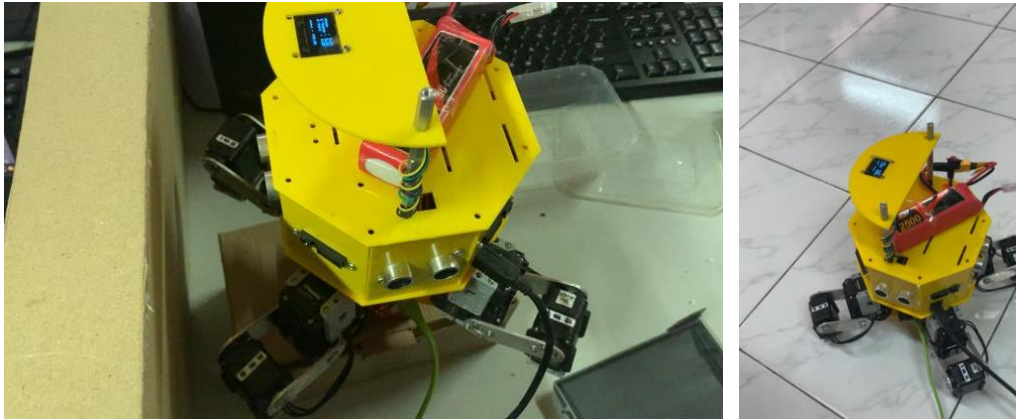


Jarak 35 cm

Tahapan pengukuran jarak sensor infrared bertujuan untuk menguji kemampuan sensor dalam membaca jarak objek terhadap robot.

Kegiatan ini dilakukan di
Laboratorium Mikrokontroler dan Mikroprocessor,
Gedung Bengkel/Lab Teknik Elektro,
Politeknik Negeri Sriwijaya, Februari – Juli 2022

LAMPIRAN B.4 – DOKUMENTASI PROJEK LA
PENGUJIAN ROBOT DENGAN METODE FUZZY LOGIC
Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



Pengujian gerakan dasar robot (gerakan maju) dengan penghalang



Tampilan jarak dan output fuzzy pada OLED display

Gerakan Maju

Gerakan Mundur

Geser Kanan

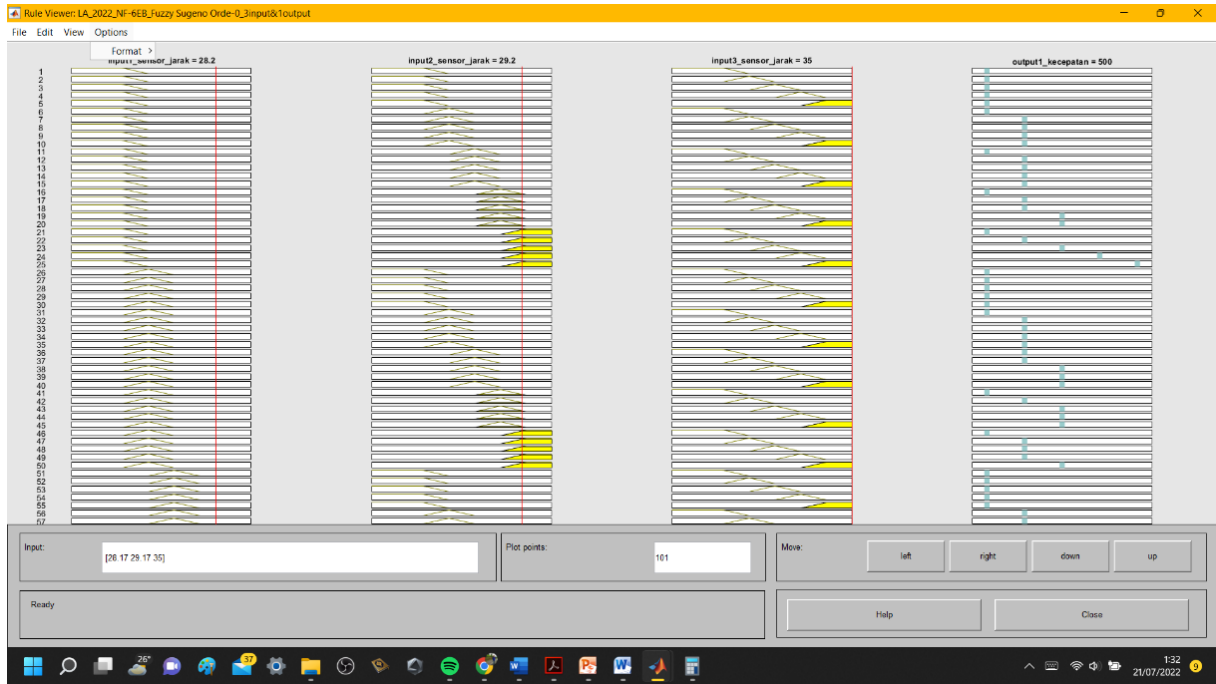
Geser Kiri



Tahapan pengujian robot dengan metode fuzzy logic

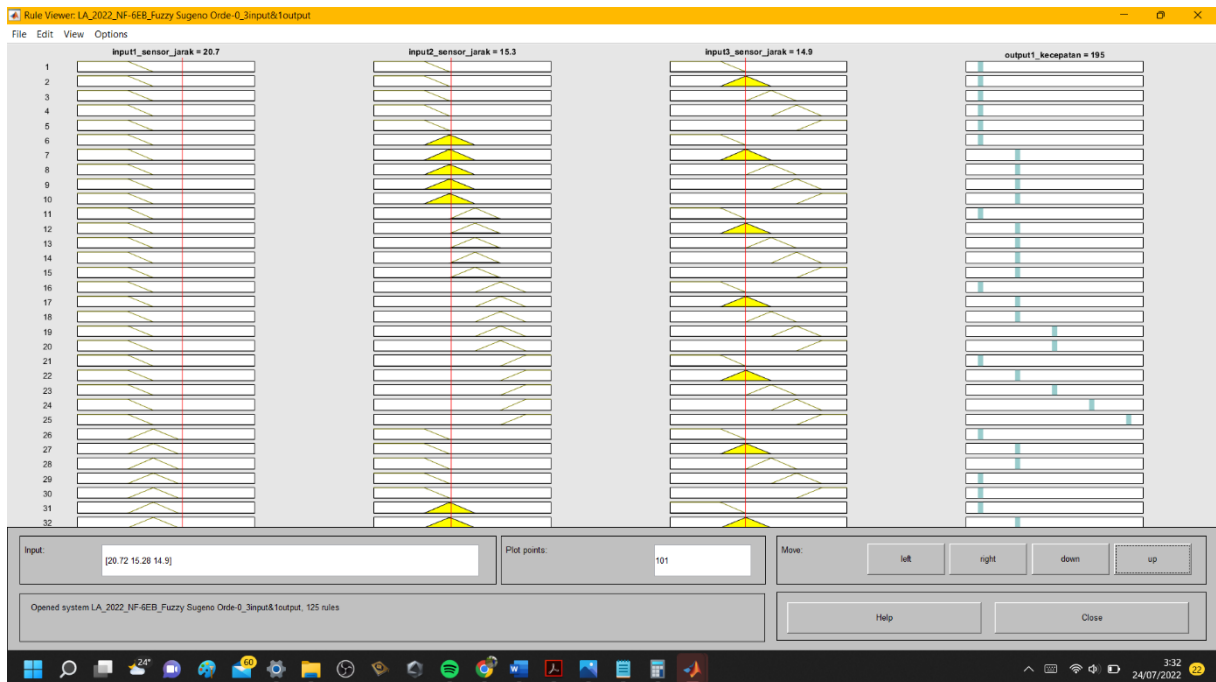
Kegiatan ini dilakukan di
Laboratorium Mikrokontroler dan Mikroprocessor,
Gedung Bengkel/Lab Teknik Elektro,
Politeknik Negeri Sriwijaya, Februari – Juli 2022

LAMPIRAN B.5 – DOKUMENTASI PROJEK LA
SIMULASI RULES METODE FUZZY PADA MATLAB
 Laporan Akhir : © 2022 - Novendra Farhan (061930320522)



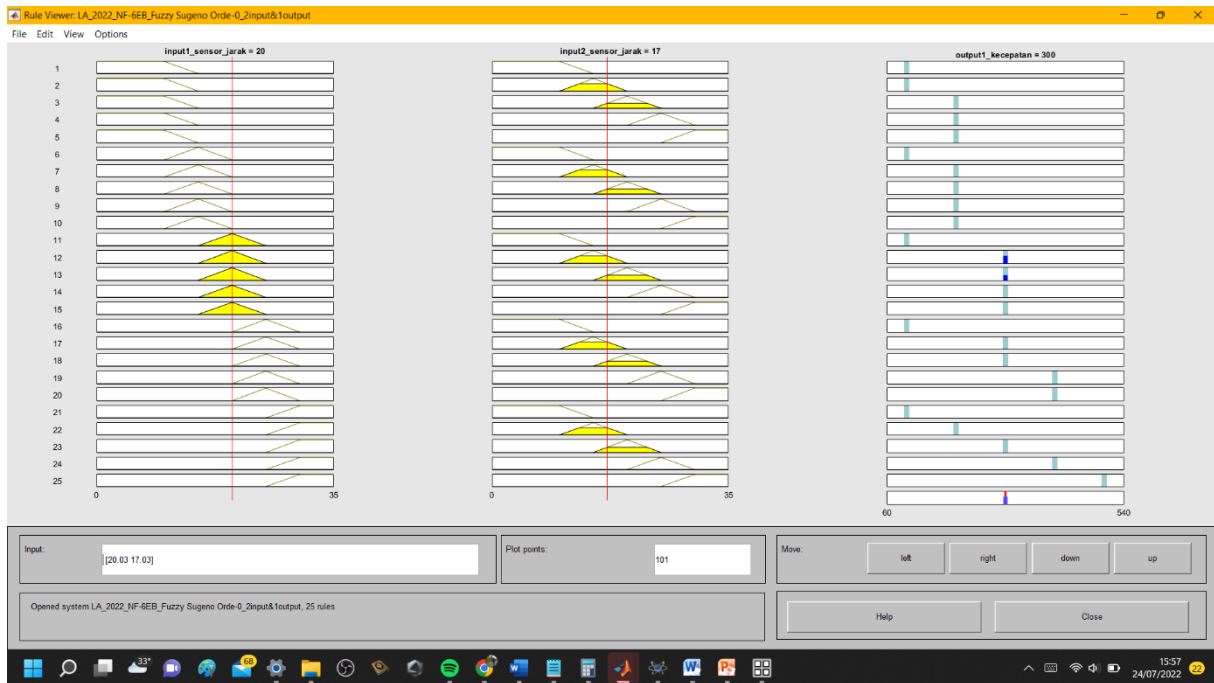
Gerak Maju (125 rules)

jarak : depan kiri 28.17cm – depan 29,17cm – depan kanan 37.61cm; output : 500

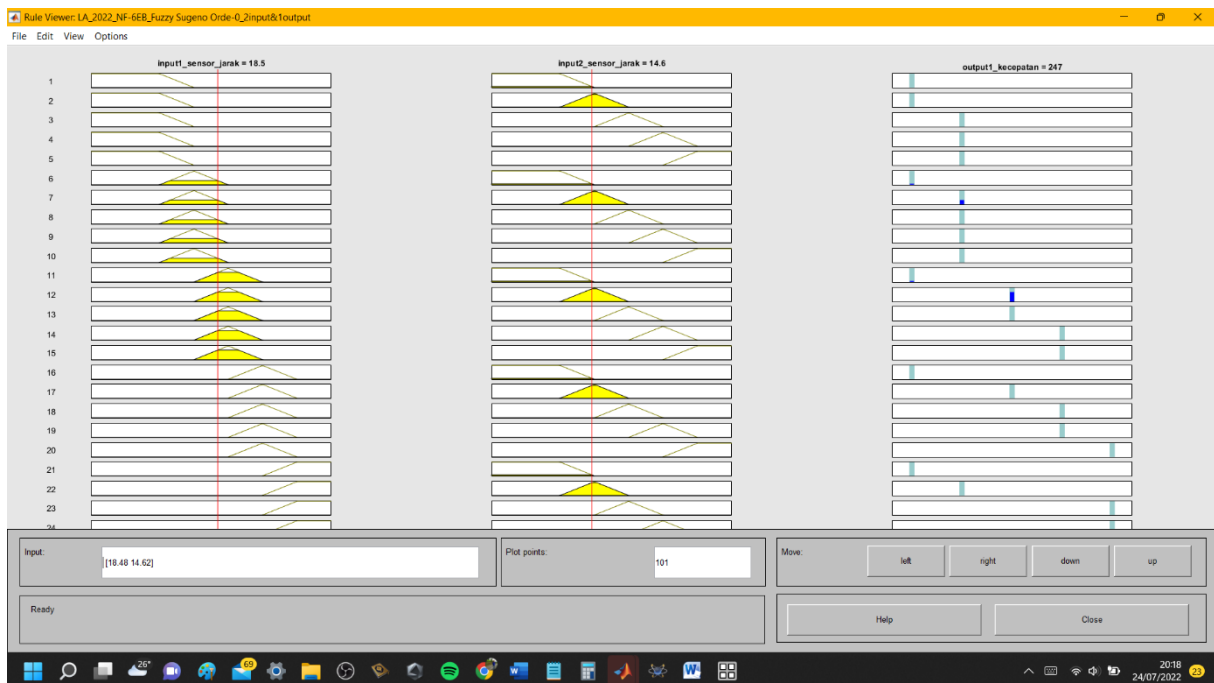


Gerak Mundur (125 rules)

jarak : depan kiri 20.72cm – depan 15.28cm – depan kanan 14.90cm; output : 195



Gerak Geser Kanan (25 rules)
 jarak : depan kanan 20.03cm – belakang kanan 17.03; output : 300



Gerak Geser Kiri (25 rules)
 jarak : depan kiri 18.48cm – belakang kiri 14.62cm; output : 247cm

Kegiatan ini dilakukan di
 Laboratorium Mikrokontroler dan Mikroprocessor,
 Gedung Bengkel/Lab Teknik Elektro,
 Politeknik Negeri Sriwijaya, Februari – Juli 2022

LAMPIRAN B.6 – DOKUMENTASI PROJEK LA
PROGRAM ROBOT (METODE FUZZY – ARDUINO MEGA 2560)
Laporan Akhir : © 2022 - Novendra Farhan (061930320522)

```
// LAPORAN AKHIR 2022 - D3 Elektronika'2019
// =====
// * Novendra Farhan_061930320522                2 Juli 2022
// * Kelas 6EB
// =====
// Dosen Pembimbing 1 : Sabilal Rasyad, S.T., M.Kom
// Dosen Pembimbing 2 : Amperawan, S.T., M.T
// =====
// * IMPLEMENTASI FUZZY LOGIC TERHADAP AKSELERASI PERGERAKAN
//   ROBOT SAR QUADRUPED
// =====
// * Fuzzy 6 input  : Sensor Jarak   (Infra Red - Sharp GP )
// *                 1 output : Kecepatan Robot (Motor Servo Dynamixel)
// =====

// Tampilan OLED=====
#include<Wire.h>                                // Lib i2c
#include<Adafruit_GFX.h>                        // Lib Grafik OLED
#include<Adafruit_SSD1306.h>                   // Lib Tipe OLED
const int lebar = 128, tinggi = 64, reset = 4;
Adafruit_SSD1306 oled(lebar, tinggi, &Wire, reset);

#define NUMFLAKES 10
#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

// logo polsri, 128x64px
const unsigned char epd_bitmap_polsri_128x64 [] PROGMEM = {
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xef, 0xf7, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xf3, 0xff,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xf3, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcf,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xcf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfb,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfb,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xdf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfb,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xef, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf3,
  0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xf7, 0xf3, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf7,
  0xef, 0xef, 0xff, 0xff,
  0xff, 0xff, 0xbf, 0xfe, 0x67, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe6,
  0x7f, 0xfd, 0xff, 0xff,
  0xff, 0xfe, 0xff, 0xff, 0x9f, 0xff, 0xff, 0xe7, 0xe7, 0xff, 0xff, 0xfd,
  0xff, 0xff, 0x7f, 0xff,
  0xff, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xee, 0xe7, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xdf, 0xff,
  0xff, 0xf7, 0xff, 0xff, 0xff, 0xf9, 0xee, 0x26, 0x60, 0xff, 0x9f, 0xff,
  0xff, 0xff, 0xe7, 0xff,
```


0xff, 0xcf, 0xff, 0xff, 0xff, 0xdf, 0x4f, 0x06, 0x61, 0xff, 0xbb, 0xff,
0xff, 0xff, 0xfb, 0xff,
0xff, 0xbf, 0xff, 0xff, 0xfe, 0x67, 0x07, 0x22, 0x49, 0xff, 0x3e, 0x7f,
0xff, 0xff, 0xfd, 0xff,
0xfe, 0xff, 0xff, 0xff, 0xfb, 0x8b, 0x99, 0xbf, 0xfd, 0xfc, 0x07, 0xdf,
0xff, 0xff, 0xff, 0x7f,
0xf9, 0xff, 0xff, 0xff, 0xef, 0xc3, 0xdf, 0x00, 0x00, 0xf9, 0x0e, 0x77,
0xff, 0xff, 0xff, 0x9f,
0xf7, 0xff, 0xff, 0xff, 0x98, 0xf0, 0x60, 0x00, 0x00, 0x07, 0x38, 0x99,
0xff, 0xff, 0xff, 0xef,
0xdf, 0xff, 0xff, 0xff, 0x74, 0x7b, 0x00, 0x01, 0x80, 0x00, 0xe2, 0x7e,
0xff, 0xff, 0xff, 0xfb,
0x3f, 0xff, 0xff, 0xfc, 0xcf, 0x9c, 0x00, 0x7f, 0xfe, 0x00, 0x33, 0xe3,
0x3f, 0xff, 0xff, 0xfc,
0x3f, 0xff, 0xff, 0xf9, 0xe1, 0xf0, 0x07, 0xff, 0xff, 0xe0, 0x0f, 0x18,
0x9f, 0xff, 0xff, 0xfc,
0xbf, 0xff, 0xff, 0xf3, 0xfc, 0xc0, 0x7f, 0xd9, 0x9b, 0xfe, 0x03, 0x25,
0xcf, 0xff, 0xff, 0xfd,
0xbf, 0xff, 0xff, 0xe6, 0x7d, 0x80, 0xfd, 0x3b, 0xdc, 0xbf, 0x01, 0x07,
0xe7, 0xff, 0xff, 0xfd,
0xbf, 0xff, 0xff, 0xef, 0x81, 0x01, 0xfb, 0x9d, 0xb9, 0xdf, 0x80, 0xfc,
0x37, 0xff, 0xff, 0xfd,
0xbf, 0xff, 0xff, 0xdf, 0xfe, 0x03, 0xdd, 0xcd, 0xb3, 0xbb, 0xc0, 0x40,
0x9b, 0xff, 0xff, 0xfd,
0xff, 0xff, 0xff, 0xd9, 0xcc, 0x07, 0xdf, 0x75, 0xa6, 0xfb, 0xe0, 0x24,
0xfb, 0xff, 0xff, 0xfd,
0xff, 0xff, 0xff, 0x99, 0xe4, 0x0f, 0x7b, 0x90, 0x0b, 0xfe, 0xf0, 0x37,
0xf9, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xbc, 0x0c, 0x7f, 0x7f, 0x63, 0xc2, 0xfe, 0xfe, 0x30,
0x0d, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xbf, 0xf8, 0x1f, 0x7f, 0x0f, 0xf0, 0xfe, 0xf8, 0x1c,
0x4d, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xb0, 0x78, 0x1e, 0xff, 0x0f, 0xf0, 0xff, 0x78, 0x13,
0xfd, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xb0, 0x0c, 0x1f, 0x3f, 0x87, 0xe1, 0xfc, 0xf8, 0x11,
0xfd, 0xff, 0xff, 0xff,
0xdf, 0xff, 0xff, 0xbf, 0xfc, 0x0f, 0x7c, 0xc0, 0x03, 0xfe, 0xf0, 0x3f,
0xdd, 0xff, 0xff, 0xfb,
0xdf, 0xff, 0xff, 0x9f, 0xfc, 0x0f, 0x8f, 0xb4, 0x2d, 0xf1, 0xf0, 0x3f,
0xf9, 0xff, 0xff, 0xfb,
0xef, 0xff, 0xff, 0xdf, 0xfe, 0x07, 0xde, 0xed, 0xb7, 0x7b, 0xe0, 0x7f,
0xfb, 0xff, 0xff, 0xf7,
0xf7, 0xff, 0xff, 0xdf, 0xfe, 0x07, 0xfb, 0xdd, 0xbb, 0xdf, 0xe0, 0x7f,
0xfb, 0xff, 0xff, 0xef,
0xf9, 0xff, 0xff, 0xef, 0xff, 0x0f, 0xfb, 0xb9, 0x9d, 0xdf, 0xf0, 0xff,
0xf7, 0xff, 0xff, 0x9f,
0xff, 0xf4, 0x7f, 0xe7, 0xff, 0xc0, 0x7f, 0xb9, 0xdd, 0xfe, 0x03, 0xff,
0xe7, 0xfc, 0xf3, 0xff,
0xff, 0xff, 0xbf, 0xf3, 0xff, 0xe0, 0x1f, 0xff, 0xff, 0xf8, 0x07, 0xff,
0xcf, 0xfd, 0xff, 0xff,
0xff, 0xff, 0x7f, 0xf9, 0xff, 0xb8, 0x03, 0xff, 0xff, 0xc0, 0x1f, 0xff,
0x9f, 0xfc, 0xff, 0xff,
0xff, 0xf9, 0xff, 0xfc, 0xfe, 0x0e, 0x00, 0xfb, 0xf3, 0x00, 0x70, 0xff,
0x3f, 0xff, 0x9f, 0xff,
0xff, 0xef, 0xff, 0xff, 0x7b, 0x3f, 0xc0, 0x80, 0x01, 0x03, 0xd8, 0x0e,
0xff, 0xff, 0xe7, 0xff,
0xff, 0xdf, 0xff, 0xff, 0x98, 0x78, 0x78, 0x00, 0x00, 0x1f, 0x9c, 0x79,
0xff, 0xff, 0xf3, 0xff,
0xff, 0xdf, 0xff, 0xff, 0xef, 0xe0, 0x37, 0xf8, 0x1f, 0xee, 0x0e, 0x77,
0xff, 0xff, 0xfb, 0xff,
0xff, 0xdf, 0xff, 0xff, 0xfb, 0xc8, 0xe4, 0x9f, 0xfd, 0xc3, 0xe7, 0xdf,
0xff, 0xff, 0xfb, 0xff,
0xff, 0xdf, 0xff, 0xff, 0xfe, 0x79, 0xcc, 0x13, 0x7d, 0xc0, 0xfe, 0x7f,
0xff, 0xff, 0xfb, 0xff,
0xff, 0xef, 0xff, 0xff, 0xff, 0xdf, 0x9c, 0x07, 0x7c, 0xe6, 0x7b, 0xff,
0xff, 0xff, 0xf7, 0xff,
0xff, 0xf7, 0xff, 0xff, 0xff, 0xf9, 0xfc, 0xcf, 0x70, 0xef, 0x9f, 0xff,
0xff, 0xff, 0xe7, 0xff,

```

    0xff, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0xfb, 0xff, 0xff,
    0xff, 0xff, 0xdf, 0xff,
    0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xff, 0xb7, 0xed, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xbf, 0xff,
    0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xfe, 0x7f, 0xff,
    0xff, 0xff, 0xbf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xfd, 0xff, 0xff,
    0xff, 0xff, 0xcf, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xf3, 0xff, 0xff,
    0xff, 0xff, 0xe7, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xe7, 0xff, 0xff,
    0xff, 0xff, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xdf, 0xff, 0xff,
    0xff, 0xff, 0xfd, 0xff, 0xff, 0xff, 0xff, 0xfd, 0xbf, 0xff, 0xff, 0xff,
    0xff, 0xbf, 0xff, 0xff,
    0xff, 0xff, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xfb, 0xdf, 0xff, 0xff, 0xff,
    0xff, 0x7f, 0xff, 0xff,
    0xff, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xf3, 0xcf, 0xff, 0xff, 0xff,
    0xfe, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe7, 0xef, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xdf, 0xfb, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xf9, 0x7a, 0xff, 0xff, 0x5c, 0x9f, 0xff,
    0xff, 0xff, 0xff, 0xff
};

// Array of all bitmaps for convenience. (Total bytes used to store
images in PROGMEM = 1040)
const int epd_bitmap_allArray_LEN = 1;
const unsigned char* epd_bitmap_allArray[1] = {
    epd_bitmap_polsri_128x64
};

//SENSOR SHARP-GP2Y0A41SK0F=====
#define SharpF  A0 // sensor depan
#define SharpR  A1 // sensor depan_kanan
#define SharpL  A2 // sensor depan_kiri
#define SharpB  A3 // sensor belakang
#define SharpR2 A4 // sensor belakang_kanan
#define SharpL2 A5 // sensor belakang_kiri

//FUZZY=====
#include <Fuzzy.h> // Library fuzzy (eFLL)
Fuzzy *fuzzy = new Fuzzy(); // Library fuzzy

float val_SharpF, val_SharpR, val_SharpL; // baca sensor sharp gp
float map_SharpF, map_SharpR, map_SharpL;
// F: depan, R: depan kanan, L: depan kiri

float val_SharpB, val_SharpR2, val_SharpL2; // baca sensor sharp gp
float map_SharpB, map_SharpR2, map_SharpL2;
// B: belakang, R2: belakang kanan, L2: belakang kiri

float out_fuzzy; // nilai defuz input 6 sensor
float out_fuzzy1; // defuz gerak maju
float out_fuzzy2; // defuz gerak mundur
float out_fuzzy3; // defuz gerak geser kanan
float out_fuzzy4; // defuz gerak geser kiri

// Membership range jarak 5-35cm
// FuzzyInput >> Sensor1 : depan
FuzzySet *sangatdekat = new FuzzySet(0, 0, 10, 15); // trapesium
FuzzySet *dekat = new FuzzySet(10, 15, 15, 20); // segitiga
FuzzySet *lumayan = new FuzzySet(15, 20, 20, 25); // segitiga
FuzzySet *jauh = new FuzzySet(20, 25, 25, 30); // segitiga

```

```

FuzzySet *sangatjauh = new FuzzySet(25, 30, 35, 35); // trapesium
// FuzzyInput >> Sensor2 : depan kanan
FuzzySet *sangatdekat2 = new FuzzySet(0, 0, 10, 15);
FuzzySet *dekat2 = new FuzzySet(10, 15, 15, 20);
FuzzySet *lumayan2 = new FuzzySet(15, 20, 20, 25);
FuzzySet *jauh2 = new FuzzySet(20, 25, 25, 30);
FuzzySet *sangatjauh2 = new FuzzySet(25, 30, 35, 35);
// FuzzyInput >> Sensor3 : depan kiri
FuzzySet *sangatdekat3 = new FuzzySet(0, 0, 10, 15);
FuzzySet *dekat3 = new FuzzySet(10, 15, 15, 20);
FuzzySet *lumayan3 = new FuzzySet(15, 20, 20, 25);
FuzzySet *jauh3 = new FuzzySet(20, 25, 25, 30);
FuzzySet *sangatjauh3 = new FuzzySet(25, 30, 35, 35);
// FuzzyInput >> Sensor4 : belakang
FuzzySet *sangatdekat4 = new FuzzySet(0, 0, 10, 15);
FuzzySet *dekat4 = new FuzzySet(10, 15, 15, 20);
FuzzySet *lumayan4 = new FuzzySet(15, 20, 20, 25);
FuzzySet *jauh4 = new FuzzySet(20, 25, 25, 30);
FuzzySet *sangatjauh4 = new FuzzySet(25, 30, 35, 35);
// FuzzyInput >> Sensor5 : belakang kanan
FuzzySet *sangatdekat5 = new FuzzySet(0, 0, 10, 15);
FuzzySet *dekat5 = new FuzzySet(10, 15, 15, 20);
FuzzySet *lumayan5 = new FuzzySet(15, 20, 20, 25);
FuzzySet *jauh5 = new FuzzySet(20, 25, 25, 30);
FuzzySet *sangatjauh5 = new FuzzySet(25, 30, 35, 35);
// FuzzyInput >> Sensor6 : belakang kiri
FuzzySet *sangatdekat6 = new FuzzySet(0, 0, 10, 15);
FuzzySet *dekat6 = new FuzzySet(10, 15, 15, 20);
FuzzySet *lumayan6 = new FuzzySet(15, 20, 20, 25);
FuzzySet *jauh6 = new FuzzySet(20, 25, 25, 30);
FuzzySet *sangatjauh6 = new FuzzySet(25, 30, 35, 35);

// FuzzyOutput >> Motor Servo 100 - 500
FuzzySet *sangatlambat = new FuzzySet (100, 100, 100, 100); singleton
FuzzySet *lambat = new FuzzySet (200, 200, 200, 200);
FuzzySet *sedang = new FuzzySet (300, 300, 300, 300);
FuzzySet *cepat = new FuzzySet (400, 400, 400, 400);
FuzzySet *sangatcepat = new FuzzySet (500, 500, 500, 500);

//=====
//=====
void setup() {
  Serial.begin (9600);
  //OLED
  oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  oled.clearDisplay();
  oled.drawBitmap(0, 0, epd_bitmap_polsri_128x64, 128, 64, 1);
  oled.display();
  delay (2500);

  oled.clearDisplay();
  oled.setTextColor(WHITE);
  oled.setTextSize(2);
  oled.setCursor(12, 15); oled.println("ROBOT SAR");
  oled.setCursor(12, 33); oled.println("QUADRUPED");
  oled.display(); delay (2500);

  oled.clearDisplay();
  oled.display();
  oled.setTextColor(WHITE);
  oled.setTextSize(1);
  oled.setCursor(5, 15); oled.println("Parni Handayani_8ELA");
  oled.setCursor(5, 30); oled.println("Novendra Farhan_6EB");
  oled.setTextSize(1.5);
  oled.setCursor(45, 50); oled.println("POLSRI");
  oled.display(); delay (2000);
}

```

```

// FuzzyInput
// =====
// Fuzzy Input Sensor1
FuzzyInput *jarak = new FuzzyInput(1);
jarak->addFuzzySet(sangatdekat);
jarak->addFuzzySet(dekat);
jarak->addFuzzySet(lumayan);
jarak->addFuzzySet(jauh);
jarak->addFuzzySet(sangatjauh);
fuzzy->addFuzzyInput(jarak);

// Fuzzy Input Sensor2
FuzzyInput *jarak2 = new FuzzyInput(2);
jarak2->addFuzzySet(sangatdekat2);
jarak2->addFuzzySet(dekat2);
jarak2->addFuzzySet(lumayan2);
jarak2->addFuzzySet(jauh2);
jarak2->addFuzzySet(sangatjauh2);
fuzzy->addFuzzyInput(jarak2);

// Fuzzy Input Sensor3
FuzzyInput *jarak3 = new FuzzyInput(3);
jarak3->addFuzzySet(sangatdekat3);
jarak3->addFuzzySet(dekat3);
jarak3->addFuzzySet(lumayan3);
jarak3->addFuzzySet(jauh3);
jarak3->addFuzzySet(sangatjauh3);
fuzzy->addFuzzyInput(jarak3);

// Fuzzy Input Sensor4
FuzzyInput *jarak4 = new FuzzyInput(4);
jarak4->addFuzzySet(sangatdekat4);
jarak4->addFuzzySet(dekat4);
jarak4->addFuzzySet(lumayan4);
jarak4->addFuzzySet(jauh4);
jarak4->addFuzzySet(sangatjauh4);
fuzzy->addFuzzyInput(jarak4);

// Fuzzy Input Sensor5
FuzzyInput *jarak5 = new FuzzyInput(5);
jarak5->addFuzzySet(sangatdekat5);
jarak5->addFuzzySet(dekat5);
jarak5->addFuzzySet(lumayan5);
jarak5->addFuzzySet(jauh5);
jarak5->addFuzzySet(sangatjauh5);
fuzzy->addFuzzyInput(jarak5);

// Fuzzy Input Sensor6
FuzzyInput *jarak6 = new FuzzyInput(6);
jarak6->addFuzzySet(sangatdekat6);
jarak6->addFuzzySet(dekat6);
jarak6->addFuzzySet(lumayan6);
jarak6->addFuzzySet(jauh6);
jarak6->addFuzzySet(sangatjauh6);
fuzzy->addFuzzyInput(jarak6);

// Fuzzy Output
//=====
// Fuzzy Output Kecepatan Gerak Robot
FuzzyOutput *motor = new FuzzyOutput(1);
motor->addFuzzySet(sangatlambat);
motor->addFuzzySet(lambat);
motor->addFuzzySet(sedang);
motor->addFuzzySet(cepat);
motor->addFuzzySet(sangatcepat);
fuzzy->addFuzzyOutput(motor);

```

```

//RULES
//=====
// rule_maju(), rule_mundur();
// rule_geser_kanan(), rule_geser_kiri();

void rule_geser_kanan() {

    //RULES-1////////////////////////////////////
    FuzzyRuleAntecedent *sangatdekat_sangatdekat2 = new
    FuzzyRuleAntecedent();
    sangatdekat_sangatdekat2->joinWithAND(sangatdekat, sangatdekat2);
    FuzzyRuleConsequent *sangatlambat_1 = new FuzzyRuleConsequent();
    sangatlambat_1->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule1 = new FuzzyRule(1, sangatdekat_sangatdekat2,
    sangatlambat_1);
    fuzzy->addFuzzyRule(fuzzyRule1);

    //RULES-2////////////////////////////////////
    FuzzyRuleAntecedent *sangatdekat_dekat2 = new FuzzyRuleAntecedent();
    sangatdekat_dekat2->joinWithAND(sangatdekat, dekat2);
    FuzzyRuleConsequent *sangatlambat_2 = new FuzzyRuleConsequent();
    sangatlambat_2->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule2 = new FuzzyRule(2, sangatdekat_dekat2,
    sangatlambat_2);
    fuzzy->addFuzzyRule(fuzzyRule2);

    //RULES-3////////////////////////////////////
    FuzzyRuleAntecedent *sangatdekat_lumayan2 = new FuzzyRuleAntecedent();
    sangatdekat_dekat2->joinWithAND(sangatdekat, lumayan2);
    FuzzyRuleConsequent *sangatlambat_3 = new FuzzyRuleConsequent();
    sangatlambat_3->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule3 = new FuzzyRule(3, sangatdekat_lumayan2,
    sangatlambat_3);
    fuzzy->addFuzzyRule(fuzzyRule3);

    //RULES-4////////////////////////////////////
    FuzzyRuleAntecedent *sangatdekat_jauh2 = new FuzzyRuleAntecedent();
    sangatdekat_jauh2->joinWithAND(sangatdekat, jauh2);
    FuzzyRuleConsequent *sangatlambat_4 = new FuzzyRuleConsequent();
    sangatlambat_4->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule4 = new FuzzyRule(4, sangatdekat_jauh2,
    sangatlambat_4);
    fuzzy->addFuzzyRule(fuzzyRule4);

    //RULES-5////////////////////////////////////
    FuzzyRuleAntecedent *sangatdekat_sangatjauh2 = new
    FuzzyRuleAntecedent();
    sangatdekat_sangatjauh2->joinWithAND(sangatdekat, sangatjauh2);
    FuzzyRuleConsequent *sangatlambat_5 = new FuzzyRuleConsequent();
    sangatlambat_5->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule5 = new FuzzyRule(5, sangatdekat_sangatjauh2,
    sangatlambat_5);
    fuzzy->addFuzzyRule(fuzzyRule5);

    //RULES-6////////////////////////////////////
    FuzzyRuleAntecedent *dekat_sangatdekat2 = new FuzzyRuleAntecedent();
    dekat_sangatdekat2->joinWithAND(dekat, sangatdekat2);
    FuzzyRuleConsequent *sangatlambat_6 = new FuzzyRuleConsequent();
    sangatlambat_6->addOutput(sangatlambat);

    FuzzyRule *fuzzyRule6 = new FuzzyRule(6, dekat_sangatdekat2,
    sangatlambat_6);
    fuzzy->addFuzzyRule(fuzzyRule6);

```



```

//RULES-7////////////////////////////////////
FuzzyRuleAntecedent *dekat_dekat2 = new FuzzyRuleAntecedent();
dekat_dekat2->joinWithAND(dekat, dekat2);
FuzzyRuleConsequent *lambat_7 = new FuzzyRuleConsequent();
lambat_7->addOutput(lambat);

FuzzyRule *fuzzyRule7 = new FuzzyRule(7, dekat_dekat2, lambat_7);
fuzzy->addFuzzyRule(fuzzyRule7);

//RULES-8////////////////////////////////////
FuzzyRuleAntecedent *dekat_lumayan2 = new FuzzyRuleAntecedent();
dekat_lumayan2->joinWithAND(dekat, lumayan2);
FuzzyRuleConsequent *lambat_8 = new FuzzyRuleConsequent();
lambat_8->addOutput(lambat);

FuzzyRule *fuzzyRule8 = new FuzzyRule(8, dekat_lumayan2, lambat_8);
fuzzy->addFuzzyRule(fuzzyRule8);

//RULES-9////////////////////////////////////
FuzzyRuleAntecedent *dekat_jauh2 = new FuzzyRuleAntecedent();
dekat_jauh2->joinWithAND(dekat, jauh2);
FuzzyRuleConsequent *lambat_9 = new FuzzyRuleConsequent();
lambat_9->addOutput(lambat);

FuzzyRule *fuzzyRule9 = new FuzzyRule(9, dekat_jauh2, lambat_9);
fuzzy->addFuzzyRule(fuzzyRule9);

//RULES-10////////////////////////////////////
FuzzyRuleAntecedent *dekat_sangatjauh2 = new FuzzyRuleAntecedent();
dekat_sangatjauh2->joinWithAND(dekat, sangatjauh2);
FuzzyRuleConsequent *lambat_10 = new FuzzyRuleConsequent();
lambat_10->addOutput(lambat);

FuzzyRule *fuzzyRule10 = new FuzzyRule(10, dekat_sangatjauh2,
lambat_10);
fuzzy->addFuzzyRule(fuzzyRule10);

//RULES-11////////////////////////////////////
FuzzyRuleAntecedent *lumayan_sangatdekat2 = new FuzzyRuleAntecedent();
lumayan_sangatdekat2->joinWithAND(lumayan, sangatdekat2);
FuzzyRuleConsequent *sangatlambat_11 = new FuzzyRuleConsequent();
sangatlambat_11->addOutput(sangatlambat);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, lumayan_sangatdekat2,
sangatlambat_11);
fuzzy->addFuzzyRule(fuzzyRule11);

//RULES-12////////////////////////////////////
FuzzyRuleAntecedent *lumayan_dekat2 = new FuzzyRuleAntecedent();
lumayan_dekat2->joinWithAND(lumayan, dekat2);
FuzzyRuleConsequent *sedang_12 = new FuzzyRuleConsequent();
sedang_12->addOutput(sedang);

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, lumayan_dekat2, sedang_12);
fuzzy->addFuzzyRule(fuzzyRule12);

//RULES-13////////////////////////////////////
FuzzyRuleAntecedent *lumayan_lumayan2 = new FuzzyRuleAntecedent();
lumayan_lumayan2->joinWithAND(lumayan, lumayan2);
FuzzyRuleConsequent *sedang_13 = new FuzzyRuleConsequent();
sedang_13->addOutput(sedang);

FuzzyRule *fuzzyRule13 = new FuzzyRule(13, lumayan_lumayan2,
sedang_13);
fuzzy->addFuzzyRule(fuzzyRule13);

```

```

//RULES-14//
FuzzyRuleAntecedent *lumayan_jauh2 = new FuzzyRuleAntecedent();
lumayan_jauh2->joinWithAND(lumayan, jauh2);
FuzzyRuleConsequent *sedang_14 = new FuzzyRuleConsequent();
sedang_14->addOutput(sedang);

FuzzyRule *fuzzyRule14 = new FuzzyRule(14, lumayan_jauh2, sedang_14);
fuzzy->addFuzzyRule(fuzzyRule14);

//RULES-15//
FuzzyRuleAntecedent *lumayan_sangatjauh2 = new FuzzyRuleAntecedent();
lumayan_sangatjauh2->joinWithAND(lumayan, sangatjauh2);
FuzzyRuleConsequent *cepat_15 = new FuzzyRuleConsequent();
cepat_15->addOutput(cepat);

FuzzyRule *fuzzyRule15 = new FuzzyRule(15, lumayan_sangatjauh2,
cepat_15);
fuzzy->addFuzzyRule(fuzzyRule15);

//RULES-16//
FuzzyRuleAntecedent *jauh_sangatdekat2 = new FuzzyRuleAntecedent();
jauh_sangatdekat2->joinWithAND(jauh, sangatdekat2);
FuzzyRuleConsequent *sangatlambat_16 = new FuzzyRuleConsequent();
sangatlambat_16->addOutput(sangatlambat);

FuzzyRule *fuzzyRule16 = new FuzzyRule(16, jauh_sangatdekat2,
sangatlambat_16);
fuzzy->addFuzzyRule(fuzzyRule16);

//RULES-17//
FuzzyRuleAntecedent *jauh_dekat2 = new FuzzyRuleAntecedent();
jauh_dekat2->joinWithAND(jauh, dekat2);
FuzzyRuleConsequent *sedang_17 = new FuzzyRuleConsequent();
sedang_17->addOutput(sedang);

FuzzyRule *fuzzyRule17 = new FuzzyRule(17, jauh_dekat2, sedang_17);
fuzzy->addFuzzyRule(fuzzyRule17);

//RULES-18//
FuzzyRuleAntecedent *jauh_lumayan2 = new FuzzyRuleAntecedent();
jauh_lumayan2->joinWithAND(jauh, lumayan2);
FuzzyRuleConsequent *cepat_18 = new FuzzyRuleConsequent();
cepat_18->addOutput(cepat);

FuzzyRule *fuzzyRule18 = new FuzzyRule(18, jauh_lumayan2, cepat_18);
fuzzy->addFuzzyRule(fuzzyRule18);

//RULES-19//
FuzzyRuleAntecedent *jauh_jauh2 = new FuzzyRuleAntecedent();
jauh_jauh2->joinWithAND(jauh, jauh2);
FuzzyRuleConsequent *cepat_19 = new FuzzyRuleConsequent();
cepat_19->addOutput(cepat);

FuzzyRule *fuzzyRule19 = new FuzzyRule(19, jauh_jauh2, cepat_19);
fuzzy->addFuzzyRule(fuzzyRule19);

//RULES-20//
FuzzyRuleAntecedent *jauh_sangatjauh2 = new FuzzyRuleAntecedent();
jauh_sangatjauh2->joinWithAND(jauh, sangatjauh2);
FuzzyRuleConsequent *sangatcepat_20 = new FuzzyRuleConsequent();
sangatcepat_20->addOutput(sangatcepat);

FuzzyRule *fuzzyRule20 = new FuzzyRule(20, jauh_sangatjauh2,
sangatcepat_20);
fuzzy->addFuzzyRule(fuzzyRule20);

```

```

//RULES-21////////////////////////////////////
FuzzyRuleAntecedent *sangatjauh_sangatdekat2 = new
FuzzyRuleAntecedent();
sangatjauh_sangatdekat2->joinWithAND(sangatjauh, sangatdekat2);
FuzzyRuleConsequent *sangatlambat_21 = new FuzzyRuleConsequent();
sangatlambat_21->addOutput(sangatlambat);

FuzzyRule *fuzzyRule21 = new FuzzyRule(21, sangatjauh_sangatdekat2,
sangatlambat_21);
fuzzy->addFuzzyRule(fuzzyRule21);

//RULES-22////////////////////////////////////
FuzzyRuleAntecedent *sangatjauh_dekat2 = new FuzzyRuleAntecedent();
sangatjauh_dekat2->joinWithAND(sangatjauh, dekat2);
FuzzyRuleConsequent *sedang_22 = new FuzzyRuleConsequent();
sedang_22->addOutput(sedang);

FuzzyRule *fuzzyRule22 = new FuzzyRule(22, sangatjauh_dekat2,
sedang_22);
fuzzy->addFuzzyRule(fuzzyRule22);

//RULES-23////////////////////////////////////
FuzzyRuleAntecedent *sangatjauh_lumayan2 = new FuzzyRuleAntecedent();
sangatjauh_lumayan2->joinWithAND(sangatjauh, lumayan2);
FuzzyRuleConsequent *sangatcepat_23 = new FuzzyRuleConsequent();
sangatcepat_23->addOutput(sangatcepat);

FuzzyRule *fuzzyRule23 = new FuzzyRule(23, sangatjauh_lumayan2,
sangatcepat_23);
fuzzy->addFuzzyRule(fuzzyRule23);

//RULES-24////////////////////////////////////
FuzzyRuleAntecedent *sangatjauh_jauh2 = new FuzzyRuleAntecedent();
sangatjauh_jauh2->joinWithAND(sangatjauh, jauh2);
FuzzyRuleConsequent *sangatcepat_24 = new FuzzyRuleConsequent();
sangatcepat_24->addOutput(sangatcepat);

FuzzyRule *fuzzyRule24 = new FuzzyRule(24, sangatjauh_jauh2,
sangatcepat_24);
fuzzy->addFuzzyRule(fuzzyRule24);

//RULES-25////////////////////////////////////
FuzzyRuleAntecedent *sangatjauh_sangatjauh2 = new
FuzzyRuleAntecedent();
sangatjauh_sangatjauh2->joinWithAND(sangatjauh, sangatjauh2);
FuzzyRuleConsequent *sangatcepat_25 = new FuzzyRuleConsequent();
sangatcepat_25->addOutput(sangatcepat);

FuzzyRule *fuzzyRule25 = new FuzzyRule(25, sangatjauh_sangatjauh2,
sangatcepat_25);
fuzzy->addFuzzyRule(fuzzyRule25);
}

//=====

// Data Komunikasi Serial: Arduino Mega2560 <-> Open CM9.04
DDRC = B11111111; // gerak
DDRA = B11111111; // data
}

void loop() {
//gerak_biasa();
maju_fuz (); /// OK-(5 JULI 2022)
mundur_fuz ();
geser_kanan_fuz ();
geser_kiri_fuz ();
}

```