

LAMPIRAN

Listing Program Arduino

```
#include <LiquidCrystal.h>
const int rs = A1, en = A2, d4 = A3, d5 = A4, d6 = A5, d7 = A6;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int jumlah_kardus = 0;
// variable variable untuk servo
#include <Servo.h>
Servo gate_botol;
Servo tutup_kardus;
Servo potong_lakban;

#define bukaGate 130
#define tutupGate 20
#define normalKardus 50
#define tutupKardus 27
#define normalLakban 180
#define potongLakban 70

//button start stop
const int p_button = 13;
bool kondisiButton = 0;

// variable variable untuk stepper
const int pulsa_per_cm = 39; // nilai diameter menunggu mekanik ,
satuan cm
const int P_pulse_stepper = 5; // pin pul + di tb6600
const int P_directional_stepper = 4; // pin dir + di tb6600
const int P_enable_stepper = 3; // pin ena + di tb6600
const int periode = 2000; // untuk mengatur kecepatan, semakin besar
nilainya semakin lambat, semakin kecil nilainya semakin cepat

// sensor infrared pin
const int P_detekGate = 6;
const int P_detekKardus = 7;
const int P_detekTutup = 8;
const int P_detekLakban = 9;
```

```

// variable untuk data sensor
bool irDetekGate = true;
bool irDetekKardus = true;
bool irDetekTutup = true;
bool irDetekLakban = true;

// variable untuk logic
bool ada_kardus = false;
int botol_dikardus = 0;
int jumlah_botol = 0;
unsigned long waktuTutup;
bool belakangLakban = false;
bool boolean_tutup_kardus = false;
unsigned long waktuLakban;
int posisi = 1;

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  Serial.begin(9600);
  setupSensor();
  setupStepper();
  setupServo();
}

void loop() {
  printlcd();
  bacaSensor(); // pembacaan sensor infrared
  // print_sensor();
  if (digitalRead(p_button) == LOW) {
    delay(500);
    if (digitalRead(p_button) == LOW)while (!digitalRead(p_button));
    kondisiButton = !kondisiButton;
  }
  // kondisiButton = 1;
  if (kondisiButton == 1) {
    switch (posisi) {
      case 1: //kondisi awal, yaitu mendeteksi ada kardus dibawah botol

```

```

if (irDetekKardus == 0) {
    jalankan_stepper(1);
    matikan_stepper();
    jumlah_kardus++;

    posisi++;
}
else if (irDetekKardus == 1) {
    jalankan_stepper_kontinue();
}
break;
case 2: //kondisi jika terdapat kardus dibawah botol
if (botol_dikardus <= 4) {
    gate_botol.write(bukaGate); //membuka servo gate
    if (irDetekGate == 0) { //ir gate mendeteksi botol
        botol_dikardus++; //menambahkan jumlah botol dalam
kardus
        jumlah_botol++;
        delay(1500);
        gate_botol.write(tutupGate); //menutup gate botol
        jalankan_stepper(6.5); // menjalankan stepper sejauh 1 botol
    }
    else {
        matikan_stepper();
    }
    if (botol_dikardus == 4) {
        posisi++;
        //        botol_dikardus = 0;
    }
}
break;
case 3:// kondisi jika botol didalam kardus sudah 4
jalankan_stepper_kontinue(); // selalu menjalankan stepper
if (irDetekTutup == 0 && boolean_tutup_kardus == false) { //
mendeteksi kardus untuk menutup belakang kardus
    boolean_tutup_kardus = true;
    tutup_kardus.write(tutupKardus);
    waktuTutup = millis() + 2000;
}
}

```

```

        if (boolean_tutup_kardus == true && millis() > waktuTutup) { //
tunggu 2 detik sebelum servo kembali
            tutup_kardus.write(normalKardus);
            boolean_tutup_kardus = false;
            posisi++;
        }
        break;
    case 4:
        jalankan_stepper_kontinue(); // selalu menjalankan stepper
        if (irDetekLakban == 0 && belakangLakban == false) { // kondisi
mendeteksi depan kardus
            belakangLakban = true;
        }
        else if (irDetekLakban == 1 && belakangLakban == true) { //
kondisi mendeteksi belakang kardus
            delay(100);
            bacaSensor();
            if (irDetekLakban == 1 && belakangLakban == true) { //
mengecek kembali kondisi mendeteksi belakang kardus
                jalankan_stepper(2);
                jalankan_stepper(-2);
                potong_lakban.write(potongLakban);
                delay(1000);
                potong_lakban.write(normalLakban);
                belakangLakban = false;
                posisi = 1;
                botol_dikardus = 0;
            }
        }
        break;
    }
}
else if (kondisiButton == 0)matikan_stepper();
}

void print_logic() {
    Serial.print("ada_kardus: ");
    Serial.print(ada_kardus);
    Serial.print("\tbotol_dikardus: ");

```

```
Serial.print(botol_dikardus);
Serial.print("\tbelakangLakban: ");
Serial.println(belakangLakban);
}
unsigned long waktu_lcd;
void printlcd() {
  if (millis() > waktu_lcd) {
    waktu_lcd = millis() + 1000;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Jumlah Kardus ");
    lcd.print(jumlah_kardus);
    lcd.setCursor(0, 1);
    lcd.print("Jumlah Botol ");
    lcd.print(jumlah_botol);
  }
}
```

Infrared sensor

```
void setupSensor() {
  pinMode(P_detekGate, INPUT_PULLUP);
  pinMode(P_detekKardus, INPUT_PULLUP);
  pinMode(P_detekTutup, INPUT_PULLUP);
  pinMode(P_detekLakban, INPUT_PULLUP);
  pinMode(p_button, INPUT_PULLUP);
}

void bacaSensor() {
  irDetekGate = digitalRead(P_detekGate);
  irDetekKardus = digitalRead(P_detekKardus);
  irDetekTutup = digitalRead(P_detekTutup);
  irDetekLakban = digitalRead(P_detekLakban);
}

void print_sensor() {
  Serial.print("irDetekGate: ");
  Serial.print(irDetekGate);
  Serial.print("\tirDetekKardus: ");
  Serial.print(irDetekKardus);
  Serial.print("\tirDetekTutup: ");
  Serial.print(irDetekTutup);
  Serial.print("\tirDetekLakban: ");
  Serial.println(irDetekLakban);
}
```

Servo

```
void setupServo(){  
  gate_botol.attach(12);  
  tutup_kardus.attach(11);  
  potong_lakban.attach(10);  
  tutup_kardus.write(normalKardus);  
  gate_botol.write(tutupGate);  
  potong_lakban.write(normalLakban);  
}
```

TB6600

```
void setupStepper() {
  pinMode(P_pulse_stepper, OUTPUT);
  pinMode(P_directional_stepper, OUTPUT);
  pinMode(P_enable_stepper, OUTPUT);
  digitalWrite(P_directional_stepper, LOW); // atur arah stepper menjadi
maju
  digitalWrite(P_enable_stepper, HIGH); // matikan stepper
}

void jalankan_stepper(float cm) {
  int total_pulsa = absoul(cm) * pulsa_per_cm; // hitung total pulsa
dengan hitungan jarak
  if(cm>0)digitalWrite(P_directional_stepper, LOW); // atur arah stepper
menjadi maju
  else{
    digitalWrite(P_directional_stepper, HIGH); // atur arah stepper
menjadi maju
  }
  digitalWrite(P_enable_stepper, LOW); // nyalakan stepper
  for (int i = 0; i < total_pulsa; i++) {
    digitalWrite(P_pulse_stepper, HIGH);
    delayMicroseconds(periode);
    digitalWrite(P_pulse_stepper, LOW);
    delayMicroseconds(periode);
  }
  digitalWrite(P_enable_stepper, HIGH); // matikan stepper
}

void jalankan_stepper_kontinue() {
  digitalWrite(P_directional_stepper, LOW);
  digitalWrite(P_enable_stepper, LOW); // nyalakan stepper
  digitalWrite(P_pulse_stepper, HIGH);
  delayMicroseconds(periode-200);
  digitalWrite(P_pulse_stepper, LOW);
  delayMicroseconds(periode-200);
}
```



```
void matikan_stepper(){
    digitalWrite(P_enable_stepper, HIGH); // matikan stepper
}
float absoult(float x){
    if(x<0)x *= -1;
    return x;
}
```