

DAFTAR PUSTAKA

- [1] Wongkar, Ferdinan. Alat pengisi botol secara otomatis berbasis mikrokontroler. Diss. Widya Mandala Catholic University Surabaya, 2018.
- [2] Sianturi, Libianko, et al. "Disain Dan Implementasi Sistem Penyortir Botol Minuman Otomatis Menggunakan Sensor Berbasis Arduino Uno." *Jurnal Visi Eksakta* 3.1 (2022): 21-34.
- [3] Yusuf, Muhamad Maulana, Mardiono Mardiono, and Sri Wiji Lestari. "Rancang Bangun Alat Pemilah Barang Berdasarkan Warna Dan Berat." *Jurnal Teknologi* 6.2 (2019): 119-135.
- [4] Setiawan, Hasbi Ade, and Tri Rijanto. "Rancang Bangun Sistem Kontrol Pengisian Air Minum dalam Kemasan Menggunakan Arduino Uno dengan Sensor *Load cell*." *Jurnal Teknik Elektro* 8.3 (2019).
- [5] Ramadhan, Fajrian. "Perancangan Penyortiran Barang Berdasarkan Berat dengan Sistem Pick And Place Berbasis Mikrokontroler." *JTEV (Jurnal Teknik Elektro dan Vokasional)* 6.2 (2020): 168-180.
- [6] Lestari, Ayu, and Oriza Candra. "Prototype Sistem Pensortir Barang di Industri Menggunakan Loadcell berbasis Arduino Uno." *JTEV (Jurnal Teknik Elektro Dan Vokasional)* 7.1 (2021): 27-36.
- [7] Yusri, Muhammad, et al. "Rancang Bangun Sistem Sortir Ikan Berdasarkan Berat Berbasis PLC." *Mechatronics Journal in Professional and Entrepreneur (MAPLE)* 4.2 (2022): 48-53.
- [8] Samudra, Kiki Julia. *RANCANG BANGUN MODUL PRAKTIKUM BERBASIS ARDUINO MEGA 2650*. Diss. Politeknik Negeri Sriwijaya, 2021.
- [9] A. Appiani, "Arduino ® MEGA 2560 Rev3 Features," Arduino ® MEGA 2560, pp. 1–18, 2022, [Online]. Available: www.arduino.cc
- [10] Edy Supriyadi, E. S. Laporan Penelitian yang tidak dipublikasikan tetapi tersimpan di Perpustakaan dengan Judul Rancang Bangun Alat Untuk Sistem Sortir Dimensi, Berat Dan Barcode Kota Tujuan Berbasis Mikrokontroler Arduino Mega 2560, TH 2021-2022.
- [11] Hendrawan, Roni, Arief Syaichu Rohman, and Denny Hidayat Tri Nugroho. "SISTEM MONITORING BERAT PADA ALAT

- ORGANIC WASTE CHOPPER (GASPER) DENGAN SENSOR BERAT (*LOAD CELL*) BERBASIS ARDUINO MEGA 2560."
- [12] S. P. Description and T. Sensitivity, "Load cell Weight Sensor 1Kg," vol. 2000, 2000.
- [13] Heryanto, Budi. *PENGEMBANGAN TEMPAT SAMPAH OTOMATIS DENGAN PEMILAH JENIS SAMPAH ORGANIK, ANORGANIK DAN LOGAM MENGGUNAKAN ARDUINO*. Diss. Politeknik Harapan Bersama Tegal, 2021.
- [14] Ray'onaldo, C. RANCANG BANGUN ROBOT PEMINDAH BARANG BERBASIS MIKROKONTROLER UNTUK MONITORING STOK GUDANG.
- [14] 14Core, "Wiring the E18-D80NK Infrared Distance Ranging Sensor," pp. 17–18, 2018, [Online]. Available: <https://www.14core.com/wiring-the-e18-d80nk-infrared-distance-ranging-sensor/>
- [15] Pulsar, "PS 12V / 5A enclosed switch mode power supply Features of the power supply unit :," *Www.Pulsar.Pl*, pp. 1–4, 2014, [Online]. Available: <http://www.pulsar.pl/en/wycofane/PS-601250 catalogue sheet - 2018-05-24.pdf>
- [16] Trisetiyanto, Adi Nova. "Rancang Bangun Alat Penyemprot Disinfektan Otomatis untuk Mencegah Penyebaran Virus Corona." *Journal of Informatics Education* 3.1 (2020): 45-51.
- [17] M. M. Wurfel, "SG996R High Torque Metal Gear Dual Ball Bearing Servo," *Electonic Caldas*, no. 6 V, pp. 1–2, 2015.
- [18] Ray'onaldo, C. RANCANG BANGUN ROBOT PEMINDAH BARANG BERBASIS MIKROKONTROLER UNTUK MONITORING STOK GUDANG.
- [19] Fatimah, Q. I., Marselino, R., & Asnil, A. (2021). Web-Based DC Motor Speed Design and Control. *MOTIVECTION: Journal of Mechanical, Electrical and Industrial Engineering*, 3(3), 101-112.
- [20] Jufriyanto, Jufriyanto, et al. "Rancang Bangun Media Pembelajaran Penyortiran Benda Berbasis Mikrokontroler." *Joule (Journal of Electrical Engineering)* 1.1 (2020): 32-40.
- [21] Aprilyani, Safah Tasya, Irianto Irianto, and Epyk Sunarno. "Desain dan Komparasi Kontrol Kecepatan Motor DC." *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)* 7.2 (2020): 127-134.
- [22] Kamolan, Aries, and Limbran Sampebatu. "Rancang Bangun

- Prototipe Pengaman Ruangan dengan *Input* Kode PIN dan Multi Sensor Berbasis Mikrokontroler." *Jurnal Ampere* 6.1 (2021): 22-31.
- [23] Subagyo, Lucky Aggazi, and Bambang Suprianto. "Sistem Monitoring Arus Tidak Seimbang 3 Fasa Berbasis Arduino Uno." *Jurnal Teknik Elektro* 6.3 (2017): 213-21.
- [24] N. N. Sam, M. Rifaldi, N. R. Wibowo, M. Nur, and P. Bosowa, "Rancang Bangun Modul Praktik *Load cell* dengan Kapasitas 20 Kg Berbasis Arduino Nano," *Mechatronics J. Prof. Entrep.*, vol. 2, no. 1, pp. 21–26, 2020.

LAMPIRAN A

Listing Program Arduino

```
#include <LiquidCrystal.h>
const int rs = 48, en = 46, d4 = 44, d5 = 42, d6 = 40, d7 = 38;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#include <Servo.h>
Servo tutup_botol;
Servo dorong_tutup;
Servo cap;
Servo dorong_cap;

#define p_infra_air  A0
#define p_infra_tutup A1
#define p_infra_cap  A2

#define P_ls_A 53 // ls besar
#define P_ls_B A3 // if botol besar
#define P_ls_a A4 // btl kecil atas if
#define P_ls_b 52 // bawah ls botol kecil

#define detek 0
#define notdetek 1

float valLoadCell;

char key;
bool sudah_penuh = false;
bool sudah_di_tutup = false;
bool sudah_sorting = false;

int mode;
int setAir;

void setupMotor();
void motor_tutup(bool kondisi); // 1 for start, 0 for stop
void motor_naikturun(char kondisi); // n for naik, t for turun, s for stop
void motor_conv_load(bool kondisi); // 1 for start, 0 for stop
```

```

void motor_conv(bool kondisi); // 1 for start, 0 for stop
void motor_pompa(bool kondisi); // 1 for start, 0 for stop
void setupServo(); // just for setup
void servo_dorong_tutup(char data); // 'd' for dorong, t for tarik
void servo_tutup_botol(char data); // 'n' for naik, t for turun
void servo_cap(char data); // 'n' for naik, t for turun
void servo_dorong_cap(bool data); // { 1 for dorong, 0 for lanjut
float setGram;
float berat_btl_kecil = 15;
float berat_btl_besar = 21;
void setup() {
  Serial.begin(9600);
  setupLcd();
  setupInfrared();
  setupServo();
  setupLoadCell();
  setupMotor();

  // motor_conv_load(0);
  // //Serial.println("system will be running");
  tampilkan_menu();
  if (mode == 1) { // mode botol 600ml
    setGram = map(setAir, 0, 100, berat_btl_besar, 607);
    while (r_sen(P_ls_A) == notdetek) { // naik terus sampe ls bawah
kecil
      motor_naikturun('n');
      loopLoadCell_1();
      tampilan_berat_ml();
    }
    motor_naikturun('s');
  }
  if (mode == 2) { // mode botol 330ml
    setGram = map(setAir, 0, 100, berat_btl_kecil, 330);
    while (r_sen(P_ls_b) == notdetek) { // naik terus sampe ls bawah
kecil
      loopLoadCell_1();
      tampilan_berat_ml();
      motor_naikturun('t');
    }
  }
}

```

```

    motor_naikturun('s');
    while (r_sen(P_ls_a) == notdetek) { // naik terus sampe ls bawah kecil
        motor_naikturun('n');
        loopLoadCell_1();
        tampilan_berat_ml();
    }
    motor_naikturun('s');
}
Serial.println("Jenis Botol,Persen Air,Berat Perhitungan,Berat
Terbaca,Hasil Sortir,Botol Tersortir,Botol Tidak Tersortir");
}

void loop() {
    if (mode == 1) { // mode botol 600ml
        loop_mode1();
    }
    else if (mode == 2) { // mode botol 330ml
        loop_mode2();
    }
    tampilan_berat_ml();
}

void loop_mode1() {
    if (aksi_isi_air_600ml()); //Serial.println("Sudah mengisi air");
    else if (aksi_tutup_botol_600ml()); //Serial.println("Sudah menutup
botol");
    else if (aksi_sorting_botol_600ml()); //Serial.println("Sudah menyorting
botol");
    else {
        if (sudah_di_tutup == true) {
            motor_conv_load(1);
            motor_conv(1);
        }
        else {
            motor_conv(1);
            motor_conv_load(0);
        }
    }
}
}
}

```

```
void loop_mode2() {
  if (aksi_isi_air());//Serial.println("Sudah mengisi air");
  else if (aksi_tutup_botol());//Serial.println("Sudah menutup botol");
  else if (aksi_sorting_botol());//Serial.println("Sudah menyorting
botol");
  else {
    if (sudah_di_tutup == true) {
      motor_conv_load(1);
      motor_conv(1);
    }
    else motor_conv(1);
  }
}
```

KEYPAD

```
#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {A8, A9, A10, A11}; //connect to the row
pinouts of the keypad
byte colPins[COLS] = {A12,A13,A14, A15}; //connect to the column
pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins,
ROWS, COLS );

void getkey(){
  char newkey = keypad.getKey();
  if(newkey) key = newkey;
  else key = 'x';
}
```


MENU AWAL

```
void setupLcd() {
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" System Start ");
}

void tampilkan_menu() {
  lcd.begin(16, 2);
  lcd.clear();
  while (1) {
    lcd.setCursor(0, 0);
    lcd.print("A utk btl 600 ml");
    lcd.setCursor(0, 1);
    lcd.print("B utk btl 330 ml");
    getKey();
    if (key == 'A') {
      mode = 1;
      break;
    }
    else if (key == 'B') {
      mode = 2;
      break;
    }
  }
  lcd.begin(16, 2);
  lcd.clear();
  String data_keypad = "";
  while (1) {
    getKey();
    if (key == 'D') {
//      Serial.println(data_keypad);
      if (data_keypad.length() > 0) {
        data_keypad.remove(data_keypad.length() - 1);
      }
//      Serial.println(data_keypad);
      delay(100);
      lcd.clear();
    }
  }
}
```

```

    }
    else if (key == '0' || key == '1' || key == '2' || key == '3' || key == '4' ||
key == '5' || key == '6' || key == '7' || key == '8' || key == '9') {
        data_keypad += key;
        delay(100);
    }
    else if (key == '#') {
        setAir = constrain(data_keypad.toInt(),0,100); // nilai persen input
dimasukan ke variable setAir
        delay(100);
        break;
    }

    lcd.setCursor(0, 0);
    lcd.print("Input Air dlm %");
    lcd.setCursor(4, 1);
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print(data_keypad);
}
lcd.begin(16, 2);
lcd.clear();

while (1) {
    getkey();
    if (key == '#') break;
    lcd.setCursor(0, 0);
    if (mode == 2)lcd.print("330ml");
    else if (mode == 1)lcd.print("600ml");
    lcd.print(" isi ");
    lcd.print(setAir);
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print("Tkn # utk mulai");
}
lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0, 1);
lcd.print("Sistem Berjalan");

```

```
lcd.setCursor(0, 0);  
lcd.print("Btl ");  
if (mode == 2)lcd.print("330ml ");  
else if (mode == 1)lcd.print("600ml ");  
lcd.print(setAir);  
lcd.print("%");  
}
```

MOTOR

```
#define p_m_tutup 6
#define p_m_naikturun1 4
#define p_m_naikturun2 5
#define p_m_pompa 3
#define p_m_load 7
#define p_m_conv 8
#define p_m_conv1 9

void setupMotor() {
  pinMode(p_m_tutup, OUTPUT);
  pinMode(p_m_naikturun1, OUTPUT);
  pinMode(p_m_naikturun2, OUTPUT);
  pinMode(p_m_pompa, OUTPUT);
  pinMode(p_m_load, OUTPUT);
  pinMode(p_m_conv, OUTPUT);
  pinMode(p_m_conv1, OUTPUT);

  motor_conv(0);
  motor_conv_load(0);
  motor_tutup(0);
  motor_naikturun('s');
  motor_pompa(0);
}

void motor_tutup(bool kondisi) {
  if (kondisi == true)digitalWrite(p_m_tutup, LOW);
  else digitalWrite(p_m_tutup, HIGH);
}

void motor_naikturun(char kondisi) {
  if (kondisi == 'n') { // naik
    digitalWrite(p_m_naikturun1, HIGH);
    digitalWrite(p_m_naikturun2, LOW);
  }
  else if (kondisi == 't') {
    digitalWrite(p_m_naikturun1, LOW);
    digitalWrite(p_m_naikturun2, HIGH);
  }
}
```

```

    }
    else {
        digitalWrite(p_m_naikturun1, LOW);
        digitalWrite(p_m_naikturun2, LOW);
    }
}

void motor_pompa(bool kondisi) {
    if (kondisi == true) analogWrite(p_m_pompa, 150);
    else digitalWrite(p_m_pompa, LOW);
}

bool sudahjalan_load;
int makspwm_load = 160;
int i_load = 0;
void motor_conv_load(bool kondisi) {
    if (kondisi == 1) {
        analogWrite(p_m_load, makspwm_load);
    }
    else if (kondisi == 0) {
        sudahjalan_load = false;
        i_load = 0;
        analogWrite(p_m_load, 0);
    }
}

bool sudahjalan, sudahberhenti;
int makspwm = 90;
int i = 0;

void motor_conv(int kondisi) {
    if (kondisi == 1) {
        if (sudahjalan == true) {
            analogWrite(p_m_conv, makspwm);
            digitalWrite(p_m_conv1, LOW);
        }
        else {
            analogWrite(p_m_conv1, LOW);
            if (i < makspwm) {

```

```
        i++;
        delay(5);
        analogWrite(p_m_conv, i);
    }
    else {
        sudahjalan = true;
    }
}
}
else if (kondisi == 0) {
    sudahjalan = false;
    i = 0;
    digitalWrite(p_m_conv, LOW);
    digitalWrite(p_m_conv1, LOW);
}
else if (kondisi == 2) {
    sudahjalan = false;
    i = 0;
    digitalWrite(p_m_conv, 255);
    digitalWrite(p_m_conv1, LOW);
}
}
```

INFRARED

```
void setupInfrared(){  
  pinMode(P_ls_b,INPUT_PULLUP);  
  pinMode(P_ls_A,INPUT_PULLUP);  
}
```

```
bool r_sen(int pin){  
  return digitalRead(pin);  
}
```

ISI BOTOL

```
bool aksi_isi_air() { // botol kecil
  if (r_sen(p_infra_air) == detek && sudah_penuh == false) {
    motor_conv(0);
    while (r_sen(P_ls_b) == notdetek) { // turun terus sampe ls bawah
      kecil
        tampilan_berat_ml();
        motor_naikturun('t');
      }
      delay(5500);
      motor_naikturun('s');
      motor_pompa(1);
      float input_persen = setAir; //50%
      float mlAir = (input_persen / 100) * 330;
      float total_delay = map(mlAir, 73, 225, 1450, 4545);
      delay(total_delay + 500);
      motor_pompa(0);
      while (r_sen(P_ls_a) == notdetek) { // turun terus sampe ls bawah
        kecil
          motor_naikturun('n');
          tampilan_berat_ml();
        }
        motor_naikturun('s');
        motor_conv(1);
        sudah_penuh = true;
        sudah_sorting = false;
        return true;
      }
    }
  else {
    return false;
  }
}

bool aksi_isi_air_600ml() { // botol besar
  if (r_sen(p_infra_air) == detek && sudah_penuh == false) {
    delay(150);
    motor_conv(0);
  }
}
```



```

    while (r_sen(P_ls_B) == notdetek) { // turun terus sampe ls bawah
    besar
        tampilan_berat_ml();
        motor_naikturun('t');
    }
    delay(5500);
    motor_naikturun('s');

    motor_pompa(1);
    float input_persen = setAir;
    float mlAir = (input_persen / 100) * 620;
    float total_delay = map(mlAir, 73, 225, 1450, 4398);
    delay(total_delay + 500);
    motor_pompa(0);

    while (r_sen(P_ls_A) == notdetek) { // naik terus sampe ls atas besar
        motor_naikturun('n');
        tampilan_berat_ml();
    }
    motor_naikturun('s');
    motor_conv(1);
    sudah_penuh = true;

    sudah_sorting = false;
    return true;
}
else {
    return false;
}
}

```

PENUTUP

```
#define detek 0
#define notdetek 1
bool aksi_tutup_botol(){
  if (r_sen(p_infra_tutup) == detek && sudah_di_tutup == false) {
    motor_conv(0); // hentikan conveyor
    delay(100);
    while (r_sen(P_ls_a) == notdetek) {
      motor_naikturun('n'); // turun terus sampe ls bawah kecil
      tampilan_berat_ml();
    }
    motor_naikturun('t');
    delay(8000);
    tampilan_berat_ml();
    motor_naikturun('s');

    servo_dorong_tutup('d'); // putar servo untuk mengisi tutup botol
    delay(1000);
    servo_tutup_botol('t');

    motor_tutup(1); // turun ke bawah sambil putar tutup
    motor_naikturun('t');
    delay(1500); // ini perlu diatur
    motor_naikturun('s');
    delay(1500); // ini perlu diatur
    servo_tutup_botol('n');

    while (r_sen(P_ls_a) == notdetek) { // naik terus sampe ls atas kecil
      motor_naikturun('n');
      tampilan_berat_ml();
    }
    delay(3000);
    motor_tutup(0);
    motor_naikturun('s');

    motor_conv(1); // jalankan kembali conveyor
    servo_dorong_tutup('t');
    motor_naikturun('s');
```

```

    sudah_di_tutup = true;
    sudah_penuh = false;
    return true;
}
else return false;
}

bool aksi_tutup_botol_600ml() {
    if (r_sen(p_infra_tutup) == detek && sudah_di_tutup == false) {
        delay(50);
        motor_conv(0); // hentikan conveyor
        delay(750);
        while (r_sen(P_ls_A) == notdetek) {
            motor_naikturun('n'); // turun terus sampe ls bawah kecil
            tampilan_berat_ml();
        }
        motor_naikturun('t');
        delay(8000);
        tampilan_berat_ml();
        motor_naikturun('s');

        servo_dorong_tutup('d'); // putar servo untuk mengisi tutup botol
        delay(1000);
        servo_tutup_botol('t');

        motor_tutup(1); // turun ke bawah sambil putar tutup
        motor_naikturun('t');
        delay(1500); // ini perlu diatur
        motor_naikturun('s');
        delay(3000); // ini perlu diatur
        servo_tutup_botol('g');

        while (r_sen(P_ls_A) == notdetek) { // naik terus sampe ls atas kecil
            motor_naikturun('n');
            tampilan_berat_ml();
        }
        delay(3000);
        servo_tutup_botol('n');
    }
}

```

```
motor_tutup(0);  
motor_naikturun('s');  
  
motor_conv(1); // jalankan kembali conveyor  
servo_dorong_tutup('t');  
motor_naikturun('s');  
  
sudah_di_tutup = true;  
sudah_penuh = false;  
  
return true;  
}  
else return false;  
}
```

LOAD CELL

```
#include <HX711_ADC.h>
#define HX711_dout 32 //mcu > HX711 dout pin
#define HX711_sck 33 //mcu > HX711 sck pin
#define factor_kalibrasi_lc 820.00

HX711_ADC LoadCell(HX711_dout, HX711_sck);
void setupLoadCell() {
  LoadCell.begin();
  LoadCell.setReverseOutput();
  unsigned long stabilizingtime = 2000;
  boolean _tare = true;
  LoadCell.start(stabilizingtime, _tare);
  if (LoadCell.getTareTimeoutFlag()) {
    // Serial.println("load cell tidak terdeteksi");
  }
  else {
    LoadCell.setCalFactor(factor_kalibrasi_lc);
    // Serial.println("Kalibrasi selesai");
  }
}

void loopLoadCell() {
  boolean newDataReady = 0;
  for (int i = 0; i < 600; i++) {
    if (LoadCell.update()) newDataReady = true;
    if (newDataReady) {
      valLoadCell = LoadCell.getData();
      newDataReady = false;
    }
    delay(10);
  }
  tampilan_berat_ml();
}

void loopLoadCell_1() {
  boolean newDataReady = 0;
  if (LoadCell.update()) newDataReady = true;
```

```

if (newDataReady) {
    valLoadCell = LoadCell.getData();
    newDataReady = false;
}
}

void kalibrasiLoadCell() {
    LoadCell.tareNoDelay();
    while (!LoadCell.getTareStatus()) {
        static boolean newDataReady = 0;
        if (LoadCell.update()) newDataReady = true;
        if (newDataReady) {
            float i = LoadCell.getData();
            newDataReady = false;
        }
    }
}

unsigned long lcdtime;
void tampilan_berat_ml() {
    if (millis() > lcdtime) {
        lcdtime = millis() + 1000;
        lcd.begin(16, 2);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Btl ");
        if (mode == 2)lcd.print("330ml ");
        else if (mode == 1)lcd.print("600ml ");
        lcd.print(setAir);
        lcd.print("%");

        lcd.setCursor(0, 1);
        lcd.print("Br:");
        lcd.print(int(valLoadCell));
        lcd.print("g");
        lcd.print(" V1:");
        if (mode == 1)lcd.print(constrain(int(valLoadCell - berat_btl_besar),
0, 1000));
        if (mode == 2)lcd.print(constrain(int(valLoadCell - berat_btl_kecil),
0, 1000));
    }
}

```

```
    lcd.print("mL");  
  }  
}
```

SORTING

```
int offsideGram = 30;
int tersortir, tidak_tersortir;
bool aksi_sorting_botol() { //botol kecil
  if (!r_sen(p_infra_cap) && sudah_sorting == false) {
    delay(100);
    motor_conv(0);
    motor_conv_load(0);
    while (r_sen(P_ls_a) == notdetek) {
      motor_naikturun('n'); // turun terus sampe ls bawah kecil
      tampilan_berat_ml();
    }
    motor_naikturun('s');
    loopLoadCell();

    // var setGram untuk menghitung berat botol seharusnya
    // 20 / 100 * 330
    // 66
    // setAir = 100 %
    //     50 > 172
    //     100 > 344 + 15
    //     359
    //     setGram = map(setAir, 0, 100, 0, 344) + berat_btl_kecil;

    // lcd.clear();
    // lcd.setCursor(0, 0);
    // lcd.print("Berat Htg ");
    // lcd.print(int(setGram));
    // lcd.setCursor(0, 1);
    // lcd.print("Berat Btl ");
    // lcd.print(int(valLoadCell));
    // lcd.print(" g");
    tampilan_berat_ml();

    // offsideGram yang dimaksud yaitu toleransi

    //dibaca sensor = 84
    // berat seharusnya sesuai perhitungan 66
```



```

// toleransi = 30
// diantara (beratseharusnya - toleransi) sampai (beratseharusnya +
toleransi)
// 66-30 sampai 66 + 30
// 36 sampai 96

// persen toleransi = toleransi/330*100%
// persen toleransi = 30/330*100%
// persen toleransi = +-9%
// 350
// 359-30
// 329
// 359+30
// 589
// 329-389

setGram = map(setAir, 0, 100, 0, 330) + berat_btl_besar;

float masa_jenis_air = 1000; //kg/m3
float masa_jenis_air_gram_ml = masa_jenis_air * 1000 / 1000000;
//merubah kg/m3 menjadi gram/ml
float masa_berat = valLoadCell; // hasil pembacaan loadcell dalam
gram
float volume = masa_berat / masa_jenis_air_gram_ml; //mL

if (volume > (setGram - offsideGram) && volume < (setGram +
offsideGram)) {
  tersortir++;
  Serial.println("330ml," + String(setAir) + "," + String(setGram) + ","
+ String(valLoadCell) + ",Tersortir," + String(tersortir) + "," +
String(tidak_tersortir));
  while (r_sen(P_ls_b) == notdetek) {
    motor_naikturun('t');// turun terus sampe ls bawah kecil
    tampilan_berat_ml();
  }
  motor_naikturun('s');
  servo_cap('t');
  servo_dorong_cap(0);
  delay(1000);
}

```

```

servo_cap('n');
while (r_sen(P_ls_a) == notdetek) motor_naikturun('n'); // turun
terus sampe ls bawah kecil
motor_naikturun('s');
delay(500);
motor_conv_load(1);
delay(5000);
servo_dorong_cap(0);
}
else {
tidak_tersortir++;
Serial.println("330ml," + String(setAir) + "," + String(setGram) + ","
+ String(valLoadCell) + ",Tidak Lolos," + String(lolos) + "," +
String(tidak_lolos));
servo_dorong_cap(1);
motor_conv_load(1);
delay(10000);
servo_dorong_cap(0);
}
tampilan_berat_ml();
sudah_sorting = true;
sudah_di_tutup = false;

return true;
}
else return false;
}

bool aksi_sorting_botol_600ml() {
if (!r_sen(p_infra_cap) && sudah_sorting == false) {
delay(100);
motor_conv(0);
motor_conv_load(0);
while (r_sen(P_ls_A) == notdetek) {
motor_naikturun('n'); // turun terus sampe ls bawah kecil
tampilan_berat_ml();
}
motor_naikturun('s');
loopLoadCell();
}

```

```

setGram = map(setAir, 0, 100, 0, 607) + berat_btl_besar;
float masa_jenis_air = 1000; //kg/m3
float masa_jenis_air_gram_ml = masa_jenis_air * 1000 / 1000000;
//merubah kg/m3 menjadi gram/ml
float masa_berat = valLoadCell; // hasil pembacaan loadcell dalam
gram
float volume = masa_berat / masa_jenis_air_gram_ml; //mL

if (volume > (setGram - offsideGram) && volume < (setGram +
offsideGram)) {
    tersortir++;
    Serial.println("600ml," + String(setAir) + "," + String(setGram) + ","
+ String(valLoadCell) + ",Tersortir," + String(tersortir) + "," +
String(tidak_tersortir));
    while (r_sen(P_ls_B) == notdetek) {
        motor_naikturun('t'); // turun terus sampe ls bawah kecil
        tampilan_berat_ml();
    }
    motor_naikturun('s');
    servo_cap('t');
    servo_dorong_cap(0);
}
else {
    tidak_tersortir++;
    servo_dorong_cap(1);
    Serial.println("600ml," + String(setAir) + "," + String(setGram) + ","
+ String(valLoadCell) + ",Tidak Lolos," + String(lolos) + "," +
String(tidak_lolos));
}

delay(1000);
servo_cap('n');
while (r_sen(P_ls_A) == notdetek) motor_naikturun('n'); // turun terus
sampe ls bawah kecil
motor_naikturun('s');
delay(500);
motor_conv_load(1);
delay(5000);
servo_dorong_cap(0);

```

```
tampilan_berat_ml();
sudah_sorting = true;
sudah_di_tutup = false;

return true;
}
else return false;
}
```

STEMPEL

```
void aksi_detek_cap() {
  if (r_sen(p_infra_cap) == detek) {

    motor_conv(0);
    motor_conv_load(0);

    if (valLoadCell > 320) {
      servo_cap('t');
      delay(300);
      servo_cap('n');
      delay(300);

      motor_conv_load(1);
    }
    else if (valLoadCell > 100) {
      servo_dorong_cap(1);
      delay(500);
      motor_conv_load(1);
      delay(3000);
      servo_dorong_cap(0);
    }
  }
}

void aksi_detek_cap_600ml() {
  if (r_sen(p_infra_cap) == detek) {
    motor_conv(0);
    motor_conv_load(0);

    if (valLoadCell > 320) {
      servo_cap('t');
      delay(300);
      servo_cap('n');
      delay(300);

      motor_conv_load(1);
    }
    else if (valLoadCell > 100) {
```

```
servo_dorong_cap(1);  
delay(500);  
motor_conv_load(1);  
delay(3000);  
servo_dorong_cap(0);  
}  
}  
}
```

SERVO

```
void setupServo() {
  dorong_tutup.attach(26);
  tutup_botol.attach(23);
  cap.attach(24);
  dorong_cap.attach(25);

  servo_dorong_tutup('t');
  servo_tutup_botol('n');
  servo_cap('n');
  servo_dorong_cap(0);
}

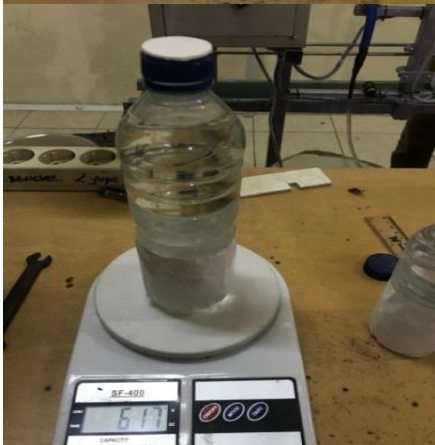
void servo_tutup_botol(char data) {
  if (data == 'n')tutup_botol.write(50);
  else if (data == 'g')tutup_botol.write(180);
  else tutup_botol.write(170);
}

void servo_cap(char data) {
  if (data == 'n')cap.write(140);
  else cap.write(60);
}

void servo_dorong_cap(bool data) { // 1 for dorong, 0 for lanjut
  if (data) dorong_cap.write(115);
  else dorong_cap.write(0);
}

void servo_dorong_tutup(char data) {
  if (data == 'd') {
    for (int i = 0; i <= 105; i++) {
      dorong_tutup.write(i);
      delay(5);
    }
  }
  else dorong_tutup.write(0);
}
```

LAMPIRAN B



BIODATA PENULIS



Nama : Desna Fitrilia
Tempat/Tanggal Lahir : Cilacap, 25 Desember 2001
Alamat : Jl. Kebon Pepaya RT01/RW 01
Kec. Kesugihan, Kab. Cilacap
Email : desnafitrilia25@gmail.com
Telepon/HP : 089602505924
Hobi : Menonton, membaca, travelling
Motto : Belajar dari pengalaman

Riwayat Pendidikan

- SD Negeri Kesugihan 01 Tahun 2008-2013
- SD Negeri Percobaan Bandung Tahun 2013-2014
- SMP VIP Al- Huda Tahun 2014-2017
- SMA Negeri 1 Maos Tahun 2017-2020
- Politeknik Negeri Cilacap Tahun 2020-2023
Prodi D3 Teknik Elektronika

Penulis telah mengikuti seminar hasil pada tanggal 16 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md)

