DAFTAR PUSTAKA

[1] D. D. Silfi and K. Pengantar, “Program Studi Teknik Lingkungan Universitas Sumatera Utara,” 2014.

[2] S. B. Dr. dr. P. Sudiharto, “PENGEMBANGAN TEKNOLOGI KESEHATAN UNTUK MENJAWAB TANTANGAN DAN KEBUTUHAN MASA DEPAN DEMI KEMANDIRIAN BANGSA,” vol. 2, no. 5, p. 255, 2009, [Online].

[3] D. Handayani, D. R. Hadi, F. Isbaniah, E. Burhan, and H. Agustin, “Corona Virus Disease 2019,” *J. Respirologi Indones.*, vol. 40, no. 2, pp. 119–129, 2020, doi: 10.36497/jri.v40i2.101.

[4] P. D. A. N. Sunarwan, “Perancangan dan pembuatan otomatisasi layanan informasi objek museum menggunakan rfid,” 2017.

[5] D. Nurdiansyah, N. Ikhsani, and I. Afriliana, “Sistem Absensi Menggunakan Rfid Dan Esp32 Cam Berbasis Iot,” 2019.

[6] Viera Astry and kiki Prawiroredjo, “Mesin Kartu Antrian Berbasis Mikrokontroler Atmega16a,” vol. 12, no. 2, pp. 33–44, 2015.

[7] Sheila Frasisca, “Penerapan Sensor HC-SR04 dan *Printer Thermal* pada Sistem Antrian Bank Menggunakan *Mikrokontroller* Arduino,” vol. 19, 2021.

[8] Kadek joni purnawan, “PROTOTYPE MESIN ANTRIAN DENGAN ARDUINO TUGAS,” vol. 15, no. 2, pp. 1–23, 2021.

LAMPIRAN A

DAFTAR PROGRAM ARDUINO

A. Program Arduino Untuk *Printer Thermal*

```
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <SoftwareSerial.h>

#define RST_PIN 10
#define SS_PIN 9

const byte ROWS = 5;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'f', 'v', '#', '*'},
  {'1', '2', '3', 'n'},
  {'4', '5', '6', 'u'},
  {'7', '8', '9', 's'},
  {'<', '0', '>', 'x'}
};
byte rowPins[ROWS] = {A0, A1, A2, A3, 8};
byte colPins[COLS] = {4, 5, 6, 7};

String datarfid, antrian_skrg, dataserial, datavalid;
String data_daftar, data_nohp;
int no_antri, data_antri;
byte flag, en;
long proveus, proveus1, timer;
char customKey;

Keypad customKeypad = Keypad(
makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

```
MFRC522::MIFARE_Key key;

SoftwareSerial serial(2, 3);

void setup() {
  //inialisasi
  Serial.begin(57600);
  serial.begin(57600);
  Wire.begin();
  SPI.begin();
  mfrc522.PCD_Init();
  lcd.init();
  lcd.backlight();
}

void loop() {
  // indikator serial
  if ((millis() - proveus1) > 1000) {
    Serial.println("datavalid: " + datavalid);
    Serial.println(customKey);
    Serial.println("flag: " + String(flag));
    Serial.println("dataserial: " + dataserial);
    proveus1 = millis();
  }
  mainn();
}
```

B. Program Input Untuk *Keypad* dan RFID

```
//baca keypad
void keypad() {
  char customKey = customKeypad.getKey();
  if (customKey) {
    Serial.println(customKey);
  }

}

//baca rfid
```

```
void rfid() {
  String content = "";
  byte letter;
  if (!mfrc522.PICC_IsNewCardPresent() ||
!mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  Serial.print("UID tag :");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    komunikasi();
    proveus = millis();
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10
? "0" : ""));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }

  en = 1;
  Serial.println();
  datarfid = content;
  serial.println("{" + data_daftar + "/" + data_nohp + "/"
+ datarfid + "/" + String() + "}");
  Serial.println("{" + data_daftar + "/" + data_nohp + "/"
+ datarfid + "/" + String() + "}");
  Serial.println("datarfid: " + datarfid);
  komunikasi();
  lcd.clear();
  delay(1000);
  flag = 1;
}
```

C. Program Komunikasi Serial Arduino

```
//komunikasi serial
void komunikasi() {
  if (serial.available() > 0) {
    dataserial = serial.ReadStringUntil('\n');
    datavalid = dataserial;
    Serial.println("dataserial: " + dataserial);
  }
}
```

D. Program Pengolahan Data Arduino

```
//logic kepad
void mainn() {
  char customKey = customKeypad.getKey();
  if (customKey) {
    Serial.println(customKey);
  }

  if (customKey == '*') {
    lcd.clear();
    flag = 2;
  }
  else if (customKey == 's') {
    lcd.clear();
    data_nohp = "";
    flag = 2;
  }
  else if (customKey == '<') {
    lcd.clear();
    data_nohp = "";
    data_daftar = "";
    flag = 0;
  }

  if (flag == 0) {
    data_daftar = "";
    data_nohp = "";
    if ((millis() - proveus) > 1000) {
      rfid();
      komunikasi();
      proveus = millis();
     }
    komunikasi();
    lcd.setCursor(0, 0);
    lcd.print("No Antri: " + String(data_antri)) + " ";
    lcd.setCursor(0, 1);
    lcd.print("Tap Kartu Anda  ");
   }

  else if (flag == 1) {
    komunikasi();
    delay(2000);
    flag = 5;
   }

  else if (flag == 5) {
    if (datavalid.toInt() == 1) {
      komunikasi();
      data_antri++;
      antrian_skrg = String(data_antri);
      Serial.println("no antri: " + String(data_antri));
      serial.println("{" + data_daftar + "/" + data_nohp + "/" +
datarfid + "/" + String(data_antri) + "}");
      lcd.setCursor(0, 0);
      lcd.print("Sedang Cetak  ");
      lcd.setCursor(0, 1);
      lcd.print("No Antri      ");
      delay(2000);
      flag = 6;
      en = 0;
      datavalid = "";
     }
    else if (datavalid.toInt() == 0) {
      komunikasi();
      antrian_skrg = antrian_skrg;
      Serial.println("data tidak valid");
      lcd.setCursor(0, 0);
      lcd.print("Invalid        ");
      lcd.setCursor(0, 1);
      lcd.print("                ");
      delay(2000);
      flag = 6;
      en = 0;
      datavalid = "";
    }
  }

  else if (flag == 6) {
    delay(1000);
    lcd.clear();
    flag = 0;
  }

  else if (flag == 2) {
    if (customKey == '0' || customKey == '1' || customKey == '2' ||
customKey == '3' || customKey == '4' || customKey == '5' ||
customKey == '6' || customKey == '7' || customKey == '8' ||
customKey == '9') {
      data_nohp = data_nohp + customKey;
      Serial.println("data_nohp: " + data_nohp);
    }
    else if (customKey == 'x') {
      lcd.clear();
      flag = 3;
    }
    Serial.println("masukan no hp");
    lcd.setCursor(0, 0);
    lcd.print("Masukan No Hp: ");
    lcd.setCursor(0, 1);
    lcd.print(data_nohp);
  }

  else if (flag == 3) {
    Serial.println("tap kartu");
    lcd.setCursor(0, 0);
    lcd.print("Tap Kartu Anda");
    String content = "";
    byte letter;
    if (!mfrc522.PICC_IsNewCardPresent() ||
!mfrc522.PICC_ReadCardSerial()) {
      return;
    }
    Serial.print("UID tag :");
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
      Serial.print(mfrc522.uid.uidByte[i], HEX);
      content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" :
""));
      content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }

    Serial.println();
    data_daftar = content;
    Serial.println("data_daftar: " + data_daftar);
    lcd.clear();
    flag = 4;
  }

  else if (flag == 4) {
    serial.println("{" + data_daftar + "/" + data_nohp + "/" +
datarfid + "/" + String(data_antri) + "}");
    lcd.setCursor(0, 0);
    lcd.print("Daftar Sukses");
    Serial.println(data_daftar + "," + data_nohp);
    Serial.println("kartu berhasil didaftarkan");
    delay(2000);
    lcd.clear();
    flag = 0;
  }
}
```

E. Program Perhitungan Antrean

```
#include <WiFi.h>
#include <DMD32.h>
#include "fonts/SystemFont5x7.h"
#include "fonts/Arial_black_16.h"

#define DISPLAYS_ACROSS 1
#define DISPLAYS_DOWN 1
#define    LED0     2
#define    BUTTON    4
#define    MAXSC     10
```

```
unsigned long awal = 0, akhir = 1000;
char*       ESPssid;
char*       ESPpassword;
String      Message;
byte b;
String data;
int dataint;

hw_timer_t * timer = NULL;
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);
WiFiServer  ESPServer(9001);
WiFiClient  ESPClient[MAXSC];

void IRAM_ATTR triggerScan()
{
  dmd.scanDisplayBySPI();
}

void setUP(void)
{
  //inialisasi
  Serial.begin(115200);
  pinMode(LED0, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  Serial.println("I/O Pins Modes Set .... Done");
  SetWifi("Mesin Antrian", "mesinantrian");
  uint8_t cpuClock = ESP.getCpuFreqMHz();
  timer = timerBegin(0, cpuClock, true);
  timerAttachInterrUPt(timer, &triggerScan, true);
  timerAlarmWrite(timer, 300, true);
  timerAlarmEnable(timer);
  dmd.clearScreen( true );
}

void loop(void)
{
  //terima kirim data
  AvailableClients();
  AvailableMessage();
  dataint = Message.toInt();
  if (dataint > 200) {
    dataint = 0;
  }

  //tampil nilai Display
  dmd.selectFont(Arial_Black_16);
  dmd.dikasiInfoMaseh(dataint);
}

//open wifi akses poin
void SetWifi(char* Name, char* Password)
{
  WiFi.disconnect();
  WiFi.mode(WIFI_AP_STA);
  Serial.println("WIFI Mode : AccessPoint Station");
  ESPssid      = Name;
  ESPpassword  = Password;
  WiFi.softAP(ESPssid, ESPpassword);
  Serial.println("WIFI < " + String(ESPssid) + " > ... Started");
  delay(500);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AccessPoint IP : ");
  Serial.println(IP);
  Serial.print("AccessPoint MC : ");
  Serial.println(String(WiFi.softAPmacAddress()));
  ESPServer.begin();
  ESPServer.setNoDelay(true);
  Serial.println("Server Started");
}

//deteksi client konek
void AvailableClients()
{
  if (ESPServer.hasClient())
  {
    if (digitalRead(LED0) == HIGH) digitalWrite(LED0, LOW);
    for (uint8_t i = 0; i < MAXSC; i++)
    {
      if (!ESPClient[i] || !ESPClient[i].connected())
      {
```

```
    if (ESPClient[i])
    {
     ESPClient[i].stop();
    }
    if (ESPClient[i] = ESPServer.available())
    {
     Serial.println("New Client: " + String(i + 1));
    }
    continue;
   }
  }
  WiFiClient ESPClient = ESPServer.available();
  ESPClient.stop();
 }
 else
 {
  digitalWrite(LED0, HIGH);
  delay(250);
  digitalWrite(LED0, LOW);
  delay(250);
 }
}

//deteksi pesan
void AvailableMessage()
{
 for (uint8_t i = 0; i < MAXSC; i++)
 {
  if (ESPClient[i] && ESPClient[i].connected() &&
ESPClient[i].available())
  {
   while (ESPClient[i].available())
   {
    Message = ESPClient[i].ReadStringUntil('\n');
    ESPClient[i].flush();
    ClientNumber();
    dmd.clearScreen( true );
    Serial.println(Message);
   }
  }
 }
```

```
}

//nomor client
void  ClientNumber() {
  if (Message == "<Cliente 01-1>") {
    Serial.println("datamasuk1");
  }
  if (Message == "<Cliente 01-0>") {
    Serial.println("datamasuk0");
  }
}
```

F. Program ESP32 Untuk *Printer*

```
#include <WiFi.h>
#include <SoftwareSerial.h>
#include <BluetoothSerial.h>
#include "FS.h"
#include "SPIFFS.h"
#include "Adafruit_Thermal.h"
#include <EEPROM.h>
#include <Wire.h>
#define LED0 2

String dataserial, data_daftar, data_nohp, datarfid, data_antri,
cek_datarfid;
long proveus2;
bool en;
const String  ClientType       = "Printer RFID";
char*         ESPssid;
char*         ESPpassword;
int           ESPServerPort  = 9001;
String kondisi;

uint8_t address[6]  = {0x86, 0x67, 0x7a, 0x00, 0x76, 0xa2};
String name = "RPP02N";
char *pin = "0000"; //<- standard pin would be provided by
default
bool connectedd, a;

BluetoothSerial SerialBT;
```

```
Adafruit_Thermal printer(&SerialBT);
IPAddress      ESPServer(192, 168, 4, 1);
WiFiClient     ESPClient;
SoftwareSerial arduino;
SoftwareSerial esp;

void setUP() {
  //inialisai
  Serial.begin(115200);
  arduino.begin(57600, SWSERIAL_8N1, 16, 17, false);
  esp.begin(57600, SWSERIAL_8N1, 32, 33, false);
  SPIFFS.begin();
  delay(100);
  pinMode(LED0, OUTPUT);
  digitalWrite(LED0, !LOW);
  Serial.println("\nI/O Pins Modes Set .... Done");

  //  WiFi.mode(WIFI_STA);
  //  WiFi.begin("Mesin Printer", "mesinprinter");
  //  while (WiFi.status() == WL_CONNECTED)
  //  {
  //    WiFi.disconnect();
  //    WiFi.mode(WIFI_OFF);
  //    delay(500);
  //  }
  //
  //  WiFi.mode(WIFI_STA);
  //  CheckWiFiConnectivity();

 //inidikatpr konek bluetooth
  Serial.println("!--- Connecting To " + WiFi.SSID() + " ---!");
  digitalWrite(LED0, !HIGH);
  Serial.println("!-- Client Device Connected --!");

  Serial.println("Connected To      : " + String(WiFi.SSID()));
  Serial.println("Signal Strenght   : " + String(WiFi.RSSI()) + "
dBm");
  Serial.print  ("Server IP Address : ");
  Serial.println(ESPServer);
  Serial.print  ("Server Port Num   : ");
  Serial.println(ESPServerPort);
  Serial.print  ("Device MC Address : ");
  Serial.println(String(WiFi.macAddress()));
  Serial.print  ("Device IP Address : ");
  Serial.println(WiFi.localIP());

  SerialBT.setPin(pin);
  SerialBT.begin("ESP32test", true);
  Serial.println("The device started in master mode, make sure
remote BT device is on!");

  connectedd = SerialBT.connect(address);

  if (connectedd) {
    Serial.println("Connected Succesfully!");
  } else {
    while (!SerialBT.connected(10000)) {
      Serial.println("Failed to connect. Make sure remote device
is available and in range, then restart app.");
    }
  }
  if (SerialBT.disconnect()) {
    Serial.println("Disconnected Succesfully!");
  }

  SerialBT.connect();
  printer.begin();
}

void loop() {
  //  CheckWiFiConnectivity();
  //komunikasi
  komunikasi();
}

void komunikasi() {
  // {data_daftar/data_nohp/datarfid/data_antri}

 //serial komunikasi
  if (arduino.available()) {
    dataserial = arduino.ReadStringUntil('\n');
    en = 1;
  }

  // parsing data
  int batas1 = dataserial.indexOf('{');
  int batas2 = dataserial.indexOf('/', batas1 + 1);
  int batas3 = dataserial.indexOf('/', batas2 + 1);
  int batas4 = dataserial.indexOf('/', batas3 + 1);
  int batas5 = dataserial.indexOf('}', batas4 + 1);

  data_daftar = dataserial.substring(batas1 + 1, batas2);
  data_nohp = dataserial.substring(batas2 + 1, batas3);
  datarfid = dataserial.substring(batas3 + 1, batas4);
  data_antri = dataserial.substring(batas4 + 1, batas5);

  // logic baca penyimpanan nomor
  if (data_nohp == "" and datarfid != "" and data_daftar == ""
and data_antri == "" and en == 1) {
    cek_datarfid = datarfid;
    if (String(ReadFile(SPIFFS, String("/" + String(cek_datarfid)
+ ".txt").c_str())) != "") {
      delay(250);
      kondisi = "1";
      arduino.println(kondisi);
    }
    else {
      en = 0;
      a = 0;
      delay(250);
      kondisi = "0";
      arduino.println(kondisi);
      delay(250);
      Serial.println("tidak valid");
      kondisi = "";
      dataserial = "";
    }
  }
  // logic simpan nomor
  else if (data_nohp == "" and datarfid != "" and data_daftar ==
"" and data_antri != "" and en == 1) {
    delay(250);
    send_data(String(data_antri) + String(ReadFile(SPIFFS,
String("/" + String(cek_datarfid) + ".txt").c_str())));
    esp.println("{" + String(data_antri) + "/" +
String(ReadFile(SPIFFS, String("/" + String(cek_datarfid) +
".txt").c_str())) + "}");
    Serial.println("{" + String(data_antri) + "/" +
String(ReadFile(SPIFFS, String("/" + String(cek_datarfid) +
".txt").c_str())) + "}");
    delay(250);
    Serial.println("valid, print antrian");
    data_nohp = ReadFile(SPIFFS, String("/" +
String(cek_datarfid) + ".txt").c_str());
    printerx();
    a = 1;
    en = 0;
    kondisi = "";
    dataserial = "";
  }
  else if (data_nohp != "" and data_daftar != "" and datarfid !=
"" and data_antri != "") {
    WriteFile(SPIFFS, String("/" + data_daftar + ".txt").c_str(),
String(data_nohp).c_str());
    Serial.println("save data: " + data_daftar + ", " + data_nohp);
    Serial.println("save sukses");
  }

  //logic tampil monitor
  if ((millis() - proveus2) > 1000) {
    Serial.println("data_daftar: " + data_daftar);
    Serial.println("data_nohp: " + data_nohp);
    Serial.println("datarfid: " + datarfid);
    Serial.println("data_antri: " + data_antri);
    Serial.println("en: " + en);
    Serial.println("kondisi: " + kondisi);
    Serial.println("dataserial: " + dataserial);
    arduino.println(kondisi);
    proveus2 = millis();
  }
}

// send data to Display
void send_data(String pesan) {
  ESPClient.println(pesan);
  ESPClient.flush();
}

//cek konek wifi
void CheckWiFiConnectivity()
{
  while (WiFi.status() != WL_CONNECTED)
  {
    for (int i = 0; i < 10; i++)
    {
      digitalWrite(LED0, !HIGH);
      delay(250);
      digitalWrite(LED0, !LOW);
      delay(250);
      Serial.print(".");
    }
    Serial.println("");
  }
}

//kirim pesan
void ESPRequest()
{
  ESPClient.stop();
  if (ESPClient.connect(ESPServer, ESPServerPort))
  {
    Serial.println    ("<" + ClientType + "- CONNECTED>");
    ESPClient.println ("<" + ClientType + "- CONNECTED>");
  }
}

 //printer
void printerx() {
  printer.setFont('B');
  printer.setSize('M');
  printer.justify('C');
  printer.underlineOn();
  printer.println("<<<  ANTRIAN OTOMATIS >>>");
  printer.underlineOff();
  printer.println("                  ");

  printer.setFont('B');
  printer.setSize('S');
  printer.justify('C');
  printer.print("No hp: ");
  printer.print(data_nohp);
  printer.println("");
  printer.println("                  ");

  printer.setFont('B');
  printer.setSize('M');
  printer.justify('C');
  printer.println("-----------NOMOR ANTRIAN-----------");


  printer.setFont('B');
  printer.setSize('L');
  printer.justify('C');
  printer.println(data_antri);

  printer.setFont('B');
  printer.setSize('M');
  printer.justify('C');
  printer.println("----------TERIMA KASIH-----------");
  printer.println("                  ");

  printer.feed();
  printer.setDefault();
  delay(1000);
}


// baca
// contoh   sp_humd_bawah = ReadFile(SPIFFS,
"/sp_humd_bawah.txt").toFloat();
String ReadFile(fs::FS &fs, const char * path) {
  File file = fs.open(path, "r");
  if (!file || file.isDirectory())
  {
    return String();
  }
  String fileContent;
  while (file.available())
  {
    fileContent += String((char)file.Read());
  }
  return fileContent;
}

//simpan
//contoh   WriteFile(SPIFFS, "/sp_humd_bawah.txt", "30");
void WriteFile(fs::FS &fs, const char * path, const char *
message)
{
  File file = fs.open(path, "w");
  if (!file)
  {
    return;
  }
  file.print(message);
}
```

G. Program ESP DF*Player mini*

```
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DFPlayer_Mini_Mp3.h>
#include <SoftwareSerial.h>

#define      LED0    2
#define      call    5
#define      recal   15
#define      prefius  4

//rx 16 = tx 16
//tx 17 = rx 17

#define belas             12 //belas.mp3
#define puluh              13 //puluh.mp3
#define seratus            14 //seratus.mp3
```

```
#define ratus              15 //ratus.mp3
#define seribu             16 //seribu.mp3
#define ribu               17 //ribu.mp3
#define koma               18 //Koma.mp3
#define antrianNomor       101

int          ButtonState;
int          LastButtonState   = LOW;
int          LastDebounceTime  = 0;
int          DebounceDelay     = 50;
const String  ClientType       = "Led P10";
int          LEDState         = LOW;
unsigned long CurrMillis       = 0;
unsigned long PrevMillis       = 0;
unsigned long Interval         = 1000;
char*        ESPssid;
char*        ESPpassword;
int          ESPServerPort  = 9001;
long count = 0, proveus;
byte en;

LiquidCrystal_I2C lcd(0x27, 16, 2);
IPAddress       ESPServer(192, 168, 4, 1);
WiFiClient      ESPClient;
SoftwareSerial dfPlayer;
SoftwareSerial esp;

void setUP()
{
  //inisialisasi
  Serial.begin(115200);
  dfPlayer.begin(9600, SWSERIAL_8N1, 32, 33, false);
  esp.begin(57600, SWSERIAL_8N1, 16, 17, false);
  mp3_set_serial (dfPlayer);
  mp3_set_volume(30);
  lcd.init();
  lcd.backlight();
  pinMode(LED0, OUTPUT);
  pinMode(call, INPUT_PULLUP);
  pinMode(recal, INPUT_PULLUP);
  pinMode(prefius, INPUT_PULLUP);
```

```
digitalWrite(LED0, !LOW);
Serial.println("\nI/O Pins Modes Set .... Done");

//lcd tampil konek wifi
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connecting to...");
lcd.setCursor(0, 1);
lcd.print("Server");
WiFi.mode(WIFI_STA);
WiFi.begin("Mesin Antrian", "mesinantrian");

while (WiFi.status() == WL_CONNECTED)
{
  WiFi.disconnect();
  WiFi.mode(WIFI_OFF);
  lcd.setCursor(15, 1);
  lcd.print(">");
  delay(500);
  lcd.setCursor(15, 1);
  lcd.print(" ");
  delay(500);
  lcd.setCursor(15, 1);
  lcd.print(">");
}

WiFi.mode(WIFI_STA);
WiFi.begin("Mesin Antrian", "mesinantrian");
CheckWiFiConnectivity();

//lcd tampil sukses konek
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connected.");
delay(2000);
lcd.clear();

//indikator konek sukses
Serial.println("!--- Connecting To " + WiFi.SSID() + " ---!");
digitalWrite(LED0, !HIGH);
Serial.println("!-- Client Device Connected --!");
```

```
  Serial.println("Connected To      : " + String(WiFi.SSID()));
  Serial.println("Signal Strenght   : " + String(WiFi.RSSI()) + "
dBm");
  Serial.print  ("Server IP Address : ");
  Serial.println(ESPServer);
  Serial.print  ("Server Port Num   : ");
  Serial.println(ESPServerPort);
  Serial.print  ("Device MC Address : ");
  Serial.println(String(WiFi.macAddress()));
  Serial.print  ("Device IP Address : ");
  Serial.println(WiFi.localIP());

  ESPRequest();
}

void loop()
{
  CheckWiFiConnectivity();
  ReadButton();

  //kirim data antrian ke esp Whatsapp
  if ((millis() - proveus) > 1000) {
    esp.println(String(count));
    Serial.println(count);
    proveus = millis();
  }
}

void ReadButton() // logic panggil antrian dan suara
{
  int caller = digitalRead(call);
  int back = digitalRead(prefius);
  int callback = digitalRead(recal);
  Serial.println(String(caller) + String(" ") + String(back) +
String(" ") + String(callback));
  if ((millis() - PrevMillis) > 2000) {
    lcd.init();
    lcd.clear();
    PrevMillis = millis();
  }
```

```
lcd.setCursor(0, 0);
lcd.print("antrian saat ini");
lcd.setCursor(7, 1);
lcd.print(count);

if (caller == 0)
{
  count = count + 1;
  if (count > 200) {
    count = 0;
  }
  else if (count < 0) {
    count = 200;
  }
  esp.println(count);
  ESPClient.println(count);
  ESPClient.flush();
  mp3_play (101);
  delay(2000);
  suaraBilangan(count);
  delay(1000);
  Serial.println   (count);
}
else if (back == 0) {
  count = count - 1;
  if (count > 200) {
    count = 0;
  }
  else if (count < 0) {
    count = 200;
  }
  delay(500);
  Serial.println(count);
  esp.println(count);
  ESPClient.println(count);
  ESPClient.flush();
  mp3_play (101);
  delay(2000);
  suaraBilangan(count);
  delay(1000);
  Serial.println("ok");
  }
  else if (callback == 0) {
    Serial.println   (count);
    mp3_play (101);
    delay(2000);
    suaraBilangan(count);
    delay(1000);
    //    ESPClient.println(count);
    //    ESPClient.flush();
    Serial.println("ok");
  }
}


//cek konekting wifi
void CheckWiFiConnectivity()
{
  while (WiFi.status() != WL_CONNECTED)
  {
    for (int i = 0; i < 10; i++)
    {
      digitalWrite(LED0, !HIGH);
      delay(250);
      digitalWrite(LED0, !LOW);
      delay(250);
      Serial.print(".");
    }
    Serial.println("");
  }
}

void ESPRequest()
{
  ESPClient.stop();
  if (ESPClient.connect(ESPServer, ESPServerPort))
  {
    Serial.println    ("<" + ClientType + "- CONNECTED>");
    ESPClient.println ("<" + ClientType + "- CONNECTED>");
  }
}
```

```
//logic suara mp3
void mp3_play_and_wait(uint16_t num) {
  mp3_play (101);
  delay(3000);
  mp3_play (num);
  delay(1000);
}

void suaraBilangan(uint32_t Bilangan)
{
  if (Bilangan < 100)
  {
    suaraPuluhan(Bilangan);
  }
  else if (Bilangan < 1000)
  {
    suaraRatusan(Bilangan);
  }
  else
  {
    suaraRibuan(Bilangan);
  }
}
void suaraPuluhan(uint8_t Bilangan)
{
  if (Bilangan < 12)
  {
    mp3_play_and_wait(Bilangan);
  }
  else if (Bilangan < 20)
  {
    mp3_play_and_wait(Bilangan);
    mp3_play (Bilangan - 10);
    delay(3000);
    mp3_play(belas);
    delay(1000);
  }
  else
  {
    uint8_t puluhan = Bilangan / 10;
    mp3_play_and_wait(Bilangan);
    mp3_play(puluhan);
    delay(3000);
    mp3_play(puluh);
    delay(1000);

    puluhan *= 10;
    if (Bilangan - puluhan != 0)
    {
      mp3_play((Bilangan - puluhan));
      delay(1000);
    }
  }
}
void suaraRatusan(uint16_t Bilangan)
{
  uint8_t ratusan = (uint8_t)(Bilangan / 100);
  if (ratusan == 1)
  {
    mp3_play_and_wait(seratus);
    delay(1000);
  }
  else
  {
    mp3_play_and_wait(ratusan);
    delay(1000);
    mp3_play_and_wait(ratus);
    delay(1000);
  }
  if (Bilangan % 100)
  {
    suaraPuluhan(Bilangan - (ratusan * 100));
    delay(1000);
  }
}
void suaraRibuan(uint32_t Bilangan)
{
  uint16_t ribuan = (uint16_t)(Bilangan / 1000);
  if (ribuan == 1)
  {
    mp3_play_and_wait(seribu);
  }
  else if (ribuan < 100)
  {
    suaraPuluhan(ribuan);
    delay(1000);
    mp3_play_and_wait(ribu);
  }
  else
  {
    suaraRatusan(ribuan);
    delay(1000);
    mp3_play_and_wait(ribu);
  }
  if (Bilangan % 1000)
  {
    suaraRatusan(Bilangan - (ribuan * 1000));
  }
}
```

H. Program ESP32 Untuk Pesan *Whatsapp*

```
#include <SoftwareSerial.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <HTTPClient.h>
#include "FS.h"
#include "SPIFFS.h"

#define    LED0      2
#define    MAXSC     2

const char* ssid     = "p";
const char* password = "qwertyuiop";

String apikey = "359518818b2476c92b9f54c2";
String userkey = "546f3d32dc96";
String to = "089672396259";
String message = "Nomor Antrian Anda Akan Segera
Dipanggil";

int responehttp;
int interval_wa = 5;
int panggil;
int antrian_skrg;

String payload;
String data_antrian;
String data_printer;
String Message;
String Read_antrian;
String no_antrian;
String no_hp;
String en_wa;
String no_antri;
bool en;

char* ESPssid;
char* ESPpassword;

long proveus3;

HTTPClient http;
SoftwareSerial esp;
SoftwareSerial printer;
WiFiServer  ESPServer(9001);
WiFiClient  ESPClient[MAXSC];

void setUP() {
  //inialisasi serial
  Serial.begin(115200);
  esp.begin(57600, SWSERIAL_8N1, 32, 33, false);
  printer.begin(57600, SWSERIAL_8N1, 17, 16, false);
  SPIFFS.begin();
  delay(100);

  //indikator
  pinMode(LED0, OUTPUT);
  Serial.println("I/O Pins Modes Set .... Done");
  //  SetWifi("Mesin Printer", "mesinprinter");
  //konek wifi printer
  SetWifi("Mesin Printer", "0000000000000");

  //konek wifi
  Serial.print("Attempting to connect to SSID: ");
  WiFi.disconnect();
  delay(50);
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(250);
  }

  //indikator konek
  Serial.print("Connected to ");
  Serial.println(ssid);
}

void loop() {
  komunikasi();
  mainn();
}
```

I. Program Untuk Komunikasi ESP32 Ke *Server* Zenziva

```
void komunikasi() { //full komunikasi
  if (esp.available() > 0) {
    data_antrian = esp.ReadStringUntil('\n');
  }
  Serial.println("data_antrian: " + data_antrian);
  delay(250);

  if (printer.available() > 0) {
    data_printer = printer.ReadStringUntil('\n');
  }
  Serial.println("data_printer: " + data_printer);
  delay(250);

  if (data_antrian.toInt() > antrian_skrg){
    en = 1;
  }

  int batas1 = data_printer.indexOf('{');
  int batas2 = data_printer.indexOf('/', batas1 + 1);
  int batas3 = data_printer.indexOf('}', batas2 + 1);
  //format {no_antrian/no_hp}
  no_antrian = data_printer.substring(batas1 + 1, batas2);
  no_hp = data_printer.substring(batas2 + 1, batas3);
  delay(100);
  WriteFile(SPIFFS, String("/" + String(no_antrian) +
".txt").c_str(), no_hp.c_str());

  Serial.println("data: " + data_printer);
  Serial.println("no_antrian: " + no_antrian);
  Serial.println("no_hp: " + no_hp);
  Serial.println("en: " + String(en));
  Serial.println("antrian_skrg: " + String(antrian_skrg));
}

void server() {
  AvailableClients();
  AvailableMessage();
}

void SetWifi(char* Name, char* Password) // set wifi server
{
  WiFi.disconnect();
  WiFi.mode(WIFI_AP_STA);
  Serial.println("WIFI Mode : AccessPoint Station");
  ESPssid      = Name;
  ESPpassword  = Password;
  WiFi.softAP(ESPssid, ESPpassword);
  Serial.println("WIFI < " + String(ESPssid) + " > ... Started");
  delay(500);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AccessPoint IP : ");
  Serial.println(IP);
  Serial.print("AccessPoint MC : ");
  Serial.println(String(WiFi.softAPmacAddress()));
  ESPServer.begin();
  ESPServer.setNoDelay(true);
  Serial.println("Server Started");
}

void AvailableClients() //deteksi client
{
 if (ESPServer.hasClient())
 {
  if (digitalRead(LED0) == HIGH) digitalWrite(LED0, LOW);
  for (uint8_t i = 0; i < MAXSC; i++)
  {
   if (!ESPClient[i] || !ESPClient[i].connected())
   {
    if (ESPClient[i])
    {
     ESPClient[i].stop();
    }
    if (ESPClient[i] = ESPServer.available())
    {
     Serial.println("New Client: " + String(i + 1));
    }
    continue;
   }
  }
  WiFiClient ESPClient = ESPServer.available();
  ESPClient.stop();
 }
 else
 {
  digitalWrite(LED0, HIGH);
  delay(250);
  digitalWrite(LED0, LOW);
  delay(250);
 }
}

void AvailableMessage() / deteksi pesan
{
 for (uint8_t i = 0; i < MAXSC; i++)
 {
  if (ESPClient[i] && ESPClient[i].connected() &&
ESPClient[i].available())
  {
   while (ESPClient[i].available())
   {
    Message = ESPClient[i].ReadStringUntil('\n');
    ESPClient[i].flush();
```

```
        Serial.println(Message);
      }
    }
  }

  //  int batas1 = Message.indexOf('{');
  //  int batas2 = Message.indexOf('/', batas1 + 1);
  //  int batas3 = Message.indexOf('}', batas2 + 1);
  //
  //  //format {no_antrian/no_hp}
  //  no_antrian = Message.substring(batas1 + 1, batas2);
  //  no_hp = Message.substring(batas2 + 1, batas3);
  //  delay(100);
  //  WriteFile(SPIFFS, String("/" + String(no_antrian) +
".txt").c_str(), no_hp.c_str());
  //
  //  Serial.println("data: " + Message);
  //  Serial.println("no_antrian: " + no_antrian);
  //  Serial.println("no_hp: " + no_hp);
}

// contoh   sp_humd_bawah = ReadFile(SPIFFS,
"/sp_humd_bawah.txt").toFloat();
String ReadFile(fs::FS &fs, const char * path) { //baca data ke
spiffs
  File file = fs.open(path, "r");
  if (!file || file.isDirectory())
  {
    return String();
  }
  String fileContent;
  while (file.available())
  {
    fileContent += String((char)file.Read());
  }
  return fileContent;
}

//contoh   WriteFile(SPIFFS, "/sp_humd_bawah.txt", "30");
void WriteFile(fs::FS &fs, const char * path, const char *
message) // tulis data spiffs
```

```
{
  File file = fs.open(path, "w");
  if (!file)
  {
    return;
  }
  file.print(message);
}
```

J. Program Pengolahan Data Antrean *Whatsapp*

```
void mainn() {
  // data_antrian = antrian sekarang
  // no_antrian = antrian menunggu
  // interval_wa = kirim pesan/menunggu
  // rumus = (antrian sekarang - antrian mrnunggu) >
interval_wa
  panggil = data_antrian.toInt() + interval_wa;
  Serial.println("data_panggl: " + String(panggil));

  if (panggil <= no_antrian.toInt() && en == 1) { //logic kirim
whatsaap
    antrian_skrg = data_antrian.toInt();
    Serial.println("Panggil no: ");
    Serial.println(ReadFile(SPIFFS, String("/" + String(panggil)
+ ".txt").c_str()));
    to = ReadFile(SPIFFS, String("/" + String(panggil) +
".txt").c_str());
    delay(500);
    Whatsapp();
    delay(500);
    Serial.println("Panggil OK");
  }
}
```

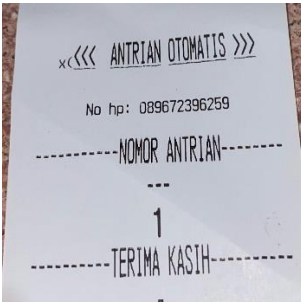
K. Program Kirim Pesan *Whatsapp*

```
void Whatsapp() { // full kirim pesan whatsaap
  if (WiFi.status() == WL_CONNECTED) {

http.begin("https://console.zenziva.net/wareguler/api/sendWA/"
);
    delay(50);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    String data = "userkey=" + userkey + "&passkey=" + apikey + "&to=" + to + "&message=" + message;
    int httpcode = http.POST(data);
    responehttp = http.GET();

    if (responehttp > 0) {
      Serial.print("HTTP Response code: ");
      Serial.println(responehttp);
      payload = http.getString();
      Serial.println(payload);
    }
    else {
      Serial.print("Error code: ");
      Serial.println(responehttp);
    }
  }

  else {
    Serial.println("WiFi Disconnected");
  }
  delay(1000);
  en = 0;
}
```

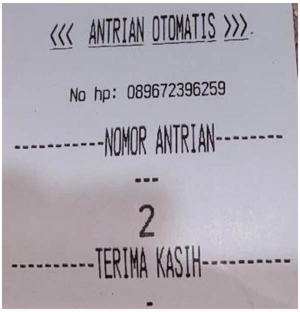
L. Program *Printer Whatsapp*

```
void Whatsapp() { // full kirim pesan whatsaap
  if (WiFi.status() == WL_CONNECTED) {

http.begin("https://console.zenziva.net/wareguler/api/sendWA/");
    delay(50);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    String data = "userkey=" + userkey + "&passkey=" + apikey + "&to=" + to + "&message=" + message;
    int httpcode = http.POST(data);
    responehttp = http.GET();

    if (responehttp > 0) {
      Serial.print("HTTP Response code: ");
      Serial.println(responehttp);
      payload = http.getString();
      Serial.println(payload);
    }
    else {
      Serial.print("Error code: ");
      Serial.println(responehttp);
    }
  }

  else {
    Serial.println("WiFi Disconnected");
  }
  delay(1000);
  en = 0;
}
```

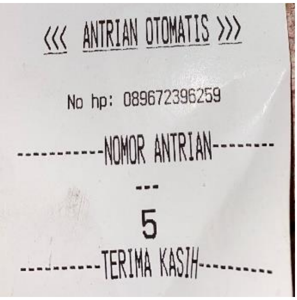
LAMPIRAN B
HASIL PRINT NOMOR ANTREAN



```
x(<<<  ANTRIAN OTOMATIS >>>

      No hp: 089672396259

-----------NOMOR ANTRIAN--------
               ---
                1
----------TERIMA KASIH----------
```



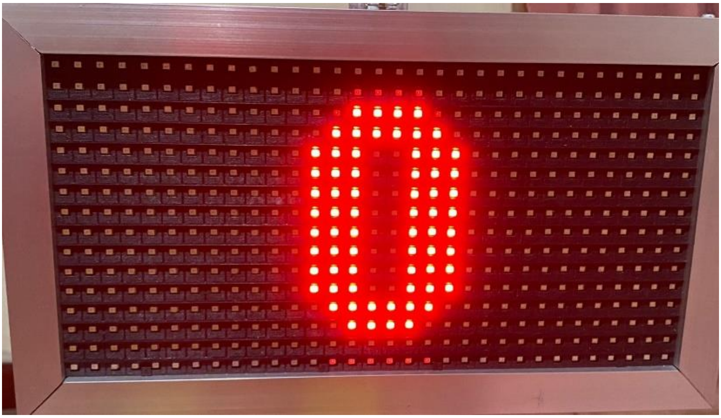
```
 <<<  ANTRIAN OTOMATIS >>>.

      No hp: 089672396259

-----------NOMOR ANTRIAN--------
               ---
                2
----------TERIMA KASIH----------
                -
```



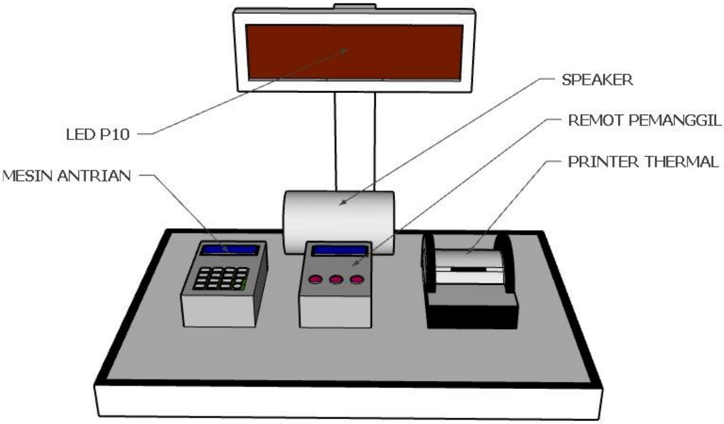
```
 <<<  ANTRIAN OTOMATIS >>>

      No hp: 089672396259

-----------NOMOR ANTRIAN--------
               ---
                5
----------TERIMA KASIH----------
```

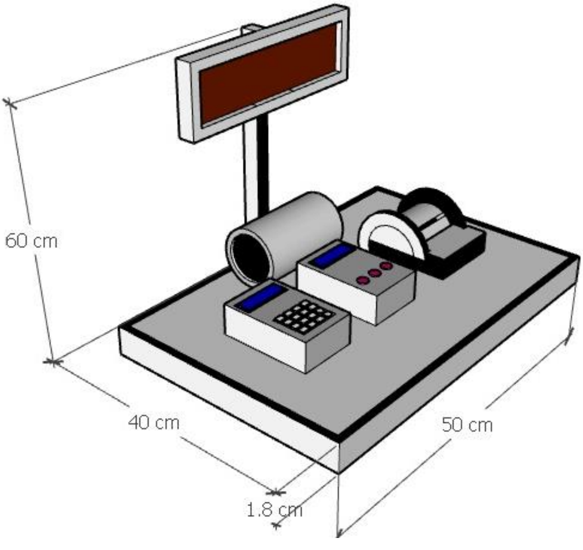
LAMPIRAN C

TAMPILAN PANEL *DISPLAY* P10



LAMPIRAN D
DESAIN MEKANIK ALAT





BIODATA PENULIS

Nama	: Bara Arya Putra Khahanan
Tempat/Tanggal Lahir	: Cilacap, 31 Desember 2001
Alamat	: JL.Soekarno-Hatta No.141 RT 05 / RW 01, Kec, Kesugihan, Kab. Cilacap
Email	: baraarya88@gmail.com
Telepon/Hp	: 085941387808
Hobi	: Berenang, Memasak, Membaca
Motto	: Kalo boleh pengen punya *passive income* yang gede dari *rest area.*
Riwayat Pendidikan	:

- SD Negeri Menganti 01Tahun 2007 – 2013
- SMP Negeri 7 Cilacap Tahun 2013 – 2016
- SMK Dr.Soetomo Cilacap Tahun 2016 – 2019
- Politeknik Negeri Cilacap Tahun 2019 – 2022

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 26 Juli 2022 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md)