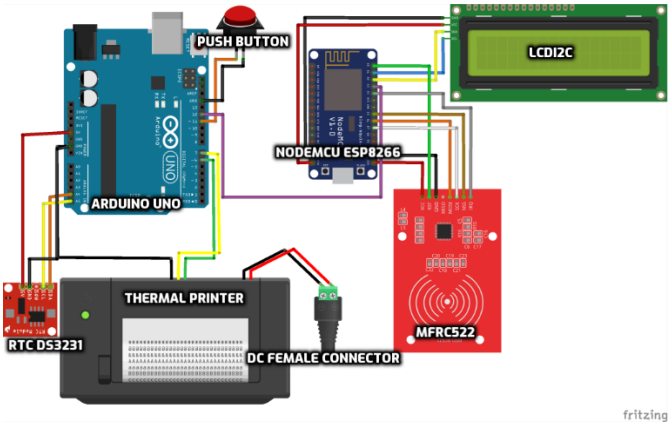
File Ciptaan

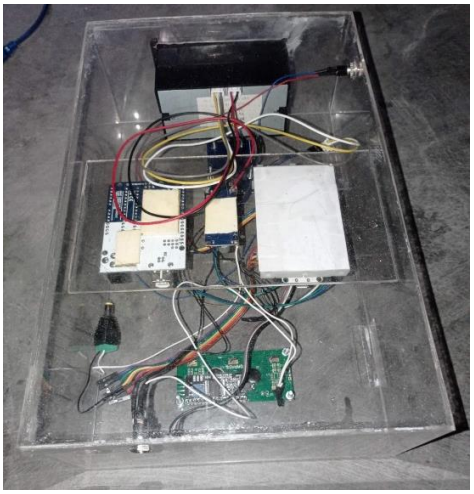
Lampiran 1

Gambar *Pemrogram Sistem Antrian Pelayanan Berbasis E-KTP dan Bukti Pajak*



Gambar Pengkawatan Keseluruhan Alat

Gambar Alat



Gambar Tampak Belakang Alat

Gambar Pengujian Alat

Gambar Pengujian Cetak Nomor Antrian

Lampiran 2.

KODE PROGRAM Nodemcu ESP8266

```
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>

#include <SoftwareSerial.h>
#include <WiFiClientSecure.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
#define SS_PIN D4
#define RST_PIN D0    // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance
#define PRINTER_PIN D3
const char* host = "script.google.com";
const int httpsPort = 443;
const char* fingerprint  = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6"; // for https

//**********Things to change*******************
const char* ssid = "Yoot";
const char* password = "gitarmuda";
String GOOGLE_SCRIPT_ID = "AKfycbyflZes3v8DKDrh2dYMpfcPmz8yxEvPSfErVgTZ"; // Replace by your
GAS service id
const String unitName = "headquarter"; // any name without spaces and special characters
//**********Things to change*******************
uint64_t PrinterMillis = 0;
WiFiClientSecure client;

void LcdClearAndPrint(String text)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(text);
}
```

```
void setup() {

  pinMode(PRINTER_PIN, OUTPUT);
  Serial.begin(115200);

  lcd.init(); // Init with pin default ESP8266 or ARDUINO
 //lcd.begin(2, 1); //ESP8266-01 I2C with pin 0-SDA 2-SCL
// Turn on the blacklight and print a message.
  lcd.backlight();
  LcdClearAndPrint("Loading");

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  Serial.println("Started");
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522
  delay(4);       // Optional delay. Some board do need more time after init to be ready, see Readme
  mfrc522.PCD_DumpVersionToSerial();  // Show details of PCD - MFRC522 Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
  lcd.setCursor(0, 0);
  lcd.print("Silahkan Tempel");
  lcd.setCursor(0, 1);
  lcd.print("E-KTP Anda...");
}
byte readCard[4];

void HandleDataFromGoogle(String data)
```

```
{
  int ind = data.indexOf(":");
  String access = data.substring(0, ind);
  int nextInd = data.indexOf(":", ind + 1);
  String name = data.substring(ind + 1, nextInd);
  String text = data.substring(nextInd + 1, data.length());

  Serial.println(name);
  LcdClearAndPrint(name);
  lcd.setCursor(0, 1);
  lcd.print(text);
  if (access=="-1")
  {
    lcd.print(" " + String("denied"));
//    Siren();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Silahkan Menuju");
    lcd.setCursor(0, 1);
    lcd.print("Ruangan Kadus");
    delay(5000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Silahkan Tempel");
    lcd.setCursor(0, 1);
    lcd.print("E-KTP Anda...");
  }
  else if(access=="any")
  {

    lcd.print(" " + String("Silahkan antri"));
    StartPrint();
  }
}
```

```
void StartPrint()
{
  PrinterMillis = millis()+5000;
  digitalWrite(PRINTER_PIN, HIGH);
  delay(5000);
}

void StopPrint()
{
  PrinterMillis = 0;
  digitalWrite(PRINTER_PIN, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Silahkan Tempel");
  lcd.setCursor(0, 1);
  lcd.print("E-KTP Anda...");
}

void loop() {
  if (PrinterMillis > 0 && PrinterMillis < millis())
  {
    StopPrint();
  }



  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
  if (!mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  Serial.println(F("Scanned PICC's UID:"));
  String uid = "";
```

```
  for (uint8_t i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
    Serial.print(readCard[i], HEX);
    uid += String(readCard[i],HEX);
  }
  Serial.println("");

  LcdClearAndPrint("Silahkan Tunggu-");
  String data = sendData("id=" + unitName + "&uid=" + uid,NULL);
  HandleDataFromGoogle(data);


  mfrc522.PICC_HaltA();


}

String sendData(String params, char* domain) {
  //google scripts requires two get requests
  bool needRedir = false;
  if (domain == NULL)
  {
    domain=(char*)host;
    needRedir = true;
    params = "/macros/s/" + GOOGLE_SCRIPT_ID + "/exec?" + params;
  }

  Serial.println(*domain);
  String result = "";
  client.setInsecure();
  Serial.print("connecting to ");
  Serial.println(host);
  if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return "";
  }
```

```
if (client.verify(fingerprint, domain)) {
}

Serial.print("requesting URL: ");
Serial.println(params);

client.print(String("GET ") + params + " HTTP/1.1\r\n" +
  "Host: " + domain + "\r\n" +
  "Connection: close\r\n\r\n");

Serial.println("request sent");
while (client.connected()) {

  String line = client.readStringUntil('\n');
  //Serial.println(line);
  if (needRedir) {

  int ind = line.indexOf("/macros/echo?user");
  if (ind > 0)
  {
    Serial.println(line);
    line = line.substring(ind);
    ind = line.lastIndexOf("\r");
    line = line.substring(0, ind);
    Serial.println(line);
    result = line;
  }
  }

  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
}
while (client.available()) {
```

```
    String line = client.readStringUntil('\n');
    if(!needRedir)
    if (line.length() > 5)
      result = line;
    //Serial.println(line);

    }
  if (needRedir)
    return sendData(result, "script.googleusercontent.com");
  else return result;


}
```

Lampiran 3

Kode Program Arduino Uno

```
#include <PString.h>
#include "Adafruit_Thermal.h"
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 RTC;

#include <SoftwareSerial.h>
#define TX_PIN 7
#define RX_PIN 6

SoftwareSerial mySerial(RX_PIN, TX_PIN);
Adafruit_Thermal printer(&mySerial);

int hitung=0;

const int NodemcuTrig = 12;
const int resetpb = 11;

int buttonState=0;
int lastButtonState = LOW;
long lastDebounceTime = 0;
long debounceDelay = 50;

void setup() {
  Serial.begin(9600);
   Wire.begin();
    RTC.begin();
    if (! RTC.isrunning())
  {
    Serial.println("RTC is NOT running!");

    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
```

```
 pinMode(NodemcuTrig, INPUT_PULLUP);
 pinMode(resetpb, INPUT_PULLUP);
 pinMode(NodemcuTrig, LOW);
  mySerial.begin(9600);



  printer.begin();

}

void loop() {

  buttonState = digitalRead(NodemcuTrig);
   if (buttonState == HIGH) {
     printer.wake();
  printer.setDefault();
    hitung+=1;
    char buffer[40];
  PString str(buffer, sizeof(buffer));
  if(hitung<100)
  {
    str.print(0);
    if(hitung<10)
    {
      str.print(0);
    }
  }
  else if(hitung<10)
    {
      str.print(0);
    }
  str.print(hitung);

printer.boldOn();
  printer.justify('C');
```

```
 printer.setSize('M');
printer.println(F("Nomor Antrian :"));

printer.setSize('L');
printer.println(str);
 printer.setSize('S');

 DateTime now = RTC.now();
str.begin();
str.print("tgl.");
str.print(now.day(), DEC);
str.print('-');
str.print(now.month(), DEC);
str.print('-');
str.print(now.year(), DEC);
str.print(' ');
int z=0;
z=now.hour(), DEC;
if(z<10)str.print('0');

str.print(z);
str.print(':');
z=now.minute(), DEC;
if(z<10)str.print('0');
str.print(now.minute(), DEC);
str.print(':');
z=now.second(), DEC;
if(z<10)str.print('0');
str.print(now.second(), DEC);
printer.println(str);
printer.println("PEMERINTAHAN");
printer.println("DESA RANDUBANGO");
  printer.feed(6);
printer.sleep();
delay(5000);
printer.wake();
```

```
  printer.setDefault();



  }
   if(!(digitalRead(resetpb))){
  printer.wake();
  printer.setDefault();
  printer.justify('C');

  printer.setSize('M');
  printer.println(F("Printer Direset"));
  hitung=0;
  printer.feed(6);
  printer.sleep();
  delay(3000);
  printer.wake();
  printer.setDefault();

   }
}
```

Lampiran 4

Kode Program Spreadsheet

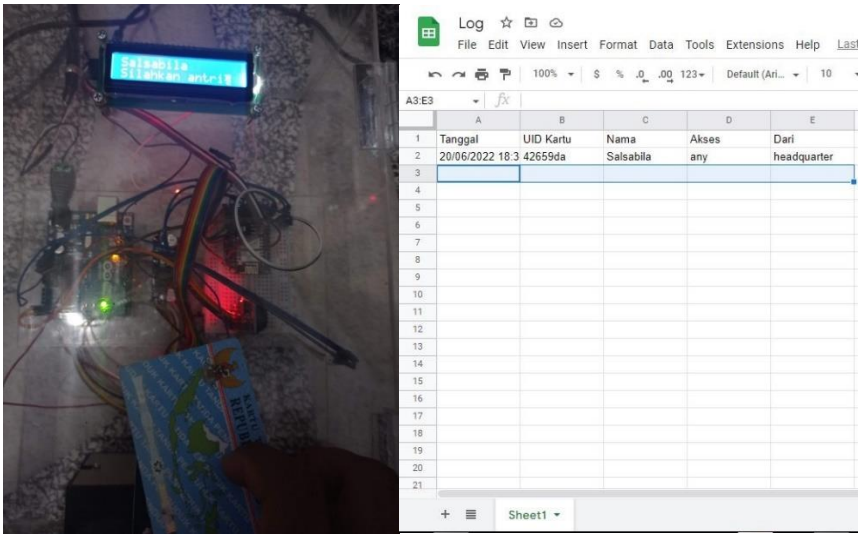
```
var timeZone="GMT+7";
var dateTimeFormat="dd/MM/yyyy HH:mm:ss";
var logSpreadSheetId="1djWlqhgvAZ3pCyB_wMvguAIwQcmm70HSA1oYI6NRPJI";

function sendEmail(message, id) {
    var subject = 'Something wrong with ' + id;
    MailApp.sendEmail(emailAddress, subject, message);

}

function doGet(e) {
   var access="-1";
      var text='go home';
      var name='Who are you?';
  var json;
  var error="idk";
    Logger.log(JSON.stringify(e)); // view parameters
    var result = 'Ok'; // assume success
    if (e.parameter == 'undefined') {
        result = 'No Parameters';
    } else {

        var uid = '';
        var onlyPing=false;
        var id = 'headquarter';
        var error = '';
        for (var param in e.parameter) {

            var value = stripQuotes(e.parameter[param]);

            switch (param) {
                case 'uid':
                    uid = value;
                    break;
```

```
            case 'id':
               id = value;
               break;

            default:
               result = "unsupported parameter";
         }
      }


    var sheet=SpreadsheetApp.getActive().getActiveSheet();

    var data = sheet.getDataRange().getValues();
  if (data.length == 0)
     return;
  for (var i = 0; i < data.length; i++) {

     if (data[i][0] ==uid)
     {
       name=data[i][1];
       access=data[i][2];
       text=data[i][3];
       break;
     }

  }


    addLog(uid,id,name,access);


     }
//    json = {
//   'access':access,
//   'name': name,
//   'text':text,
```

```
//    'error':error}

     result=(access+":"+name+":"+text);
  return ContentService.createTextOutput(result);
//  return ContentService.createTextOutput(JSON.stringify(json)
).setMimeType(ContentService.MimeType.JSON);
}


function addLog(uid,entrance, name,result) {


    var spr=SpreadsheetApp.openById(logSpreadSheetId);
    var sheet = spr.getSheets()[0];
    var data = sheet.getDataRange().getValues();

    var pos = sheet.getLastRow() + 1;

    var rowData = [];
  rowData[0] = Utilities.formatDate(new Date(), timeZone, dateTimeFormat);
  rowData[4]=entrance;
  rowData[1] = uid;
  rowData[2] = name;
  rowData[3] = result;
    var newRange = sheet.getRange(pos, 1, 1, rowData.length);
    newRange.setValues([rowData]);


}



/**
 * Remove leading and trailing single or double quotes
 */
function stripQuotes(value) {
```

```
    return value.replace(/^["']|["']$/g, "");
}
```

DISKRIPSI ALAT



Setiap warga yang akan minta pelayanan surat menyurat terlebih dahulu menuju ruang administrasi untuk mendapatkan antrianKTP akan terdeteksi dan terdapat nama sesuai dengan penamaan saat pendaftaran pada Data Base. E-KTP terdeteksi, contoh dengan UID “42659da” dengan nama “Salsabila” pada Data Base, pada LCD juga menampilkan nama dan juga Text silahkan antri. KTP yang terdaftar adalah LTP yang sudah bayar pajak. Jika belum bayar pajak, tidak mendapat nomor antrian sebelum pajaknya terbayarkan.