DAFTAR PUSTAKA

[1] N. Ramsari and A. Rifaldi, "Rancang Bangun aplikasi Penjadwalan Kegiatan Akademik Disertai Sistem Reminder Berbasis Responsive Web Design," *Fiki*, vol. IX, no. 1, pp. 1–11, 2018.

[2] S. Syaharani, "RANCANG BANGUN SMART MIRROR BERBASIS RASPBERRY PI 4 UNTUK HOME AUTOMATION," vol. 3, no. March, p. 6, 2021.

[3] "No Home, Mengenal Lebih Jauh Tentang Google Home," *SERBA SERBI, TECHNOLOGY*, 2018.

[4] R. Ramdhan Nugraha and I. Ikbal, "PEMBANGUNAN SMART MIRROR MENGGUNAKAN RASPBERRY PI BERBASIS IOT," *Pembang. SMART MIRROR MENGGUNAKAN RASPBERRY PI Berbas. IOT*, vol. 25, p. 8, 2019.

[5] A. Hanani and M. A. Hariyadi, "Smart Home Berbasis IoT Menggunakan Suara Pada Google Assistant," *J. Ilm. Teknol. Inf. Asia*, vol. 14, no. 1, p. 49, 2020, doi: 10.32815/jitika.v14i1.456.

[6] A. S. Nataprawira, A. Rizal, and A. S. Wibowo, "Perancangan Display Led Dot Matrix Via Wi-Fi Menggunakan Aplikasi Mobile Android," *Intech*, vol. 1, no. 1, pp. 1–7, 2020.

[7] M. F. D. Prayogo and W. S. Aji, "Design and Build Reminder for Arduino Based Drug Schedule Consumption System," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 2, no. 2, p. 75, 2020, doi: 10.12928/biste.v2i2.1031.

[8] D. Hadi, T. Aditya, and S. Winardi, "Rancang Bangun Jam Digital Pengingat Waktu Ibadah Berbasis Arduino," vol. 8, no. 1, p. 12, 2018.

[9] Y. Widiawati and P. H. Islam, "Pemanfaatan RTC (Real Time Clock) DS3231 Untuk Menghemat Daya," *Pros. Semin. Nas. Tek. Elektro*, vol. 3, pp. 287–289, 2018.

[10] D. F. W. Ilham Gantar Friansyah, Safe'I, "Implementasi Sistem Bluetooth Menggunakan Android dan Arduino untuk Kendali Peralatan Elektronik," vol. 2, no. 2, pp. 121–127, 2021.

[11] M. Albertus and M. Muliady, “Pengaturan Fan Speed dan Suhu Air Conditioner Melalui Ucapan Dengan Layanan Google Assistant API,” *TESLA J. Tek. Elektro*, vol. 21, no. 2, p. 170, 2020, doi: 10.24912/tesla.v21i2.7189.

[12] B. L. Indonesia, “LED Strip EPISTAR SMD 5050 | 12V IP67 Outdoor Waterproof,” *DEPO LED*, 2022.

[13] S. M. Theme, “Pengertian, Fungsi, dan Simbol LED (Light Emmiting Diode) | Gambar Skema Rangkaian Elektronika,” 2022.

[14] S. Dwiyatno, R. Iskandar, and E. Nuryani, “Pengendali Lampu Kantor Menggunakan Google Assistant Dan Adafruit. Io Berbasis Nodemcu Esp8266,” *J. Ilm. Sains dan Teknol.*, vol. 5, no. 1, pp. 14–23, 2020, doi: 10.47080/saintek.v5i1.1195.

[15] Z. Lubis and I. I. T. Pustaka, “Model Terbaru menggunakan perintah suara Untuk menstater Mesin Mobil dan keamanannya menggunakan SmartPhone Berbasis Arduino UNO,” vol. 7, no. 2, pp. 100–104, 2022.

[16] S. Huda, L. d. Mahfudz, and S. Kismiati, “Pengaruh Step down Protein dan Penambahan Acidifier pada Pakan terhadap Performans Ayam Broiler Effect of Step down Protein and Addition Acidifier of Feed on Broiler Chickens Performance S. Huda 1 , L. D. Mahfudz 2 dan S. Kismiati 2,” *J. Sain Peternak. Indones.*, vol. 14, no. 4, pp. 404–410, 2019.

[17] M. N. D. Satria, F. Saputra, and D. Pasha, “Mit App Invertor Pada Aplikasi Score Board Untuk Pertandingan Olahraga Berbasis Android,” *J. Teknoinfo*, vol. 14, no. 2, p. 81, 2020, doi: 10.33365/jti.v14i2.665.

[18] A. Rifa’i, “Sistem Pemantauan Dan Kontrol Otomatis Kualitas Air Berbasis Internet of Things (Iot) Menggunakan Platform Node-Red Untuk Budidaya Udang,” *JTT (Jurnal Teknol. Ter.*, vol. 7, no. 1, p. 19, 2021, doi: 10.31884/jtt.v7i1.317.

[19] M. Darwis, “Penambahan Fitur Tampilan LCD dan Micro SD Card Reader pada mesin Laser Engraver and Cutter di Laboratorium Pengemudian Listrik,” *J. Pengelolaan Lab. Pendidik.*, vol. 2, no. 1, pp. 8–18, 2020, doi: 10.14710/jplp.2.1.8-18.

[20] R. Ruuhwan, R. Rizal, and R. Kurniawan, "Pendeteksi Gerakan Menggunakan Sensor PIR untuk Sistem Keamanan di Ruang Kamar Berbasis SMS," *J. Inform. Univ. Pamulang*, vol. 5, no. 3, p. 281, 2020, doi: 10.32493/informatika.v5i3.5706.

LAMPIRAN A

```
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>
#include <Wire.h>
#include <RTCDS3231.h>
#include <ESP8266Firebase.h>
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <NTPClient.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define _SSID "zheviazhevi"
#define _PASSWORD "belladisna"
#define PROJECT_ID "smart-mirror-817c1-default-rtdb"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define IO_USERNAME  "smartmirrorpnc"
#define IO_KEY       "aio_QPyr16LHqDpoUZzlXS5Om1blmqQm"

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 4
#define CLK_PIN   0
#define CS_PIN    2
#define DATA_PIN  14
#define CLK_PIN1   12
#define CS_PIN1    13
#define DATA_PIN1  15
#define Relay 16

uint8_t scrollSpeed = 25;    // default frame delay value
textEffect_t scrollEffect = PA_SCROLL_LEFT;
textPosition_t scrollAlign = PA_LEFT;
uint16_t scrollPause = 2000; // in milliseconds

#define  BUF_SIZE  75
char curMessage[BUF_SIZE] = { "" };
char newMessage[BUF_SIZE] = { "Hello! Enter new message?" };
bool newMessageAvailable = true;

const uint16_t WAIT_TIME = 1000;
char datestring[20];
char datestring1[20];
char datestring2[20];

float suara;
int setpoin = 26;
int currentYear;
String currentDate;
String currentMonthName;
int currentMonth;
int monthDay;
String weekDay;
String data;
String text;
String mode;
int currentSecond;
int currentMinute;
int currentHour;
String formattedTime;
String Data_turnon, Data_turnoff;
long proveus, proveus1, proveus4, proveus5;
int count, count1;
int tanggal, bulan, tahun, jam, menit;
String xtanggal1, xbulan1, xtahun1, xjam1, xmenit1;
String xtanggal2, xbulan2, xtahun2, xjam2, xmenit2;
String xtanggal3, xbulan3, xtahun3, xjam3, xmenit3;
String xtanggal4, xbulan4, xtahun4, xjam4, xmenit4;
String xtanggal5, xbulan5, xtahun5, xjam5, xmenit5;
String xtanggal6, xbulan6, xtahun6, xjam6, xmenit6;
String xtanggal7, xbulan7, xtahun7, xjam7, xmenit7;
String acara1, acara2, acara3, acara4, acara5, acara6, acara7;
String waktu1, waktu2, waktu3, waktu4, waktu5, waktu6, waktu7;
String weekDays[7] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
```

```
String months[12] = {"January", "February", "March", "April",
"May", "June", "July", "August", "September", "October",
"November", "December"};
String datawaktu1, datawaktu2, datawaktu3, datawaktu4, datawaktu5,
datawaktu6, datawaktu7;
String saklar;
String data_jam, data_tanggal, data_tahun;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
Firebase firebase(PROJECT_ID);
MD_Parola P = MD_Parola(HARDWARE_TYPE, DATA_PIN,
CLK_PIN, CS_PIN, MAX_DEVICES);
MD_Parola P1 = MD_Parola(HARDWARE_TYPE, DATA_PIN1,
CLK_PIN1, CS_PIN1, MAX_DEVICES);
RTCDS3231<TwoWire> RTC(Wire);
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER,
AIO_SERVERPORT, IO_USERNAME, IO_KEY);
Adafruit_MQTT_Subscribe sub_turnon =
Adafruit_MQTT_Subscribe(&mqtt, IO_USERNAME
"/feeds/smartmirror");
Adafruit_MQTT_Publish pub_turnon =
Adafruit_MQTT_Publish(&mqtt, IO_USERNAME
"/feeds/smartmirror");
Adafruit_MQTT_Subscribe sub_turnoff =
Adafruit_MQTT_Subscribe(&mqtt, IO_USERNAME
"/feeds/turnoff");
Adafruit_MQTT_Publish pub_turnoff =
Adafruit_MQTT_Publish(&mqtt, IO_USERNAME "/feeds/turnoff");

void setup(void)
{
 Serial.begin(115200);
 WiFi.mode(WIFI_STA);
 WiFi.disconnect();
 delay(1000);
 Serial.println();
 Serial.println();
```

```
  Serial.print("Connecting to: ");
  Serial.println(_SSID);
  WiFi.begin(_SSID, _PASSWORD);

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

  Serial.println("");
  Serial.println("WiFi Connected");

  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

  mqtt.subscribe(&sub_turnon);
  mqtt.subscribe(&sub_turnoff);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(Relay, OUTPUT);
  digitalWrite(Relay, HIGH);
  digitalWrite(LED_BUILTIN, HIGH);
  P.begin();
  P1.begin();
  P1.displayText(curMessage, scrollAlign, scrollSpeed, scrollEffect,
scrollEffect, scrollEffect);
  setup_RTC();
  timeClient.begin();
  timeClient.setTimeOffset(25200);
}

void loop(void)
{
  parsing();
  loop_waktu();
  // 01/01/2000 00:00:00 bulan/hari/tahun jam:menit:detik
  tanggal = monthDay;
  bulan = currentMonth;
```

```
tahun = currentYear;
jam = currentHour;
menit = currentMinute;

data_jam = String(jam) + ":" + String(menit);
data_tanggal = String(tanggal) + "/" + String(bulan);
data_tahun =  String(tahun);

if (P1.displayAnimate())
{
  if (newMessageAvailable)
  {
    strcpy(curMessage, text.c_str());
    newMessageAvailable = false;
  }
  P1.displayReset();
}

MQTT_connect();
Adafruit_MQTT_Subscribe *subscription;

if ((subscription = mqtt.readSubscription(0))) {
  if (subscription == &sub_turnon) {
    Data_turnon = (char *)&sub_turnon.lastread;
    Serial.println("turn on: " + Data_turnon);
    if (Data_turnon == "1") {
      firebase.setString("switch", "1");
      saklar = firebase.getString("switch");
      mode = "turnon";
      newMessageAvailable = true;
      Serial.println("mode: " + mode);
      if (! pub_turnon.publish(0)) {
        Serial.println(F("Failed"));
      }
      else {
        Serial.println(F("OK!"));
      }
    }
  }
```

```
    else if (subscription == &sub_turnoff) {
      Data_turnoff = (char *)&sub_turnoff.lastread;
      Serial.println("turn off: " + Data_turnoff);
      if (Data_turnoff == "1") {
        firebase.setString("switch", "0");
        saklar = firebase.getString("switch");
        mode = "turnoff";
        Serial.println("mode: " + mode);
        P.print("            ");
        digitalWrite(Relay, HIGH);
        if (! pub_turnoff.publish(0)) {
          Serial.println(F("Failed"));
        }
        else {
          Serial.println(F("OK!"));
        }
      }
    }
  }

  if ((millis() - proveus4) > 10000) {
    P1.begin();
    P1.displayText(curMessage, scrollAlign, scrollSpeed, scrollEffect,
scrollEffect, scrollEffect);
    data = firebase.getString("data");
    saklar = firebase.getString("switch");
    Serial.println("data: " + data);
    proveus4 = millis();
  }

  if (mode == "turnon") {
    if (mode == "turnoff" or saklar == "0") {
      firebase.setString("switch", "0");
      mode = "turnoff";
    }
    Serial.println("Relay on");
    digitalWrite(Relay, LOW);
  if (tanggal == xtanggal1.toInt() and bulan == xbulan1.toInt()) {
   //    P1.print(acara1);
   text = "\\\\" + acara1;
  }
  else if (tanggal == xtanggal2.toInt() and bulan == xbulan2.toInt())
{
   //    P1.print(acara2);
   text = "\\\\" + acara2;
  }
  else if (tanggal == xtanggal3.toInt() and bulan == xbulan3.toInt())
{
   //    P1.print(acara3);
   text = "\\\\" + acara3;
  }
  else if (tanggal == xtanggal4.toInt() and bulan == xbulan4.toInt())
{
   //    P1.print(acara4);
   text = "\\\\" + acara4;
  }
  else if (tanggal == xtanggal5.toInt() and bulan == xbulan5.toInt())
{
   //    P1.print(acara5);
   text = "\\\\" + acara5;
  }
  else if (tanggal == xtanggal6.toInt() and bulan == xbulan6.toInt())
{
   //    P1.print(acara6);
   text = "\\\\" + acara6;
  }
  else if (tanggal == xtanggal7.toInt() and bulan == xbulan7.toInt())
{
   //    P1.print(acara7);
   text = "\\\\" + acara7;
  }
  else {
   text = "tidak ada acara";
   //    Serial.println("scrool");
  }
  if ((millis() - proveus1) > 1000) {
    newMessageAvailable = true;
    count1++;
    proveus1 = millis();
  }

  if (P1.displayAnimate())
  {
    if (newMessageAvailable)
    {
      strcpy(curMessage, text.c_str());
      newMessageAvailable = false;
    }
    P1.displayReset();
  }

  if ((millis() - proveus) > 3000) {
    count++;
    proveus = millis();
  }

  if (count > 3) {
    count = 1;
  }
  else if (count == 0) {
    count = 1;
  }

  if (count == 1) {
    P.print(" " + String(data_tanggal));
  }
  else if (count == 2) {
    P.print(" " + String(data_jam));
  }
  else if (count == 3) {
    P.print(" " + String(data_tahun));
  }
}
else if (mode == "turnoff") {
    if (mode == "turnon" or saklar == "1") {
      firebase.setString("switch", "1");
      mode = "turnon";
    }
    Serial.println("Relay off");
    P.print("          ");
    digitalWrite(Relay, HIGH);
    P.begin();
    P1.begin();
    P1.displayText(curMessage, scrollAlign, scrollSpeed, scrollEffect,
scrollEffect, scrollEffect);
  }
  else if (mode == "") {
    if (mode == "turnon" or saklar == "1") {
      firebase.setString("switch", "1");
      mode = "turnon";
    }
  }

}

void parsing() {
  //  \"{rapat\\/1-2-2019 12:1\\/guru\\/6-7-2022 12:30\\/ngajar\\/3-11-
2012 12:15\\/tishr\\/3-11-2012 12:15\\/masak\\/3-5-2021
21:15\\/ngising\\/13:12\\/madang\\/1-2-2018 21:58}\"

  int batas1 = data.indexOf('{');
  int batas2 = data.indexOf('/', batas1 + 1);
  int batas3 = data.indexOf('/', batas2 + 1);
  int batas4 = data.indexOf('/', batas3 + 1);
  int batas5 = data.indexOf('/', batas4 + 1);
  int batas6 = data.indexOf('/', batas5 + 1);
  int batas7 = data.indexOf('/', batas6 + 1);
  int batas8 = data.indexOf('/', batas7 + 1);
  int batas9 = data.indexOf('/', batas8 + 1);
  int batas10 = data.indexOf('/', batas9 + 1);
  int batas11 = data.indexOf('/', batas10 + 1);
  int batas12 = data.indexOf('/', batas11 + 1);
  int batas13 = data.indexOf('/', batas12 + 1);
int batas14 = data.indexOf('/', batas13 + 1);
int batas15 = data.indexOf('}', batas14 + 1);

acara1 = data.substring(batas1 + 1, batas2);
waktu1 = data.substring(batas2 + 1, batas3);
acara2 = data.substring(batas3 + 1, batas4);
waktu2 = data.substring(batas4 + 1, batas5);
acara3 = data.substring(batas5 + 1, batas6);
waktu3 = data.substring(batas6 + 1, batas7);
acara4 = data.substring(batas7 + 1, batas8);
waktu4 = data.substring(batas8 + 1, batas9);
acara5 = data.substring(batas9 + 1, batas10);
waktu5 = data.substring(batas10 + 1, batas11);
acara6 = data.substring(batas11 + 1, batas12);
waktu6 = data.substring(batas12 + 1, batas13);
acara7 = data.substring(batas13 + 1, batas14);
waktu7 = data.substring(batas14 + 1, batas15);

datawaktu1 = "{" + waktu1 + "}";
datawaktu2 = "{" + waktu2 + "}";
datawaktu3 = "{" + waktu3 + "}";
datawaktu4 = "{" + waktu4 + "}";
datawaktu5 = "{" + waktu5 + "}";
datawaktu6 = "{" + waktu6 + "}";
datawaktu7 = "{" + waktu7 + "}";

int bat1 = datawaktu1.indexOf('{');
int bat2 = datawaktu1.indexOf('-', bat1 + 1);
int bat3 = datawaktu1.indexOf('-', bat2 + 1);
int bat4 = datawaktu1.indexOf('-', bat3 + 1);
int bat5 = datawaktu1.indexOf(':', bat4 + 1);
int bat6 = datawaktu1.indexOf('}', bat5 + 1);

xtanggal1 = datawaktu1.substring(bat1 + 1, bat2);
xbulan1 = datawaktu1.substring(bat2 + 1, bat3);
xtahun1 = datawaktu1.substring(bat3 + 1, bat4);
xjam1 = datawaktu1.substring(bat4 + 1, bat5);
xmenit1 = datawaktu1.substring(bat5 + 1, bat6);
```

```
int bat12 = datawaktu2.indexOf('{');
int bat22 = datawaktu2.indexOf('-', bat12 + 1);
int bat32 = datawaktu2.indexOf('-', bat22 + 1);
int bat42 = datawaktu2.indexOf('-', bat32 + 1);
int bat52 = datawaktu2.indexOf(':', bat42 + 1);
int bat62 = datawaktu2.indexOf('}', bat52 + 1);

xtanggal2 = datawaktu2.substring(bat12 + 1, bat22);
xbulan2 = datawaktu2.substring(bat22 + 1, bat32);
xtahun2 = datawaktu2.substring(bat32 + 1, bat42);
xjam2 = datawaktu2.substring(bat42 + 1, bat52);
xmenit2 = datawaktu2.substring(bat52 + 1, bat62);

int bat13 = datawaktu3.indexOf('{');
int bat23 = datawaktu3.indexOf('-', bat13 + 1);
int bat33 = datawaktu3.indexOf('-', bat23 + 1);
int bat43 = datawaktu3.indexOf('-', bat33 + 1);
int bat53 = datawaktu3.indexOf(':', bat43 + 1);
int bat63 = datawaktu3.indexOf('}', bat53 + 1);

xtanggal3 = datawaktu3.substring(bat13 + 1, bat23);
xbulan3 = datawaktu3.substring(bat23 + 1, bat33);
xtahun3 = datawaktu3.substring(bat33 + 1, bat43);
xjam3 = datawaktu3.substring(bat43 + 1, bat53);
xmenit3 = datawaktu3.substring(bat53 + 1, bat63);

int bat14 = datawaktu4.indexOf('{');
int bat24 = datawaktu4.indexOf('-', bat14 + 1);
int bat34 = datawaktu4.indexOf('-', bat24 + 1);
int bat44 = datawaktu4.indexOf('-', bat34 + 1);
int bat54 = datawaktu4.indexOf(':', bat44 + 1);
int bat64 = datawaktu4.indexOf('}', bat54 + 1);

xtanggal4 = datawaktu4.substring(bat14 + 1, bat24);
xbulan4 = datawaktu4.substring(bat24 + 1, bat34);
xtahun4 = datawaktu4.substring(bat34 + 1, bat44);
xjam4 = datawaktu4.substring(bat44 + 1, bat54);
xmenit4 = datawaktu4.substring(bat54 + 1, bat64);
```

```
  int bat15 = datawaktu5.indexOf('{');
  int bat25 = datawaktu5.indexOf('-', bat15 + 1);
  int bat35 = datawaktu5.indexOf('-', bat25 + 1);
  int bat45 = datawaktu5.indexOf('-', bat35 + 1);
  int bat55 = datawaktu5.indexOf(':', bat45 + 1);
  int bat65 = datawaktu5.indexOf('}', bat55 + 1);

  xtanggal5 = datawaktu5.substring(bat15 + 1, bat25);
  xbulan5 = datawaktu5.substring(bat25 + 1, bat35);
  xtahun5 = datawaktu5.substring(bat35 + 1, bat45);
  xjam5 = datawaktu5.substring(bat45 + 1, bat55);
  xmenit5 = datawaktu5.substring(bat55 + 1, bat65);

  int bat16 = datawaktu6.indexOf('{');
  int bat26 = datawaktu6.indexOf('-', bat16 + 1);
  int bat36 = datawaktu6.indexOf('-', bat26 + 1);
  int bat46 = datawaktu6.indexOf('-', bat36 + 1);
  int bat56 = datawaktu6.indexOf(':', bat46 + 1);
  int bat66 = datawaktu6.indexOf('}', bat56 + 1);

  xtanggal6 = datawaktu6.substring(bat16 + 1, bat26);
  xbulan6 = datawaktu6.substring(bat26 + 1, bat36);
  xtahun6 = datawaktu6.substring(bat36 + 1, bat46);
  xjam6 = datawaktu6.substring(bat46 + 1, bat56);
  xmenit6 = datawaktu6.substring(bat56 + 1, bat66);

  int bat17 = datawaktu7.indexOf('{');
  int bat27 = datawaktu7.indexOf('-', bat17 + 1);
  int bat37 = datawaktu7.indexOf('-', bat27 + 1);
  int bat47 = datawaktu7.indexOf('-', bat37 + 1);
  int bat57 = datawaktu7.indexOf(':', bat47 + 1);
  int bat67 = datawaktu7.indexOf('}', bat57 + 1);

  xtanggal7 = datawaktu7.substring(bat17 + 1, bat27);
  xbulan7 = datawaktu7.substring(bat27 + 1, bat37);
  xtahun7 = datawaktu7.substring(bat37 + 1, bat47);
  xjam7 = datawaktu7.substring(bat47 + 1, bat57);
  xmenit7 = datawaktu7.substring(bat57 + 1, bat67);
}
```

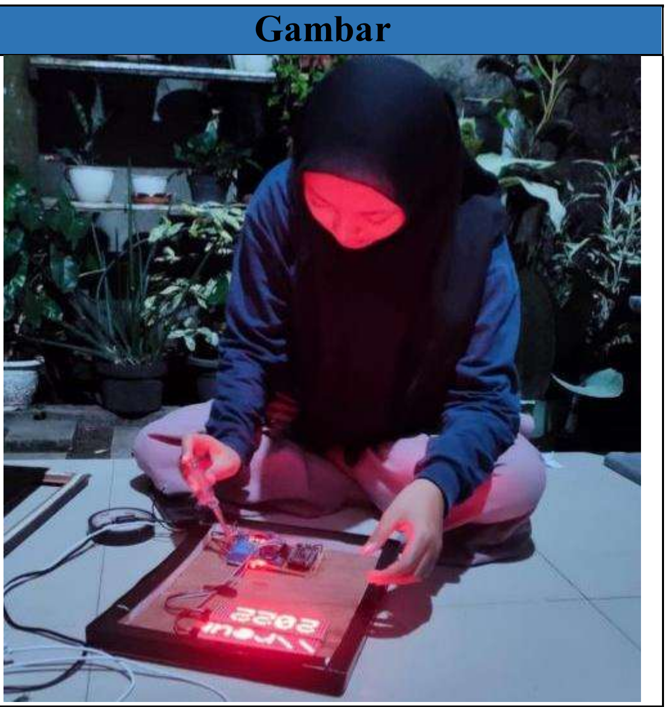
```
#define countof(a) (sizeof(a) / sizeof(a[0]))

void printDateTime(const RTCDateTime & dt)
{
  snprintf_P(datestring,
           countof(datestring),
           PSTR("%02u/%02u"),
           dt.Day(),
           dt.Month());

  snprintf_P(datestring1,
           countof(datestring1),
           PSTR("%02u:%02u"),
           dt.Hour(),
           dt.Minute() );

  snprintf_P(datestring2,
           countof(datestring2),
           PSTR("%04u"),
           dt.Year() );

}

void loop_waktu() {
  timeClient.update();

  time_t epochTime = timeClient.getEpochTime();
  formattedTime = timeClient.getFormattedTime();
  currentHour = timeClient.getHours();
  currentMinute = timeClient.getMinutes();
  currentSecond = timeClient.getSeconds();
  weekDay = weekDays[timeClient.getDay()];

  struct tm *ptm = gmtime ((time_t *)&epochTime);
  monthDay = ptm->tm_mday;
  currentMonth = ptm->tm_mon + 1;
  currentMonthName = months[currentMonth - 1];
  currentYear = ptm->tm_year + 1900;
  currentDate = String(currentYear) + "-" + String(currentMonth) + "-
" + String(monthDay);

  if ((millis() - proveus5) > 1000) {
    Serial.println("menitreal: " + String(menit));
    Serial.println("jamreal: " + String(jam));
    Serial.println("tahunreal: " + String(tahun));
    Serial.println("bulanreal: " + String(bulan));
    Serial.println("tanggalreal: " + String(tanggal));
    Serial.println("acara1: " + acara1);
    Serial.println("acara2: " + acara2);
    Serial.println("acara3: " + acara3);
    Serial.println("acara4: " + acara4);
    Serial.println("acara5: " + acara5);
    Serial.println("acara6: " + acara6);
    Serial.println("acara7: " + acara7);
    Serial.println("waktu1: " + waktu1);
    Serial.println("waktu2: " + waktu2);
    Serial.println("waktu3: " + waktu3);
    Serial.println("waktu4: " + waktu4);
    Serial.println("waktu5: " + waktu5);
    Serial.println("waktu6: " + waktu6);
    Serial.println("waktu7: " + waktu7);
    Serial.println("xtanggal1: " + xtanggal1);
    Serial.println("xbulan1: " + xbulan1);
    Serial.println("xtahun1: " + xtahun1);
    Serial.println("xjam1: " + xjam1);
    Serial.println("xmenit1: " + xmenit1);
    Serial.println("xtanggal2: " + xtanggal2);
    Serial.println("xbulan2: " + xbulan2);
    Serial.println("xtahun2: " + xtahun2);
    Serial.println("xjam2: " + xjam2);
    Serial.println("xmenit2: " + xmenit2);
    Serial.println("xtanggal3: " + xtanggal3);
    Serial.println("xbulan3: " + xbulan3);
    Serial.println("xtahun3: " + xtahun3);
    Serial.println("xjam3: " + xjam3);
    Serial.println("xmenit3: " + xmenit3);
Serial.println("xtanggal4: " + xtanggal4);
Serial.println("xbulan4: " + xbulan4);
Serial.println("xtahun4: " + xtahun4);
Serial.println("xjam4: " + xjam4);
Serial.println("xmenit4: " + xmenit4);
Serial.println("xtanggal5: " + xtanggal5);
Serial.println("xbulan5: " + xbulan5);
Serial.println("xtahun5: " + xtahun5);
Serial.println("xjam5: " + xjam5);
Serial.println("xmenit5: " + xmenit5);
Serial.println("xtanggal6: " + xtanggal6);
Serial.println("xbulan6: " + xbulan6);
Serial.println("xtahun6: " + xtahun6);
Serial.println("xjam6: " + xjam6);
Serial.println("xmenit6: " + xmenit6);
Serial.println("xtanggal7: " + xtanggal7);
Serial.println("xbulan7: " + xbulan7);
Serial.println("xtahun7: " + xtahun7);
Serial.println("xjam7: " + xjam7);
Serial.println("xmenit7: " + xmenit7);
Serial.print("Epoch Time: ");
Serial.println(epochTime);
Serial.print("Format Waktu: ");
Serial.println(formattedTime);
Serial.print("jam: ");
Serial.println(currentHour);
Serial.print("menit: ");
Serial.println(currentMinute);
Serial.print("detik: ");
Serial.println(currentSecond);
Serial.print("hari: ");
Serial.println(weekDay);
Serial.print("Month day: ");
Serial.println(monthDay);
Serial.print("bulan: ");
Serial.println(currentMonth);
Serial.print("nama bulan: ");
Serial.println(currentMonthName);
Serial.print("tahun: ");
```

```
  Serial.println(currentYear);
  Serial.print("Format Tanggal: ");
  Serial.println(currentDate);
  Serial.print(datestring);
  Serial.print(datestring1);
  Serial.println(datestring2);
  Serial.println("suara: " + String(suara));
  Serial.println("saklar: " + saklar);
  Serial.println("");
  proveus5 = millis();
 }
}

void setup_RTC() {
 Serial.print("compiled: ");
 Serial.print(__DATE__);
 Serial.println(__TIME__);
 RTC.Begin();
 RTCDateTime compiled = RTCDateTime(__DATE__, __TIME__);
 printDateTime(compiled);
 Serial.println();

 if (!RTC.IsDateTimeValid())
 {
  if (RTC.LastError() != 0)
  {
   Serial.print("RTC communications error = ");
   Serial.println(RTC.LastError());
  }
  else
  {
   Serial.println("RTC lost confidence in the DateTime!");
   RTC.SetDateTime(compiled);
  }
 }

 if (!RTC.GetIsRunning())
 {
  Serial.println("RTC was not actively running, starting now");
```

```
    RTC.SetIsRunning(true);
  }

  RTCDateTime now = RTC.GetDateTime();
  if (now < compiled)
  {
    Serial.println("RTC is older than compile time!  (Updating
DateTime)");
    RTC.SetDateTime(compiled);
  }
  else if (now > compiled)
  {
    Serial.println("RTC is newer than compile time. (this is expected)");
  }
  else if (now == compiled)
  {
    Serial.println("RTC is the same as compile time! (not expected but
all is fine)");
  }

  RTC.Enable32kHzPin(false);
  RTC.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
}

void MQTT_connect() {
  int8_t ret;

  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);
  retries--;
  if (retries == 0) {
    while (1);
  }
}
Serial.println("MQTT Connected!");
```

LAMPIRAN B

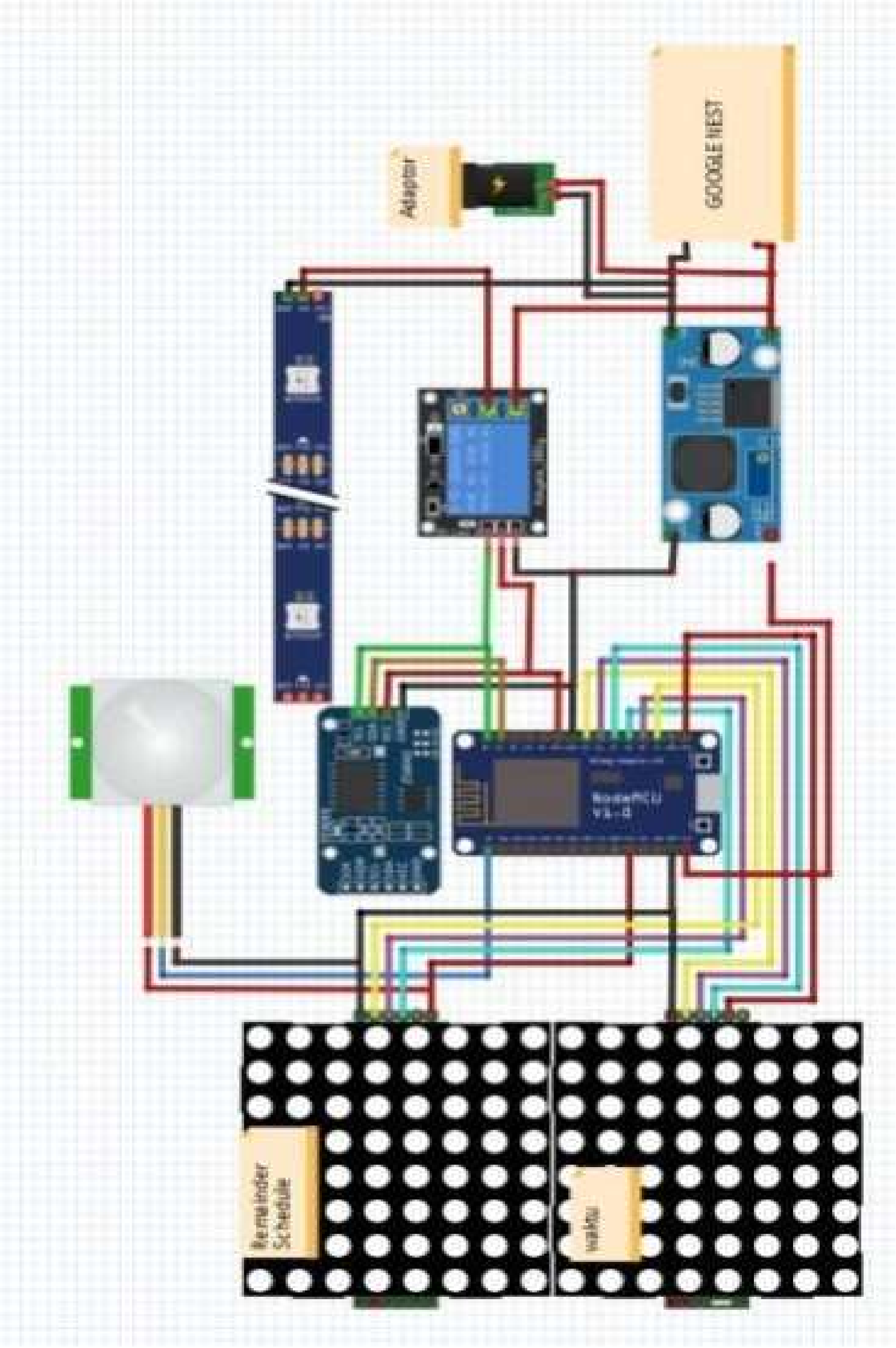
Dokumentasi kegiatan

NO	Keterangan	Gambar
1.	Kegiatan uji coba alat	
2.	Pemasangan rangakain elektrik	

3.	Melakukan pengambilan data	

LAMPIRAN C

Rangkaian Elektrik Keseluruhan



BIODATA PENULIS

Nama	: Zhevia Bella Cantika
Tempat/ Tanggal Lahir	: Cilacap, 05 September 2001
Alamat	: Jl Damar, Rt 01/10, Karang Talun, Cilacap Utara, Jawa Tengah
Telephone/ Hp	: 0858-6913-2300
Hobi	: Melukis, Menyanyi, Kuliner,Traveling
Motto	: Tuhan lebih tau yang terbaik untuk Hamba-NYA
Kontak Person	
Instagram	: @zheviabela
Email	: zheviab@gmail.com

Riwayat Pendidikan

• SD Bukit Bestari 12 Tanjung pinang	: 2008 - 2011
• SDN Gumilir 01 Cilacap	: 2011- 2013
• SMP Negeri 8 Cilacap	: 2014 - 2016
• SMK Negeri 1 Cilacap	: 2017- 2019
• Politeknik Negeri Cilacap	: 2019 - 2022

Penulis telah mengikuti Tugas Akhir pada 15 Agustus 2022 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.md).