DAFTAR LAMPIRAN

LAMPIRAN – *1* *Source code ESP32* menggunakan *visual studio code*

```
#include <Arduino.h>
#include <ArduinoJson.h>
#include <WiFi.h>
#include "AsyncJson.h"
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <DMD32.h>
#include <WaktuSholat.h>
#include "time.h"
#include <LITTLEFS.h>
#include "WebPage.h"
#include "fonts/HoboStd12.h"
#include "fonts/HoboStd16.h"
#include "fonts/CooperBlack10.h"
#include "fonts/ElektronMart5x6.h"
#include "fonts/ElektronMart6x16.h"

#define SPIFFS LITTLEFS

const char *ssid = "OPPO A31";
const char *password = "FADIL12345";
AsyncWebServer server(80);

struct Config
{
  double latitude;
  double longitude;
  int zonaWaktu;
  String listInformasi;
};

struct Iqomah
{
  int subuh;
  int zuhur;
  int asar;
  int magrib;
  int isya;
};

const char *fileConfig = "/config.json";
Config config;
Config configShow;
```

```
const char *fileIqomah =
"/iqomah.json";
Iqomah iqomah;

bool updateInformasi;
bool retInformasi = true;
int indexInformasi = 0;

const char *ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 28800; //
GMT +8
const int daylightOffset_sec = 3600;

struct tm timeinfo;

struct Waktu
{
  int tahun;
  int bulan;
  int tanggal;
  int jam;
  int menit;
  int detik;
};
Waktu waktu;
Waktu waktuIqomah;

double times[sizeof(TimeName) /
sizeof(char *)];
int indexTime[] = {0, 1, 2, 3, 5, 6, 7};
int indexTimeActive = 0;

// 0 = tampil tanggal; 1 = tampil waktu
sholat; 2 = tampil adzan; 3 = tampil
iqomah;
int indexCetakTampil = 0;

#define
INTERVAL_CETAK_WAKTU_SHOL
AT 5000
unsigned long time_cetak_waktu = 0;
unsigned long time_cetak_informasi =
0;
unsigned long time_iqomah = 0;

// Fire up the DMD library as dmd
#define KOLOM 3
#define BARIS 1
```

```
DMD dmd(KOLOM, BARIS);

// Timer setup
// create a hardware timer  of ESP32
hw_timer_t *timer = NULL;

void initLED(boolean first);
void initWaktuSholat();
void loadFile();
void saveFile(const char *filePath,
JsonObject json);
void setWaktu(int yr, int month, int
mday, int hr, int minute, int sec, int
isDst);
void updateWaktu();
void cetakWaktu();
void cetakTanggal();
void cetakJamSholat();
void cetakInfo();
void cetakAdzan();
void cetakIqomah();
void IRAM_ATTR triggerScan();
void textCenter(int y, char *Msg);
void
handleGetIndex(AsyncWebServerReque
st *request);
void
handlePostWaktu(AsyncWebServerReq
uest *request);
void
handleGetPostLokasi(AsyncWebServer
Request *request);
void
handleGetPostInformasi(AsyncWebServ
erRequest *request);
void
handleGetPostIqomah(AsyncWebServer
Request *request);

void setup(void)
{
  Serial.begin(115200);

  delay(1000);

  // Initialize SPIFFS
  if (!SPIFFS.begin(true))
  {
```

```
    Serial.println("An Error has occurred
while mounting SPIFFS");
    return;
  }

  loadFile();

  // connect to WiFi
  Serial.printf("Menghubungkan ke %s
...", ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() !=
WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" Terhubung");
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  server.on("/", handleGetIndex);
  server.on("/waktu", handlePostWaktu);
  server.on("/lokasi",
handleGetPostLokasi);
  server.on("/informasi",
handleGetPostInformasi);
  server.on("/iqomah",
handleGetPostIqomah);

  // Start server

DefaultHeaders::Instance().addHeader("
Access-Control-Allow-Origin", "*");
  server.begin();

  // init and get the time
  configTime(gmtOffset_sec,
daylightOffset_sec, ntpServer);
  updateWaktu();

  Serial.println(&timeinfo, "%A, %B %d
%Y %H:%M:%S");

  initWaktuSholat();
  initLED(true);
}
```

```
void loop(void)
{
  updateWaktu();
  cetakWaktu();
  cetakJamSholat();
  cetakTanggal();
  cetakInfo();
  cetakAdzan();
  cetakIqomah();
}

//
============================
========= WEB SERVER
==================//
void
handleGetIndex(AsyncWebServerReque
st *request)
{
  request->send_P(200, "text/html",
webpage);
}

void
handlePostWaktu(AsyncWebServerReq
uest *request)
{
  AsyncResponseStream *response =
request-
>beginResponseStream("application/jso
n");

  waktu.tahun = request-
>getParam("tahun", true)-
>value().toInt();
  waktu.bulan = request-
>getParam("bulan", true)-
>value().toInt();
  waktu.tanggal = request-
>getParam("tanggal", true)-
>value().toInt();
  waktu.jam = request->getParam("jam",
true)->value().toInt();
  waktu.menit = request-
>getParam("menit", true)-
>value().toInt();

  timerDetachInterrupt(timer);
```

```
  setWaktu(waktu.tahun, waktu.bulan,
waktu.tanggal, waktu.jam, waktu.menit,
0, 0);

  StaticJsonDocument<512> json;

  json["error"] = false;
  json["message"] = "Waktu berhasil
diperbaharui";

  serializeJson(json, *response);
  json.clear();

  request->send(response);

  initWaktuSholat();
  delay(500);
  initLED(false);
}

void
handleGetPostLokasi(AsyncWebServer
Request *request)
{
  AsyncResponseStream *response =
request-
>beginResponseStream("application/jso
n");

  StaticJsonDocument<512> json;

  if (request->method() == HTTP_GET)
  {
    json["latitude"] = config.latitude;
    json["longitude"] = config.longitude;
    json["zonaWaktu"] =
config.zonaWaktu;
  }
  else if (request->method() ==
HTTP_POST)
  {
    config.latitude = request-
>getParam("latitude", true)-
>value().toDouble();
    config.longitude = request-
>getParam("longitude", true)-
>value().toDouble();
```

```
    config.zonaWaktu = request-
>getParam("zonaWaktu", true)-
>value().toInt();

    json["latitude"] = config.latitude;
    json["longitude"] = config.longitude;
    json["zonaWaktu"] =
config.zonaWaktu;
    json["listInformasi"] =
config.listInformasi;

    saveFile(fileConfig,
json.as<JsonObject>());
  }

  serializeJson(json, *response);
  json.clear();

  request->send(response);
}

void
handleGetPostInformasi(AsyncWebServ
erRequest *request)
{
  AsyncResponseStream *response =
request-
>beginResponseStream("application/jso
n");

  StaticJsonDocument<512> json;

  if (request->method() == HTTP_GET)
  {
    deserializeJson(json,
config.listInformasi);
  }
  else if (request->method() ==
HTTP_POST)
  {
    config.listInformasi = request-
>getParam("listInformasi", true)-
>value();

    Serial.println(config.listInformasi);

    json["latitude"] = config.latitude;
    json["longitude"] = config.longitude;
    json["zonaWaktu"] =
config.zonaWaktu;
```

```
    json["listInformasi"] =
config.listInformasi;
    saveFile(fileConfig,
json.as<JsonObject>());
    updateInformasi = true;
  }

  // json.shrinkToFit();
  serializeJson(json, *response);
  json.clear();

  request->send(response);
}

void
handleGetPostIqomah(AsyncWebServer
Request *request)
{
  AsyncResponseStream *response =
request-
>beginResponseStream("application/jso
n");

  StaticJsonDocument<512> json;
  if (request->method() == HTTP_GET)
  {
    json["subuh"] = iqomah.subuh;
    json["zuhur"] = iqomah.zuhur;
    json["asar"] = iqomah.asar;
    json["magrib"] = iqomah.magrib;
    json["isya"] = iqomah.isya;
  }
  else if (request->method() ==
HTTP_POST)
  {
    iqomah.subuh = request-
>getParam("subuh", true)-
>value().toInt();
    iqomah.zuhur = request-
>getParam("zuhur", true)-
>value().toInt();
    iqomah.asar = request-
>getParam("asar", true)->value().toInt();
    iqomah.magrib = request-
>getParam("magrib", true)-
>value().toInt();
    iqomah.isya = request-
>getParam("isya", true)->value().toInt();
```

```
    json["subuh"] = iqomah.subuh;
    json["zuhur"] = iqomah.zuhur;
    json["asar"] = iqomah.asar;
    json["magrib"] = iqomah.magrib;
    json["isya"] = iqomah.isya;
    saveFile(fileIqomah,
json.as<JsonObject>());
  }

  // json.shrinkToFit();
  serializeJson(json, *response);
  json.clear();

  request->send(response);
}

//
============================
============================
========== //

// Load File Config dan Iqomah
void loadFile()
{
  // Load File Config
  File configFile =
SPIFFS.open(fileConfig);

  if (!configFile)
  {
    Serial.println("Gagal membuka file
config");
    return;
  }

  size_t sizeConfig = configFile.size();
  std::unique_ptr<char[]>
bufConfig(new char[sizeConfig]);
  configFile.readBytes(bufConfig.get(),
sizeConfig);

  DynamicJsonDocument doc(1024);
  DeserializationError error =
deserializeJson(doc, bufConfig.get());

  if (error)
  {
    Serial.println("Gagal parse file
config");
    return;
```

```
  }

  Serial.println("Load data file
config...");

  config.latitude = doc["latitude"];
  config.longitude = doc["longitude"];
  config.zonaWaktu =
doc["zonaWaktu"];
  config.listInformasi =
doc["listInformasi"].as<String>();

  configFile.close();
  doc.clear();

  // Load File Iqomah
  File iqomahFile =
SPIFFS.open(fileIqomah);

  if (!iqomahFile)
  {
    Serial.println("Gagal membuka file
iqomah");
    return;
  }

  size_t sizeIqomah = iqomahFile.size();
  std::unique_ptr<char[]>
bufIqomah(new char[sizeIqomah]);
  iqomahFile.readBytes(bufIqomah.get(),
sizeIqomah);

  error = deserializeJson(doc,
bufIqomah.get());

  if (error)
  {
    Serial.println("Gagal parse file
iqomah");
    return;
  }

  Serial.println("Load data file
iqomah...");

  iqomah.subuh = doc["subuh"];
  iqomah.zuhur = doc["zuhur"];
  iqomah.asar = doc["asar"];
  iqomah.magrib = doc["magrib"];
  iqomah.isya = doc["isya"];
```

```
  configFile.close();
  doc.clear();
}

void saveFile(const char *filePath,
JsonObject json)
{
  timerDetachInterrupt(timer);
  File configFile =
SPIFFS.open(filePath, "w");

  if (!configFile)
  {
    Serial.println("Gagal membuka file
config");
    json["error"] = true;
    json["message"] = "Gagal membuka
file config";
  }

  serializeJson(json, configFile);
  configFile.close();
  json.clear();
  //
  json["error"] = false;
  json["message"] = "Data berhasil
disimpan";

  initWaktuSholat();
  delay(500);
  initLED(false);
  time_cetak_waktu = 0;
}

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

void initLED(boolean first = true)
{
  if (first)
  {
    uint8_t cpuClock =
ESP.getCpuFreqMHz();
    timer = timerBegin(0, cpuClock,
true);
  }
```

```
  timerAttachInterrupt(timer,
&triggerScan, true);
  timerAlarmWrite(timer, 300, true);

  timerAlarmEnable(timer);

  if (first)
    dmd.clearScreen(true);
}

void initWaktuSholat()
{
  set_calc_method(Custom);
  set_asr_method(Shafii);

set_high_lats_adjust_method(AngleBase
d);
  set_fajr_angle(20);
  set_isha_angle(18);

  get_prayer_times(waktu.tahun,
waktu.bulan, waktu.tanggal,
config.latitude, config.longitude,
config.zonaWaktu, times);
}

/*--------------------------------------------
----------------------------------------
  Fungsi untuk menampilkan waktu ke
LED
--------------------------------------------
----------------------------------*/
void cetakWaktu()
{
  dmd.selectFont(ElektronMart6x16);

  char jam[3], menit[3];

  sprintf(jam, "%02d", waktu.jam);
  sprintf(menit, "%02d", waktu.menit);

  dmd.drawString(0, 0, jam, 3,
GRAPHICS_NORMAL);

  dmd.selectFont(HoboStd16);
  dmd.drawChar(14, -2, ':', waktu.detik
% 2 == 0 ? GRAPHICS_OR :
GRAPHICS_NOR);
```

```
  dmd.selectFont(ElektronMart6x16);
  dmd.drawString(19, 0, menit, 2,
GRAPHICS_NORMAL);
}

void cetakTanggal()
{
  if (indexCetakTampil != 0)
    return;

  dmd.selectFont(ElektronMart5x6);

  char teks[11];
  sprintf(teks, "%02d-%02d-%d",
waktu.tanggal, waktu.bulan,
waktu.tahun);

  textCenter(0, teks);
  if (millis() >= time_cetak_waktu +
INTERVAL_CETAK_WAKTU_SHOL
AT * 2)
  {
    time_cetak_waktu +=
INTERVAL_CETAK_WAKTU_SHOL
AT;
    indexCetakTampil = 1;
  }
}

void cetakJamSholat()
{
  if (indexCetakTampil != 1)
    return;

  int hh, mm;

  int index =
indexTime[indexTimeActive];

  if (millis() >= time_cetak_waktu +
INTERVAL_CETAK_WAKTU_SHOL
AT)
  {
    time_cetak_waktu +=
INTERVAL_CETAK_WAKTU_SHOL
AT;
    get_float_time_parts(times[index], hh,
mm);

    dmd.selectFont(ElektronMart5x6);
```

```
   // clear
   dmd.drawFilledBox(32, 0, 3 *
DMD_PIXELS_ACROSS, 7,
GRAPHICS_INVERSE);

   char teks[strlen(TimeName[index]) +
5];
   sprintf(teks, "%s %02d:%02d",
TimeName[index], hh, mm);

   textCenter(0, teks);

   // Fungsi untuk menampilkan jadwal
sholat berikutnya
   if (indexTimeActive <
(sizeof(indexTime) / sizeof(int *) - 1))
   {
    indexTimeActive++;
   }
   else
   {
    indexTimeActive = 0;
    indexCetakTampil = 0;
    // clear
      dmd.drawFilledBox(32, 0, 3 *
DMD_PIXELS_ACROSS, 7,
GRAPHICS_INVERSE);
    }
  }
}

void cetakInfo()
{
  StaticJsonDocument<512>
jsonInformasi;
  deserializeJson(jsonInformasi,
config.listInformasi);

  dmd.selectFont(CooperBlack10);
  const char *MSG =
jsonInformasi[indexInformasi];

  if (retInformasi)
  {
    dmd.drawMarquee(MSG,
strlen(MSG),
(DMD_PIXELS_ACROSS * KOLOM),
8);
    retInformasi = false;
```

```
  }

  if (!retInformasi && millis() >=
time_cetak_informasi + 70)
  {
    time_cetak_informasi += 70;

    dmd.selectFont(CooperBlack10);
    retInformasi =
dmd.stepSplitMarquee(9, 15, 39);

    if (retInformasi)
      if (indexInformasi <
jsonInformasi.size() - 1)
        indexInformasi++;
      else
        indexInformasi = 0;
  }

  if (updateInformasi)
  {
    indexInformasi = 0;
    retInformasi = true;
    dmd.clearScreen(true);
  }
    updateInformasi = false;

    jsonInformasi.clear();
}

void cetakAdzan()
{
  if (indexCetakTampil != 2)
    return;

  int hh, mm;


get_float_time_parts(times[indexTimeA
ctive], hh, mm);

  char
teks[strlen(TimeName[indexTimeActive
]) + 5];
  sprintf(teks, "%s %02d:%02d",
TimeName[indexTimeActive], hh, mm);

  dmd.selectFont(ElektronMart5x6);
```

```
  if (waktu.detik % 2 != 0)
    dmd.drawFilledBox(32, 0, 3 *
DMD_PIXELS_ACROSS, 7,
GRAPHICS_INVERSE);
  else
    textCenter(0, teks);

  // berali ke iqomah setelah 1 menit
  if (waktu.menit > mm)
  {
    dmd.drawFilledBox(32, 0, 3 *
DMD_PIXELS_ACROSS, 7,
GRAPHICS_INVERSE);

    indexCetakTampil = 3;
    time_iqomah = millis();
    if (indexTimeActive == 0)
      waktuIqomah.menit =
iqomah.subuh;
    else if (indexTimeActive == 2)
      waktuIqomah.menit = iqomah.zuhur;
    else if (indexTimeActive == 3)
      waktuIqomah.menit = iqomah.asar;
    else if (indexTimeActive == 5)
      waktuIqomah.menit =
iqomah.magrib;
    else if (indexTimeActive == 6)
      waktuIqomah.menit = iqomah.isya;
  }
}

void cetakIqomah()
{
  if (indexCetakTampil != 3)
    return;

  if (millis() >= time_iqomah + 1000)
  {
    time_iqomah += 1000;

    char teks[13];
    sprintf(teks, "IQOMAH %02d:%02d",
waktuIqomah.menit,
waktuIqomah.detik);

    dmd.selectFont(ElektronMart5x6);
    textCenter(0, teks);
```

```
    if (waktuIqomah.menit == 0 &&
waktuIqomah.detik == 0)
    {
      indexCetakTampil = 1;
    }
    else if (waktuIqomah.detik == 0)
    {
      waktuIqomah.detik = 59;
      waktuIqomah.menit--;
    }
    else
    {
      waktuIqomah.detik--;
    }
  }
}

void setWaktu(int yr, int month, int
mday, int hr, int minute, int sec, int
isDst)
{
  struct tm tm;

  tm.tm_year = yr - 1900;
  tm.tm_mon = month - 1;
  tm.tm_mday = mday;
  tm.tm_hour = hr;
  tm.tm_min = minute;
  tm.tm_sec = sec;
  tm.tm_isdst = isDst; // 1 or 0
  time_t t = mktime(&tm);
  Serial.printf("Set Waktu: %s",
asctime(&tm));
  struct timeval now = {.tv_sec = t};
  settimeofday(&now, NULL);
}

void updateWaktu()
{
  if (!getLocalTime(&timeinfo))
  {
    Serial.println("Gagal mengambil
waktu");
    return;
  }

  waktu.tahun = timeinfo.tm_year +
1900;
  waktu.bulan = timeinfo.tm_mon + 1;
  waktu.tanggal = timeinfo.tm_mday;
```

```
  waktu.jam = timeinfo.tm_hour;
  waktu.menit = timeinfo.tm_min;
  waktu.detik = timeinfo.tm_sec;

  // periksa jika masuk waktu sholat
  int indexWaktuSholat[] = {0, 2, 3, 5,
6};
  if (waktu.detik == 0)
  {
    for (size_t i = 0; i <
sizeof(indexWaktuSholat) / sizeof(int *);
i++)
    {
      char tmp[10];
      int hh, mm;

get_float_time_parts(times[indexWaktu
Sholat[i]], hh, mm);

      if (waktu.jam == hh &&
waktu.menit == mm)
      {
        indexCetakTampil = 2;
        indexTimeActive =
indexWaktuSholat[i];
        break;
      }
    }
  }
}

void textCenter(int y, char *Msg)
{
  int width = 0;
  for (size_t i = 0; i < strlen(Msg); i++)
  {
    width += dmd.charWidth(Msg[i]);
  }

  int center = int(((30 * (KOLOM + 1)) -
width) / 2);

  dmd.drawString(center, y, Msg,
strlen(Msg), GRAPHICS_NORMAL);
}
```

Lampiran – *2* *source code untuk tampilan web (visual studio code)*

```
const char webpage[] PROGMEM =
   R"=====(
   <!DOCTYPE html>
<html>
  <head>
   <meta charset="utf-8" />
   <meta name="viewport"
content="width=device-width, initial-
scale=1" />
   <meta http-equiv=”refresh”
content=”5" />
   <title>Jadwal Waktu Sholat</title>
   <link
    rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bulma
@0.9.3/css/bulma.min.css"
   />
  </head>

  <body>
   <div id="app" class="container is-
fluid">
    <section class="section">
     <div class="container">
      <h1 class="title">Jadwal Waktu
Sholat</h1>
      <p class="subtitle">ESP32 Panel
P10</p>
     </div>
    </section>

    <div class="columns is-mobile is-
multiline">
     <div class="column is-full">
      <div class="card">
       <header class="card-header">
        <p class="card-header-title">
         <span class="icon is-large
has-text-info">
          <ion-icon name="person-
circle" size="large"></ion-icon>
         </span>
         Profil
        </p>
       </header>
       <div class="card-content">
        <div class="content">
         <div class="columns is-
mobile" v-for="profile in listProfile">
          <div
           class="column is-one-fifth-
desktop is-one-third-mobile"
           v-text="profile.title"
          ></div>
          <div class="column is-1" v-
text="`:`"></div>
          <div class="column" v-
text="profile.text"></div>
         </div>
        </div>
       </div>
      </div>

     <div class="column is-full">
      <div class="card">
       <header class="card-header">
        <p class="card-header-title">
         <span class="icon is-large
has-text-info">
          <ion-icon name="stopwatch"
size="large"></ion-icon>
         </span>
         Pengaturan Waktu
        </p>
        <div class="card-header-icon">
         <button
          class="button is-info"
          @click="postWaktu"
          :disabled="!tanggal || !jam"
         >
          <span class="icon">
           <ion-icon
name="save"></ion-icon>
          </span>
         </button>
        </div>
       </header>
       <div class="card-content">
        <div class="content">
         <div class="columns">
          <div class="column">
           <div class="field">
            <label
class="label">Tanggal</label>
```

```
                <div class="control">
                  <input
                    v-model="tanggal"
                    class="input"
                    type="date"
                    placeholder="Masukkan
Tanggal"
                  />
                </div>
              </div>
            </div>
            <div class="column">
              <div class="field">
                <label
class="label">Jam</label>
                <div class="control">
                  <input
                    v-model="jam"
                    class="input"
                    type="time"
                    placeholder="Masukkan
Jam"
                  />
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>

      <div class="column is-full">
        <div class="card">
          <header class="card-header">
            <p class="card-header-title">
              <span class="icon is-large
has-text-info">
                <ion-icon name="location"
size="large"></ion-icon>
              </span>
              Pengaturan Lokasi
            </p>
            <div class="card-header-icon">
              <button class="button is-info"
@click="postLokasi">
                <span class="icon">
                  <ion-icon
name="save"></ion-icon>
                </span>
              </button>
            </div>
          </header>
          <div class="card-content">
            <div class="content">
              <div class="columns">
                <div class="column">
                  <div class="field">
                    <label
class="label">Latitude</label>
                    <div class="control">
                      <input
                        v-model="latitude"
                        class="input"
                        type="number"
                        placeholder="Masukkan
Latitude"
                      />
                    </div>
                  </div>
                </div>
                <div class="column">
                  <div class="field">
                    <label
class="label">Longitude</label>
                    <div class="control">
                      <input
                        v-model="longitude"
                        class="input"
                        type="number"
                        placeholder="Masukkan
Longitude"
                      />
                    </div>
                  </div>
                </div>
                <div class="column">
                  <div class="field">
                    <label class="label">Zona
Waktu</label>
                    <div class="control">
                      <input
                        v-model="zonaWaktu"
                        class="input"
                        type="number"
                        placeholder="Masukkan
Zona Waktu"
                      />
                    </div>
                  </div>
```

```
            </div>
           </div>
          </div>
         </div>
        </div>

        <div class="column is-full">
         <div class="card">
          <header class="card-header">
           <p class="card-header-title">
            <span class="icon is-large
has-text-info">
             <ion-icon
name="information-circle"
size="large"></ion-icon>
            </span>
            Pengaturan Informasi
           </p>
           <div class="card-header-icon">
            <button class="button is-info"
@click="postListInformasi">
             <span class="icon">
              <ion-icon
name="save"></ion-icon>
             </span>
            </button>
           </div>
          </header>
          <div class="card-content">
           <div class="content">
            <div v-for="(_, i) in
listInformasi" class="columns is-
mobile">
             <div class="column is-10">
              <input
               v-model="listInformasi[i]"
               class="input"
               type="text"
               placeholder="Pesan
Informasi"
              />
             </div>
             <div class="column is-2">
              <button class="button"
@click="hapusInformasi(i)">
               <span class="icon">
                <ion-icon
name="close"></ion-icon>
               </span>
              </button>
               
              <button
               v-if="i ==
listInformasi.length -1"
               class="button"

@click="tambahInformasi"
              >
               <span class="icon">
                <ion-icon
name="add"></ion-icon>
               </span>
              </button>
             </div>
            </div>
           </div>
          </div>
         </div>
        </div>

        <div class="column is-full">
         <div class="card">
          <header class="card-header">
           <p class="card-header-title">
            <span class="icon is-large has-
text-info">
             <ion-icon name="volume-
medium" size="large"></ion-icon>
            </span>
            Pengaturan Iqomah
           </p>
           <div class="card-header-icon">
            <button class="button is-info"
@click="postIqomah">
             <span class="icon">
              <ion-icon
name="save"></ion-icon>
             </span>
            </button>
           </div>
          </header>
          <div class="card-content">
           <div class="content">
            <div class="columns">
             <div v-for="(waktu, iqo) in
iqomah" class="column">
              <div class="field">
```

```
                <label
class="label">{{iqo.toUpperCase()}}</l
abel>
                <div class="control">
                  <input
                    v-model="iqomah[iqo]"
                    class="input"
                    type="number"
                    placeholder="Menit"
                  />
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>

      <div class="modal">
        <div class="modal-
background"></div>
        <div class="modal-card">
          <header class="modal-card-
head">
            <p class="modal-card-
title">Informasi</p>
          </header>
          <section class="modal-card-
body">
            <div
              class="notification"
              :class="[response.error ? `is-
danger`: `is-info`]"
              v-text="response.message"
            ></div>
          </section>
        </div>
      </div>
    </div>

    <script
src="https://unpkg.com/vue@next"></sc
ript>
    <script
      type="module"

src="https://unpkg.com/ionicons@5.5.2/
dist/ionicons/ionicons.esm.js"
    ></script>
    <script
      nomodule

src="https://unpkg.com/ionicons@5.5.2/
dist/ionicons/ionicons.js"
    ></script>
  </body>
  <script setup>
    const { ref, createApp, onMounted }
= Vue;
    createApp({
      setup() {
        const loading = ref(true);
        const tanggal = ref(null);
        const jam = ref(null);
        const latitude = ref(null);
        const longitude = ref(null);
        const zonaWaktu = ref(null);
        const listInformasi = ref([]);
        const iqomah = ref({});
        const response = ref({
          error: false,
          message: null,
        });

        const url = new
URL(document.URL);

        const listProfile = ref([
          {
            title: "Nama",
            text: "Muhammad Fadhil
Arham",
          },
          {
            title: "Nim",
            text: "217280175",
          },
          {
            title: "Judul",
            text: "Sistem Pengendali
Running Text P10 Berbasis Android",
          },
          {
            title: "Pembimbing 1",
            text: "Ade Hastuty,
ST.,S.kom.,M.Kom",
          },
          {
            title: "Pembimbing 2",
```

```
          text: "Hj. Andi Irmayani P,
ST.,MT",
        },
        {
          title: "Penguji 1",
          text: "Ir. Untung Suwardoyo,
S.kom.,M.Kom",
        },
        {
          title: "Penguji 1",
          text: "Muh. Zainal, ST.,MT",
        },
      ]);

      function openModal() {
        const modal =
document.querySelector(".modal");
        modal.classList.add("is-active");

        setTimeout(() =>
modal.classList.remove("is-active"),
2000);
      }

      const post = async (endpoint,
formData) => {
        const post = await
fetch(`${url.href}${endpoint}`, {
          method: "POST",
          body: formData,
        });
        const res = await post.json();
        response.value = res;
        openModal();
      };

      const postWaktu = async () => {
        const date = new
Date(`${tanggal.value} ${jam.value}`);

        const formData = new
FormData();
        formData.append("tahun",
date.getFullYear());
        formData.append("bulan",
date.getMonth() + 1);
        formData.append("tanggal",
date.getDate());
        formData.append("jam",
date.getHours());
        formData.append("menit",
date.getMinutes());

        post("waktu", formData);
      };

      const getLokasi = async () => {
        const get = await
fetch(`${url.href}lokasi`);

        const res = await get.json();

        latitude.value = res.latitude;
        longitude.value = res.longitude;
        zonaWaktu.value =
res.zonaWaktu;
      };

      const postLokasi = async () => {
        const formData = new
FormData();
        formData.append("latitude",
latitude.value);
        formData.append("longitude",
longitude.value);
        formData.append("zonaWaktu",
zonaWaktu.value);

        post("lokasi", formData);
      };

      const getListInformasi = async ()
=> {
        const get = await
fetch(`${url.href}informasi`);

        const res = await get.json();

        listInformasi.value = res;
      };

      const postListInformasi = async ()
=> {
        const formData = new
FormData();
        formData.append("listInformasi",
JSON.stringify(listInformasi.value));

        post("informasi", formData);
      };
```

```
      const getIqomah = async () => {
        const get = await
fetch(`${url.href}iqomah`);

        const res = await get.json();

        iqomah.value = res;
      };

      const postIqomah = async () => {
        const formData = new
FormData();

        for (const key in iqomah.value) {
          formData.append(key,
iqomah.value[key]);
        }

        post("iqomah", formData);
      };

      onMounted(() => {
        getLokasi();
        getListInformasi();
        getIqomah();
      });

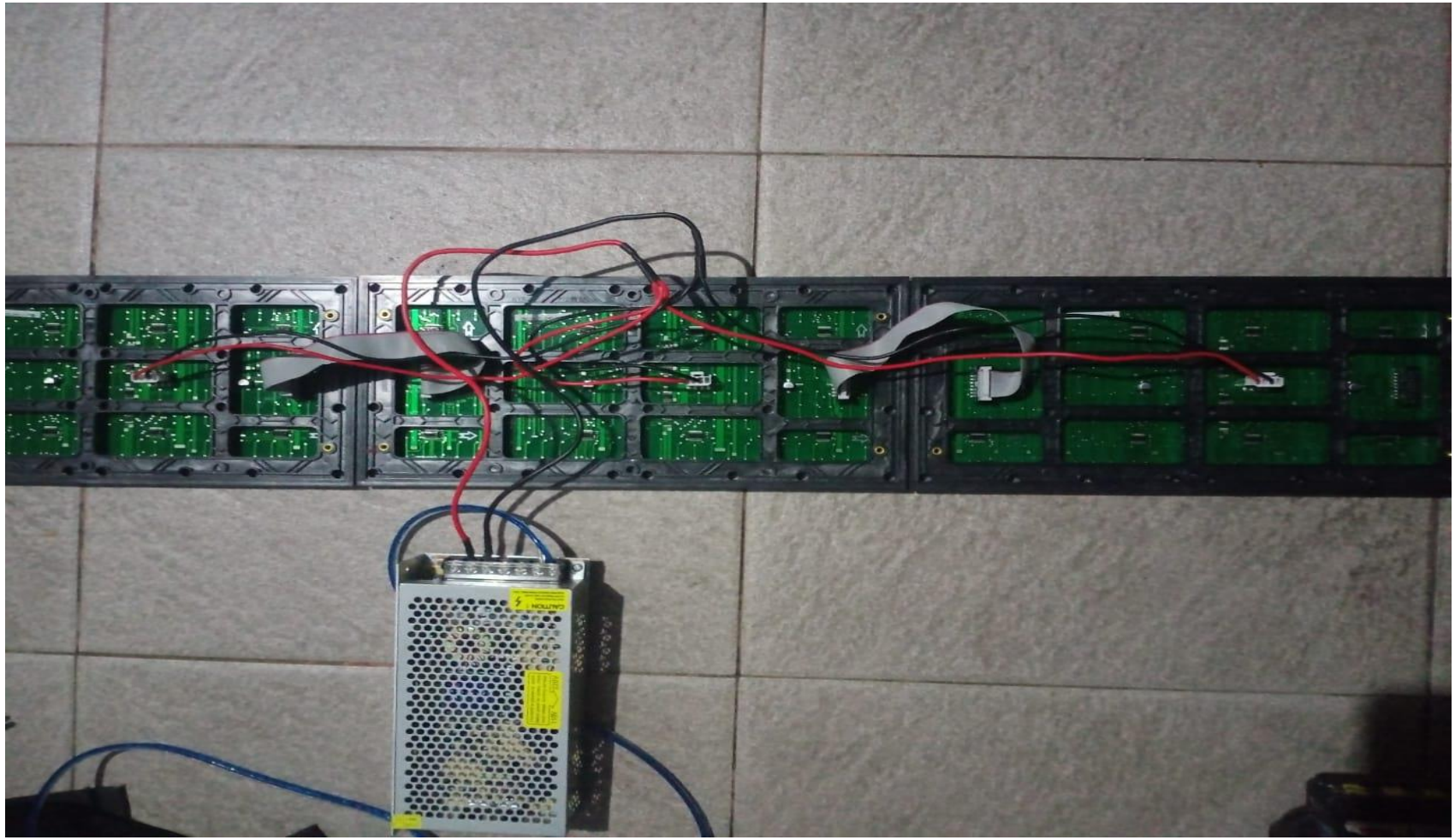
      const tambahInformasi = () => {
        listInformasi.value.push([""]);
      };

      const hapusInformasi = (index) =>
{
        listInformasi.value.splice(index,
1);
      };

      return {
        loading,
        tanggal,
        jam,
        latitude,
        longitude,
        zonaWaktu,
        iqomah,
        listProfile,
        listInformasi,
        postWaktu,
        postLokasi,
        postListInformasi,
        postIqomah,
        tambahInformasi,
        hapusInformasi,
        response,
      };
    },
  }).mount("#app");
 </script>
</html>

)=====";
```

LAMPIRAN – 3 Gambar pemasangan alat running text P10



LAMPIRAN – 4 Tampilan running text P10

