

LAMPIRAN

Lampiran 1 Gambar Observasi di RSUD Cilacap



Lampiran 2 Surat Serah Terima Barang



**PEMERINTAH KABUPATEN CILACAP
DINAS KESEHATAN
UPTD RSUD CILACAP**

Jalan Jenderal Gatot Subroto Nomor 28 Telepon (0282) 533010 Faximile (0282) 520755
website : www.rsud.cilacapkab.go.id E-mail : bludrsudcilacap@gmail.co.id

CILACAP

Kode Pos 53223

BERITA ACARA SERAH TERIMA

BANTUAN.....

Pada Hari Jumat tanggal 3-2-2023 telah dilaksanakan serah terima bantuan berupa pewaktu pada Ruang Operasi

Dari

Nama : Tar Fikriyani
No HP : 087878584788
Yayasan / instansi : Universitas Tahdlatul Ulama Al Ghazali

Kepada : **Panitia Pengelolaan Barang Hibah RSUD Cilacap**

Yang Mewakili :

Nama : Buana Anton
Jabatan : Koordinator dan Logistik
Instansi : RSUD Cilacap

Dengan rincian barang yang diterima sebagai berikut :

| NO | NAMA BARANG | JUMLAH |
|----|-----------------------------------|----------|
| 1. | <u>Pewaktu pada Ruang Operasi</u> | <u>1</u> |
| | | |
| | | |
| | | |

Cilacap, 3 Februari 2023

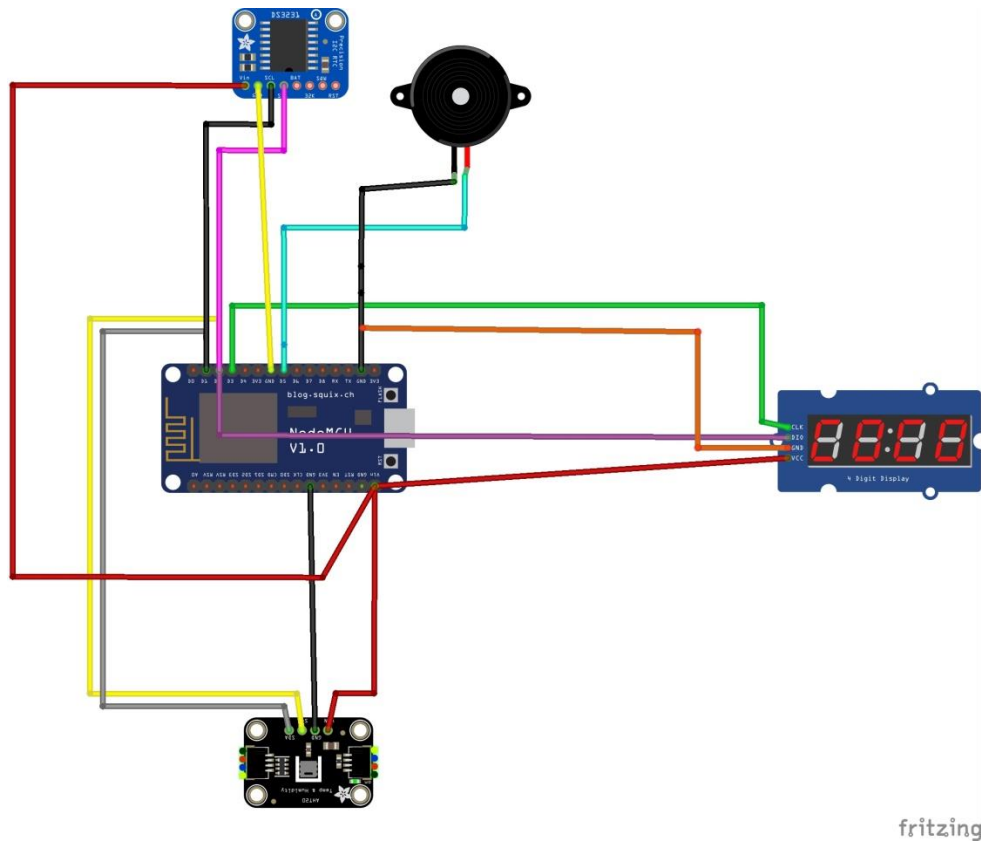
Yang Menyerahkan

Tar Fikriyani

Yang Menerima

Buana Anton

Lampiran 3 Perancangan Alat



| 7-Segment | NodeMCU ESP8266 |
|-----------|--------------------|
| DI0 | D2 |
| GND | GND |
| CLK | D3 |
| VCC | VIN |

| AHT10 | NodeMCU ESP8266 |
|-------|--------------------|
| VIN | 3V |
| GND | GND |
| SCL | D2 |
| SDA | D1 |

| Buzzer | NodeMCU ESP8266 |
|---------|--------------------|
| Positif | D5 |
| Negatif | GND |

| RTC DS3231 | NodeMCU ESP8266 |
|---------------|--------------------|
| VIN | VIN |
| GND | GND |
| SDA | D1 |
| SCL | D2 |

Lampiran 4 Kode Program Alat

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include "index.h"
#include <EEPROM.h>
/*=====
ACCESS POINT in SURGERY TIME
=====*/
const char* ssid = "Surgery Time";
const char* password = "12345678";
ESP8266WebServer server(80);

/*=====*/
#include <AHT10.h>
AHT10 myAHT10(AHT10_ADDRESS_0X38);
float dataSuhu, dataKelembaban;

/* ===== */
#define J_R1 D5 // GPIO14
#define J_ST D6 // GPIO12
#define J_SH D7 // GPIO13
#define Relay D4 // GPIO2
#define DIP D8 // GPIO15

boolean tandaDip;
int diPlast, Temp_Harian, dataDigit, Data_Satuan, Data_Puluhan;
int humidS, humidP, suhuS, suhuP, timerSd, timerPd, timerSm;
int timerPm, timerSj, timerPj, tandaS, tandaP;

/* ===== */
byte decToBcd(byte val) {
    return ( val / 10 * 16) + (val % 10);
}
byte bcdToDec(byte val) {
    return ( val / 16 * 10) + (val % 16);
}

/*=====*/
void handleRoot() {
    server.send(200, "text/html", MAIN_page); //Send web page
}
/*=====
Real Time Clock DS3231
=====*/
#include "RTClib.h"
RTC_DS3231 RTC;
DateTime now;
/*=====*/
void handleGetDS3231() {
    DateTime now = RTC.now();
    char istr[6];
    sprintf(istr, "%d : %d : %d", now.hour(), now.minute(), now.second());
    server.send(200, "text/plain", istr);
}

void handleGetTanggal () {
    DateTime now = RTC.now();
    char str[6];
    sprintf(str, "%d : %d : %d", now.day(), now.month(), now.year());
    server.send(200, "text/plain", str);
}

void handleSetWaktu() {
    String rtc_state = server.arg("RTCset");
    Serial.println(rtc_state);
    uint8_t detik = (((rtc_state[ 0] - 48) * 10) + (rtc_state[ 1] - 48));
    uint8_t menit = (((rtc_state[ 2] - 48) * 10) + (rtc_state[ 3] - 48));
    uint8_t jam = (((rtc_state[ 4] - 48) * 10) + (rtc_state[ 5] - 48));
    uint8_t hari = ( rtc_state[ 6] - 48) + 1;
    uint8_t tanggal = (((rtc_state[ 7] - 48) * 10) + (rtc_state[ 8] - 48));
    uint8_t bulan = (((rtc_state[ 9] - 48) * 10) + (rtc_state[10] - 48) + 1); // format hex dari komputer
    uint8_t tahun = (((rtc_state[13] - 48) * 10) + (rtc_state[14] - 48));
    RTC.adjust(DateTime(tahun, bulan, tanggal, jam, menit, detik));
}

```

```

    bep();
}
/*=====*/

/*=====
                SURGERY TIME LAYANAN WEB SERVER
=====*/
uint8_t elapsedJam, elapsedMenit, elapsedDetik;
/*=====*/
void handleGetElapsed() {
    char str[6];
    sprintf(str, "%d : %d : %d", elapsedJam, elapsedMenit, elapsedDetik);
    server.send(200, "text/plain", str);
}

/*=====*/
uint8_t alamatTime1 = 1;
uint8_t alamatTime2 = 2;
uint8_t alamatTime3 = 3;
/*=====*/
void handleGetTime1() {
    char str[2];
    sprintf(str, "%d", EEPROM.read(alamatTime1));
    server.send(200, "text/plain", str);
}
void handleGetTime2() {
    char str[2];
    sprintf(str, "%d", EEPROM.read(alamatTime2));
    server.send(200, "text/plain", str);
}
void handleGetTime3() {
    char str[2];
    sprintf(str, "%d", EEPROM.read(alamatTime3));
    server.send(200, "text/plain", str);
}

/*=====*/
void handleSetJadwal() {
    String timeState = server.arg("sendJadwal");
    Serial.println(timeState);
    EEPROM.write (alamatTime1, (((timeState[ 0] - 48) * 100) + (timeState[ 1] - 48) * 10) + (timeState[ 2] - 48));
    EEPROM.write (alamatTime2, (((timeState[ 3] - 48) * 100) + (timeState[ 4] - 48) * 10) + (timeState[ 5] - 48));
    EEPROM.write (alamatTime3, (((timeState[ 6] - 48) * 100) + (timeState[ 7] - 48) * 10) + (timeState[ 8] - 48));
    EEPROM.commit();
    bep();
}

void handleResetter() {
    EEPROM.write (alamatTime1, (0));
    EEPROM.write (alamatTime2, (0));
    EEPROM.write (alamatTime3, (0));
    EEPROM.commit();
    Serial.println("Semua Data Dihapus. Lakukan Setting Ulang");
    bep();
}

/*=====
    Program pengambilan data timer yang tersimpan dan
    menjalankan program
*/
int dataTimer;
int elapsedJam1, elapsedMenit1;
int elapsedJam2, elapsedMenit2;
int elapsedJam3, elapsedMenit3;
int timerJam, timerMenit;
int tandaMem = 0;
float angkaTimer = 0;
boolean tandaReload = false;
boolean tandaTimer = false;

```

```

/*=====*/
void handleStart() {
  if (tandaTimer == false) {
    Serial.println(" ");
    Serial.println("Timer yang tersimpan");
    Serial.println("=====");

    Serial.print("Elapsed 1 = ");
    angkaTimer = EEPROM.read(alamatTime1);
    if (angkaTimer != 0) tandaTimer = true;
    timerConvert();
    elapsedJam1 = timerJam;
    elapsedMenit1 = timerMenit;

    Serial.print("Elapsed 2 = ");
    angkaTimer = EEPROM.read(alamatTime2);
    if (angkaTimer != 0) tandaTimer = true;
    timerConvert();
    elapsedJam2 = timerJam;
    elapsedMenit2 = timerMenit;

    Serial.print("Elapsed 3 = ");
    angkaTimer = EEPROM.read(alamatTime3);
    if (angkaTimer != 0) tandaTimer = true;
    timerConvert();
    elapsedJam3 = timerJam;
    elapsedMenit3 = timerMenit;

    Serial.println("Start");
    Serial.println("=====");

    bep();
    tandaMem = 1;
    tandaReload = true;
    dataTimer = 10;
  }
}

}

/*=====*/
void timerConvert() {
  float timer = angkaTimer / 60;
  timerJam = timer;
  float timerM = (timer - timerJam);
  timerMenit = (timerM * 60);

  Serial.print(timerJam);
  Serial.print(" : ");
  Serial.println(timerMenit);
}

/*=====*/
void handleStop() {
  tandaTimer = tandaTimer ^ true;
  Serial.println("Stop / Pause");
  bep();
}

/*=====*/
void bep() {
  digitalWrite (Relay, LOW);
  delay(300);
  digitalWrite (Relay, HIGH);
}

/*=====*/
/*=====

```

```

void setup() {
  Serial.begin(9600);
  Serial.println();
  EEPROM.begin(8);

  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();

  server.begin();
  server.on("/",          handleRoot);
  server.on("/setJam",    handleSetWaktu);
  server.on("/getDS3231", handleGetDS3231);
  server.on("/getanggal", handleGetTanggal);
  server.on("/getTimeElapsed", handleGetElapsed);
  server.on("/time1",     handleGetTime1);
  server.on("/time2",     handleGetTime2);
  server.on("/time3",     handleGetTime3);
  server.on("/setJadwal", handleSetJadwal);
  server.on("/setReset",  handleResetter);
  server.on("/setStart",  handleStart);
  server.on("/setStop",   handleStop);

  /* ===== Inisialisasi Interface Komponen ===== */
  pinMode(J_ST, OUTPUT); // Latch -->Set the Display Pins as outputs 74HC595
  pinMode(J_SH, OUTPUT); // Clock
  pinMode(J_R1, OUTPUT); // Data

  pinMode(Relay, OUTPUT);
  pinMode(DIP, OUTPUT); //LED OUTPUT
  digitalWrite(Relay, HIGH);

  /* ===== Display Starting Programs ===== */
  Serial.println();
  Serial.print("Acces Point: ");

```

```

void loop() {
  server.handleClient();

  DateTime now = RTC.now();
  if (diplast != now.second()) {
    diplast = now.second();
    tandaDip = tandaDip ^ true;
    digitalWrite(DIP, digitalRead(DIP) ^ 1);
    Temp_Harian++;
    if (Temp_Harian == 20) {
      Temp_Harian = 0;
    }
  }

  timerPj = 11, timerSj = 11, timerPm = 11, timerSm = 11, timerPd = 11, timerSd = 11;

  if (tandaTimer == true) {

    /* ===== Isi Ulang variabel Timer ===== */
    if (tandaReload == true) {
      tandaReload = false;
      switch (tandaMem) {
        case 1:
          elapsedJam = elapsedJam1;
          elapsedMenit = elapsedMenit1;
          elapsedDetik = 0;
          dataTimer = 1;
          break;
        case 2:
          elapsedJam = elapsedJam2;
          elapsedMenit = elapsedMenit2;
          elapsedDetik = 0;
          dataTimer = 2;
          break;
        case 3:
          elapsedJam = elapsedJam3;
          elapsedMenit = elapsedMenit3;

```

```

        elapsedDetik = 0;
        dataTimer = 3;
        break;
    }
}

/* ===== Perhitungan Timer =====*/
if (elapsedDetik == 0) {
    elapsedDetik = 60;
    if (elapsedMenit == 0) {
        elapsedMenit = 60;
        if (elapsedJam == 0) {
            elapsedJam = 1, elapsedMenit = 1, elapsedDetik = 1;
            bep();

            tandaMem++;
            tandaReload = true;
            if (tandaMem == 4) {
                tandaMem = 0;
                tandaTimer = false;
                dataTimer = 10;
            }
        }
        elapsedJam --;
    }
    elapsedMenit --;
}
elapsedDetik --;

/* ===== Restore Nilai Timer =====*/
restoreTimerJam() ;
restoreTimerMenit();
restoreTimerDetik();

Serial.print(elapsedJam);
Serial.print(" : ");

```

```

Serial.print(elapsedJam);
Serial.print(" : ");
Serial.print(elapsedMenit);
Serial.print(" : ");
Serial.println(elapsedDetik);
}

/* ===== AHT10 check =====*/
dataSuhu = myAHT10.readTemperature(AHT10_FORCE_READ_DATA);
if (dataSuhu != AHT10_ERROR)
    restoreSuhu();
else dataSuhu = 30.1;

dataKelembaban = myAHT10.readHumidity(AHT10_USE_READ_DATA);
if (dataKelembaban != AHT10_ERROR)
    restoreKelembaban();
else dataKelembaban = 80.7;

/* ===== Tampilkan 7-segment =====*/
tandaP = dataTimer ;
tandaS = 10 ;
Restore_Waktu ();

}
}

```



```

int Tgl_Puluhan, Tgl_Satuan, Bln_Satuan, Bln_Puluhan, Thn_Satuan, Thn_Puluhan, Thn_Ratusan, Thn_Ribuan ;
void Hapus_Kalender () {
    Tgl_Puluhan = 10, Tgl_Satuan = 10, Bln_Puluhan = 10, Bln_Satuan = 10, Thn_Puluhan = 10;
    Thn_Satuan = 10, Thn_Ribuan = 10, Thn_Ratusan = 10;
}

int Mnt_Satuan, Mnt_Puluhan, Jam_Satuan, Jam_Puluhan ;
void Hapus_Jam () {
    Jam_Puluhan = 10, Jam_Satuan = 10;
    Mnt_Puluhan = 10, Mnt_Satuan = 10;
}

void Hapus_Timer () {
    humdS = 10, timerSd = 10, timerSj = 10;
    humdP = 10, timerPd = 10, timerPj = 10;
    suhuS = 10, timerSm = 10, tandaS = 10;
    suhuP = 10, timerPm = 10, tandaP = 10;
}

void restoreStart() {
    Tgl_Puluhan = 0x0A ; // display START
    Tgl_Satuan = 0x0A ;
    Bln_Puluhan = 0x0A ;
    Bln_Satuan = 0x22 ;
    Thn_Ribuan = 0x23 ;
    Thn_Ratusan = 0x10 ;
    Thn_Puluhan = 0x21 ;
    Thn_Satuan = 0x23 ;
}

void Restore_Waktu () { // Mengkonversi data ke setiap satuan digit

    Restore_Jam () ;
    Restore_Menit () ;

    if (Temp_Harian <= 10) {
        Restore_Tanggal ();
        Restore_Bulan ();
        Restore_Tahun ();
    }
    if (Temp_Harian > 10) {
        Hapus_Kalender ();
        Restore_Hari () ;
    }

    restoreSuhu() ;
    restoreKelembaban();

    Scanning_Segment ();
}

void Restore_Hari () {
    DateTime now = RTC.now();
    switch ( now.dayOfTheWeek() ) {
        case 0: // ahad

```

```

    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x0A ;
    Bln_Puluhan = 0x10 ;
    Bln_Satuan = 0x17 ;
    Thn_Ribuan = 0x10 ;
    Thn_Ratusan = 0x13 ;
    Thn_Puluhan = 0x0A ;
    Thn_Satuan = 0x0A ;
    break ;
case 1: // senin
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x0A ;
    Bln_Puluhan = 0x22 ;
    Bln_Satuan = 0x14 ;
    Thn_Ribuan = 0x1D ;
    Thn_Ratusan = 0x18 ;
    Thn_Puluhan = 0x1D ;
    Thn_Satuan = 0x0A ;
    break ;
case 2: //selasa
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x22 ;
    Bln_Puluhan = 0x14 ;
    Bln_Satuan = 0x1B ;
    Thn_Ribuan = 0x10 ;
    Thn_Ratusan = 0x22 ;
    Thn_Puluhan = 0x10 ;
    Thn_Satuan = 0x0A ;
    break ;
case 3: //rabu
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x0A ;
    Bln_Puluhan = 0x21 ;
    Bln_Satuan = 0x10 ;
    Thn_Ribuan = 0x11 ;
    Thn_Ratusan = 0x24 ;

    Thn_Puluhan = 0x0A ;
    Thn_Satuan = 0x0A ;
    break ;
case 4: //kamis
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x1A ;
    Bln_Puluhan = 0x10 ;
    Bln_Satuan = 0x2B ;
    Thn_Ribuan = 0x2A ;
    Thn_Ratusan = 0x18 ;
    Thn_Puluhan = 0x22 ;
    Thn_Satuan = 0x0A ;
    break ;
case 5: //jumat
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x19 ;
    Bln_Puluhan = 0x24 ;
    Bln_Satuan = 0x2B ;
    Thn_Ribuan = 0x2A ;
    Thn_Ratusan = 0x10 ;
    Thn_Puluhan = 0x23 ;
    Thn_Satuan = 0x0A ;
    break ;
default: //sabtu
    Tgl_Puluhan = 0x0A ;
    Tgl_Satuan = 0x0A ;
    Bln_Puluhan = 0x22 ;
    Bln_Satuan = 0x10 ;
    Thn_Ribuan = 0x11 ;
    Thn_Ratusan = 0x23 ;
    Thn_Puluhan = 0x24 ;
    Thn_Satuan = 0x0A ;
    break;
}
}

```

```

    Thn_Puluhan = Data_Puluhan;
    Thn_Satuan = Data_Satuan ;
}

void restoreTimerJam() {
    dataDigit = elapsedJam ;
    Restoring_Digit () ;
    timerPj = Data_Puluhan;
    timerSj = Data_Satuan ;
}
void restoreTimerMenit() {
    dataDigit = elapsedMenit;
    Restoring_Digit () ;
    timerPm = Data_Puluhan;
    timerSm = Data_Satuan ;
}
void restoreTimerDetik() {
    dataDigit = elapsedDetik;
    Restoring_Digit () ;
    timerPd = Data_Puluhan;
    timerSd = Data_Satuan ;
}

void restoreSuhu() {
    dataDigit = dataSuhu ;
    Restoring_Digit () ;
    suhuP = Data_Puluhan;
    suhuS = Data_Satuan ;
}
void restoreKelembaban() {
    dataDigit = dataKelembaban ;
    Restoring_Digit () ;
    humdP = Data_Puluhan;
    humdS = Data_Satuan ;
}

void Restoring_Digit () { // Menentukan digit satuan, puluhan, ratusan dll
    if (dataDigit < 10) {
        Data_Puluhan = 0;
        Data_Satuan = dataDigit;
    }
    else if (dataDigit > 9 && dataDigit < 20) {
        Data_Puluhan = 1;
        Data_Satuan = dataDigit - 10;
    }
    else if (dataDigit > 19 && dataDigit < 30) {
        Data_Puluhan = 2;
        Data_Satuan = dataDigit - 20;
    }
    else if (dataDigit > 29 && dataDigit < 40) {
        Data_Puluhan = 3;
        Data_Satuan = dataDigit - 30;
    }
    else if (dataDigit > 39 && dataDigit < 50) {
        Data_Puluhan = 4;
        Data_Satuan = dataDigit - 40;
    }
    else if (dataDigit > 49 && dataDigit < 60) {
        Data_Puluhan = 5;
        Data_Satuan = dataDigit - 50;
    }
    else if (dataDigit > 59 && dataDigit < 70) {
        Data_Puluhan = 6;
        Data_Satuan = dataDigit - 60;
    }
    else if (dataDigit > 69 && dataDigit < 80) {
        Data_Puluhan = 7;
        Data_Satuan = dataDigit - 70;
    }
}

else if (dataDigit > 79 && dataDigit < 90) {
    Data_Puluhan = 8;
    Data_Satuan = dataDigit - 80;
}
else if (dataDigit > 89 && dataDigit < 100) {
    Data_Puluhan = 9;
    Data_Satuan = dataDigit - 90;
}
}
}

```

```

static byte displayDigit1[46] = {
0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90, 0xFF, 0xBF,
0xFF, 0xFF, 0xFF, 0xA0, 0x83, 0xC6, 0xA1, 0x86, 0x8E, 0x90, 0x8B,
0xF9, 0xE1, 0x8A, 0xC7, 0xA9, 0xC8, 0xC0, 0x8C, 0x98, 0xCE, 0x92, 0x87,
0xC1, 0xFF, 0xFF, 0x91, 0xFF, 0xD8, 0xCC, 0xC3, 0xE1
};
static byte displayDigit2[11] = {
0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78, 0x00, 0x10, 0x7F
};

void Scanning_Segment () {
digitalWrite (J_ST, LOW); // Menampilkan data ke 7-Segment driver 74HC595
// (datapin, clockpin, data)

shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [humdS] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [humdP] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [suhuS] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [suhuP] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerSd] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerPd] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerSm] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerPm] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerSj] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [timerPj] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [tandaS] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [tandaP] );

shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Thn_Satuan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Thn_Puluhan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Thn_Ratusan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Thn_Ribuan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Bln_Satuan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Bln_Puluhan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Tgl_Satuan] );
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1 [Tgl_Puluhan] );

shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1[Mnt_Satuan]);

shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1[Mnt_Satuan]);
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1[Mnt_Puluhan]);
if (tandaDip == true) {
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit2[Jam_Satuan]); // untuk KeDIP Detik
} else {
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1[Jam_Satuan]);
}
shiftOut(J_R1, J_SH, MSBFIRST, displayDigit1[Jam_Puluhan]);

digitalWrite (J_ST, HIGH);
}

/*=====
=====*/

```

Lampiran 5 Kode Program Aplikasi

```
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>

<head>
  <title>SURGERY TIME</title>
</head>

<body bgcolor="142831">

  <table width="360" height="100" bgcolor="142831" align="center" border="2" bordercolor="224855">
    <tr align="center">
      <td width="70%">
        <font size="6" color="9cd038"><span color="white" id="getDS3231">00:00:00</span></font><br>
        <font size="3" color="9cd038"><span color="white" id="getanggal">09:01:2023</span></font>
      </td>
      <td width="30%">
        <button style="width: 100%; height: 100%;" type="button" onclick="proseswaktu()">
          <font size="4">DATE</font>
        </button>
      </td>
    </tr>
  </table>

  <br>

  <table width="360" height="200" bgcolor="1a3843" align="center" border="2" bordercolor="224855" cellpadding="4">
    <tr align="center">
      <td>
        <font size="8" color="yellow"><span color="white" id="getTimeElapsed"></span></font><br>
      </td>
    </tr>
  </table>

  <table width="360" cellpadding="4" bgcolor="142831" border="2" bordercolor="224855" align="center">
    <tr align="center" bgcolor="142831" height="60px">

      <td width="20%">
        <font size="3" color="white">No.</font>
      </td>
      <td width="40%">
        <font size="3" color="white">Progress</font>
      </td>
      <td width="40%">
        <font size="3" color="white">Setting</font>
      </td>
    </tr>
    <tr bgcolor="1a3843" align="center" height="50px">
      <td>
        <font color="white">1</font>
      </td>
      <td>
        <font color="9cd038"><span id="time1">0</span></font>
      </td>
      <td><input size="3" type="text" minlength="3" maxlength="3" value="000" id="on01"></td>
    </tr>
    <tr bgcolor="142831" align="center" height="50px">
      <td>
        <font color="white">2</font>
      </td>
      <td>
        <font color="9cd038"><span id="time2">0</span></font>
      </td>
      <td><input size="3" type="text" minlength="3" maxlength="3" value="000" id="on02"></td>
    </tr>
    <tr bgcolor="1a3843" align="center" height="50px">
      <td>
        <font color="white">3</font>
      </td>
      <td>
        <font color="9cd038"><span id="time3">0</span></font>

```

```

        </td>
        <td><input size="3" type="text" minlength="3" maxlength="3" value="000" id="on03"></td>
    </tr>
</table>

<table width="360" height="60" align="center" border="2" bordercolor="224855" cellpadding="4">
    <td width="35%">
        <button style="width: 100%; height: 100%;" type="button" onclick="prosesSetTime()">
            <font size="4">SAVE</font>
        </button>
    </td>
</table>

<br>

<table width="360" height="100" align="center" border="2" bordercolor="224855" cellpadding="4">
    <td width="35%">
        <button style="width: 100%; height: 100%;" type="button" onclick="prosesStart()">
            <font size="5">START</font>
        </button>
    </td>
    <td width="30%">
        <button style="width: 100%; height: 100%;" type="button" onclick="prosesReset()">
            <font size="5">RESET</font>
        </button>
    </td>
    <td width="35%">
        <button style="width: 100%; height: 100%;" type="button" onclick="prosesStop()">
            <font size="5">PAUSE</font>
        </button>
    </td>
</table>

```

```

<script type="text/javascript">

```

```

    function checkTime(i) {
        if (i < 10) {
            i = "0" + i;
        };
        return i;
    }

    function proseswaktu() {
        var date = new Date();
        var ijam = date.getHours();
        var imenit = date.getMinutes();
        var idetik = date.getSeconds();
        var iday = date.getDay();
        var itanggal = date.getDate();
        var ibulan = date.getMonth();
        var itahun = date.getYear();

        ijam = checkTime(ijam);
        imenit = checkTime(imenit);
        idetik = checkTime(idetik);
        itanggal = checkTime(itanggal);
        ibulan = checkTime(ibulan);
        var itahun = (itahun < 1000) ? itahun + 1900 : itahun;

        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function () {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("RTCset").innerHTML =
                    this.responseText;
            }
        };
        xhttp.open("POST", "setJam?RTCset=" + idetik + imenit + ijam + iday + itanggal + ibulan + itahun, true);

```

```

    xhttp.send();
}

function prosesStart() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("dataStart").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("POST", "setStart?dataStart=", true);
    xhttp.send();
}

function prosesStop() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("dataStop").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("POST", "setStop?dataStop=", true);
    xhttp.send();
}

function prosesReset() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("dataReset").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("POST", "setReset?dataReset=", true);
}

}

function prosesSetTime() {
    var J01 = document.getElementById('on01').value;
    var J02 = document.getElementById('on02').value;
    var J03 = document.getElementById('on03').value;

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("sendJadwal").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("POST", "setJadwal?sendJadwal=" + J01 + J02 + J03, true);
    xhttp.send();
}

setInterval(function () { getData() }, 1000);
function getData() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) { document.getElementById("getDS3231").innerHTML = this.responseText }
    };
    xhttp.open("GET", "getDS3231", true);
    xhttp.send();

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) { document.getElementById("getanggal").innerHTML = this.responseText }
    };
    xhttp.open("GET", "getanggal", true);
}

```

```

var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) { document.getElementById("getTimeElapsed").innerHTML = this.responseText }
};
xhttp.open("GET", "getTimeElapsed", true);
xhttp.send();

var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) { document.getElementById("time1").innerHTML = this.responseText }
};
xhttp.open("GET", "time1", true);
xhttp.send();

var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) { document.getElementById("time2").innerHTML = this.responseText }
};
xhttp.open("GET", "time2", true);
xhttp.send();

var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) { document.getElementById("time3").innerHTML = this.responseText }
};
xhttp.open("GET", "time3", true);
xhttp.send();

}

</script>

</body>

</html>
)=====";

```