

LAMPIRAN

Lampiran 1. Coding Keseluruhan Sistem

```
#include <Wire.h>
#include <RTCLib.h>
#include "PrayerTimes.h"
#include "DFRobotDFPlayerMini.h"
#include <SoftwareSerial.h>

SoftwareSerial mySoftwareSerial(12, 13);
DFRobotDFPlayerMini myDFPlayer;
RTC_DS3231 rtc;
SoftwareSerial GPRS(10, 11);

// Variabel praytime
double times[sizeof(TimeName) /
sizeof(char*)];
byte hour, minute, second, year, month,
date;

const int relayPin = 4; // Ubah pin relay
sesuai dengan yang Anda gunakan
boolean relayState = false;

boolean audioPlaying = false;
unsigned long audioStartTime = 0;

boolean alatStatus = false;
String allowedPhoneNumber =
"082136132508";
void setup(void) {
    Serial.begin(9600);
    mySoftwareSerial.begin(9600);
    GPRS.begin(9600);
    GPRS.listen();
    delay(5000);
    Serial.print("GPRS siap...\r\n");
    GPRS.print("AT+CMGF=1\r\n");
    delay(1000);
    GPRS.print("AT+CNMI=2,2,0,0,0\r\n");
    String buffer = "";

    Wire.begin();
    rtc.begin();
    hour = rtc.now().hour();
    minute = rtc.now().minute();
    second = rtc.now().second();
    year = rtc.now().year();
    month = rtc.now().month();
    date = rtc.now().day();

    myDFPlayer.begin(mySoftwareSerial);
    myDFPlayer.volume(20);

    pinMode(relayPin, OUTPUT);
    digitalWrite(relayPin, LOW);
}
String getCallerNumber() {
```

```

GPRS.println("AT+CLCC");

delay(100);

String response =
GPRS.readStringUntil("\n");

return response;
}

void loop() {
    while (GPRS.available()) {

        String sms =
GPRS.readStringUntil("\n");

        sms.trim();

        Serial.println("SMS Diterima: " + sms);

        if (sms.equalsIgnoreCase("ON")) {
            alatStatus = true; // Hidupkan alat

            digitalWrite(relayPin, HIGH);

            sendSMS("Sound System Telah
Hidup");

        } else if
(sms.equalsIgnoreCase("TELPON")) {

            String phoneNumber =
"082136132508"; // Ganti dengan nomor
telepon yang ingin Anda hubungi

            Serial.println("Menelepon " +
phoneNumber);

            makeCall(phoneNumber);

        } else if (sms.equalsIgnoreCase("OFF"))
        {

            alatStatus = false; // Matikan alat

            digitalWrite(relayPin, LOW);

            sendSMS("Sound System Telah
Mati");

        } else if (sms.startsWith("VOL")) {

            // Periksa pesan SMS yang
mengandung perintah volume

```

```

int volumeValue = 0;

if (sscanf(sms.c_str(), "VOL%d",
&volumeValue) == 1) {

    // Memeriksa apakah pesan SMS
berisi angka volume yang valid

    if (volumeValue >= 0 &&
volumeValue <= 30) {

        // Mengatur volume DFPlayer
Mini

myDFPlayer.volume(volumeValue);

        // Kirim balasan SMS untuk
mengonfirmasi perubahan volume

        sendSMS(("Volume diatur ke " +
String(volumeValue)).c_str());

    }

}

}

}

if (GPRS.available()) {

    String callStatus =
GPRS.readStringUntil("\n");

    callStatus.trim();

    if (callStatus.startsWith("RING")) {

        delay(2000);

        String phoneNumber =
getCallerNumber();

        if
(phoneNumber.equals(allowedPhoneNumb
er)) {

            GPRS.println("ATA");

            GPRS.println("AT+CHLD=1");

            Serial.println("Mengangkat
panggilan...");

```

```

    } else {
        GPRS.println("ATH");

        Serial.println("Panggilan dari nomor
yang tidak diizinkan. Panggilan ditolak.");
    }
}

DateTime now = rtc.now();
double lintang = -3.9194525; // Latitude
double bujur = 119.7864342; //
Longitude
int Z = 8; // Zona Waktu GMT,
WIB biasanya 7

int jam, menit; // Jam sholat dan
menit sholat

set_calc_method(ISNA); // Metode
perhitungan mengikuti ISNA
set_asr_method(Shafii); // Pendapat
Imam Syafi'i

set_high_lats_adjust_method(AngleBased)
;

set_fajr_angle(20); // Sudut Waktu
Subuh

set_isha_angle(18); // Sudut Waktu
Isya

get_prayer_times(year, month, date,
lintang, bujur, Z, times);

// Jadwal sholat
// Shubuh

```

```

    get_float_time_parts(times[0], jam,
menit);

    if (now.hour() == jam && now.minute()
== (menit-20)) {
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(3);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
audioStartTime >= 3600000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }

    // Dhuhur

    get_float_time_parts(times[2], jam,
menit);

    if (now.hour() == jam && now.minute()
== (menit-15)) {
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(4);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
audioStartTime >= 3600000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }

```

```

    }

    // Ashar
    get_float_time_parts(times[3], jam,
    menit);

    if (now.hour() == jam && now.minute()
    == (menit-20)){
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);

            playAudio(5);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
    audioStartTime >= 3600000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }

    // Maghrib
    get_float_time_parts(times[4], jam,
    menit);

    if (now.hour() == jam && now.minute()
    == (menit-20)){
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(6);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }
}

```

```

        if (audioPlaying && millis() -
        audioStartTime >= 3600000) {
            digitalWrite(relayPin, LOW);
            audioPlaying = false;
        }

        // Isya
        get_float_time_parts(times[6], jam,
        menit);

        if (now.hour() == jam && now.minute()
        == (menit-20)) {
            if (!audioPlaying) {
                digitalWrite(relayPin, HIGH);
                playAudio(7);
                audioPlaying = true;
                audioStartTime = millis();
            }
        }

        if (audioPlaying && millis() -
        audioStartTime == 3600000) {
            digitalWrite(relayPin, LOW);
            audioPlaying = false;
        }

        //menentukan kapan Ramadan dimulai
        dan berakhir
        if ((now.year() == 2024 && now.month()
        == 3 && now.day() >= 12) ||
            (now.year() == 2024 && now.month()
            == 4 && now.day() <= 11)) {
            Serial.println("Saatnya memutar audio
            untuk Ramadan!");
        }

        // Shubuh
    }
}

```

```

    get_float_time_parts(times[0], jam,
menit);

    if (now.hour() == jam && now.minute()
== (menit-31)) {
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(2);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
audioStartTime == 3600000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }
}

// Maghrib

get_float_time_parts(times[4], jam,
menit);

    if (now.hour() == jam && now.minute()
== (menit-19)) {
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(1);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
audioStartTime == 3600000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;

```

```

    }
    // Isya

    get_float_time_parts(times[6], jam,
menit);

    if (now.hour() == jam && now.minute()
== (menit-20)) {
        if (!audioPlaying) {
            digitalWrite(relayPin, HIGH);
            playAudio(7);
            audioPlaying = true;
            audioStartTime = millis();
        }
    }

    if (audioPlaying && millis() -
audioStartTime >= 14400000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }
}

// Tambahkan kondisi untuk menentukan
apakah hari ini adalah hari Jumat

    if (now.dayOfTheWeek() == 5) {
        get_float_time_parts(times[2], jam,
menit);

        if (now.hour() == jam && now.minute()
== (menit-40)){
            if (!audioPlaying) {
                digitalWrite(relayPin, HIGH);
                playAudio(8);
                audioPlaying = true;
                audioStartTime = millis();
            }

```

```

    }

    if (audioPlaying && millis() -
audioStartTime >= 7200000) {
        digitalWrite(relayPin, LOW);
        audioPlaying = false;
    }
}

printDateTime(now);

delay(10000);

}

void relayHidup() {
    digitalWrite(relayPin, HIGH);
}

void relayMati() {
    digitalWrite(relayPin, LOW);
}

void playAudio(int trackNumber) {
    myDFPlayer.play(trackNumber);

    delay(500000); // Delay sementara
sebelum memeriksa status pemutaran

    while (myDFPlayer.available()) {
        myDFPlayer.readType();
    }
}

void sendSMS(const char* pesan) {
    GPRS.println("AT");

```

```

    delay(500);

    while (Serial.available()) {
        GPRS.write(Serial.read());
    }

    while (GPRS.available()) {
        Serial.write(GPRS.read());
    }

    GPRS.println("AT+CMGF=1");

    delay(500);

    while (Serial.available()) {
        GPRS.write(Serial.read());
    }

    while (GPRS.available()) {
        Serial.write(GPRS.read());
    }

    GPRS.println("AT+CMGS=\"0821361325
08\""); // Ganti dengan nomor tujuan untuk
notifikasi

    delay(500);

    while (Serial.available()) {
        GPRS.write(Serial.read());
    }

    while (GPRS.available()) {
        Serial.write(GPRS.read());
    }

    GPRS.print(pesan);

    delay(500);

```

```

while (Serial.available()) {
    GPRS.write(Serial.read());
}

while (GPRS.available()) {
    Serial.write(GPRS.read());
}

GPRS.write(26);
delay(500);
while (Serial.available()) {
    GPRS.write(Serial.read());
}

while (GPRS.available()) {
    Serial.write(GPRS.read());
}
}

void makeCall(String phoneNumber) {
    GPRS.print("ATD");
    GPRS.print(phoneNumber);
    GPRS.println(";");
    delay(5000); // Tambahkan delay agar
    panggilan terjadi dengan baik
    while (Serial.available()) {
        GPRS.write(Serial.read());
    }
    while (GPRS.available()) {
        Serial.write(GPRS.read());
    }
}

void callStatus (String phoneNumber) {
    GPRS.print("ATA");

```

```

    GPRS.print(phoneNumber);
    GPRS.println(";");
    delay(2000); // Tambahkan delay agar
    panggilan terjadi dengan baik
    while (Serial.available()) {
        GPRS.write(Serial.read());
    }
    while (GPRS.available()) {
        Serial.write(GPRS.read());
    }
}

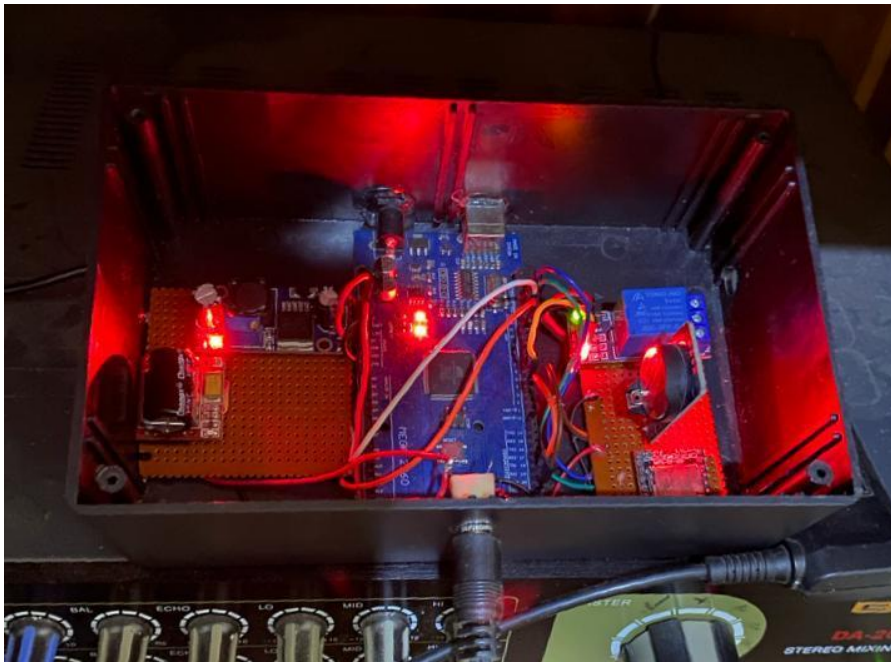
void printDateTime(const DateTime& dt)
{
    Serial.print(dt.year(), DEC);
    Serial.print('/');
    Serial.print(dt.month(), DEC);
    Serial.print('/');
    Serial.print(dt.day(), DEC);
    Serial.print(" ");
    Serial.print(dt.hour(), DEC);
    Serial.print(':');
    Serial.print(dt.minute(), DEC);
    Serial.print(':');
    Serial.print(dt.second(), DEC);
    Serial.println();
}

```

Lampiran 2. Dokumentasi Alat Keseluruhan



Lampiran 3. Dokumentasi Alat Dalam Keadaan Terbuka



Lampiran 4. Alat terpasang di Masjid

