

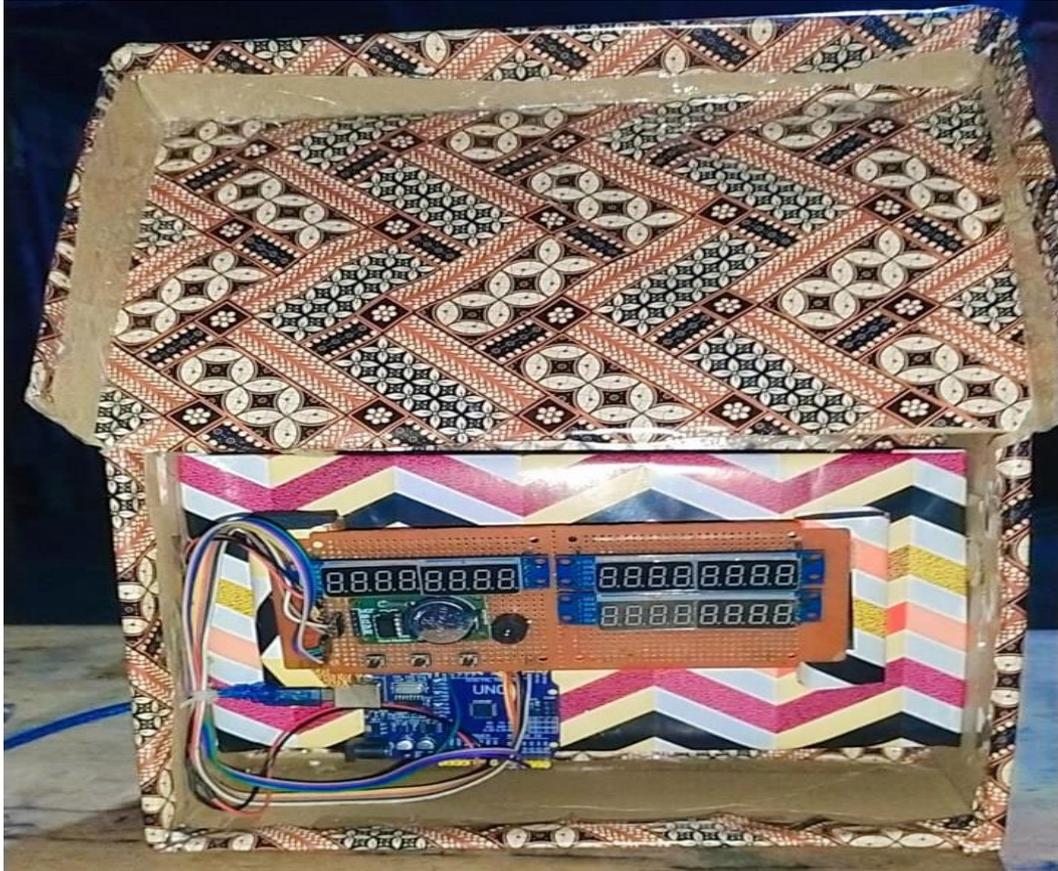
## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Impelementasi Perangkat Keras (*Hardware*)**

Pada penelitian rancang bangun pengingat waktu salat berbasis Arduino guna membantu umat Muslim dalam melaksanakan Salat tepat waktu dan lebih konsisten. Penulis menggunakan komponen mikrokontroler Arduino Uno, modul RTC DS1302 (*Real-Time Clock*), MAX7219 7 segment, buzzer, dan tombol. Untuk menciptakan alat yang dapat memberikan peringatan waktu Salat secara otomatis saat waktu Salat tiba.

Dalam penggunaannya, perangkat akan membaca waktu dari modul RTC DS1302 dan membandingkannya dengan jadwal waktu Salat yang telah ditentukan sebelumnya. Ketika waktu Salat telah tiba, perangkat akan memberikan peringatan melalui bunyi dari buzzer dan menampilkan informasi waktu Salat berikutnya pada display MAZ7219 7 segment.

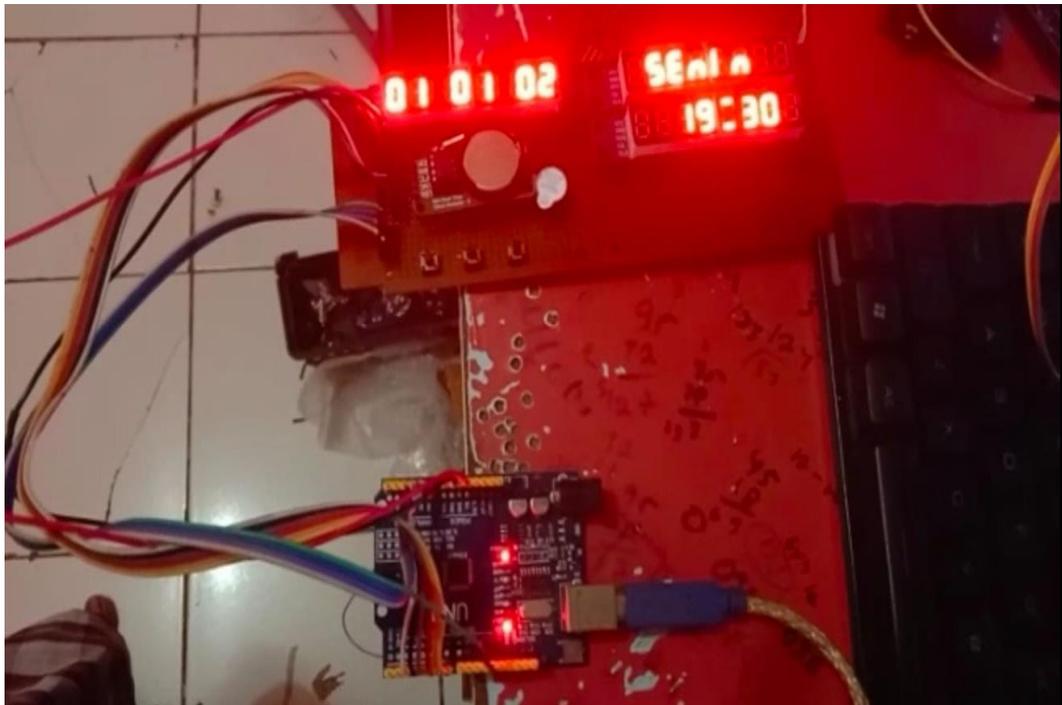


Gambar 4.1 Hardware Pengingat Waktu Salat

Gambar 4.1 adalah gambaran tentang seluruh komponen *hardware* (perangkat keras) yang telah dikonfigurasi dan disusun menjadi satu dan di letakkan pada box atau tempat penyimpanan. Terdapat beberapa komponen yang ada dalam Gambar 4.1 yaitu:

- a. Arduino
- b. MAX7219 7 Segment
- c. RTC DS 1302 (*Real-Time Clock*)
- d. MAX7219 7 Segment

- e. Buzzer
- f. Button
- g. Kabel Jumper
- h. Pcb Bolong

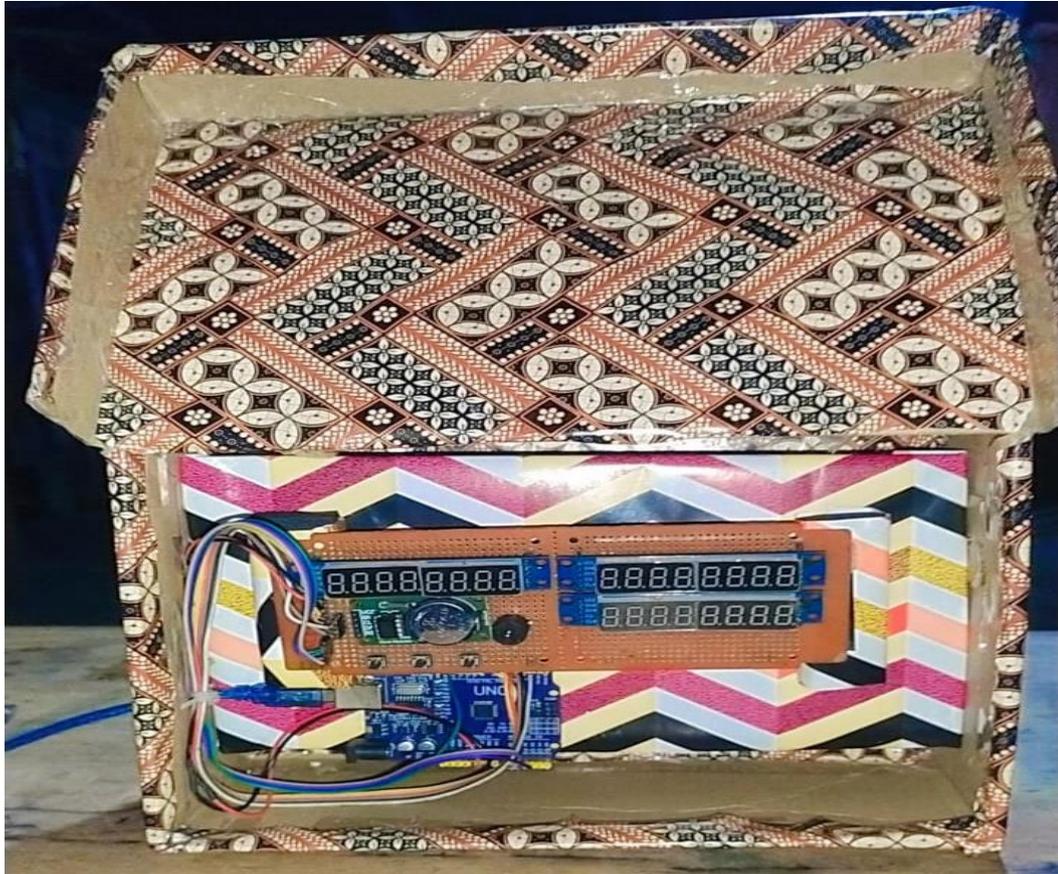


Gambar 4.2 Alat sebelum dimasukkan kedalam box



Gambar 4.3 Proses penghubungan arduino ke PCB Bolong

Penghubungan Arduino dengan PCB adalah proses menyusun komponen dan koneksi elektronik pada sebuah papan sirkuit (PCB) untuk membangun perangkat berbasis Arduino.



Gambar 4.4 Proses perakitan alat ke box penyimpanan

Gambar 4.4 menggambarkan proses perakitan sistem ke dalam box penyimpanan dengan menempelkan double tape/lem pada PCB, Arduino Uno, dan akan diletakkan di box penyimpanan. Gambar 4.5 menggambarkan hasil dari perakitan semua komponen yang telah dimasukkan ke dalam box penyimpanan guna untuk melindungi komponen.

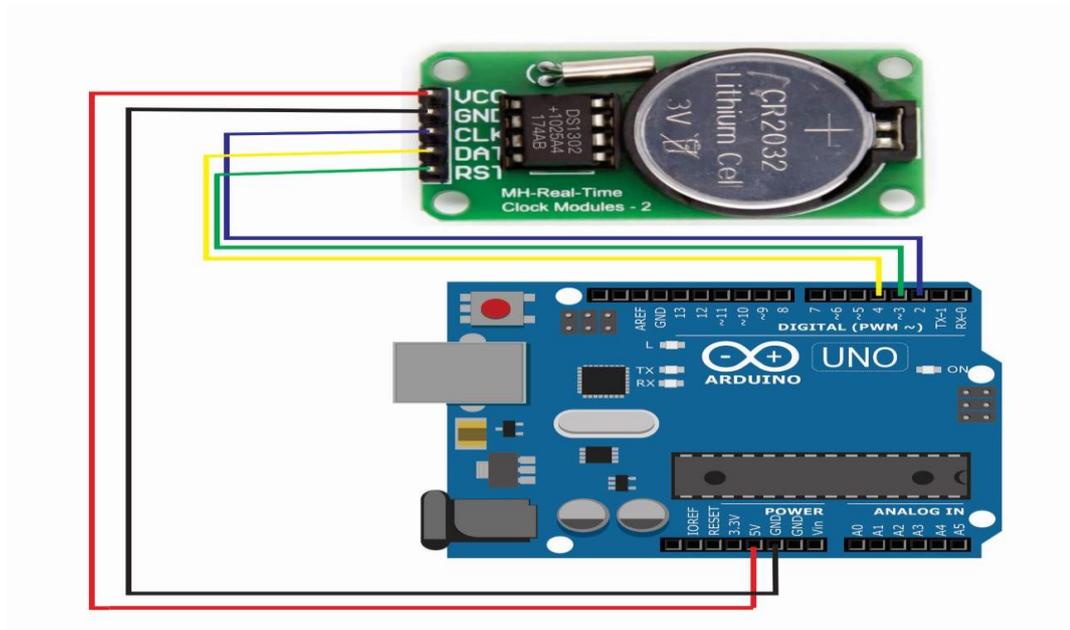


Gambar 4.5 Hasil perakitan alat

#### 4.2 Konfigurasi RTC DS1302

RTC DS1302 digunakan untuk mengatur waktu dengan akurat. Untuk menghubungkannya dengan Arduino, kita perlu menghubungkan beberapa kabel. Pertama, kita sambungkan kabel RTC CLK ke pin 2 pada Arduino untuk mengatur waktu dan sinkronisasi data dengan tepat. kabel RTC DAT dihubungkan ke pin 3 untuk mentransmisikan data antara RTC DS1302 dan Arduino, kabel RTC RST ke pin 4 untuk memberikan kemampuan reset pada modul RTC, kaki negatif (GND) RTC hubungkan ke ground (GND) pada Arduino. kaki VCC RTC dihubungkan ke pin 5V pada Arduino untuk menyediakan suplai daya 5V

yang dibutuhkan oleh RTC DS1302. Dengan pengaturan ini, Arduino dapat berkomunikasi dengan RTC DS1302 dan memastikan waktu yang tepat ditampilkan pada alat, konfigurasi dapat dilihat pada Gambar 4.6



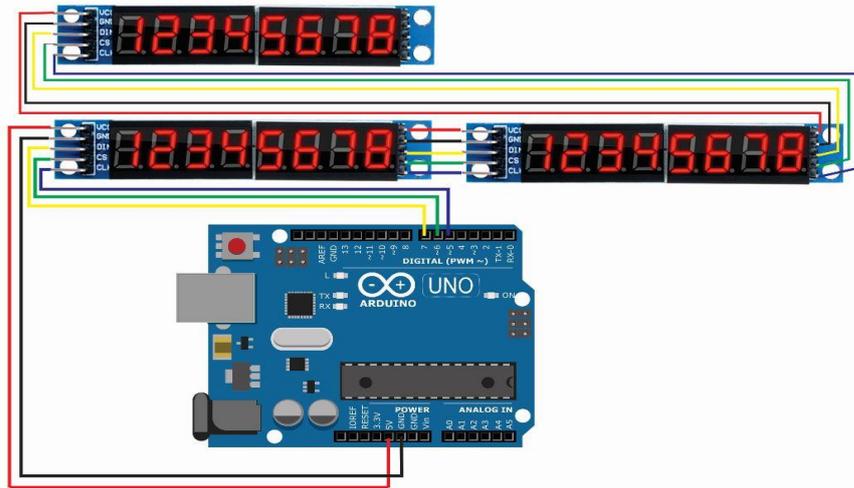
Gambar 4.6 Diagram Blok RTC DS1302

RTC DS1302	Koneksi ke Arduino Pin
VCC	5V
GND	GND
CLK	2
DAT	3
RST	4

Tabel 4.7 Konfigurasi RTC DS1302

### 4.3 Konfigurasi Arduino MAX7219 7 Segment

MAX7219 sebuah perangkat driver yang berfungsi sebagai kontroler untuk menampilkan angka dan karakter pada display LED tujuh segment



Gambar 4.8 Diagram Blok MAX7219 7 Segment

Penulis menggunakan tiga modul MAX7219. Kaki data output dari modul pertama dihubungkan ke data input modul kedua, dan seterusnya hingga modul ketiga. Ini membentuk rangkaian kaskade, memungkinkan kita mengontrol ketiga modul dengan satu pin data dari Arduino. Lalu, kita sambungkan pin DIN ke pin 7 pada Arduino sebagai koneksi data input. Kemudian, pin CS dihubungkan ke pin 6 pada Arduino. Selanjutnya, pin CLK dihubungkan ke pin 5 pada Arduino sebagai koneksi clock untuk mengatur transfer data. Terakhir, kita hubungkan kaki negatif (GND) ke ground (GND) Arduino dan pin VCC ke pin 5V Arduino untuk mendapatkan suplai daya 5V. Dengan sambungan ini, kita dapat

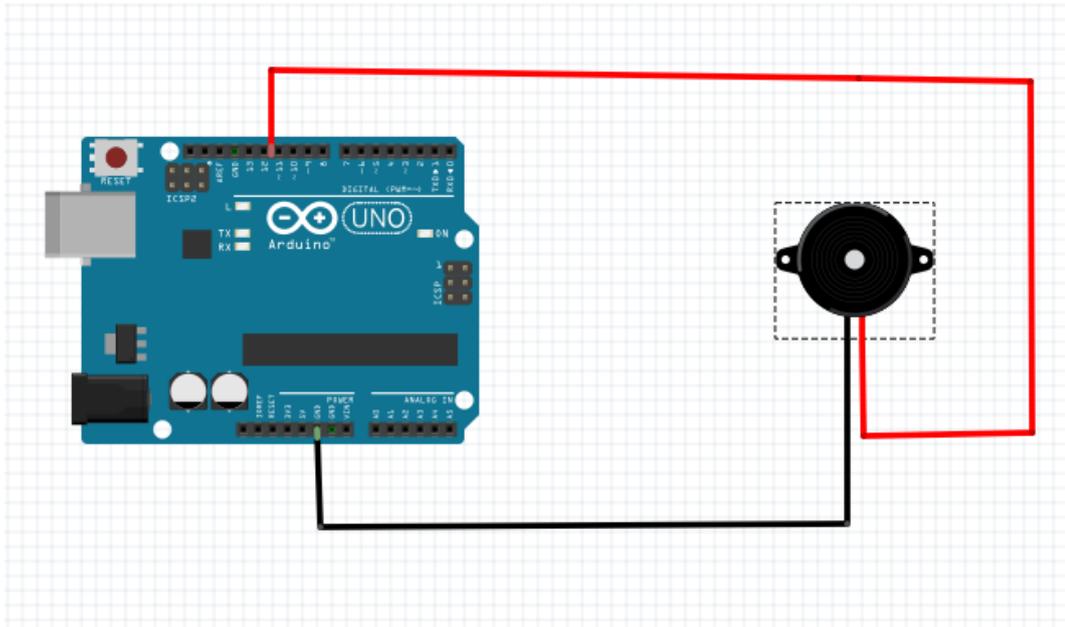
mengontrol dan menampilkan informasi pada display 7-segment dari ketiga modul. konfigurasinya dapat dilihat seperti Gambar 4.8

<b>MAX7219 7 Segment</b>	<b>Koneksi ke Arduino Pin</b>
VCC	5V
GND	GND
DIN	7
CS	6
CLK	5

Tabel 4.9 Konfigurasi MAX7219 7 Segment

#### 4.4 Konfigurasi Buzzer

Buzzer berfungsi sebagai pengingat waktu Salat dengan memberikan peringatan audio ketika waktu Salat tiba.

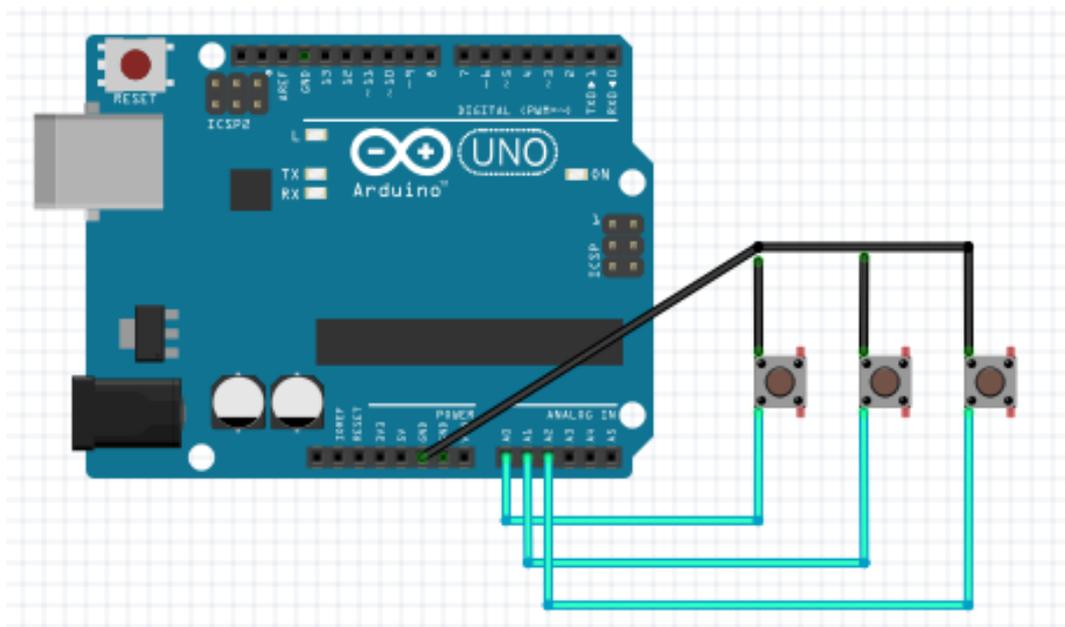


Gambar 4.10 Diagram Blok Buzzer

Buzzer memiliki 2 pin yaitu pin daya positif dan pin daya negatif. Pin daya positif dari buzzer langsung dihubungkan ke pin 12 Arduino dan pin daya negative buzzer dihubungkan ke pin GND pada Arduino dan konfigurasi nya dapat dilihat seperti Gambar 4.10

## 4.5 Konfigurasi Button

Button berfungsi sebagai input tambahan untuk mengatur waktu Salat dan mengontrol pengingat. Dengan menekan tombol, pengguna dapat memasuki mode pengaturan waktu Salat dan mengubah nilainya. Selain itu, tombol juga dapat digunakan untuk mematikan atau mereset pengingat waktu Salat sesuai kebutuhan. Tombol memberikan interaksi antara pengguna dan sistem, memudahkan pengaturan waktu Salat, dan memberikan kontrol atas pengingat yang aktif atau nonaktif.



Gambar 4.11 Diagram Blok Button

Tiga tombol (Button 1, Button 2, Button 3) dihubungkan ke Arduino pada pin A0, A1, dan A2 dengan kaki negatifnya ke ground (GND). Ketika ditekan, nilai input akan berubah, dan Arduino dapat membaca kondisi tombol tersebut untuk melakukan tindakan tertentu sesuai program yang dijalankan.

## 4.6 Pembahasan Sintak Program

Penjelasan sintak program Rancang Bangun Peningat Waktu Salat Berbasis Arduino. Menjelaskan sintak-sintak yang ada pada program. Sintak sendiri merupakan kode program yang memerintahkan sistem sehingga sistem dapat berjalan sesuai dengan apa yang diharapkan.

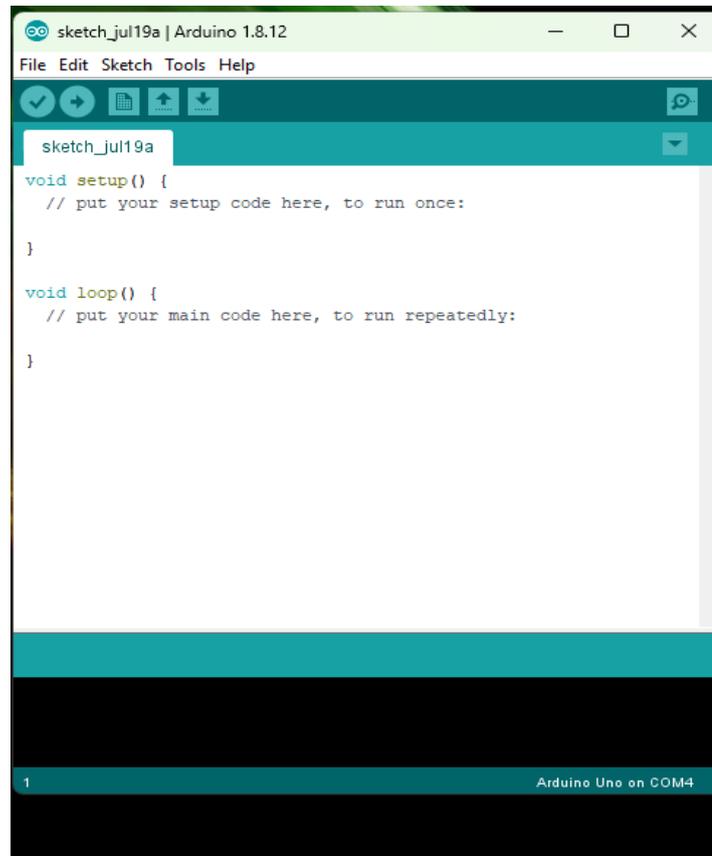
Pertama kali sebelum melakukan pemrograman, terlebih dahulu menginstall Arduino IDE agar dapat menuliskan sintak program sehingga mikrokontroler Arduino Uno dapat diperintah sesuai dengan keinginan. Setelah menginstall Arduino IDE hubungkan mikrokontroler Arduino Uno ke komputer/laptop menggunakan kabel USB dan menginstall driver Arduino Uno agar papan mikrokontroler tersebut terdeteksi sehingga sintak program yang ditulis dapat masuk ke dalam mikrokontroler.

### 4.6.1 Awal Pembuatan Program

Struktur dasar dari bahasa pemrograman arduino itu sederhana hanya terdiri dari dua bagian. Dimana *setup()* bagian untuk inisialisasi yang hanya dijalankan sekali di awal program, sedangkan *loop()* untuk mengeksekusi bagian program yang akan dijalankan berulang-ulang untuk selamanya.

Fungsi *setup()* hanya di panggil satu kali ketika program pertama kali di jalankan. Ini digunakan untuk pendefinisian mode pin atau memulai komunikasi serial. Fungsi *setup()* harus di ikut sertakan dalam program walaupun tidak ada *statement* yang di jalankan. Setelah melakukan fungsi *setup()* maka secara langsung akan melakukan fungsi *loop()* secara berurutan dan melakukan instruksi-instruksi yang ada dalam fungsi *loop()* ditanamkan suatu program bernama

*Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan *mikrokontroler*.



## 4.12 Program Arduino

### a. void setup

berikut adalah penjelasan singkat mengenai bagian kode dalam fungsi void setup():

#### 1. LED Display:

Menghidupkan tiga modul LED display dengan menonaktifkan mode shutdown dan mengatur tingkat kecerahan ke 8.

## 2. RTC:

Mengatur mode pin untuk komunikasi dengan RTC. RTC\_RST dan RTC\_CLK diatur sebagai output, sedangkan RTC\_DAT diatur sebagai input. RTC\_RST ditetapkan dalam kondisi rendah mungkin untuk mereset RTC.

## 3. Button:

Mengatur mode pin tombol sebagai input dengan pull-up resistor yang diaktifkan. Tombol BTN\_DN, BTN\_UP, dan BTN\_SET diatur demikian.

## 4. Buzzer:

Mengatur mode pin untuk pengaturan bunyi alarm (buzzer) sebagai output. Bunyi alarm awalnya dimatikan dengan mengirimkan sinyal rendah.

## 5. Delay (1000)

```
void setup() {
    Lc.shutdown(0, false);
    Lc.setIntensity(0, 8);
    Lc.shutdown(1, false);
    Lc.setIntensity(1, 8);
    Lc.shutdown(2, false);
    Lc.setIntensity(2, 8);
    Lc.shutdown(3, false);
    Lc.setIntensity(3, 8);

    pinMode(RTC_RST, OUTPUT);
    pinMode(RTC_CLK, OUTPUT);
    pinMode(RTC_DAT, INPUT);
    digitalWrite(RTC_RST, LOW);

    pinMode(BTN_DN, INPUT_PULLUP);
    pinMode(BTN_UP, INPUT_PULLUP);
    pinMode(BTN_SET, INPUT_PULLUP);

    pinMode(AL_BUZZ, OUTPUT);
    digitalWrite(AL_BUZZ, LOW);

    delay(1000);
}
```

### 4.13 Program void setup

b. void loop menjalankan tugas-tugas ini secara berulang

```

void loop() {
  unsigned long current = millis();
  if ((current >= timeplaydatejws + 3000) && (setMode == 0)) {
    lockjws++;
    playjws();

//    playdatejwsSw ? playdate() : playjws();
//    playdatejwsSw = 1 - playdatejwsSw;
    timeplaydatejws = current;
  }
  if((current-waktuLed>=500)&&(setMode == 0)){
statusLed? titik_of():titik_on();
statusLed=1-statusLed;
waktuLed=millis();
}
if((current-waktupdate>=2000)&&(setMode == 0)){
statusplay? playdate():displayhari();
statusplay=1-statusplay;
waktupdate=millis();
}
  if ((current >= timeUpdate + 1000) && (setMode == 0)) {
    if (ss == 0) {
      updateDisplay();
      if ((hh == jwssbh2) && (mm == jwssbh1)){
        bezzer =1;
      }
    }
  }
}

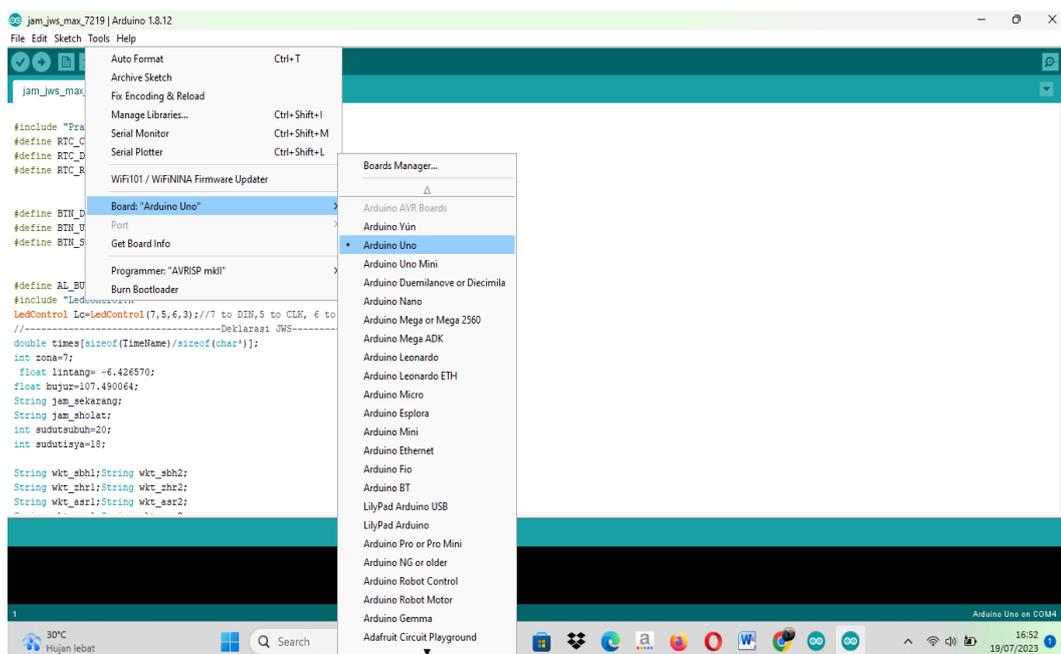
```

---

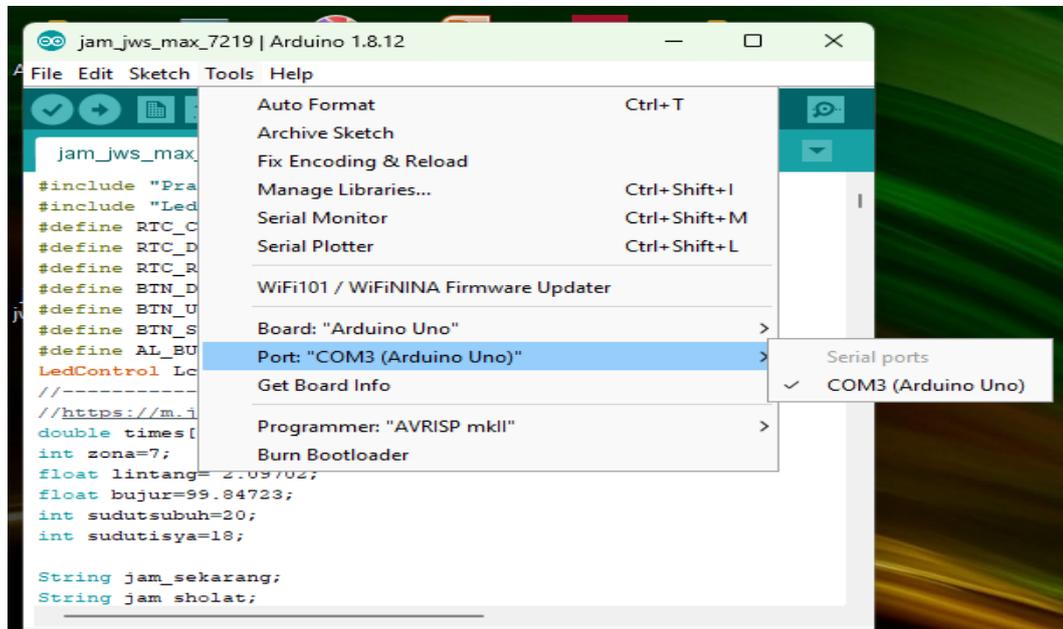
#### 4.14 Program void loop

## 4.6.2 Mengupload Program

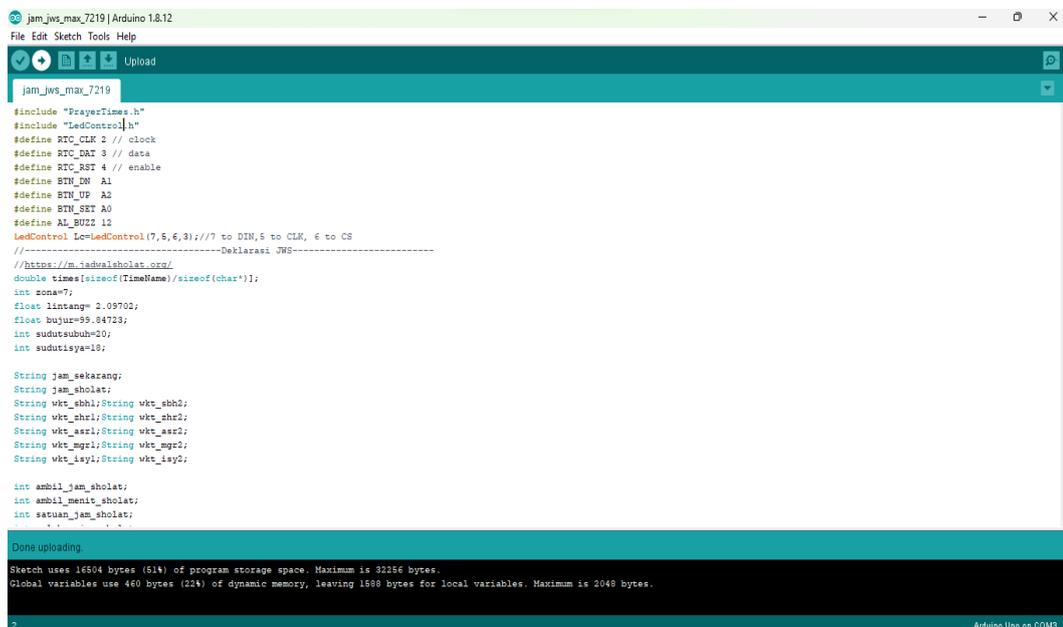
Sebelum mengupload program selesaikan dahulu programnya. Terdapat 5 (lima) menu utama yang terdapat pada bagian atas IDE yaitu *file*, *edit*, *sketch*, *tools* dan *help*. Untuk men-setting *board* yang digunakan, Klik pada menu “Tools”, pada bagian “Board”, pilih sesuai dengan *board* arduino yang digunakan (dalam kasus ini adalah Arduino Uno). Selanjutnya adalah setting port, pada bagian “Port”, pilih *port* yang tersedia (dalam kasus ini adalah COM3). Untuk beberapa keadaan, nama *port* yang ditampilkan bisa berbeda. Setelah itu Upload *sketch* tahap dimana program ditanamkan pada board arduino. diasumsikan bahwa kita telah selesai menulis program yang akan ditanamkan pada board arduino. Untuk mulai meng-upload program, hubungkan board arduino ke Laptop melalui USB, klik pada *icon* Panah Kanan.



Gambar 4.15 Board Arduino Uno



Gambar 4.16 Port COM3

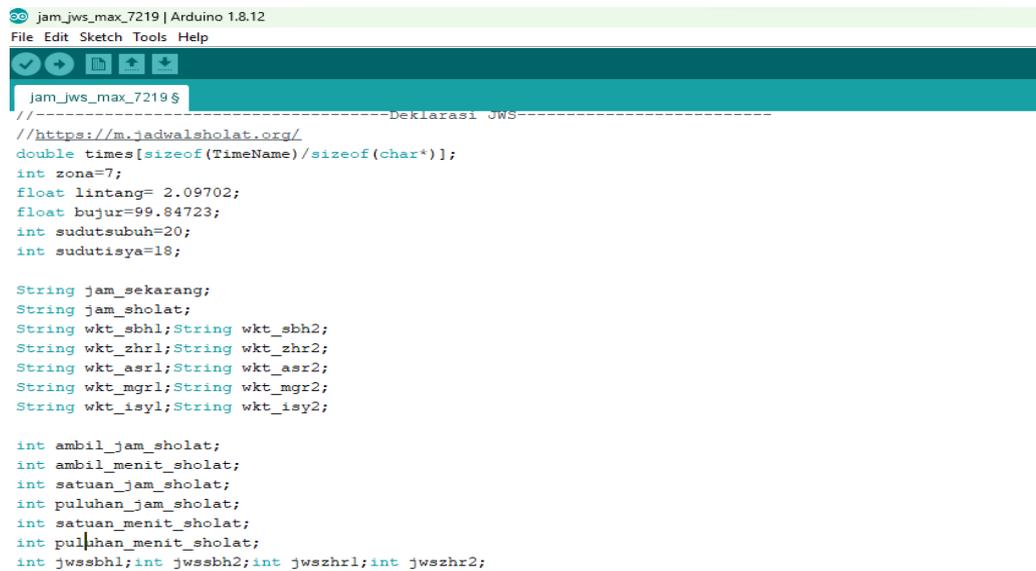


Gambar 4.17 Proses berhasil mengupload program

Atur lokasi untuk perhitungan waktu Salat sesuai dengan lokasi kita

-Garis Lintang-*latitude* = 2.09702; // sesuai dengan lintang lokasi kita

-Garis Bujur-*longitude* =99.84723; // sesuai dengan bujur lokasi kita



```

jam_jws_max_7219 | Arduino 1.8.12
File Edit Sketch Tools Help
jam_jws_max_7219 $
//-----Deklarasi JWS-----
//https://m.jadwalsholat.org/
double times[sizeof(TimeName)/sizeof(char)];
int zona=7;
float lintang= 2.09702;
float bujur=99.84723;
int sudutsubuh=20;
int sudutisya=18;

String jam_sekarang;
String jam_sholat;
String wkt_sbhl;String wkt_sbh2;
String wkt_zhrl;String wkt_zhr2;
String wkt_asr1;String wkt_asr2;
String wkt_mgr1;String wkt_mgr2;
String wkt_isy1;String wkt_isy2;

int ambil_jam_sholat;
int ambil_menit_sholat;
int satuan_jam_sholat;
int puluhan_jam_sholat;
int satuan_menit_sholat;
int puluhan_menit_sholat;
int jwsbh1;int jwsbh2;int jwszhr1;int jwszhr2;

```

Gambar 4.18 Mengatur waktu pada RTC DS1302

```

pinMode (BTN_DN, INPUT_PULLUP);
pinMode (BTN_UP, INPUT_PULLUP);
pinMode (BTN_SET, INPUT_PULLUP);

pinMode (AL_BUZZ, OUTPUT);
digitalWrite (AL_BUZZ, LOW);
delay (1000);

```

Gambar 4.19 Untuk mengatur Buzzer

```

void setup() {
    Lc.shutdown(0, false);
    Lc.setIntensity(0, 8);

    Lc.shutdown(1, false);
    Lc.setIntensity(1, 8);
    Lc.shutdown(2, false);
    Lc.setIntensity(2, 8);
    Lc.shutdown(3, false);
    Lc.setIntensity(3, 8);
}

```

Gambar 4.20 Mengendalikan tampilan MAX7219 7 segment

```

if (current >= timeButton + 10) {
    if ((digitalRead(BTN_SET) == LOW)) {
        buttonPress = true;
        setMode++; // Pindah ke mode pengaturan berikutnya
        setVariable();
    }
    if ((digitalRead(BTN_UP) == LOW) && (setMode) < 4)
        //Tombol untuk menambah nilai pengaturan ditekan
        buttonPress = true;
        adjustVariable(1);
    }
    if ((digitalRead(BTN_DN) == LOW) && (setMode) > 0) {
        //Tombol untuk mengurangi nilai pengaturan ditekan
        buttonPress = true;
        adjustVariable(-1);
    }
    timeButton = current;
    if (buttonPress) {
        timeButton += 200; // Tunda tekanan tombol berikutnya sebesar 200 ms
        buttonPress = false;
    }
}
}

```

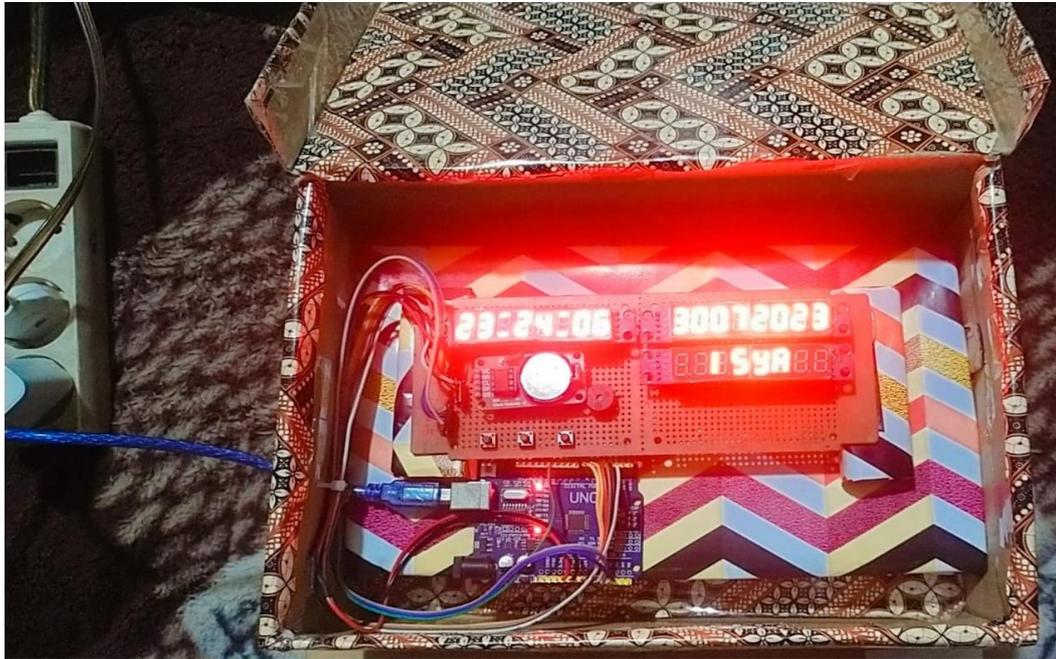
Gambar 4.21 Mengatur Button

## **4.7 Pengujian perangkat**

Fungsi dari pengujian perangkat untuk memastikan bahwasannya “Rancang Bangun Pengingat Waktu Salat Berbasis Arduino” telah sesuai dengan yang diharapkan dan seluruh fungsi yang ada di dalam sistem dapat berjalan lancar. Selain itu akan membahas kekurangan dan kelebihan dari sistem ini sehingga nantinya dapat menjadi bahan acuan untuk penelitian selanjutnya.

### **4.7.1 Pengujian RTC DS1302**

Tujuan pengujian RTC (*Real Time Clock*) adalah untuk memastikan bahwa RTC berfungsi dengan akurat dan andal dalam menghitung waktu secara terus-menerus. Pengujian ini bertujuan untuk memverifikasi bahwa RTC dapat menyimpan dan menampilkan waktu dengan tepat, termasuk tanggal dan jam yang sesuai dengan waktu yang sebenarnya. Dengan melakukan pengujian, kita dapat memastikan bahwa RTC DS1302 beroperasi sesuai yang diharapkan, dapat diatur dengan benar melalui Arduino, serta mampu menjaga waktu yang tepat bahkan saat alat mati dan tidak mendapat pasokan daya listrik. Keakuratan dan keandalan RTC sangat penting dalam membuat “Rancang Bangun Pengingat Waktu Salat Berbasis Arduino”.



Gambar 4.22 Tampilan waktu pada alat

No	Nama Pengujian	Hasil Pengujian
1	Akurasi Waktu	Akurat dan konsisten
2	Penyimpanan Data	Data waktu tetap tersimpan
3	Pengaturan Waktu	Dapat diatur dengan benar
4	Backup Baterai	Backup baterai berfungsi baik
5	Akurasi Kalender	Menampilkan tanggal dengan benar

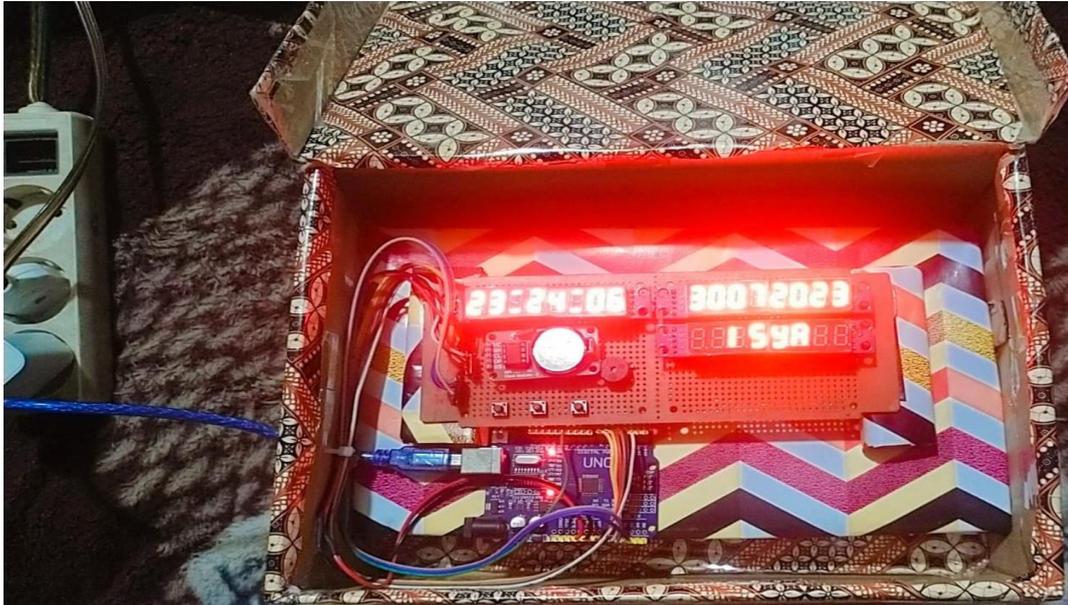
Gambar 4.23 Tabel pengujian RTC DS1302

No	Waktu Salat	Tanggal	Waktu Tampilkan RTC	Waktu Rantauprapat	Selisih Waktu (RTC - Referensi)
1	Subuh	08/08/2023	05:02	05:02	00:00:00
2	Dzuhur	08/08/2023	12:28	12:29	1 menit
3	Ashar	08/08/2023	15:49	15:50	1 menit
4	Maghrib	08/08/2023	18:34	18:35	1 menit
5	Isya	08/08/2023	19:46	19:46	00:00:00

Gambar 4.24 Tabel akurasi waktu pada RTC DS1302

#### 4.7.2 Pengujian MAX7219 7 Segment

Tujuan pengujian MAX7219 adalah untuk memeriksa apakah modul ini bekerja dengan baik dan dapat mengendalikan layar LED dengan benar. Pengujian bertujuan untuk memastikan bahwa MAX7219 dapat menampilkan angka atau gambar dengan jelas, sesuai dengan waktu yang diatur, dan memiliki kecerahan yang tepat. Selain itu, pengujian dilakukan untuk memeriksa apakah modul merespons tombol atau kontrol tambahan, serta mengonsumsi daya listrik secara efisien. Dengan melakukan pengujian ini, kita dapat memastikan bahwa MAX7219 dapat diandalkan dan sesuai untuk digunakan dalam pembuatan “Rancang Bangun Pengingat Waktu Salat Berbasis Arduino”



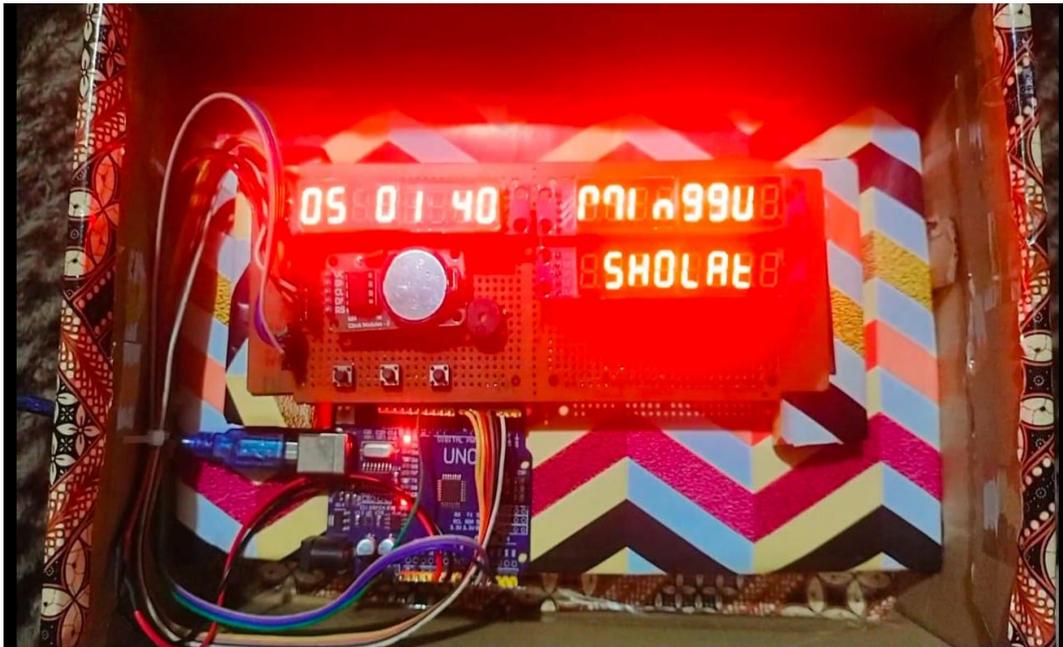
Gambar 4.25 Tampilan MAX7219 7 Segment

No.	Pengujian	Hasil	Keterangan
1	Tes Tampilan LED	LED matrix menyala dengan baik	Modul MAX7219 berfungsi
2	Kontrol dengan Arduino	Tampilan angka sesuai dengan kode	Komunikasi Arduino-MAX7219 berhasil
3	Uji Kecerahan	Kecerahan LED dapat diatur	Fitur pengaturan kecerahan berfungsi
4	Tes Responsif Tombol	Modul merespons tombol dengan benar	Kontrol tambahan berfungsi
5	Pengujian Waktu (Terintegrasi dengan RTC)	Tampilan waktu sesuai dengan RTC	Integrasi RTC-MAX7219 berhasil
6	Pengujian Daya	Konsumsi daya dalam batas yang wajar	Modul efisien dalam penggunaan daya
7	Tes Stabilitas	Modul beroperasi stabil selama 12 jam	Stabilitas kinerja terverifikasi
8	Tes Layar Penuh	Semua LED menyala tanpa cacat	LED matrix berfungsi penuh
9	Verifikasi Konsistensi	Modul konsisten selama pengujian	Kualitas dan keandalan terpenuhi

Gambar 4.26 Tabel Pengujian MAX7219 7 Segment

### 4.7.3 Pengujian Buzzer

Tujuan menguji buzzer untuk memastikan bahwa buzzer berfungsi dengan baik dan sesuai dengan kebutuhan. Pengujian bertujuan untuk memverifikasi kualitas suara yang dihasilkan oleh buzzer, serta memastikan bahwa buzzer dapat berbunyi pada frekuensi dan intensitas yang diinginkan. Selain itu, pengujian juga dilakukan untuk memastikan buzzer responsif terhadap perintah dan dapat diandalkan dalam penggunaannya dalam jangka waktu yang lama. Gambar 4.21 menunjukkan masuknya waktu Salat dan Buzzer menyala



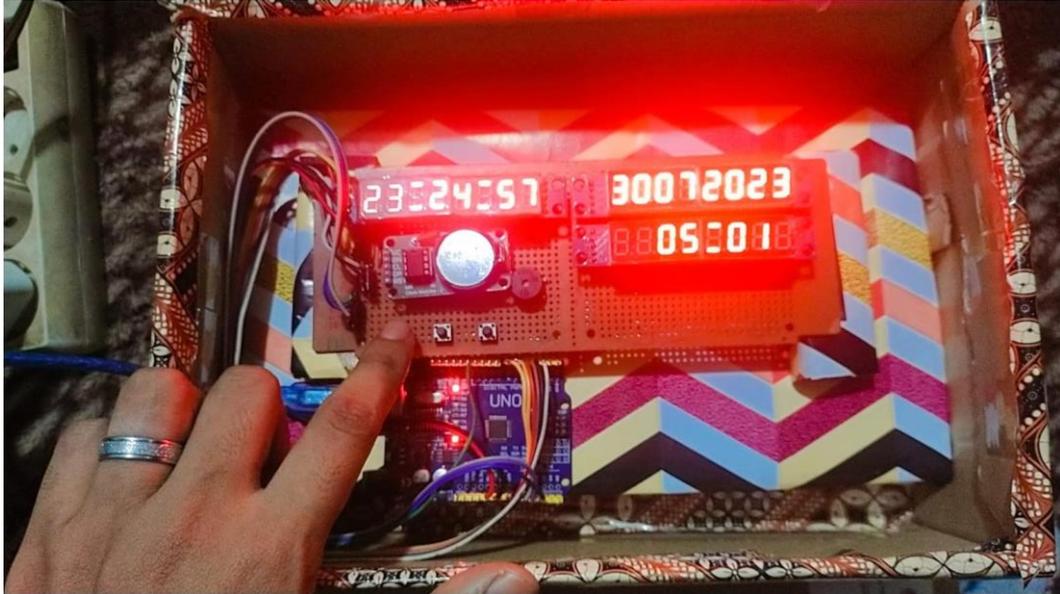
Gambar 4.27 Tampilan Masuknya Waktu Salat

No.	Pengujian	Hasil Pengujian	Keterangan
1	Tes Bunyi Buzzer	Buzzer berbunyi	Buzzer berfungsi dengan baik
2	Kualitas Suara	Suara jernih dan jelas	Kualitas suara buzzer memadai
3	Frekuensi Bunyi	Buzzer berbunyi dengan frekuensi yang diinginkan	Frekuensi bunyi sesuai dengan yang diharapkan
4	Intensitas Bunyi	Buzzer dapat diatur intensitasnya	Intensitas bunyi bisa disesuaikan
5	Pengendalian Buzzer	Buzzer berbunyi saat tombol ditekan	Buzzer dapat dikendalikan dengan baik
6	Responsifitas	Buzzer merespons perintah dengan cepat	Buzzer responsif dan tidak ada penundaan yang signifikan
7	Ketahanan	Buzzer tetap berfungsi selama pengujian yang panjang	Buzzer dapat diandalkan dalam jangka waktu lama

Gambar 4.28 Tabel pengujian Buzzer

#### 4.7.4 Pengujian Button

Tujuan pengujian button untuk memastikan bahwa tombol berfungsi dengan baik dan merespon dengan tepat ketika ditekan. Pengujian dilakukan untuk memverifikasi kualitas, performa, dan keandalan tombol dalam berbagai aspek. Dengan pengujian ini, kita dapat memastikan tombol siap digunakan dengan baik pada alat, serta mencegah masalah yang mungkin timbul di kemudian hari.



Gambar 4.29 Uji coba Button

No	Pengujian	Hasil Pengujian	keterangan
1	Fungsi Hardware	Tombol berfungsi	Tombol berfungsi dengan baik
2	Responsivitas	Tombol Merespon cepat	Tidak ada penundaan yang signifikan

Gambar 4.30 Tabel uji coba Button

#### 4.8 Jadwal Salat Rantauprapa

Gambar dibawah ini jadwal Salat pada Rantauprapat lengkap dengan bujur dan lintang nya.

Jadwal Sholat untuk Rantauprapat, GMT +7								
Agustus 2023								
← sebelum		Pilih Kota <input type="text" value="Rantauprapat"/>					sesudah →	
Tanggal	Imsyak	Shubuh	Terbit	Dhuha	Dzuhur	Ashr	Maghrib	Isya
01	04:51	05:01	06:19	06:43	12:29	15:52	18:36	19:48
02	04:52	05:02	06:19	06:43	12:29	15:52	18:35	19:48
03	04:52	05:02	06:19	06:43	12:29	15:51	18:35	19:48
04	04:52	05:02	06:19	06:43	12:29	15:51	18:35	19:48
05	04:52	05:02	06:19	06:43	12:29	15:51	18:35	19:47
06	04:52	05:02	06:19	06:43	12:29	15:50	18:35	19:47
07	04:52	05:02	06:18	06:42	12:29	15:50	18:35	19:47
08	04:52	05:02	06:18	06:42	12:29	15:50	18:35	19:46
09	04:52	05:02	06:18	06:42	12:28	15:49	18:34	19:46
10	04:52	05:02	06:18	06:42	12:28	15:49	18:34	19:46
11	04:52	05:02	06:18	06:42	12:28	15:48	18:34	19:46
12	04:52	05:02	06:18	06:42	12:28	15:48	18:34	19:45
13	04:52	05:02	06:18	06:42	12:28	15:47	18:34	19:45
14	04:52	05:02	06:18	06:42	12:28	15:47	18:33	19:45
15	04:52	05:02	06:18	06:42	12:27	15:46	18:33	19:44
16	04:52	05:02	06:18	06:42	12:27	15:46	18:33	19:44
17	04:52	05:02	06:17	06:41	12:27	15:45	18:33	19:43
18	04:52	05:02	06:17	06:41	12:27	15:45	18:32	19:43
19	04:52	05:02	06:17	06:41	12:27	15:44	18:32	19:43
20	04:52	05:02	06:17	06:41	12:26	15:43	18:32	19:42
21	04:52	05:02	06:17	06:41	12:26	15:43	18:31	19:42
22	04:52	05:02	06:17	06:41	12:26	15:42	18:31	19:42
23	04:52	05:02	06:16	06:40	12:26	15:41	18:31	19:41
24	04:52	05:02	06:16	06:40	12:25	15:41	18:31	19:41
25	04:52	05:02	06:16	06:40	12:25	15:40	18:30	19:40
26	04:52	05:02	06:16	06:40	12:25	15:39	18:30	19:40
27	04:51	05:01	06:16	06:40	12:25	15:38	18:30	19:39
28	04:51	05:01	06:15	06:39	12:24	15:38	18:29	19:39
29	04:51	05:01	06:15	06:39	12:24	15:37	18:29	19:39
30	04:51	05:01	06:15	06:39	12:24	15:36	18:28	19:38
31	04:51	05:01	06:15	06:39	12:23	15:35	18:28	19:38

**:: Parameter**

Untuk Kota **Rantauprapat** 2°15' LU 99°50' BT

Arah : 293.27 ° ke Mekah

Jarak : 6827.527 km ke Mekah

**:: Pilihan Fiqh**

Penetapan Waktu Shubuh : 20.0 deg. Kemiringan Matahari

Penetapan Waktu Ashr : Perbandingan bayangan 1 (Shaffi dan lainnya)

Penetapan Waktu Isya : 18.0 deg. Kemiringan Matahari

Penetapan Waktu Imsyak : 10.0 min. Jarak Waktu dari Shubuh

Jadwal sudah diberi : 2 menit untuk waktu ihtiyati (pengaman)

**:: Print/Cetak**

**PRINT**

copyright © 2003-2023 by cahyadsn Modified by JadwalSholat.org

Gambar 4.31 Jadwal Salat Rantauprapat