

BAB I

PENDAHULUAN

A. Latar Belakang

Masjid merupakan tempat beribadah umat Islam. Masjid mengalami kemajuan yang pesat, baik dalam bentuk bangunan maupun fungsi dan peranannya. Masjid yang telah berdiri di berbagai tempat, baik besar maupun kecil telah menjadi sarana untuk berkumpul, menuntut ilmu, menyampaikan informasi, bertukar pengalaman, pusat dakwah dan tempat beribadah.

Beribadah adalah kegiatan yang wajib di laksanakan oleh umat beragama khususnya bagi para umat muslim. Umat Islam mengartikan ibadah merupakan bentuk menyembah kepada Allah sehingga manusia diciptakan untuk beribadah adalah untuk mengabdikan seluruh aktivitas kehidupannya dalam rangka beribadah kepada Allah (Kallang, 2018). Melaksanakan ibadah sholat lima waktu setiap hari. Sholat lima waktu diantaranya waktu subuh, dzuhur, ashar, maghrib dan isya. Setiap hari jum'at seorang muslim laki-laki diwajibkan untuk melaksanakan sholat Jum'at di waktu dzuhur. Kegiatan ibadah sholat diawali dengan berkumandangnya adzan. Adzan berfungsi sebagai pertanda waktu sholat sudah tiba contohnya pada saat sholat wajib dan waktu tertentu pun dikumandangkan. Adzan adalah simbol pengingat di kehidupan muslim dan muslimah. Pengumandang adzan bisa dikenal juga sebagai seorang muadzin (Jabbar, 2015). Dalam memakmurkan masjid maka tidak lepas dari peranan pengurus masjid, apabila pengurus masjid menjalankan tugasnya dengan baik dalam memakmurkan masjid, maka masjid tersebut bisa

dikatakan makmur dan mengalami kemajuan. Apabila pengurus masjid tidak melaksanakan tugasnya dengan baik maka masjid tersebut tidak bisa dikatakan makmur dan mengalami kemajuan akan tetapi Kegiatan manusia tidak hanya melaksanakan ibadah namun hidup untuk mencari ilmu dan mencari nafkah demi meningkatkan ekonomi untuk dirinya maupun keluarga. Kegiatan selain beribadah yaitu bekerja, sekolah, bersosialisasi, dan sebagainya. Kegiatan pengurus masjid yang cukup banyak terkadang melupakan akan kewajibannya untuk mengurus masjid terutama umat muslim yang memiliki kewajiban ibadah setiap harinya.

Dengan majunya teknologi mobile saat ini, mengakibatkan manusia berusaha berfikir lebih maju dan mencari tau hal yang memudahkan untuk memenuhi kebutuhannya. Produk dengan cara kerja yang efisien sudah menjadi idaman manusia. Perkembangan teknologi saat ini tidak hanya digunakan manusia di pabrik yang memproduksi barang, namun sudah bersentuhan dengan kebutuhan sehari-hari. Salah satunya teknologi smart home, sekarang teknologi ini sudah mulai banyak ditemui dan diterapkan yang didukung dengan gabungan teknologi ataupun sistem aplikasi guna memberikan keefisienan, keamanan dan kenyamanan pada manusia (Adi & Herlina, 2019).

Saat ini moderenisasi lebih mengutamakan rasa nyaman dan kecepatan dalam system kerja otomatisasi yang bertambah banyak. Sistem otomatisasi ini tidak jarang dianggap menjadi tanpa memakai energi manusia. Pemahaman otomatis biasa disimpulkan semacam, metode serta perlengkapan yang digunakan untuk melaksanakan control maupun keadaan dikendalikan ataupun dioperasikan secara otomatis. Dalam dunia industri, system otomatis pula sangat diminati sebab

dapat memperpendek waktu produksi, menjamin kualitas produk yang dihasilkan, mengurangi biaya 2 untuk tenaga kerja manusia (Adam, dkk. 2016). Ada pun rancangan sistem otomasi perangkat audio masjid ini untuk memudahkan pengurus masjid dalam mengelola masjid berbasis *telephone*. Pada rancangan ini kami akan mengimplementasikan di masjid BABURRAHMAN Desa Compong agar pengurus masjid menggunakan teknologi canggih dalam upaya meramaikan dan memakmurkan masjid .

Perancangan sistem otomasi perangkat audio masjid ini kami menggunakan *DF mini player* dan *Module GSM SIM800L* sebagai alat yang mengeluarkan output suara, *Ardiuno Mega 2560* sebagai *mikrokontroler*, *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dapat menghidupkan dan mematikan perangkat audio, *RTC* memberikan informasi waktu sholat sehingga *mikrokontroler* dapat mengoperasikannya.

Rancang bangun ini tentang sistem otomasi perangkat audio masjid yang dapat menyampaikan kegiatan masjid secara *real time*. Dengan *Module GSM SIM800* perangkat audio dapat dikontrol dengan *SMS* dan *telephone* untuk memyampaikan kegiatan masjid secara *real time*.

Berdasarkan dari latar belakang diatas peneliti bermaksud untuk mengangkat penelitian ini dengan **“Rancang Bangun Sistem Pengaturan Otomasi Perangkat Audio Masjid Dengan SMS Dan Telepon”**.

B. Rumusan Masalah

Berdasarkan apa yang telah diuraikan diatas dapat diketahui beberapa rumusan masalah yang muncul dari latar belakang ialah sebagai berikut :

1. Bagaiman cara membuat sistem otomasi pada perangkat audio masjid?
2. Bagaimana cara menyampaikan informasi kegiatan masjid dari jarak jauh?

C. Tujuan Penelitian

Adapun Tujuan penelitian sebagai berikut:

1. Untuk Mengetahui cara menggunakan *Module GSM SIM800L* untuk *SMS* dan *telephone*.
2. Mengetahui cara membuat sistem otomasi perangkat audio masjid.

D. Batasan Masalah

Adapun beberapa batasan masalah pada penelitian ini sebagai berikut:

1. Sistem tidak menampilkan dan mengirim notifikasi waktu sholat ke pengurus masjid.
2. Sistem otomasi perangkat audio masjid tidak berfungsi tanpa sumber listrik.
3. Pengumuman kegiatan masjid di sampaikan secara *Real Time*.

E. Manfaat Penelitian

Pada penelitian ini memiliki beberapa manfaat sebagai berikut:

1. Manfaat Untuk Peneliti.
 - a. Peneliti mengetahui ilmu-ilmu yang dapat digunakan untuk merancang sebuah produk dan dapat menerapkannya.

- b. Peneliti Dapat mengetahui kinerja otomasi perangkat audio masjid menggunakan *DFPlayer mini* dan *GSM SIM800L* serta menambah pengetahuan untuk mengaplikasikan ilmu yang didapat selama perkuliahan.
2. Manfaat Untuk Masjid.
 - a. Masjid mampu menyampaikan informasi ke masyarakat dengan cepat.
 - b. Dengan adanya sistem otomasi perangkat audio masjid ini diharapkan dapat memakmurkan masjid.
 - c. Membantu pengurus masjid dalam mengelola masjid.
 3. Manfaat Untuk Institusi.
 - a. Menghasilkan alat yang dapat meningkatkan keimanan umat islam.
 - b. Dengan adanya penelitian ini diharapkan dapat menjadi referensi untuk penelitian yang serupa atau dapat dikembangkan lagi.

F. Sistematika Penelitian

Secara garis besar sistematika penulisan yang dapat disajikan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat, serta sistematika penulis.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang teori-teori dasar dan komponen-komponen yang di gunakan dalam penelitian ini.

BAB III METODE PENELITIAN

Bab ini diuraikan mengenai jenis penelitian, lokasi dan waktu penelitian, parameter pengujian, tahap-tahap metode penelitian, dan bagan alir penelitian.

BAB IV HASIL PENGUJIAN

Bab ini membahas tentang pengujian alat yang terhubung pada Mikrokontroller dan pengambilan data.

BAB V PENUTUP

Bab ini membahas tentang kesimpulan hasil dan pengujian yang telah dilakukan, serta saran bagi penulis guna untuk memperbaiki kesalahan terhadap perancangan yang telah dilakukan.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Penelitian sebelumnya bertujuan untuk mendapatkan gambaran umum dan memperkuat kerangka berfikir dalam pembahasan. Selain itu, tujuannya adalah untuk mendapatkan bahan untuk perbandingan dan referensi, maka dalam kajian pustaka ini penelitian mencantumkan beberapa hasil penelitian terdahulu yang relevan mengenai analisa dari beberapa sistem kerja alat.

1. Penelitian oleh (Andhika Wahyu Putra, Rizki Nuryanto & Asni Tafrikhatin, 2020) yang berjudul “Fitur Pengingat Kegiatan Masjid Dengan Kontrol WI-FI Berbasis ESP-32 Pada Jam Digital” Pada Penelitian ini teknologi jam digital tidak hanya menampilkan waktu, jam digital dapat memberikan informasi berupa output suara. Proses konfigurasi juga sangat sederhana dengan mengkoneksikan mikrokontroler WI-FI berbasisi ESP-32 pada jam digital ke smartphone lalu browser otomatis akan memunculkan tampilan pengaturan jam pada smartphone kita. Setelah itu mikrokontroler akan menjalankan perintah dengan menampilkan waktu pada LED dan memberikan alarm peringatan waktu dengan output suara pada speaker yang sudah di atur pada smartphone. Penggunaan aplikasi arduino juga berperan penting dalam mengatur coding pada mikrokontroler guna mengatur output suara dan tampilan LED pada alat. Jarak maksimal antara gawai dengan jam digital

tanpa halangan sepanjang 20 meter, sedangkan dengan halangan sepanjang 6 meter.

2. Penelitian oleh (Muhammad Anton, Sulistiyanto, Muhammad Hasan Basri & Yuli Prasetyo, 2020) yang berjudul “Perancangan Jam Istiwa Otomatis Menggunakan *Running Text* Dan *Speaker* Sebagai Alat Bantu Waktu Sholat Di Masjid Nurul Hidayah Al-Takwa" Pada Penelitian ini Jam adalah petunjuk waktu yang menjadi acuan waktu sholat 5 waktu, yaitu subuh, dzuhur, ashar, magrib, dan isya. Itulah mengapa waktu sholat itu sangat penting adanya, maka sekarang sudah banyak masjid dan rumah yang terdapat kalender dan penampil waktu sholat. Maka dari itu diperlukan cara bagaimana merancang sistem control penanda datangnya waktu sholat secara otomatis untuk lebih mempermudah masjid dalam membantu masyarakat melaksanakan ibadah sholat berjamaah. Berdasarkan apa yang telah dibahas diatas dapat diketahui bahwa running text dapat membaca pada percobaan hari ke dua pada jam 04.14 WIB waktu sholat shubuh dan speaker dapat mengeluarkan suara azan. Pada percobaan hari ke ketiga running dapat membaca pada jam 04.14 WIB dan 14.50 WIB maka secara otomatis speaker mengeluarkan suara azan. Pada percobaan hari ke empat running text dapat membaca pada jam 04.14 WIB, 11.29 WIB, dan 14.50 WIB maka secara otomatis speaker mengeluarkan suara azan. Pada uji coba hari ke lima running text dapat membaca pada jam 04.14 WIB, 11.29 WIB, 14.50 WIB, dan 18.37 WIB dan secara otomatis speaker mengeluarkan suara azan. Pada percobaan hari enam hari enam *running text* dapat membaca waktu sholat secara sempurna dan mengeluarkan

suara azan secara otomatis. Maka dari semua percobaan dapat diketahui *running text* dapat membaca secara sempurna pada percobaan hari ke enam dan speaker dapat mengeluarkan suara azan secara otomatis pada waktu tertentu.

B. Kajian Teori

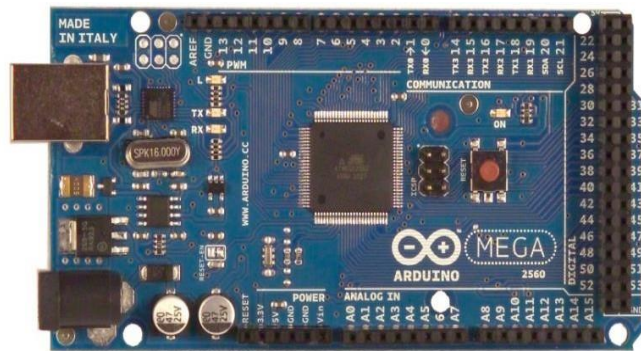
1. Sound System

Sound system adalah sebuah istilah yang digunakan untuk menyebut sekumpulan alat elektronik yang memiliki fungsi untuk memproduksi suara, baik dengan cara *live stage* maupun *recording*. *sound system* bukan hanya berupa satu alat *compact*, melainkan didukung dengan *device* lain yang membuatnya semakin komplit diantaranya adalah *microphone*, *mixer*, *equalizer*, *power amplifier*, *speaker system*, dan aksesoris lainnya. Setiap perangkat sound itu harus bisa bekerja dengan baik.

2. Arduino Mega 2560

Arduino Mega 2560 adalah papan pengembangan *mikrokontroler* yang berbasis *Arduino* dengan menggunakan *chip ATmega2560*. *Board* ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah *digital I/O pin* (15 pin diantaranya adalah *PWM*), 16 pin *analog input*, 4 pin *UART (serial port hardware)*. *Arduino Mega 2560* dilengkapi dengan sebuah *oscillator 16 Mhz*, sebuah *port USB*, *power jack DC*, *ICSP header*, dan tombol reset.

Skema *Arduino Mega 2560* tampak dari atas dapat dilihat pada Gambar di bawah dengan karakteristik sebagai berikut



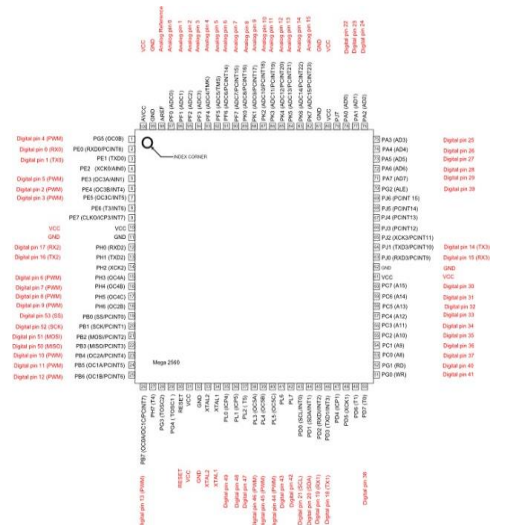
Gambar 2. 1 Ardiuno Mega

(Sumber : *ArduinoMega2560Datasheet*)

- a. Mikrokontroler : ATmega2560
- b. Tegangan Operasional : 5V
- c. Tegangan Masukan (direkomendasi) : 7-12V
- d. Tegangan Masukan (batas) : 6-20V
- e. *Pin Digital I/O* : 54 (14 pin untuk keluaran *PWM*)
- f. *Analog Input Pins* : 16
- g. Arus DC per I/O Pin : 40 Ma
- h. Arus DC for 3.3V Pin : 50 mA
- i. *Memori Flash* : 256 KB (8 KB digunakan untuk *bootloader*)
- j. *SRAM* : 8 KB
- k. *EEPROM* : 4 KB
- l. *Clock Speed* : 16 MHz

Mikrokontroler yang digunakan pada Arduino Mega 2560 ini adalah Mikrokontroler ATmega 2560. Mikrokontroler ini menjadi komponen utama dari sistem minimum Arduino Mega 2560. Setiap pin mikrokontroler ATmega 2560 dipetakan sesuai dengan kebutuhan standar Arduino pada umumnya.

Pemetaan pin (*pin mapping*) ATmega 2560 dapat dilihat pada Gambar 2.2. dibawah berikut :



Gambar 2. 2 Arduino Mega 2560

(Sumber : *ArduinoMega2560Datasheet*)

Tabel 2. 1 Deskripsi masing-masing pin ATMEGA 2560

PinNumber	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)

PinNumber	Pin Name	Mapped Pin Name
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)

PinNumber	Pin Name	Mapped Pin Name
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

3. Module GSM SIM800L

Modul GSM SIM800L adalah perangkat yang bisa digunakan untuk menggantikan fungsi handphone. Untuk komunikasi data antara sistem jaringan

seluler, maka digunakan *Modul GSM SIM800L* yang digunakan sebagai media panggilan *telephone cellular*. Protokol komunikasi yang digunakan adalah komunikasi standart modem yaitu *AT Command*. Adapun beberapa fitur *Modul GSM SIM800L* antara lain:

- a. Antarmuka: UART
- b. Support AT command
- c. Suara :Tricodec, AMR, Hand
- d. free operation
- e. SMS: SMS Broadcast, mode teks dan mode Protocol Data Unit (PDU)
- f. Catu Daya: 3.2~4.8 V
- g. Fitur tambahan: Analog Audio, Antena pad
- h. Konsumsi daya: 1.0 mA (pada sleepmode)

Modul GSM SIM800L di Indonesia banyak digunakan pada industri bisnis rumahan dan bahkan skala besar, mulai dari fungsi untuk *controller* berbasis SMS, WEB, Call sistem hingga sebagai penggerak perangkat elektronik jarak jauh. Beberapa kegunaan modem ini di masyarakat adalah antara lain:

- a. *Telemetry*
- b. *M2M integration*
- c. *SMS polling*
- d. SMS quiz application
- e. *SMS auto-reply*
- f. Aplikasi server pulsa

- g. *Payment point data*
- h. *SMS broadcast application*
- i. *PPOB*, dan sebagainya

sudah diproduksi dengan bermacam bentuk dan tipe modul adapter, seperti untuk arduino, neo, dan modul trainer kit.



Gambar 2. 3 Module GSM SIM800L

Sumber: GSM SIM800L Datasheet(www.dev.faranux.com)

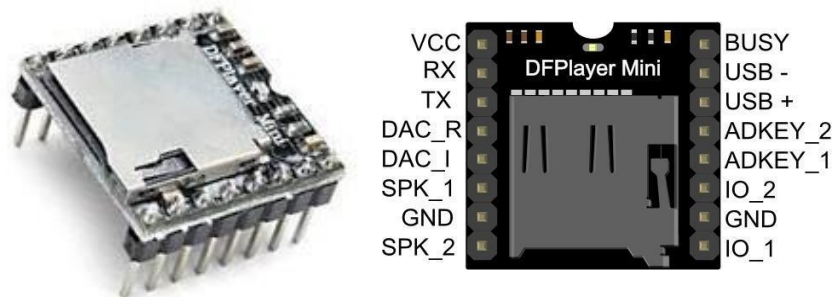
Tabel 2. 2 Fungsi Setiap Pin Modul GSM SIM800L

Pin Name	Mapped PiN Name
Ring	Ring Indikator
DTR	Data Terminal Ready
MIC+	Diferensial Input Audio MIC Positif
MIC -	Diferensial Input Audio MIC Negatif
SPEAKER +	Diferensial Input Audio Speaker Positif
SPEAKER -	Diferensial Input Audio Speaker Negatif
NET (Antena)	Pin Antena Modul GSM
VCC	Input Tegangan 3.4 V – 4.4 V
Reset	Pin Reset Modul GSM
RX	Menerima data serial
TX	Mengirim data serial
GND	Sistem Ground

4. DFPlayer Mini

DFPlayer Mini merupakan module pemutar *file audio / module sound player music* dengan *support* format audio seperti *file.mp3* yang sudah umum dikenal oleh khalayak umum. Bentuk fisik dari *DFPlayer mini* ini berbentuk persegi dengan ukuran 20 x 20 mm yang dimana memiliki 16 kaki pin. *Output* pada *module mp3* mini ini dapat langsung dihubungkan dengan speaker mini ataupun amplifier sebagai penguat suaranya.

DFPlayer mini dapat dioperasikan secara *standalone* (berdiri sendiri) ataupun dioperasikan menggunakan *microcontroller* misalnya *Arduino* melalui komunikasi serial.



Gambar 2. 4 DFPlayer Mini

Sumber: DFPlayer Datasheet(www.dev.faranux.com)

Tabel 2. 3 Spesifikasi DFPlayer Mini

Nama	Deskripsi	Catatan
VCC	<i>Input</i> Tegangan	DC 3,2-5.0V
RX	UART <i>input</i> serial	
TX	UART <i>output</i> serial	
DAC_R	<i>Output</i> audio saluran kanan	<i>Earphone drive</i> dan amplifier
DAC_L	<i>Output</i> audio saluran kiri	
SPK2	Speaker	Speaker <i>power</i> (<3W)
GND	<i>Ground</i>	<i>Power ground</i>
SPK1	Speaker	Speaker <i>power</i> (<3w)

Nama	Deskripsi	Catatan
I0 1	<i>Trigger port 1</i>	Tekan sebentar untuk memainkan laguberikutnya (tahan lama untuk mengurangivolume)
GND	<i>Ground</i>	<i>Power ground</i>
I02	<i>Trigger port 2</i>	Tekan lama untuk memainkan laguberikutnya (tahan lamauntuk meningkatkanvolume)
ADKEY1	<i>AD port 1</i>	Memicu memainkan segmen pertama
ADKEY2	<i>AD port 2</i>	Memicu memainkan segmen kelima
UBS +	USB + DP	<i>Port USB</i>
USB -	USB - DM	<i>Port USB</i>
<i>Busy</i>	Memainkan status	Rendah Memainkan musik
		Tinggi tidak memainkan musik

5. *Relay*

Relay adalah komponen elektronika yang berupa saklar atau switch elektrik yang dioperasikan menggunakan listrik. Relay juga biasa disebut sebagai komponen electromechanical atau elektromekanikal yang terdiri dari dua bagian utama yaitu coil atau elektromagnet dan kontak saklar atau mekanikal. Komponen relay menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang kecil atau low power, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi. Berikut adalah gambar relay.



Gambar 2. 5 Relay

Sumber: Relay Datasheet(www.dev.faranux.com)

Tabel 2. 4 Spesifikasi Relay

Nama	Deskripsi
Tegangan kerja	5V DC
Jumlah relay	1
Sinyal kontrol	<i>Active LOW</i>
Arus maksimal AC	AC250V 10A
Arus maksimal DC	DC30V 10A
Indikator LED setiap pin	1
Pin data	2

6. Kabel Jumper

Jumper pada sebuah komputer sebenarnya adalah *connector* penghubung sirkuit elektrik yang digunakan untuk menghubungkan atau memutus hubungan pada suatu sirkuit. Jumper juga digunakan untuk melakukan setting pada papan *Motherboard* elektrik seperti *motherboard* komputer. Kabel jumper adalah kabel yang lazimnya di gunakan sebagai penghubung antara Ardiuno dengan board atau Ardiuno dengan sensor yang akan digunakan. Kabel jumper menghantarkan listrik atau sinyal. Kabel jumper menghantarkan listrik atau sinyal melalui logam di dalamnya yang bersifat konduktor. Ada tiga jenis kabel jumper yang dapat dilihat dari ujungnya, yaitu:

- a. *Male-Male*
- b. *Male-Female*
- c. *Female-Female*



Gambar 2. 6 kabel Jumper

Sumber: Kabel Jumper Datasheet(www.dev.faranux.com)

7. Real Time Clock(RTC)

RTC adalah chip yang berdaya sangat rendah. Chip tersebut memiliki kode biner, jam atau kalender, 56 byte NVSRAM dan komunikasi antarmuka menggunakan rangkaian dua kabel. RTC berfungsi sebagai penyedia data dalam hitungan detik, menit, jam, hari, tanggal, bulan, dan informasi yang telah diprogram. RTC memiliki antarmuka serial dua kawat dan memiliki jalur data yang paralel, sinyal keluaran gelombang persegi, konsumsi daya lebih rendah 500nA dan menggunakan baterai cadangan osilator, tahan suhu -40 derajat C hingga 85 derajat celcius, tersedia dalam bentuk 8 pin DIP atau SOIC.



Gambar 2. 7 *Real Time Clock*

Sumber: RTC Datasheet(www.dev.faranux.com)

Tabel 2. 5 Spesifikasi RTC (Real Time Clock)

Nama	Deskripsi
Tegangan	3.3 – 5.5 Volt
Eror penghitungan waktu	± 1 menit
Clock chip	DS3231
Dimensi	$38 \times 22 \times 14$ mm
Memory chip	AT24C32

8. *Adaptor*

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan Catu daya Eksternal. Sumber listrik dipilih secara otomatis. Eksternal (*nonUSB*) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya *plug* pusat-positif 2.1 mm ke dalam *board* penghubung listrik. *Lead* dari baterai dapat dimasukkan ke dalam header pin GND dan Vin dari konektor Power.

Bord dapat beroperasi pada pasokan daya dari 6-20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun pin 5V dapat menyuplai kurang dari 5 Volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12 Volt, regulator tegangan bias panas dan merusak board. Rentang yang dianjurkan adalah 7-12 Volt.



Gambar 2. 8 Adaptor

(Sumber : tecnoarnel.com.ve)

9. Kapasitor Elektrolit

Pada tahun 1896, insinyur Amerika Serikat, Charles Pollak, menemukan elco dengan elektrolit berbasis padat, yaitu amonium borat. Jenis elco ini menunjukkan daya tahan dan stabilitas yang lebih baik dibandingkan dengan elco yang menggunakan elektrolit cair.

Pada perangkat ini, terdapat kapasitor elektrolit (Elco) yang dipasang paralel antara pin VCC dan GND pada modul SIM800L dengan memperhatikan polaritasnya. Saya menggunakan kapasitor Elco berkapasitas 100 μ F. Langkah ini diambil untuk memastikan bahwa modul menerima pasokan daya yang cukup saat melakukan transmisi yang bersifat intens dan cepat, terutama dalam waktu singkat.



Gambar. 2.9 kapasitor Elektrolit

(Sumber : duniaelektronika.com)

10. Resistor

Hambatan menjadi elemen yang sangat penting dalam studi rangkaian listrik karena setiap rangkaian memerlukan indikator untuk mengukur seberapa mudah arus dapat mengalir melaluinya. Unit pengukuran hambatan adalah Ohm (Ω). Selain itu, hambatan juga diperlukan untuk memastikan bahwa arus listrik yang mengalir dalam suatu rangkaian berada pada tingkat yang sesuai dengan sirkuit yang digunakan. Hambatan dapat muncul dalam suatu rangkaian melalui penggunaan resistor sebagai komponen penghambat atau pemberi hambatan. Semakin besar nilai hambatan atau resistansi dalam suatu rangkaian listrik, semakin sulit bagi arus listrik untuk mengalir melalui rangkaian tersebut. Sebaliknya, semakin kecil nilai hambatan, semakin mudah arus listrik dapat



mengalir, dan semakin besar arus yang mengalir dalam rangkaian tersebut.

Gambar 2. 10 Resistor

(Sumber : elektonika.com)

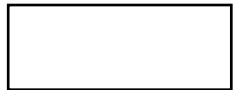
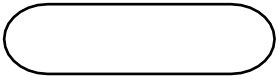
11. Flow Chart

Diagram alur (*flow chart*) adalah sebuah sistem diagram yang menggunakan serangkain bentuk simbol yang menggambarkan bagaimana data mengalir melalui suatu proses yang saling terhubung.

Flow chart adalah diagram yang menunjukkan aliran suatu proses

menggunakan simbol-simbol tingkat geometri, seperti lingkaran, persegi empat, wajik, oval, dan sebagainya untuk mempresentasikan langkah-langkah kegiatan beserta urutannya dengan menghubungkan setiap langkah tersebut menggunakan tanda panah.

Tabel 2. 6 Simbol Flow Chart

No	Simbol	Nama	Keterangan
1		<i>Input/Output</i>	Simbol input/output digunakan untuk mewakili data input/output
2		<i>Process</i>	Simbol proses, digunakan untuk mewakili suatu proses
3		<i>Predefine process</i>	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang sebelumnya ditunjukkan ditempat lain
4		<i>Terminal</i>	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses

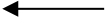
12. Unified Modelling Language (UML)

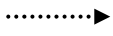
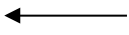
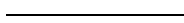
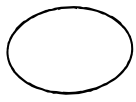
Unified Modelling Language (UML) menyediakan diagram- diagram yang sangat kaya dan dapat diperluas sesuai kebutuhan kita. Diagram adalah representasi secara grafis dari elemen-elemen tertentu beserta hubungan-hubungannya. Diagram penting karena menyediakan representasi secara grafis dari sistem atau bagiannya. Menurut Rosa Dan Shalahuddin (2017:133) “UML

(*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek.

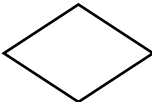
Representasi grafis sangat mempermudah pemahaman terhadap sistem. Dengan menggunakan *Unified Modelling Language (UML)* dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun. *Unified Modelling Language (UML)* memiliki beberapa diagram grafis yang diberi nama berdasarkan sudut pandang yang berbeda terhadap sistem dalam proses analisa atau rekayasa *Unified Modelling Language (UML)* yaitu sebagai berikut.

Tabel 2. 7 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang penggunaannya memainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadipada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).

No	Gambar	Nama	Keterangan
4.		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 8 Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		<i>Initial Node</i>	Bahaimana objek dibentuk atau diawali.
4.		<i>Final Node</i>	Bagaimana objek di bentuk dan di hancurkan
5.		<i>Fork Node</i>	Saluran aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		<i>Decision</i>	Memungkinkan suatu <i>usecase</i> menggunakan fungsionalitas yang dilakukan oleh <i>usecase</i> .
7.		<i>Interaction</i>	Alur

13. Arduino IDE

Arduino IDE merupakan pemrograman yang digunakan untuk menyusun program kedalam Arduino. Tugas Arduino IDE sendiri menghasilkan sebuah *file* berformat HEX dari baris kode yang dinamakan *sketch (listing program)* yang akan di *download* pada papan Arduino atau papan sistem mikrokontroler lainnya.

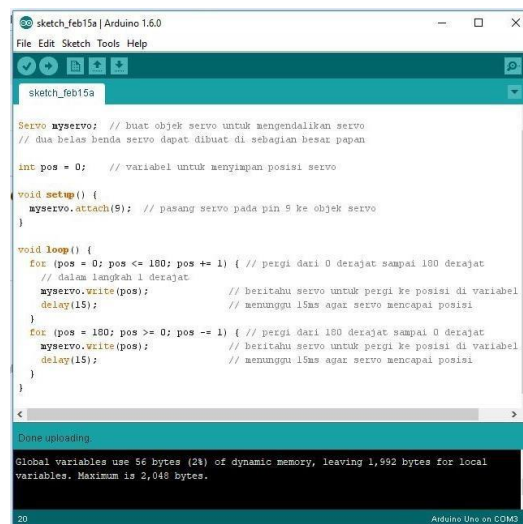
Pemrograman Arduino IDE terdiri dari 3 bagian:

- a. *Editor program*, untuk menyusun dan mengubah program dalam

Bahasa *processing*.

- b. *Compiler*, modul yang mampu mengubah kode program kedalam kode biner yang harus dipahami oleh mikrokontroler.
- c. *Uploader*, modul yang digunakan untuk memasukkan kode biner kedalam memori mikrokontroler.

Struktur perintah pada Arduino pada umumnya terdiri dari dua bagian khususnya void setup dan void loop (Menengah Mahir, 2011). Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan sementara itu void loop berisi perintah yang akan dieksekusi berulang kali selama Arduino dihidupkan.



Gambar 2. 10Ardiuno IDE

14. AT Command

AT Command adalah sejenis perintah kontrol yang digunakan untuk berkomunikasi dengan perangkat yang memiliki kemampuan untuk menerima dan memahami instruksi AT (Attention) Command. Perintah AT digunakan untuk

mengontrol dan berinteraksi dengan berbagai jenis perangkat, terutama perangkat komunikasi seperti modem, ponsel, dan modul GSM/GPRS.

MohammadRidwan(2017) memaparkan bahwa *AT Command* adalah salah satu fitur windows yang menampung perintah-perintah yang digunakan dalam komunikasi dengan serial port. Dengan *AT Command* kita dapat mengetahui vendor dari *Handphone* yang digunakan, kekuatan sinyal, membaca pesan yang ada pada *SIM Card*, mengirim pesan, mendeteksi pesan SMS baru yang masuk secara otomatis, menghapus pesan pada *SIM Card* Perintah *AT* umumnya diawali dengan karakter "AT" dan diikuti oleh sejumlah karakter tambahan yang menentukan tindakan atau konfigurasi yang diinginkan.

15. Library

Library dalam pemrograman adalah kumpulan modul atau fungsi yang telah dikodekan sebelumnya untuk digunakan kembali. Fungsinya adalah menyediakan fungsi atau rutin tertentu yang dapat diakses oleh program lain, memungkinkan pengembang menghemat waktu dan usaha tanpa perlu menulis ulang kode. Ada dua jenis library: bawaan (built-in) yang sudah disertakan dalam instalasi bahasa atau platform, dan eksternal yang dibuat oleh pihak ketiga untuk diunduh atau diintegrasikan, dalam proyek ini peneliti menggunakan library waktu sholat buatan Asmaklad. Library membantu meningkatkan produktivitas dan efisiensi dalam pengembangan perangkat lunak.

16. Bahasa C

Jamila(2018)memaparkan bahwa Pencipta bahasa C adalah Brian W.

Kernighan dan Denis M. Ritchi, sekitar tahun 1972. Penulisan program dalam bahasa C dilakukan dengan membagi dalam blok-blok, sehingga bahasa disebut dengan bahasa terstruktur. Bahasa C dapat digunakan di berbagai mesin dengan mudah, mulai dari PC sampai dengan mainframe, dengan berbagai sistem operasi misalnya DOS, UNIX, VMS dan lain-lain.

Menurut Andri Kristanto (2013) salah satu konsep yang penting dalam bahasa pemrograman C++ adalah struktur. Struktur merupakan suatu bentuk susunan dari suatu program yang dibuat. Di bawah ini merupakan contoh pemrograman sederhana sebagai berikut:

```
/* Contoh program sederhana dengan C++ */
#include <stdio.h>
    main()

    {

        Printf("Belajar Bahasa C++");

    }
```

Program C merupakan program yang berbentuk fungsi-fungsi. Sebuah program minimal mengandung sebuah fungsi. Main() merupakan nama dari suatu fungsi yang harus ada pada program dan diletakkan di bagian tertentu yang menunjukkan pada compiler dimana awal dari suatu program. Di bawah ini merupakan struktur dari suatu program.

#include<> adalah sebuah pengarah praprosesor yang digunakan untuk membaca file yang dinamakan file judul (header file), yaitu file yang berisi deklarasi dan konstanta. Tanda brace “{“ adalah tanda pembuka atau awal dari

sebuah blok fungsi atau perintah yang akan dibuat. Sedangkan tanda brance “}” merupakan tanda akhir/penutup dari suatu fungsi blok.

Dalam sebuah program bisa terdiri lebih dari satu tubuh fungsi. Untuk contoh diatas hanya terdiri dari satu buah tubuh fungsi, satu tubuh fungsi bisa diisi dengan beberapa fungsi, sedangkan suatu fungsi dapat dibuat dari satu atau lebih *statement* atau *library function* (fungsi pustaka) yang sudah tersedia. Pada contoh diatas, perintah *printf* merupakan fungsi pustaka yang dipergunakan untuk menampilkan hasil.

Pengenalan Fungsi Dasar

- a. Pengenalan Praprosessor `#include`
- b. Fungsi `main()`
- c. Fungsi Input
 - 1) Fungsi `scanf()`.
 - 2) Fungsi `gets()`
 - 3) Fungsi `getchar()`
- d. Fungsi Output
 - 1) Fungsi `printf()`
 - 2) Fungsi `puts()`
 - 3) Fungsi `Putchar()`
- e. Komentar Dalam Program

Dalam bukunya Andri Kristanto (2013) menjelaskan Tipe data adalah suatu nilai yang dapat dinyatakan dalam bentuk konstanta atau variabel dan operator yang dapat dikenakan padanya. Konstanta menyatakan nilai yang tepat,

sedangkan variabel menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung.

17. SMS (*Short Message Service*)

SMS menyiratkan bantuan layanan yang sangat luas diterapkan pada sistem komunikasi tanpa kabel, pada dasarnya *SMS* berfungsi untuk memberikan administrasi pengiriman pesan instan singkat dimulai dengan satu ponsel kemudian ke berikutnya.

Alat *SMS* yang berfungsi cocok untuk mengirim atau mendapatkan informasi antara jaringan operator seluler secara terus-menerus. *SMS* dilengkapi untuk mengirim atau menerima dari operator di seluruh dunia tanpa mengetahui batasan lokal. Pada awalnya *SMS* dibuat sebagai fitur komunikasi *GSM*, namun saat ini juga digunakan pada organisasi serbaguna seperti *UMTS (Universal Mobile Telecommunication System)*.

Kerangka *GSM* adalah sebuah inovasi komunikasi seluler yang tingkat digital. Inovasi *GSM* banyak diterapkan dalam berbagai komunikasi *mobile*, khususnya telepon seluler.

Inovasi ini menggunakan gelombang *mikro* dan transmisi sinyal dipisahkan oleh waktu, sehingga pengulangan sinyal data yang dikirim akan muncul di tujuan. *GSM* digunakan sebagai standar umum untuk komunikasi seluler serta sebagai inovasi seluler yang paling banyak digunakan individu di seluruh dunia .

GSM mempunyai keunggulan antara lain yaitu:

- a. Inovasi *GSM* memanfaatkan frekuensi radio 900 Mhz ke 1800 MHz.
- b. Sifat komunikasi antara pengguna lebih unggul menggunakan analog sistem.
- c. Kerangka kerja *GSM* mendukung transmisi informasi dengan tingkat akses sebesar 9 sampai 14,4 kbps.

18. Metode Pengujian Program

Pengujian perangkat lunak merupakan proses eksekusi program atau perangkat lunak dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi atribut dan kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau *output* yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Ada berbagai macam metode pengujian, teknik *black box* dan teknik *white box* merupakan metode pengujian yang telah dikenal dan banyak digunakan oleh pengembang perangkat lunak.

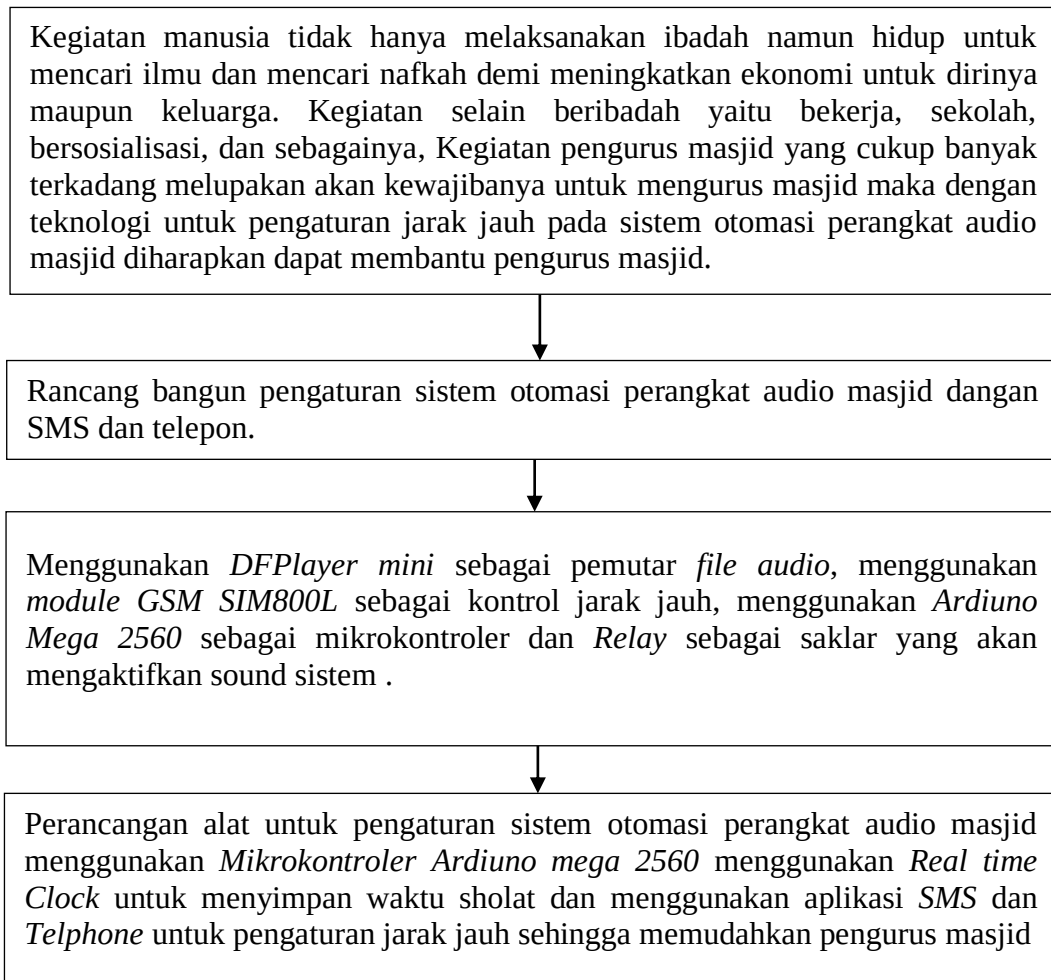
a. Metode Pengujian *Black Box*

Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut. Struktur program dan kode-kode yang ada di dalamnya tidak termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah murah dan sederhana. Namun, pengujian dengan metode ini tidak dapat mendeteksi kekurangan efektivitas pengkodean dalam suatu program.

b. Metode Pengujian *White Box*

Metode pengujian *white box* atau dapat disebut juga *glass box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah perangkat lunak atau program sebagai kotak kaca (*glass box*). Pendekatan ini akan mengevaluasi stuktur program dan kodenya yang meliputi efektivitas pengkodean, pernyataan kondisional (alur program), dan looping yang digunakan dalam program. Keuntungan dari metode pengujian ini antara lain dapat ditemukannya kode-kode tersembunyi yang menghasilkan kesalahan serta dapat menghasilkan program yang efektif.

C. KERANGKA PIKIR



BAB III

METODE PENELITIAN

A. Jenis Penelitian

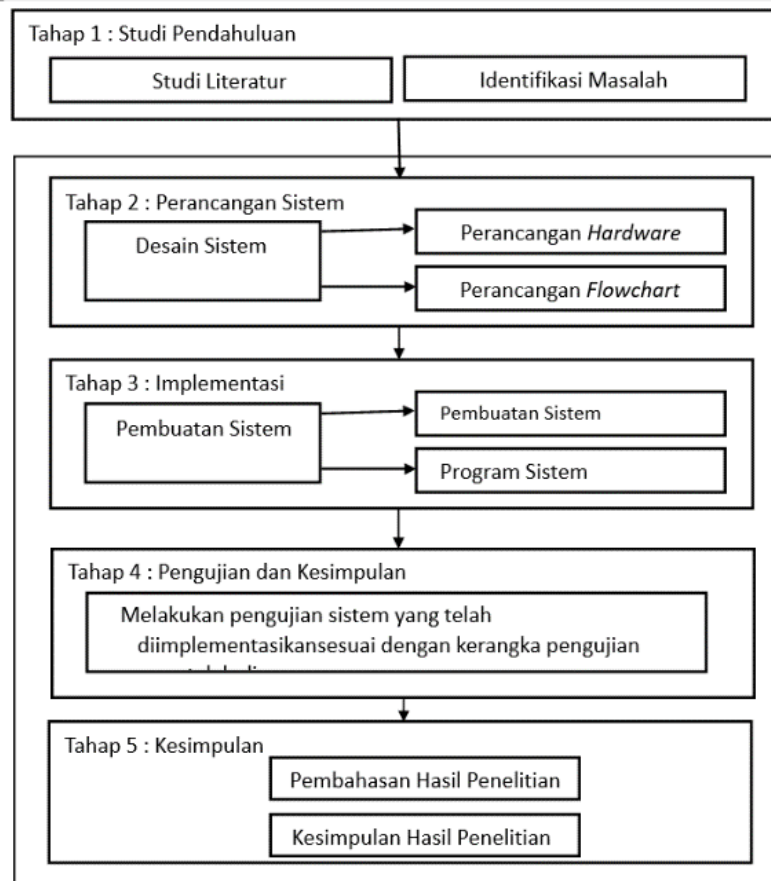
Jenis Penelitian yang digunakan dalam penelitian ini adalah penelitian kualitatif. Penelitian kualitatif merupakan penelitian rancang bangun aplikasi berbasis android untuk pengaturan jarak jauh pada sistem otomasi perangkat audio masjid menggunakan *Module GSM SIM800L*, metode kualitatif merupakan metode yang fokus pada pengamatan yang mendalam. Oleh karenanya, penggunaan metode kualitatif dalam penelitian dapat menghasilkan kajian atas suatu fenomena yang lebih komprehensif.

B. Tahapan Penelitian

Tahapan penelitian merupakan wadah pemikiran berupa kerangka alur penelitian yang dirancang secara terstruktur sesuai dengan rencana tahapan yang akan dilakukan selama penelitian berlangsung. Dalam pelaksanaannya penelitian pasti ada tahap dimana peneliti akan melakukan sedikit pemodelan (variasi) tahapan terutama pada penelitian kualitatif yang biasanya digunakan pada penelitian sosial yang cenderung abstrak. Meskipun begitu peneliti harus tetap kepada prinsip umum pada tahapan penelitian.

Tahapan penelitian adalah level atau tingkatan bisa disebut juga jenjang dalam sebuah aktivitas penelitian. Dimana tahapan tersebut terdapat memiliki proses yang dilakukan secara terstruktur, runtut, baku, logis dan sistematis.

Tahapan penelitian ini bisa dilihat dalam bentuk bagan seperti gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

Keterangan :

1. Tahapan awal dari penentuan masalah penelitian pada penelitian ini adalah studi pendahuluan dengan cara melakukan studi literatur dari berbagai sumber yang membahas mengenai masalah yang berhubungan dengan sistem otomatis perangkat audio masjid, sehingga pada tahap pertama ini penulis dapat mengidentifikasi masalah yang akan diteliti.
2. Pada tahapan kedua, hasil dari proses identifikasi masalah dan studi literatur tersebut selanjutnya menentukan rancangan sistem otomatis yang

akan digunakan pada masjid berdasarkan *flowchart* sistem dan berupa perancangan *hardware*.

3. Pada tahap implementasi, desain rancangan sistem dan rancangan hardware yang telah dibuat lalu dilakukan pembuatan sistem otomatis dan diprogram ke mikrokontroler.
4. Pada proses pengujian dilakukan dengan memastikan setiap komponen modul bekerja dengan seharusnya dan diuji juga tingkat keakurasian modul *RTC* agar dapat mengetahui waktu sholat dan *GSM SIM800L* sebagai alat yang akan menyampaikan pengumuman secara *Real Time*, serta menyimpulkan penelitian mengenai pembuatan sistem otomatis pada masjid.

C. Waktu Penelitian

Waktu yang dipergunakan untuk pelaksanaan penelitian ini selama ± 2 bulan.

D. Metode Pengumpulan data

Untuk mendapatkan informasi yang diperlukan sebagai dasar penelitian maka penulis mengumpulkan data dari lapangan 3 (Tiga) metode, yaitu :

1. Observasi adalah pengamatan langsung terhadap apa yang terjadi dilapangan sehingga penulis dapat mengkonfirmasi data yang ada.
2. Dokumentasi adalah teknik pengumpulan data dengan mengumpulkan berbagai dokumen yang ada.
3. Studi Literatur, dilakukan untuk memahami otomatisasi menggunakan *Ardiuno Mega* dan *Module GSM SIM800L*. Sumber informasi yang

digunakan untuk membuat kaidah/aturan adalah dengan menggunakan pengetahuan dari buku, jurnal dan artikel.

E. Alat dan Bahan Penelitian

Adapun alat dan bahan yang digunakan terdapat pada tabel :

Tabel 3. 1 Bahan

No	Nama Alat	Volume
1	Ardiuno Mega	1 Buah
2	DF Player Mini	1 Buah
3	Module GSM SIM800l	1 Buah
4	Relay	2 Buah
5	RTC	1 Buah
6	Kabel Power	1 Buah
7	Kabel Jumper	20 Pasang
9	Lem Tembak	1 Set
11	Timah	1 Roll
12	Jack Audio	1 Buah
13	Stuk Jack Audio	1 Buah
14	Kapasitor eletrolitik	1 Buah
15	Resistor	3 Buah

Tabel 3. 2 Alat

No.	Nama Alat	Volume
1	Solder	1 Unit
2	Laptop	1 Unit (MSI, OS Windows 10 Pro, Intel(R) Coleron(R) CPU 1019Y @ 1.00GHz, RAM 2,00 GB, SSD 120GB)

Tabel 3. 3 Perangkat Lunak

No.	Perangkat Lunak
1.	Arduino IDE
2.	Fritzing

F. Teknik Pengujian

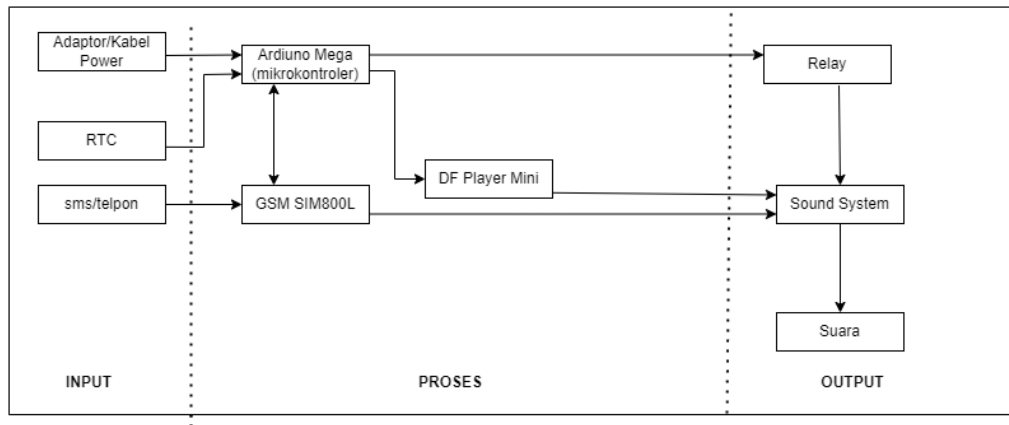
Pada metode pengujian pada penelitian ini yang digunakan adalah metode pengujian secara langsung dengan menggunakan alat sistem. Diterapkan untuk menguji fungsi-fungsi dari *software* yang dirancang. Keberhasilan *software* yang diuji hanya bisa dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi *input* yang diberikan untuk fungsi yang ada, tanpa melihat bagaimana proses mendapatkan data tersebut. Dari hasil *output* yang dihasilkan, kemampuan program dalam memenuhi kebutuhan sistem dapat diukur dan juga dapat diketahui kesalahannya.

Metode eksperimen (uji coba) dilakukan dengan perancangan rangkaian alat dan perangkat lunak agar menciptakan alat sesuai tujuan awal. Dengan melakukan eksperimen ke perancangan dan pembuatan alat ini, diharapkan akan mendapatkan rangkaian serta program yang sesuai dengan fungsi dan tujuan dari pembuatan sistem ini.

G. Desain Sistem

1. Blok Diagram

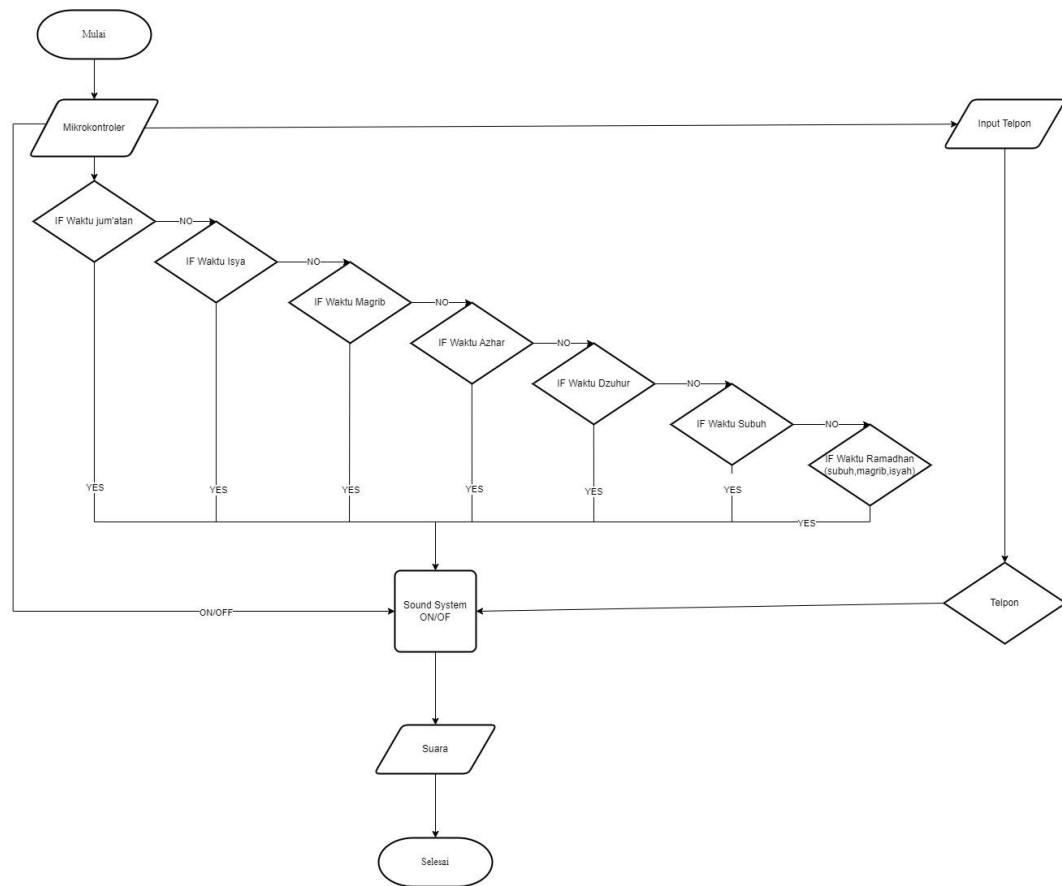
Blok diagram adalah salah satu pernyataan gambar yang diringkas dari gabungan antara masukan, keluaran dan proses dari suatu sistem. Blok diagram perancangan komponen sistem hardware sistem keamanan dapat dilihat pada gambar dibawah ini:



Gambar 3. 2 Blok Diagram

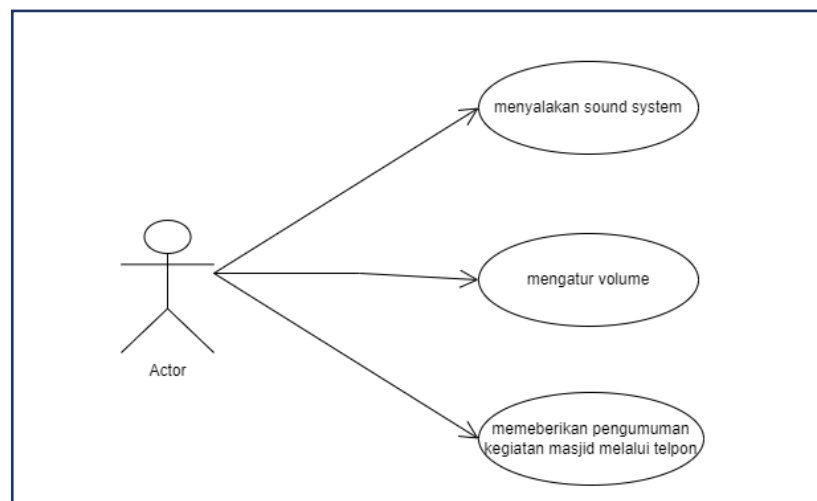
- Input* : terdiri dari Adaptor, SMS dan Telepon yang terhubung ke *Module GSM SIM800L* yang akan di proses mikrokontroler, *Real Time Clock* perangkat ini termasuk alat *input*.
- Proses* : kami menggunakan Mikrokontroler Ardiuno Mega sebagai alat proses yang memberikan perintah ke *Relay* sehingga *sound system* aktif dengan begitu *DFPlayer Mini* dan *GSM SIM 800L* memberikan *output*.
- Output* : Dengan aktifnya semua sistem otomasi Perangkat audio maka akan mengeluarkan *output* suara.

2. FlowChart



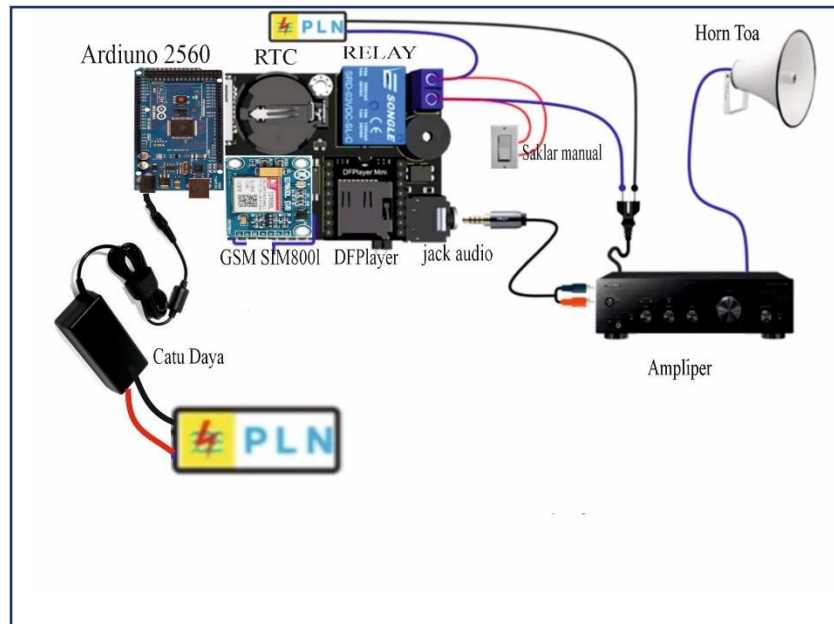
Gambar 3. 3 FlowChart

3. Usecase



Gambar 3. 4 Usecase aplikasi Android

4. Ilustrasi Sistem



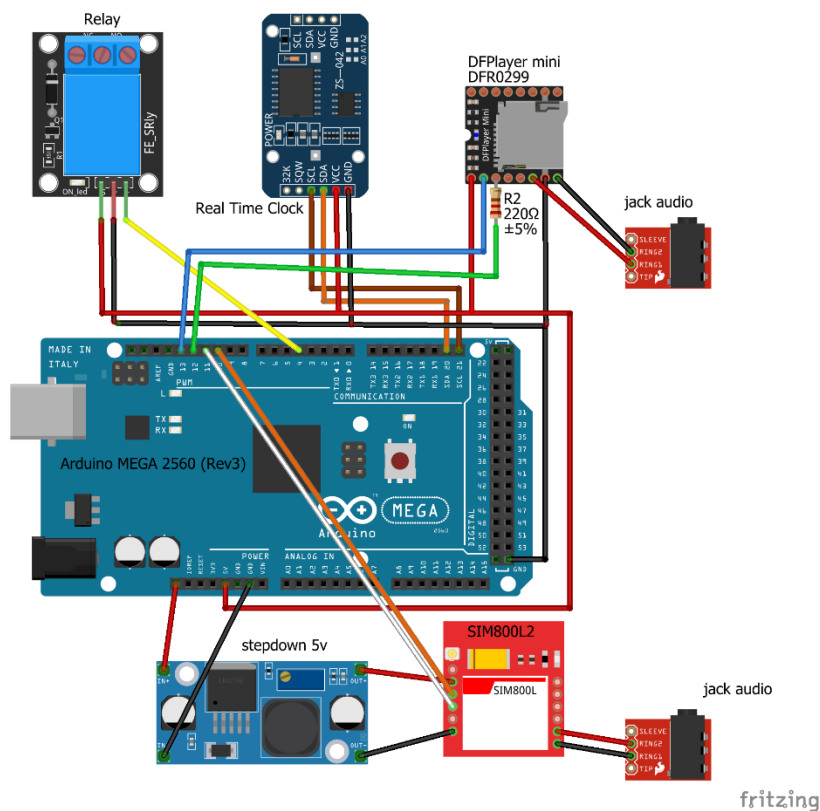
Gambar 3. 5 Ilustrasi Sistem

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Dalam bab ini, penulis akan menjelaskan hasil pengujian alat yang telah direncanakan dan melakukan pembahasan untuk menilai apakah kemampuan sistem alat sudah sesuai atau tidak. Beberapa perubahan mungkin terjadi pada desain yang awalnya tidak termasuk dalam alat sistem, seperti Resistor dan beberapa perangkat lainnya. Selain itu, bab ini juga akan mengulas hasil-hasil dari berbagai tahapan penelitian, investigasi, perencanaan, pelaksanaan, dan hasil akhirnya.



Gambar 4. 1 Skema Rangkaian Sistem

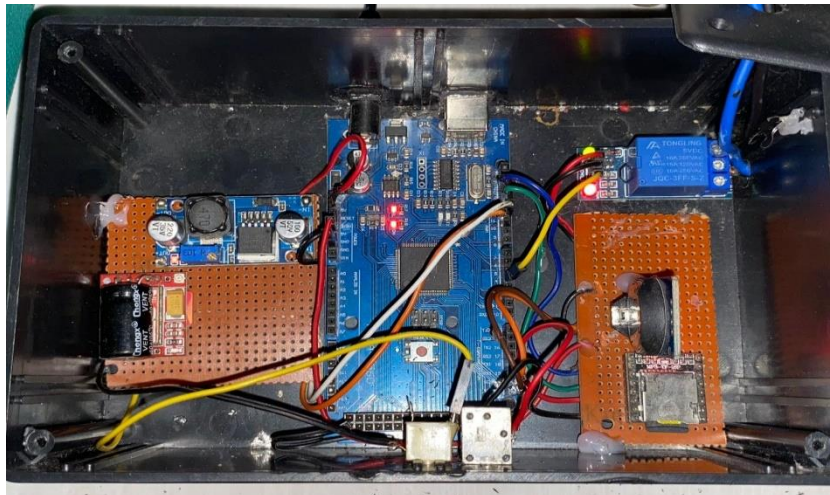
B. Perakitan Perangkat Keras

Dalam tahap pemasangan perangkat keras untuk pembuatan alat ini, setiap komponen akan dipasang secara berurutan. Peneliti akan memulai dengan memasang modul *DFPlayer Mini*, yang bertugas memutar audio tahrir, diikuti oleh *RTC (Real-Time Clock)* sebagai jam untuk mengatur waktu. Selanjutnya, peneliti akan memasang *SIM800L*, yang memungkinkan pengiriman SMS otomatis melalui jaringan *GPRS*, dan juga dapat digunakan untuk panggilan telepon. Selain itu, *relay* akan dipasang untuk mengontrol *on/off sound system*.

Peneliti juga akan memasang jack audio sebagai penghubung dengan *amplifier* untuk pengaturan suara, dan *stepdown* akan digunakan untuk mengurangi arus tegangan sesuai kebutuhan. Seluruh sistem ini akan dihubungkan ke *Arduino Mega 2560*, yang bertanggung jawab untuk mengatur jalannya program dan mengkoordinasikan berbagai komponen tersebut.



Gambar 4. 2 Keseluruhan Rangkaian



Gambar 4. 3 Bagian Dalam Rangkaian

Tabel 4. 1 Rangkaian Modul *DFPlayer mini* ke *Arduino*

Modul <i>DFPlayer mini</i>	<i>Arduino</i>
Vcc	5V
Rx	Pin 13
Tx	Pin 12
Gnd	Gnd

Tabel 4. 2 Rangkaian *SIM800L* ke *Arduino*

<i>SIM800L</i>	<i>Arduino</i>
Rx	Pin 10
Tx	Pin 11
Gnd	Gnd

Tabel 4. 3 Rangkaian Modul Relay ke Arduino

Modul Relay	Arduino
Vcc	5V
In	Pin 4
Gnd	Gnd

Tabel 4. 4 Rangkaian Modul Real Time clock ke Arduino

Modul RTC	Arduino
Vcc	5V
SDA	Pin 20
SCL	Pin 21
Gnd	Gnd

Tabel 4. 5 Rangkaian jack audio ke SIM800L/DFPlayer Mini

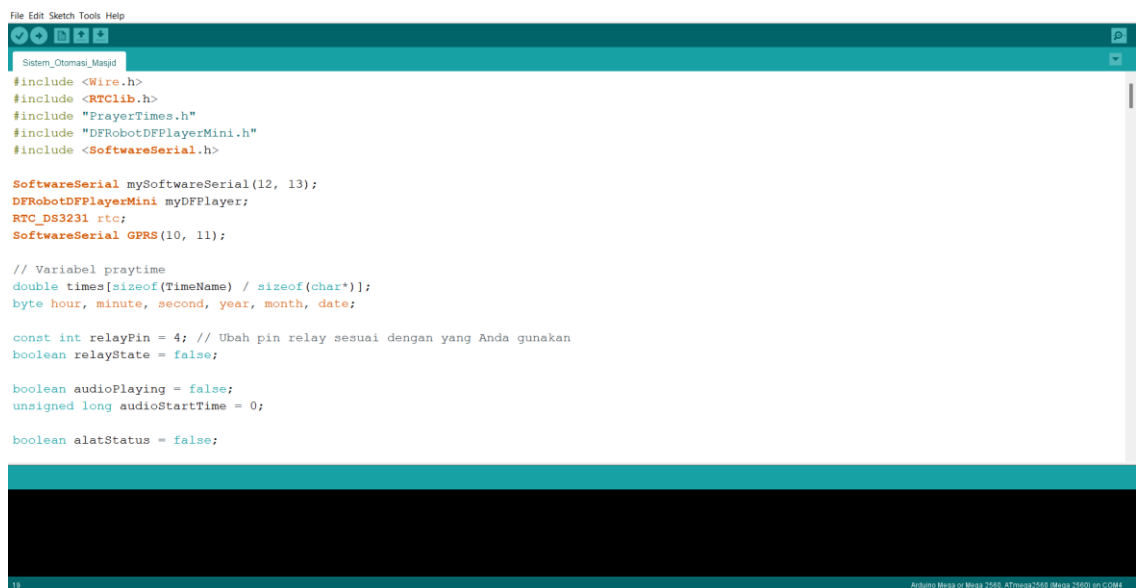
Microphone	SIM800L
Kabel +	Vcc
Kabel -	Gnd

Tabel 4. 6 Rangkaian Stepdown 4V ke SIM800L

Stepdown 4V	SIM800L
Out +	Vcc
Out -	Gnd

C. Sketch Program Arduino IDE

Peneliti menggunakan Arduino IDE untuk menulis sketch, yang kemudian diupload ke board Arduino Uno untuk memasukkan program. Dengan menggunakan aplikasi Arduino IDE dan modifikasi syntac perangkat lunak pada mikrokontroler Arduino ini membuat program yang berisi perintah yang menerima data dari smartphone dan kemudian mengeksekusi perintah tersebut. Penulisan berikut adalah hasilnya.



Gambar 4. 4 Library dan Inisialisasi

Codingan di atas merupakan bagian dari program Arduino yang menggunakan beberapa library dan mendefinisikan beberapa variabel dan objek yang akan digunakan dalam program tersebut:

1. Library dan #include Statements:

#include <Wire.h> Mendukung komunikasi I2C. **#include <RTClib.h>** Library RTC untuk mengakses modul RTC (Real-Time

Clock). "**PrayerTimes.h**" Mengimpor file header untuk mengakses fungsi terkait waktu sholat. "**DFRobotDFPlayerMini.h**" Mengimpor library untuk mengendalikan modul pemutar audio DFPlayer Mini. **<SoftwareSerial.h>** Library untuk komunikasi serial melalui pin digital.

2. Objek dan Variabel:

SoftwareSerial mySoftwareSerial(12, 13); ini untuk Membuat objek **mySoftwareSerial** untuk komunikasi serial melalui pin 12 (RX) dan 13 (TX). **DFRobotDFPlayerMini myDFPlayer;** untuk Membuat objek **myDFPlayer** untuk mengendalikan DFPlayer Mini. **RTC_DS3231 rtc;** Membuat objek **rtc** untuk RTC DS3231. **SoftwareSerial GPRS(10, 11);** Membuat objek **GPRS** untuk berkomunikasi dengan perangkat GPRS melalui pin 10 (RX) dan 11 (TX). **double times[sizeof(TimeName) / sizeof(char*)];** Array untuk menyimpan waktu sholat. Variabel **hour**, **minute**, **second**, **year**, **month**, dan **date** digunakan untuk menyimpan informasi waktu dan tanggal.

3. Relay dan Status:

const int relayPin = 4; Mendefinisikan pin untuk mengendalikan relay.

boolean relayState = false; Variabel status relay (dinyalakan atau dimatikan). **boolean alatStatus = false;** Variabel status alat secara keseluruhan.

4. **Audio:**

boolean audioPlaying = false; Variabel status pemutaran audio.

unsigned long audioStartTime = 0; Waktu mulai pemutaran audio.

5. **String allowed PhoneNumber=** Terdapat variabel ``allowedPhoneNumber`` yang menunjukkan nomor telepon yang diizinkan, yaitu `" +6282136132508"`. Kemudian, terdapat fungsi ``getCallerNumber()`` yang berfungsi untuk mengambil nomor telepon pemanggil. Langkah-langkah dalam fungsi ``getCallerNumber()`` adalah sebagai berikut: Mengirimkan perintah `"AT+CLCC"` melalui GPRS. Melakukan penundaan selama 100 milidetik menggunakan ``delay(100)``. Membuat variabel ``response`` yang awalnya kosong. Melakukan pembacaan data yang tersedia dari GPRS satu per satu. Jika karakter yang dibaca bukan merupakan karakter baris baru (``\n``) atau carriage return (``\r``), maka karakter tersebut akan ditambahkan ke dalam variabel ``response``. Mengembalikan nilai dari variabel ``response``, yang berisi respon dari perangkat terkait nomor telepon pemanggil.

Kode tersebut memiliki tujuan utama untuk mengimplementasikan sebuah sistem kompleks berbasis Arduino. Pertama, dengan menggunakan modul RTC DS3231, sistem dapat melacak waktu dan tanggal secara akurat, memastikan keakuratan waktu operasional. Selanjutnya, dengan memanfaatkan library `"PrayerTimes.h,"` sistem dapat menghitung dan menyimpan waktu sholat berdasarkan parameter waktu dan lokasi yang telah ditentukan sebelumnya.

Selain itu, kode tersebut memanfaatkan komunikasi serial menggunakan dua objek **SoftwareSerial**, yaitu **mySoftwareSerial** dan **GPRS**, untuk berinteraksi dengan perangkat eksternal. Seluruh sistem juga terintegrasi dengan modul DFPlayer Mini untuk memungkinkan pemutaran audio, sedangkan kendali relay digunakan untuk mengontrol perangkat eksternal. Sebagai tambahan, variabel **alatStatus** digunakan untuk memantau keseluruhan status alat atau sistem. Dengan demikian, kode ini menciptakan fondasi yang kuat untuk sistem Arduino yang dapat digunakan dalam berbagai konteks dan aplikasi.



Gambar 4. 5 Void Setup

Kode ini merupakan bagian dari fungsi **setup()** dalam program Arduino, yang dieksekusi sekali saat mikrokontroler pertama kali dinyalakan. Berikut adalah maksud dan tujuan dari setiap bagian kode tersebut:

1. **Serial.begin(9600);** dan **mySoftwareSerial.begin(9600);**:

Membuka komunikasi serial dengan kecepatan baud rate 9600 untuk komunikasi antara Arduino dan perangkat eksternal melalui pin default dan pin yang telah diatur dengan objek **mySoftwareSerial**.

2. **GPRS.begin(9600);**, **GPRS.listen();**, dan konfigurasi AT commands:

Menyiapkan dan mengonfigurasi komunikasi serial dengan perangkat GPRS.

GPRS.listen(); memastikan bahwa SoftwareSerial mendengarkan perangkat GPRS. Mengirim AT commands untuk mengonfigurasi mode pesan (SMS) dan notifikasi SMS.

3. **Wire.begin();** dan **rtc.begin();**:

Menginisialisasi komunikasi I2C untuk berkomunikasi dengan modul RTC menggunakan objek **Wire**. Memulai modul RTC untuk mendapatkan akses waktu real-time.

4. Inisialisasi variabel waktu dari modul RTC:

Mengambil informasi waktu (jam, menit, detik), tanggal (tahun, bulan, hari) dari modul RTC dan menyimpannya ke variabel yang sesuai.

5. **myDFPlayer.begin(mySoftwareSerial);** dan **myDFPlayer.volume(20);**:

Menginisialisasi komunikasi serial untuk mengendalikan modul pemutar audio DFPlayer Mini menggunakan objek **mySoftwareSerial**. Mengatur volume pemutar audio menjadi 20.

6. **pinMode(relayPin, OUTPUT);** dan **digitalWrite(relayPin, LOW);:**

Mengkonfigurasi pin sebagai output untuk mengendalikan relay.

Menetapkan awal status relay menjadi LOW (mati).

Tujuan keseluruhan dari fungsi **setup()** ini adalah untuk menyiapkan dan menginisialisasi semua komponen yang akan digunakan dalam program Arduino, termasuk komunikasi serial, modul RTC, modul DFPlayer Mini, perangkat GPRS,



Gambar 4. 6 Void Loop

dan pengaturan awal untuk relay. Setelah eksekusi fungsi **setup()**, Arduino siap untuk memasuki fase **loop()** dan menjalankan program utamanya.

Codingan di atas adalah bagian dari fungsi **loop()** dalam program Arduino. Fungsi **loop()** ini akan dieksekusi secara berulang-ulang setelah **setup()** selesai dijalankan. Tujuannya adalah untuk memantau dan memproses data yang masuk dari perangkat GPRS, serta melakukan tindakan berdasarkan data tersebut.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

Kode dalam fungsi **loop()** ini bertujuan untuk terus-menerus memeriksa dan menanggapi pesan SMS yang diterima melalui perangkat GPRS. Berikut adalah maksud dan tujuan dari setiap bagian kode tersebut:

1. **while (GPRS.available()) {:**

Membuka loop untuk membaca pesan SMS yang tersedia dari perangkat GPRS.

2. **String sms = GPRS.readStringUntil('\n');**

Membaca pesan SMS sampai karakter newline ('\n') dari perangkat GPRS dan menyimpannya dalam variabel **sms**.

Menghilangkan spasi ekstra di awal dan akhir pesan SMS.

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

Mencetak pesan SMS yang diterima ke Serial Monitor untuk keperluan debug.

4. **if (sms.equalsIgnoreCase("ON"))** {Jika pesan SMS adalah "ON", mengaktifkan alat dengan mengatur **alatStatus** menjadi true, menyalakan relay (**digitalWrite(relayPin, HIGH)**), dan mengirim SMS konfirmasi.

5. **else if (sms.equalsIgnoreCase("TELPON")) {:**

Jika pesan SMS adalah "TELPON", melakukan panggilan telepon ke nomor yang telah ditentukan dan mencetak informasi panggilan ke Serial Monitor.

6. **else if (sms.equalsIgnoreCase("OFF")) {:**

Jika pesan SMS adalah "OFF", mematikan alat dengan mengatur **alatStatus** menjadi false, mematikan relay (**digitalWrite(relayPin, LOW)**), dan mengirim SMS konfirmasi.

7. **else if (sms.startsWith("VOL")) {:**

Jika pesan SMS dimulai dengan "VOL", mengekstrak nilai volume dari pesan tersebut. Memeriksa apakah nilai volume valid, dan jika ya, mengatur volume pada modul DFPlayer Mini. Mengirim SMS konfirmasi perubahan volume.

Tujuan dari fungsi **loop()** ini adalah memonitor secara terus-menerus pesan SMS yang masuk melalui perangkat GPRS, dan berdasarkan isi pesan tersebut, mengambil tindakan yang sesuai, seperti mengaktifkan atau mematikan alat, melakukan panggilan telepon, atau mengatur volume pemutar audio. Selain itu, program juga akan terus membaca dan memproses data dari modul *SIM800L*.

```

Sistem_Otomasi_Masjid | Arduino 1.8.19
File Edit Sketch Tools Help

Sistem_Otomasi_Masjid

// Mendeteksi panggilan masuk dan mengangkat secara otomatis
if (GPRS.available()) {
  String callStatus = GPRS.readStringUntil('\n');
  callStatus.trim();
  if (callStatus.startsWith("RING")) {
    delay(2000);

    // Dapatkan nomor telepon dari panggilan
    String phoneNumber = getCallerNumber();

    // Periksa apakah nomor telepon sesuai dengan yang diizinkan
    if (phoneNumber.equals(allowedPhoneNumber)) {
      GPRS.println("ATA");
      GPRS.println("AT+CHLD=1");
      Serial.println("Mengangkat panggilan...");
    } else {
      GPRS.println("ATH"); // Tidak mengangkat panggilan jika nomor telepon tidak sesuai
      Serial.println("Panggilan dari nomor yang tidak diizinkan. Panggilan ditolak.");
    }
  }
}

Done compiling
Sketch uses 21532 bytes (8%) of program storage space. Maximum is 253952 bytes.
Global variables use 1172 bytes (14%) of dynamic memory, leaving 7020 bytes for local variables. Maximum is 8192 bytes.

```

Gambar 4. 7 Void Mengangkat Telepon

Jika terdeteksi adanya panggilan masuk dengan status "RING", Arduino menunda

sejenak, mendapatkan nomor telepon penelepon, dan membandingkannya dengan nomor yang diizinkan. Jika sesuai, Arduino mengangkat panggilan dan mengaktifkan fungsi hold; jika tidak sesuai, panggilan ditolak. Informasi tentang tindakan yang diambil dicetak di Serial Monitor. Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **if (GPRS.available()) {:**

Memeriksa apakah ada data yang tersedia di perangkat GPRS, menandakan adanya aktivitas panggilan. Menginisiasi pengelolaan panggilan hanya jika data tersedia untuk dibaca.

2. **String callStatus = GPRS.readStringUntil('\n');**

Membaca status panggilan dari data yang diterima hingga karakter newline ('\n') dan menyimpannya dalam variabel **callStatus**. Mendapatkan informasi terkini tentang status panggilan yang diterima.

3. **callStatus.trim();:**

Menghilangkan spasi ekstra di awal dan akhir string **callStatus**. Memastikan integritas data dan memudahkan pemrosesan lebih lanjut.

4. **if (callStatus.startsWith("RING")) {:**

Memeriksa apakah status panggilan dimulai dengan string "RING", menandakan panggilan sedang diterima. Menentukan apakah perangkat harus merespon panggilan yang masuk.

5. **delay(2000);:**

Menunda eksekusi selama 2000 milidetik (2 detik). Memberikan waktu untuk memastikan bahwa data panggilan sudah tersedia dan lengkap.

6. **String callStatus = GPRS.readStringUntil('\n');**

Membaca status panggilan dari data yang diterima hingga karakter newline ('\n') dan menyimpannya dalam variabel **callStatus**. Mendapatkan informasi terkini tentang status panggilan yang diterima.

7. **callStatus.trim();**

Menghilangkan spasi ekstra di awal dan akhir string **callStatus**. Memastikan integritas data dan memudahkan pemrosesan lebih lanjut.

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

Memeriksa apakah status panggilan dimulai dengan string "RING", menandakan panggilan sedang diterima. Menentukan apakah perangkat harus merespon panggilan yang masuk.

9. **delay(2000);**

Menunda eksekusi selama 2000 milidetik (2 detik). Memberikan waktu untuk memastikan bahwa data panggilan sudah tersedia dan lengkap.

10. **String phoneNumber = getCallerNumber();**

Memanggil fungsi **getCallerNumber()** untuk mendapatkan nomor telepon penelepon. Mendapatkan informasi nomor telepon yang dapat digunakan untuk verifikasi.

11. `if (phoneNumber.equals(allowedPhoneNumber)) {:`

Memeriksa apakah nomor telepon penelepon sama dengan nomor yang diizinkan. Menentukan apakah panggilan harus diangkat dan diaktifkan fungsi hold jika sesuai.

12. `GPRS.println("ATH");:`

Mengirim perintah "ATH" untuk menolak panggilan. Menolak panggilan jika nomor telepon penelepon tidak diizinkan.

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

Mencetak pesan ke Serial Monitor untuk memberikan informasi tentang tindakan yang diambil. Memberikan umpan balik pada Serial Monitor tentang penolakan panggilan.

14. `GPRS.println("ATA");` dan `GPRS.println("AT+CHLD=1");:`

Mengirim perintah "ATA" untuk mengangkat panggilan dan "AT+CHLD=1" untuk mengaktifkan fungsi hold. Mengambil tindakan untuk mengaktifkan panggilan jika nomor telepon penelepon diizinkan.

11. `Serial.println("Mengangkat panggilan...");:`

Mencetak pesan ke Serial Monitor untuk memberikan informasi tentang tindakan yang diambil. Memberikan umpan balik pada Serial Monitor tentang pengangkatan panggilan yang diizinkan. Melalui langkah-langkah ini, kode ini bertujuan memberikan kendali atas respon panggilan berdasarkan nomor telepon penelepon yang telah ditentukan sebelumnya.



```

Arslip_Skripsi
File Edit Sketch Tools Help

Arslip_Skripsi$

DateTime now = xtc.now();
double lintang = -3.9194525;
double bujur = 119.7864342;
int Z = 8;

int jam, menit;

set_calc_method(IZNA);
set_ar_method(Shafii);
set_high_lat_adjust_method(AngleBased);
set_fajr_angle(20);
set_isha_angle(10);
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) && now.minute() <= menit) {
  if (!audioPlaying) {
    digitalWrite(relayPin, HIGH);
    playAudio(1);
    audioPlaying = true;
    audioStartTime = millis();
  }
  if (audioPlaying && millis() - audioStartTime >= 3600000) {
    digitalWrite(relayPin, LOW);
    audioPlaying = false;
  }
}
// Shuhur
get_float_time_parts(times[2], jam, menit);
if (now.hour() == jam && now.minute() >= (menit-15) && now.minute() <= menit) {

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json

Arduino Mega or Mega 2560, ATmega328P (Mega 2560) on COM3

```

Gambar 4. 8 Void Waktu Sholat

Kode ini bertujuan untuk menghitung jadwal waktu sholat, khususnya waktu Subuh, berdasarkan parameter geografis tertentu seperti lintang dan bujur. Apabila waktu Subuh mendekati, kode akan mengaktifkan relay dan memutar audio selama 20 menit menggunakan fungsi `playAudio(3)` ini berlaku untuk lima waktu sholat, dengan memastikan bahwa pemutaran audio tidak berulang dalam satu jam. Proses ini memberikan penggunaan praktis dari waktu sholat untuk mengintegrasikan fungsi audio sebagai pengingat, dan kontrol relay memungkinkan penyesuaian perangkat tambahan pada waktu yang spesifik dalam satu siklus waktu.

Dengan demikian, kode ini berfungsi untuk mengatur Kode ini berfungsi untuk menghitung dan memeriksa waktu sholat, khususnya waktu Subuh (Shubuh). Berikut adalah penjelasan untuk setiap baris kode:

1. **DateTime now = rtc.now();:**

Mengambil waktu sekarang dari modul RTC (Real-Time Clock) dan menyimpannya dalam objek **now**.

2. **double lintang = -3.9194525;** dan **double bujur = 119.7864342;:**

Menetapkan nilai lintang (latitude) dan bujur (longitude) sesuai dengan lokasi geografis tertentu.

3. **int Z = 8;:**

Menetapkan zona waktu GMT (Greenwich Mean Time), di mana WIB (Waktu Indonesia Barat) biasanya menggunakan zona waktu 7.

4. **int jam, menit;:**

Mendeklarasikan variabel untuk menyimpan nilai jam dan menit.

5. **set_calc_method(ISNA);**, **set_asr_method(Shafii);**, dan sejenisnya:

Mengatur metode perhitungan waktu sholat dan parameter lainnya, seperti metode perhitungan (ISNA), pendapat Imam Syafi'i (Shafii), dan penyesuaian tinggi lintang (AngleBased).

6. **set_fajr_angle(20);** dan **set_isha_angle(18);:**

Mengatur sudut waktu untuk waktu Subuh (Fajr) dan waktu Isya (Isha).

7. **get_prayer_times(year, month, date, lintang, bujur, Z, times);:**

Menghitung jadwal waktu sholat berdasarkan parameter yang telah diatur sebelumnya dan menyimpannya dalam array **times**.

8. **get_float_time_parts(times[0], jam, menit);:**

Mendapatkan bagian jam dan menit dari waktu Subuh (Shubuh) yang dihitung sebelumnya.

9. **if (now.hour() == jam && now.minute() >= (menit-20) && now.minute() <= menit) {**

Kode tersebut merupakan sebuah kondisi yang menguji waktu saat ini dalam jam dan menit. Jika waktu saat ini memiliki jam yang sama dengan nilai variabel `jam`, menit saat ini berada dalam rentang 20 menit sebelum hingga waktu yang ditentukan oleh variabel `menit`, maka blok kode di dalam if statement akan dijalankan. Dengan demikian, kondisi tersebut memungkinkan eksekusi kode tertentu ketika waktu saat ini sesuai dengan kriteria yang ditetapkan, yaitu jam yang sama dengan `jam` dan menit berada dalam rentang 20 menit sebelum hingga waktu yang ditentukan.

10. **if (!audioPlaying) { ... }:**

Jika tidak sedang memutar audio, maka mengaktifkan relay dan memutar audio dengan menggunakan fungsi **playAudio(3)**.

11. **audioPlaying = true; dan audioStartTime = millis();:**

Menandai bahwa audio sedang diputar dan mencatat waktu dimulainya pemutaran audio.

12. **if (audioPlaying && millis() - audioStartTime >= 3600000) { ... }:**

Jika audio sedang diputar dan telah berlangsung selama satu jam, mematikan relay dan menyetel kembali status audioPlaying menjadi false.

pemutaran audio dan relay pada waktu Subuh, dimana audio akan diputar 20 menit sebelum waktu Subuh dan dimatikan setelah satu jam pemutaran, ini berlaku juga untuk waktu sholat yang lain.

```

//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;
    }
}
}

```

Gambar 4.9 Void Bulan Ramadan

Kode ini menentukan awal dan akhir bulan Ramadan pada tahun 2024, dengan kriteria tanggal antara 12 Maret dan 11 April, dan selama periode tersebut, pesan "Saatnya memutar audio untuk Ramadan!" dicetak di Serial Monitor. Selain itu, program ini mengelola waktu sholat, khususnya waktu Isya, dengan memulai pemutaran audio 20 menit sebelum waktu Isya. Jika kondisi waktu terpenuhi dan audio belum diputar, relay diaktifkan untuk memulai audio menggunakan fungsi `playAudio(7)`, dan status pemutaran audio diatur sebagai aktif. Terdapat juga pemeriksaan waktu tambahan agar relay dimatikan setelah pemutaran audio

selama 4 jam. Integrasi ini memberikan kendali otomatis pada pemutaran audio selama waktu Isya, sementara juga memberikan fitur khusus selama bulan Ramadan pada tahun 2024.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **if ((now.year() == 2024 && now.month() == 3 && now.day() >= 12) || (now.year() == 2024 && now.month() == 4 && now.day() <= 11)) {:**
Pernyataan kondisional untuk menentukan apakah saat ini berada dalam rentang tanggal 12 Maret hingga 11 April 2024, yang dianggap sebagai periode Ramadan.
2. **Serial.println("Saatnya memutar audio untuk Ramadan!");:** Jika kondisi Ramadan terpenuhi, pesan dicetak di Serial Monitor.
3. **// Shubuh:** Komentar untuk menandai bagian yang berkaitan dengan waktu Shubuh.
4. **get_float_time_parts(times[0], jam, menit);:** Mengambil bagian jam dan menit dari waktu Shubuh.
5. **if (now.hour() == jam && now.minute() >= (menit-30) && now.minute() <= menit):** Pernyataan kondisional untuk menentukan apakah waktu saat ini sesuai dengan waktu Shubuh dikurangi 30 menit.
6. **if (!audioPlaying) { digitalWrite(relayPin, HIGH); playAudio(2); audioPlaying = true; audioStartTime = millis(); }:** Jika kondisi waktu Shubuh terpenuhi dan audio belum diputar, relay diaktifkan, audio dimulai, dan status pemutaran audio diatur sebagai aktif.

7. **if (audioPlaying && millis() - audioStartTime == 3600000) { digitalWrite(relayPin, LOW); audioPlaying = false; }**: Pemeriksaan waktu tambahan untuk memastikan bahwa jika audio telah berjalan selama 1 jam (3600000 milidetik), relay dimatikan, dan status pemutaran audio dikembalikan ke tidak aktif.
8. **// Maghrib:** Komentar untuk menandai bagian yang berkaitan dengan waktu Maghrib.
9. **get_float_time_parts(times[4], jam, menit);**: Mengambil bagian jam dan menit dari waktu Maghrib.
10. **if (now.hour() == jam && now.minute() >= (menit-20) && now.minute() <= menit)** Pernyataan kondisional untuk menentukan apakah waktu saat ini sesuai dengan waktu Maghrib dikurangi 20 menit.
11. **if (!audioPlaying) { digitalWrite(relayPin, HIGH); playAudio(1); audioPlaying = true; audioStartTime = millis(); }**: Jika kondisi waktu Maghrib terpenuhi dan audio belum diputar, relay diaktifkan, audio dimulai, dan status pemutaran audio diatur sebagai aktif.
12. **if (audioPlaying && millis() - audioStartTime == 3600000) { digitalWrite(relayPin, LOW); audioPlaying = false; }**: Pemeriksaan waktu tambahan untuk memastikan bahwa jika audio telah berjalan selama 1 jam (3600000 milidetik), relay dimatikan, dan status pemutaran audio dikembalikan ke tidak aktif.
13. **// Isya:** Komentar untuk menandai bagian yang berkaitan dengan waktu Isya.

14. **get_float_time_parts(times[6], jam, menit);**: Mengambil bagian jam dan menit dari waktu Isya.
15. **if (now.hour() == jam && now.minute() >= (menit-20) && now.minute() <= menit);**: Pernyataan kondisional untuk menentukan apakah waktu saat ini sesuai dengan waktu Isya dikurangi 20 menit.
16. **if (!audioPlaying) { digitalWrite(relayPin, HIGH); playAudio(7); audioPlaying = true; audioStartTime = millis(); }**: Jika kondisi waktu Isya terpenuhi dan audio belum diputar, relay diaktifkan, audio dimulai, dan status pemutaran audio diatur sebagai aktif.
17. **if (audioPlaying && millis() - audioStartTime >= 14400000) { digitalWrite(relayPin, LOW); audioPlaying = false; }**: Pemeriksaan waktu tambahan untuk memastikan bahwa jika audio telah berjalan selama 4 jam (14400000 milidetik), relay dimatikan, dan status pemutaran audio dikembalikan ke tidak aktif.

```

if (now.dayOfTheWeek() == 5) {
  get_float_time_parts(times[2], jam, menit);
  if (now.hour() == jam && now.minute() >= (menit-40) && now.minute() <= menit){
    if (!audioPlaying) {
      digitalWrite(relayPin, HIGH);
      playAudio(9);
      audioPlaying = true;
      audioStartTime = millis();
    }
  }
  if (audioPlaying && millis() - audioStartTime >= 7200000) {
    digitalWrite(relayPin, LOW);
    audioPlaying = false;
  }
}

```

Gambar 4. 10 Void Sholat Jumat

Kode ini menambahkan kondisi untuk menentukan apakah hari ini adalah hari Jumat, dan jika iya, mengaktifkan relay dan memutar audio pada waktu tertentu selama Sholat Dzuhur Jumat, dengan menghentikan pemutaran

audio setelah 2 jam, serta mencetak tanggal dan waktu saat ini di Serial Monitor setiap iterasi loop.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **if (now.dayOfTheWeek() == 5) {**: Pernyataan kondisional untuk mengevaluasi apakah hari ini adalah hari Jumat (nilai 5 pada fungsi **now.dayOfTheWeek()** menunjukkan Jumat dalam kalender Arduino).
2. **get_float_time_parts(times[2], jam, menit);**: Mengambil bagian jam dan menit dari waktu Dhuhr (Sholat Dzuhur) dari array **times**.
3. **if (now.hour() == jam && now.minute() >= (menit-20) && now.minute() <= menit);**: Pernyataan kondisional untuk mengevaluasi apakah waktu saat ini sesuai dengan waktu Dhuhr dikurangi 40 menit.
4. **if (!audioPlaying) { digitalWrite(relayPin, HIGH); playAudio(8); audioPlaying = true; audioStartTime = millis(); }**: Jika kondisi waktu Dhuhr pada hari Jumat terpenuhi dan audio belum diputar, relay diaktifkan, audio dimulai menggunakan fungsi **playAudio(8)**, dan status pemutaran audio diatur sebagai aktif.
5. **if (audioPlaying && millis() - audioStartTime >= 7200000) { digitalWrite(relayPin, LOW); audioPlaying = false; }**: Pemeriksaan waktu tambahan untuk memastikan bahwa jika audio telah berjalan selama 2 jam (7200000 milidetik), relay dimatikan, dan status pemutaran audio dikembalikan ke tidak aktif.
6. **}**: Penutup dari blok kondisional yang dimulai dengan **if (now.dayOfTheWeek() == 5)**.

7. **printDateTime(now);**: Memanggil fungsi **printDateTime** untuk mencetak tanggal dan waktu saat ini di Serial Monitor.
8. **delay(10000);**: Menunda eksekusi selama 10 detik sebelum mengulangi loop.

Kode ini menambahkan kondisi untuk menentukan apakah hari ini adalah hari Jumat, dan jika iya, mengaktifkan relay dan memutar audio pada waktu tertentu selama Sholat Dzuhur Jumat, dengan menghentikan pemutaran audio setelah 2 jam, serta mencetak tanggal dan waktu saat ini di Serial Monitor setiap iterasi loop.

```
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();
    }
}
```

Gambar 4. 11 *Void Relay dan PlayAudio*

Kode ini menyediakan tiga fungsi: **relayHidup()** untuk mengaktifkan relay, **relayMati()** untuk mematikan relay, dan **playAudio(int trackNumber)** untuk memulai pemutaran audio pada trek tertentu menggunakan modul DFPlayer Mini, dengan penundaan sebelum memeriksa status pemutaran audio.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **void relayHidup() { digitalWrite(relayPin, HIGH); }**: Fungsi ini mengaktifkan relay dengan menetapkan pin relay (relayPin) ke tingkat

tinggi (HIGH), sehingga mengalirkan daya ke perangkat terkait dengan relay tersebut.

2. **void relayMati() { digitalWrite(relayPin, LOW); }:** Fungsi ini mematikan relay dengan menetapkan pin relay (relayPin) ke tingkat rendah (LOW), sehingga memutuskan aliran daya ke perangkat yang dikendalikan oleh relay.
3. **VoidplayAudio(inttrackNumber){ myDFPlayer.play(trackNumber); delay(500000);while(myDFPlayer.available()){myDFPlayer.readType(); } }:** Fungsi ini memulai pemutaran audio dengan menggunakan modul DFPlayer Mini pada trek audio tertentu (trackNumber). Setelah memulai pemutaran, terdapat delay sementara selama 500000 milidetik (setengah detik) sebelum memeriksa status pemutaran audio. Selanjutnya, ada loop while yang membaca jenis pesan dari modul DFPlayer Mini sampai tidak ada pesan lagi. Fungsi ini digunakan untuk mengaktifkan pemutaran audio dengan relay dan memberikan waktu untuk memastikan bahwa pemutaran audio telah dimulai sebelum memeriksa statusnya.

Kode ini mendefinisikan tiga fungsi dalam bahasa pemrograman Arduino.

```
Sistem_Otomasi_Masjid $
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=\"082136132508\""); // Ganti dengan nomor tujuan untuk notifikasi
    delay(500);
    while (Serial.available()) {
```

Gambar 4. 12 Void sendSMS

Fungsi **sendSMS** dalam kode Arduino bertujuan untuk mengirim pesan SMS melalui perangkat GPRS. Kode ini mengatur komunikasi serial antara Arduino dan GPRS dengan mengirimkan serangkaian perintah AT, termasuk memeriksa ketersediaan perangkat GPRS, mengatur mode pesan SMS, menentukan nomor tujuan, mengirim isi pesan, dan menandai akhir pesan dengan karakter "Ctrl-Z". Selain itu, terdapat penanganan untuk mentransfer data antara Serial Monitor dan perangkat GPRS untuk memonitor respons dan status pengiriman pesan SMS. Fungsi ini memberikan cara untuk mengimplementasikan pengiriman pesan SMS secara otomatis melalui perangkat GPRS pada proyek Arduino dengan menggantikan nomor tujuan dan isi pesan sesuai kebutuhan.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **void sendSMS(const char* pesan) {**: Mendefinisikan fungsi **sendSMS** dengan parameter pesan yang merupakan string konstan.
2. **GPRS.println("AT");**: Mengirim perintah "AT" untuk memeriksa ketersediaan perangkat GPRS.
3. **delay(500);**: Menunda eksekusi selama 500 milidetik untuk memberikan waktu respons dari perangkat GPRS.
4. **while (Serial.available()) { GPRS.write(Serial.read()); }**: Mentransfer data dari Serial Monitor ke perangkat GPRS.
5. **GPRS.println("AT+CMGF=1");**: Mengirim perintah "AT+CMGF=1" untuk mengatur mode pesan SMS.

6. **GPRS.println("AT+CMGS=\"082136132508\")**; // **Ganti dengan nomor tujuan untuk notifikasi**: Mengirim perintah untuk mengatur nomor tujuan SMS (gantilah dengan nomor yang sesuai).
7. **GPRS.print(pesan)**;: Mengirim isi pesan SMS.
8. **GPRS.write(26)**;: Mengirim karakter ASCII 26, yang merupakan karakter "Ctrl-Z" sebagai penanda akhir pesan SMS.
9. **delay(500)**;: Menunda eksekusi selama 500 milidetik untuk memastikan pesan SMS terkirim.
10. **while (Serial.available()) { GPRS.write(Serial.read()); }**: Mentransfer data dari Serial Monitor ke perangkat GPRS.
11. **while (GPRS.available()) { Serial.write(GPRS.read()); }**: Mentransfer balasan dari perangkat GPRS ke Serial Monitor.
12. **}**: Penutup dari fungsi **sendSMS**.

Fungsi ini secara keseluruhan digunakan untuk mengirim pesan SMS dengan memanfaatkan perintah AT pada komunikasi serial antara Arduino dan perangkat GPRS.

```

110 void makeCall(String phoneNumber) {
111     GPRS.print("ATD");
112     GPRS.print(phoneNumber);
113     GPRS.println(";");
114     delay(1000);
115 }

```

Gambar 4. 13 *Void MakeCall*

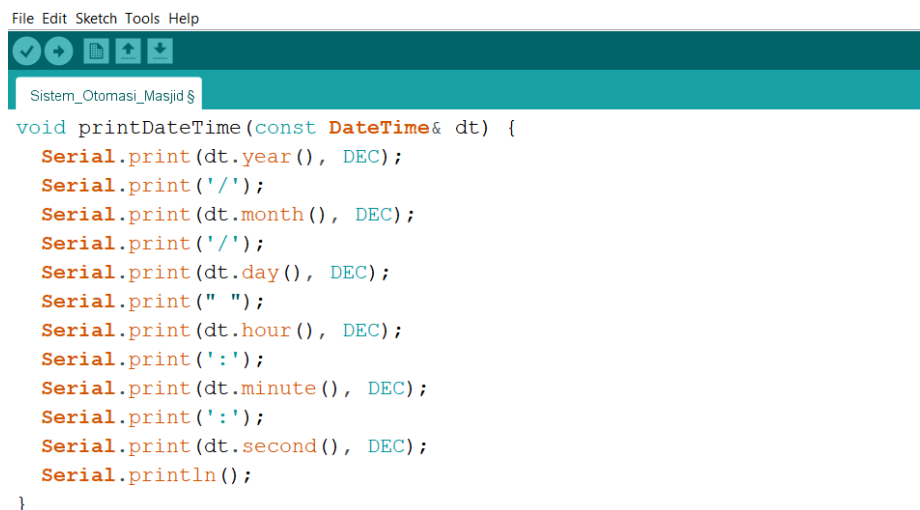
Codingan di atas adalah sebuah fungsi bernama **makeCall()** yang bertujuan untuk membuat panggilan telepon menggunakan modul GSM menggunakan protokol AT commands.

Berikut adalah maksud dan tujuan dari setiap bagian coding tersebut:

1. **void makeCall(String phoneNumber) { ... }:** Ini adalah deklarasi dan definisi fungsi **makeCall()** dengan satu parameter masukan, yaitu **phoneNumber** yang akan berisi nomor telepon yang akan dihubungi.
2. **GPRS.print("ATD");:** Baris ini menggunakan objek **GPRS** (SoftwareSerial) untuk mengirimkan perintah "ATD" ke modul GSM. Perintah "ATD" digunakan untuk memulai proses panggilan.
3. **GPRS.print(phoneNumber);:** Baris ini menggunakan objek **GPRS** (SoftwareSerial) untuk mengirimkan nomor telepon yang diinginkan (**phoneNumber**) ke modul GSM. Nomor telepon ini akan ditambahkan ke perintah "ATD", sehingga akan menciptakan perintah "ATD<phoneNumber>" yang akan dijalankan oleh modul GSM untuk melakukan panggilan ke nomor tersebut.
4. **GPRS.println(";");:** Baris ini menggunakan objek **GPRS** (SoftwareSerial) untuk mengirimkan karakter titik koma (;) ke modul GSM. Karakter titik koma digunakan sebagai penanda akhir dari nomor telepon dalam perintah "ATD<phoneNumber>". Dengan karakter ini, modul GSM akan mengetahui bahwa nomor telepon telah selesai diinput dan siap untuk melakukan panggilan.
5. **delay(1000);:** Baris ini memberi jeda selama 1000 milidetik (1 detik) sebelum melanjutkan eksekusi program berikutnya. Jeda ini diberikan untuk memberi waktu bagi modul GSM untuk memproses perintah

"ATD<phoneNumber>" dan melakukan panggilan ke nomor telepon yang telah ditentukan.

Tujuan dari codingan ini adalah untuk membuat panggilan telepon ke nomor yang telah ditentukan (**phoneNumber**). Fungsi **makeCall()** akan mengirimkan perintah "ATD<phoneNumber>;" ke modul GSM untuk memulai proses panggilan. Setelah jeda 1 detik (**delay(1000)**), proses panggilan akan dilakukan oleh modul GSM dan akan melakukan panggilan ke nomor telepon yang telah ditentukan.



Gambar 4. 14 Void Date Time

Kode ini adalah fungsi bernama **printDateTime** pada Arduino, yang bertujuan untuk mencetak tanggal dan waktu yang diberikan (objek DateTime) ke Serial Monitor. Berikut adalah penjelasan perbaris:

1. **void printDateTime(const DateTime& dt) {}**: Mendefinisikan fungsi **printDateTime** yang menerima objek DateTime sebagai parameter dan tidak mengembalikan nilai.

2. **Serial.print(dt.year(), DEC);**: Mencetak tahun dari objek DateTime ke Serial Monitor dalam format desimal.
3. **Serial.print('/');**: Mencetak karakter '/' untuk memisahkan antara tahun dan bulan.
4. **Serial.print(dt.month(), DEC);**: Mencetak bulan dari objek DateTime ke Serial Monitor dalam format desimal.
5. **Serial.print('/');**: Mencetak karakter '/' untuk memisahkan antara bulan dan hari.
6. **Serial.print(dt.day(), DEC);**: Mencetak hari dari objek DateTime ke Serial Monitor dalam format desimal.
7. **Serial.print(" ");**: Mencetak spasi sebagai pemisah antara tanggal dan waktu.
8. **Serial.print(dt.hour(), DEC);**: Mencetak jam dari objek DateTime ke Serial Monitor dalam format desimal.
9. **Serial.print(':');**: Mencetak karakter ':' untuk memisahkan antara jam dan menit.
10. **Serial.print(dt.minute(), DEC);**: Mencetak menit dari objek DateTime ke Serial Monitor dalam format desimal.
11. **Serial.print(':');**: Mencetak karakter ':' untuk memisahkan antara menit dan detik.
12. **Serial.print(dt.second(), DEC);**: Mencetak detik dari objek DateTime ke Serial Monitor dalam format desimal.

13. **Serial.println();**: Pindah baris baru untuk mengakhiri satu baris cetakan dan menyelesaikan fungsi **printDateTime**.

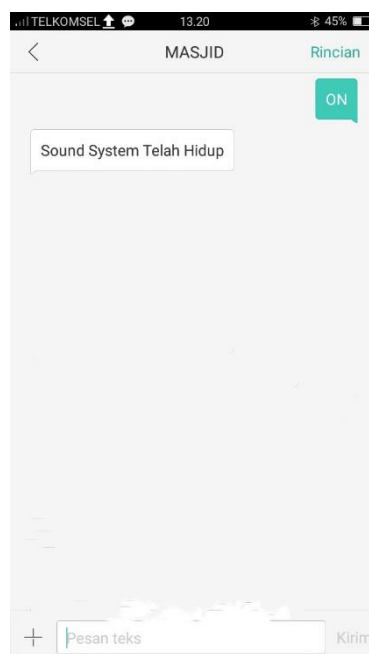
Fungsi ini memudahkan pemantauan waktu dan tanggal di Serial Monitor saat dijalankan pada mikrokontroler Arduino.

D. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah perangkat yang digunakan bekerja dengan baik atau tidak. Pengujian perangkat ini juga sangat bermanfaat untuk mengetahui kinerja fungsi perangkat tertentu, karena dilakukan pada tiap perangkat alat sehingga dapat diketahui secara akurat jika terjadi suatu kesalahan.

1. Perintah ON

Fungsi perintah ON digunakan untuk menghidupkan sistem alat tersebut, dan gambar 4.10 menunjukkan bahwa alat secara otomatis akan mengirimkan SMS ke nomor pemilik yang telah didaftarkan pada alat.



Gambar 4. 15 Perintah ON

2. Perintah Mengatur volume.

Fungsi perintah Mengatur volume jika Anda menggunakan perintah atur volume untuk mengatur volume pada sound system, alat akan mengirimkan perintah dan sound system akan mengatur volume sesuai dengan perintah yang diberikan, seperti yang ditunjukkan pada gambar 4.17.



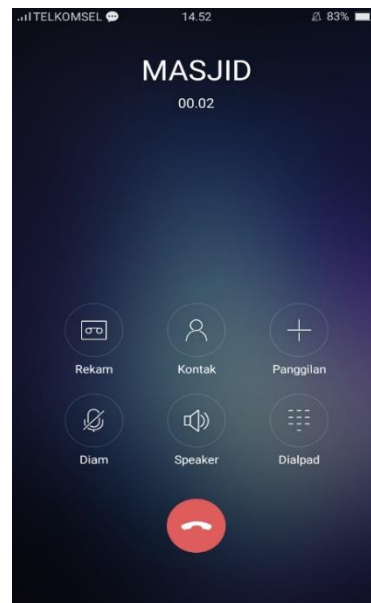
Gambar 4. 16 Perintah Mengatur volume

3. Perintah Telepon

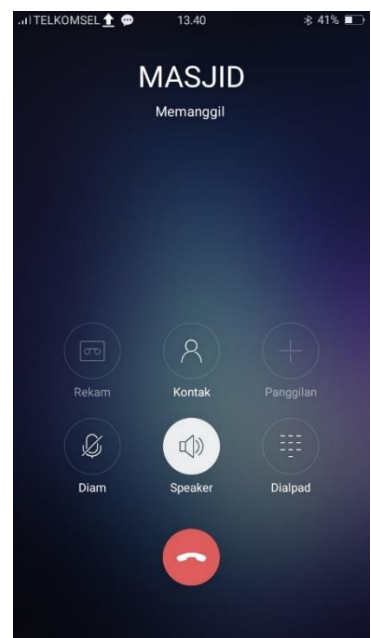
Fungsi ini memungkinkan alat untuk memberikan pengumuman kegiatan masjid melalui telephone, memanggil nomor pemilik sound system. Kemudian anda berbicara di telephone terkait pengumuman kegiatan yang akan dilaksanakan, selanjutnya alat akan menyampailan pengumuman tersebut secara otomatis melalui panngilan telephone.



Gambar 4. 17 Perintah Telpon



Gambar 4. 18 Panggilan Masuk dari Alat



Gambar 4. 19 Mengangkat Panggilan Masuk dari Alat

4. Perintah OFF

Ketika alat dimatikan oleh pemilik, Fungsi perintah Alat off dapat mematikan sistem alat tersebut dan alat akan memberi balasan SMS "Alat Telah Mati". Gambar 4.15 menunjukkan hal ini.



Gambar 4. 20 Alat mematikan sound system

Pengujian sistem alat pemantauan di beberapa jenis mobil di atas menunjukkan bahwa alat tersebut dapat memberikan titik koordinat lokasi kendaraan di berbagai lokasi dengan jarak yang cukup jauh. Di sini, peneliti menguji alat GPS terdekat, yang berjarak 1,3 km, dan yang terjauh, yang berjarak 13 km. Membutuhkan waktu 1 hingga 5 menit untuk mendapatkan pesan dari alat pemantauan sesuai dengan situasi dan kondisi di lokasi tersebut, dengan tingkat keakurasian yang masih kurang.

5. Pengujian *Black Box*

Penulis menggunakan jenis pengujian Black Box untuk mengetahui hasil pengujian sistem. Pengujian ini, yang dilakukan secara real-time, menguji

perintah pesan, penyalakan suara, dan respons perangkat sistem untuk mengetahui apakah sistem bekerja dengan normal. Tabel berikut menunjukkan hasil pengujian keseluruhan.

Tabel 4. 7 Hasil Pengujian Keseluruhan

No	Pengujian	Fungsi	Output	Status
1	Mengaktifkan alat pada sound system dengan SMS perintah ON	Pemilik Sound system dapat menerima SMS yang berupa “Sound system Telah Hidup”	Perangkat merespon melakukan proses menyalakan sound system	Ya [✓] Tidak []
2	Perintah Mengatur Volume misalnya “VOL20”	Pemilik Sound system dapat menerima sms sesuai volume yang di inginkan “VOL20”	Perangkat merespon melakukan pengaturan volume pada alat	Ya [✓] Tidak []
3	Perintah Memberikan pengumuman kegiatan masjid melalui telephone	Pemilik Sound system dapat menerima panggilan telepon dari alat	Perangkat merespon melakukan panggilan telepon ke nomor yang telah di atur	Ya [✓] Tidak []
4	Melakukan panggilan telepon	Memberikan pengumuman kegiatan masjid yang hanya dapat di lakukan oleh panitia masjid	Perangkat merespon	Ya [✓] Tidak []

No	Pengujian	Fungsi	Output	Status
5	Mematikan alat pada sound system dengan perintah OFF	Sound System Telah Mati”	Perangkat merespon mematikan sound system secara otomatis sesuai dengan perintah	Ya [✓] Tidak []
6	Murotal saat pemadaman dan pemulihan listrik	Pemulihan operasional setelah pemadaman listrik	Perangkat merespon apabila direntan waktu yang di tentukan	Ya [✓] Tidak []
7	Mutotal di waktu tertentu	Murotal di waktu bulan puasa dan sholat jum’at	Perangkat merespon sesuai fungsinya	Ya [✓] Tidak []

Hasil pengujian perangkat sistem ini menunjukkan bahwa sistem telah memenuhi fungsinya, seperti yang ditunjukkan oleh hasil pengujian keseluruhan sistem. sehingga sistem yang telah dibangun sudah dapat menghasilkan hasil yang diharapkan.

Tabel 4. 8 Pengujian Sound System Waktu Sholat

No.	Tanggal	Waktu Sholat (WITA)	Hasil Pengujian
1	Selasa, 21 Februari 2024	Subuh : 04:42	Valid
		Duhur : 12:17	Valid
		Ashar : 15:27	Valid

No.	Tanggal	Waktu Sholat (WITA)	Hasil Pengujian
		Magrib : 18:23	Valid
		Isya : 19:32	Valid
2	Rabu, 22 Februari 2024	Subuh : 04:42	Valid
		Duhur : 12:17	Valid
		Ashar : 15:27	Valid
		Magrib : 18:23	Valid
		Isya : 19:32	Valid
3	Kamis, 23 Februari 2024	Subuh : 04:42	Valid
		Duhur : 12:17	Valid
		Ashar : 15:27	Valid
		Magrib : 18:23	Valid
		Isya : 19:32	Valid
4	Jum'at, 24 Februari 2024	Subuh : 04:42	Valid
		Jum'at : 12:17	Valid
		Ashar : 15:27	Valid
		Magrib : 18:23	Valid
		Isya : 19:32	Valid
5	Sabtu, 25 Feberuari 2024	Subuh : 04:42	Valid
		Duhur : 12:17	Valid
		Ashar : 15:27	Valid
		Magrib : 18:23	Valid
		Isya : 19:32	Valid
		Subuh : 04:42	Valid

No.	Tanggal	Waktu Sholat (WITA)	Hasil Pengujian
6	Minggu, 26 Februari 2024	Duhur : 12:17	Valid
		Ashar : 15:27	Valid
		Magrib : 18:23	Valid
		Isya : 19:32	Valid

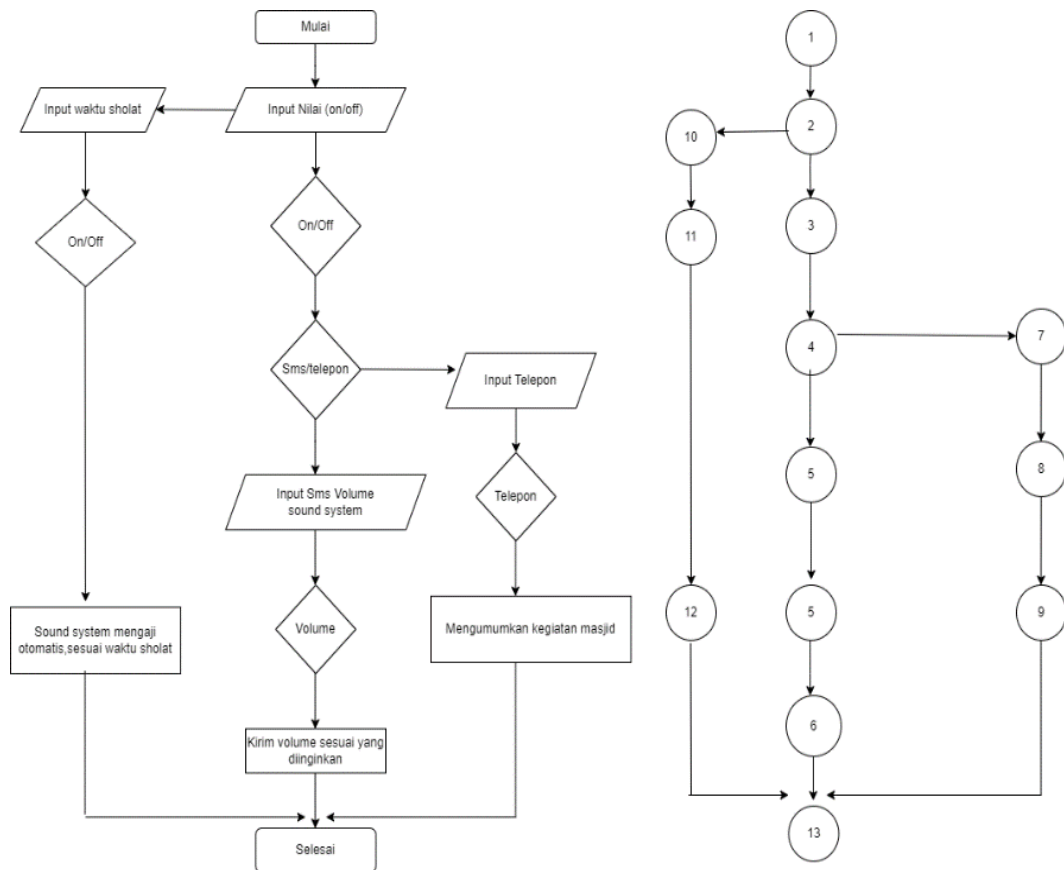
Berdasarkan hasil pengujian waktu sholat yang dilakukan di masjid Babur Rahman pada periode tertentu, dapat disimpulkan bahwa alat ini berhasil secara konsisten dalam mengukur dan memvalidasi waktu sholat pada sebagian besar hari, dengan persentase keberhasilan yang tinggi. Pada umumnya, waktu sholat Subuh, Duhur, Ashar, Magrib, dan Isya dianggap valid, menunjukkan bahwa alat ini dapat diandalkan dalam memberikan informasi yang akurat terkait jadwal sholat berupa *output* suara tahrim.

Dengan tingkat keberhasilan yang tinggi, alat ini dapat dianggap efektif dalam menyajikan jadwal waktu sholat kepada pengguna, memberikan kontribusi positif terhadap ketersediaan informasi waktu sholat secara akurat.

6. Pengujian Whitebox

Pengujian aplikasi dilakukan dengan metode pengujian White Box yang melibatkan analisis struktur internal aplikasi, termasuk pemeriksaan keseluruhan flowchart dan flowgraph aplikasi untuk memastikan kualitas dan keandalan kode program serta logika alur kerja yang terdapat di dalamnya. Metode ini memungkinkan peneliti untuk memahami secara mendalam bagaimana aplikasi beroperasi dari dalam, sehingga

memungkinkan identifikasi potensial bug, kesalahan logika, dan perbaikan yang diperlukan sebelum aplikasi diimplementasikan secara luas. Berikut adalah *Flowchart* dan *flowgraph* secara keseluruhan.



Gambar 4. 21 Flowchart & Flowgraph

Pada gambar 4.18 flowchart dan flowgraph untuk pengujian perangkat lunak, *Flowgraph* diatas dapat dilakukan perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N (Node) = 13$

$E (Edge) = 15$

$$P (\text{Predikat node}) = 3$$

$$R (\text{Region}) = 3$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 15 - 13 + 2 \\ &= 4 \end{aligned}$$

$$\begin{aligned} \text{Predikat Node (N)} &= P + 1 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

2. Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 3

3. *Independent Path* pada *Flowgraph* diatas adalah :

$$\text{Path 1} = 1-2-3-4-5-5-6-13$$

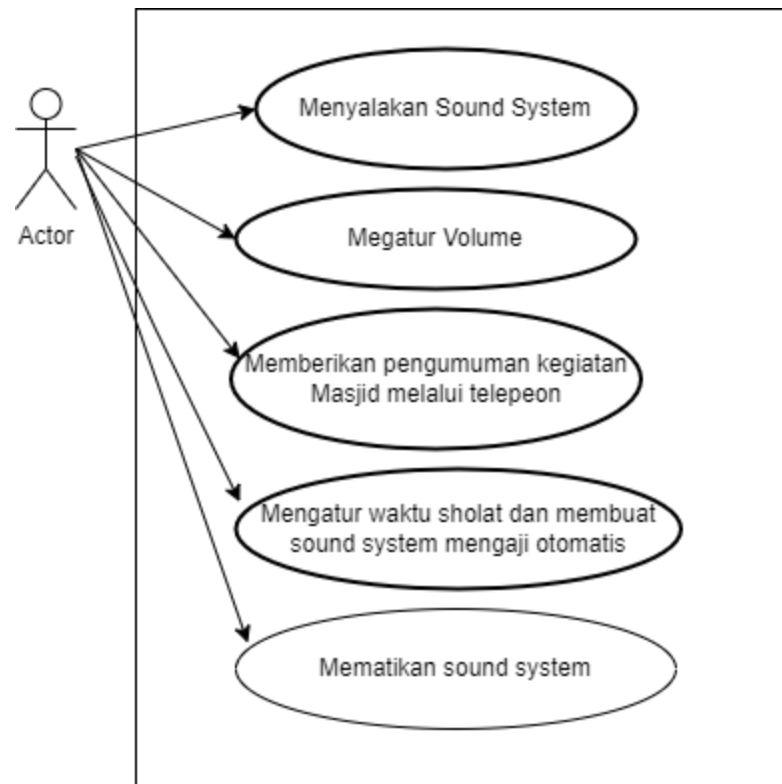
$$\text{Path 2} = 1-2-3-4-7-8-9-13$$

$$\text{Path 3} = 1-2-3-4-19-11-12-13$$

E. UML

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika



Gambar 4. 22 Use Case Diagram

dilihat menurut sudut pandang orang yang berada diluar sistem (*user*)

Keterangan Usecase:

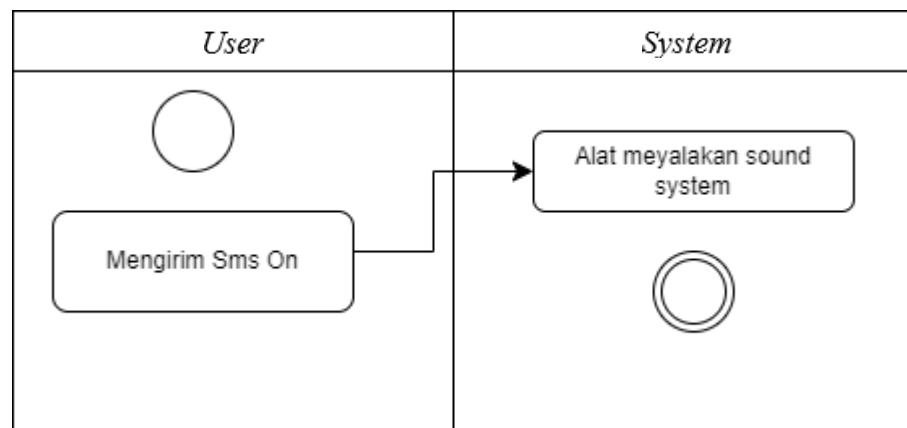
- i. Menyalakan *sound system*: Merupakan proses awal untuk menyalakan sound system menggunakan alat.
- ii. Mengatur volume: Merupakan proses mengatur volume *sound system* menggunakan alat.
- iii. Memberikan pengumuman kegiatan masjid melalui *telephone*:

Merupakan proses dimana memberikan pengumuman terkait kegiatan masjid melalui *telephone* dari alat.

- iv. Mengatur waktu sholat : Merupakan proses membuat sound system mengaji secara otomatis berdasarkan waktu sholat.
- v. Mematikan *sound system* : Merupakan proses untuk mematikan sound system menggunakan alat.

2. Activity Diagram

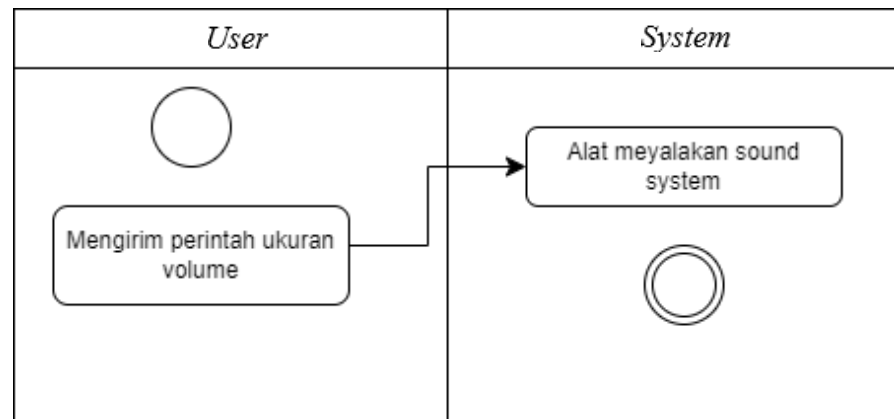
Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sistem.



Gambar 4. 23 Activity Diagram Menyalakan sound system

a. Activity Diagram Menyalakan *sound system*

Activity Diagram Diagram Menyalakan sound system bertujuan untuk mendefinisikan bagaimana user Menyalakan sound system melalui alat.

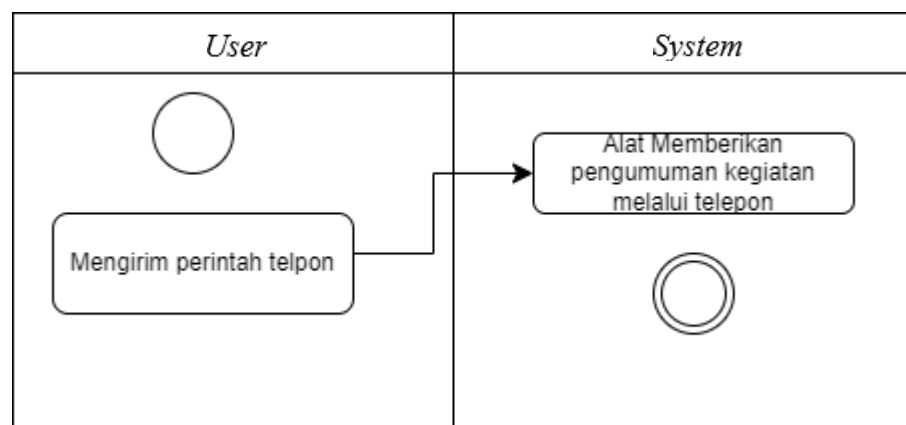


Gambar 4. 24 Activity Diagram Mengatur volume sound system

b. *Activity Diagram Mengatur volume sound system*

Activity Diagram Mengatur volume bertujuan untuk mendefinisikan bagaimana user Mengatur volume menggunakan Alat.

c. *Activity Diagram Memberikan kegiatan pengumuman masjid melalui telephone*

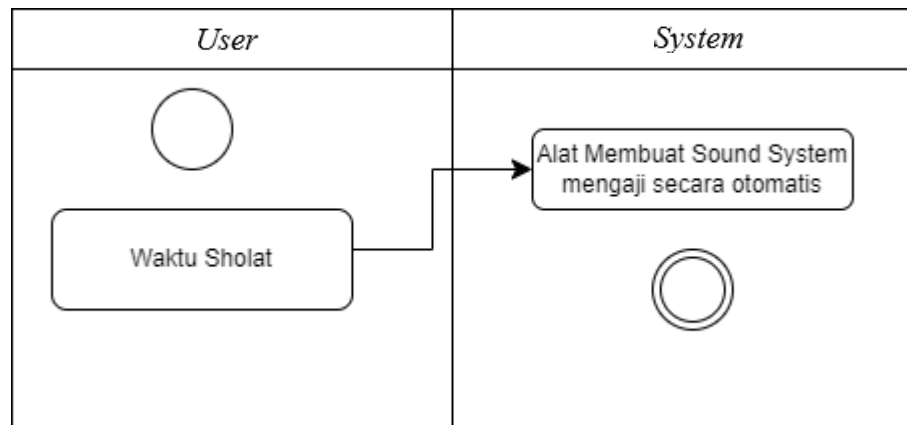


Gambar 4. 25 Activity Diagram pengumuman kegiatan masjid

Activity Diagram Memberikan pengumuman kegiatan masjid melalui telephone bertujuan untuk mendefinisikan bagaimana *user* Memberikan pengumuman kegiatan masjid melalui telephone.

d. *Activity diagram* Waktu sholat

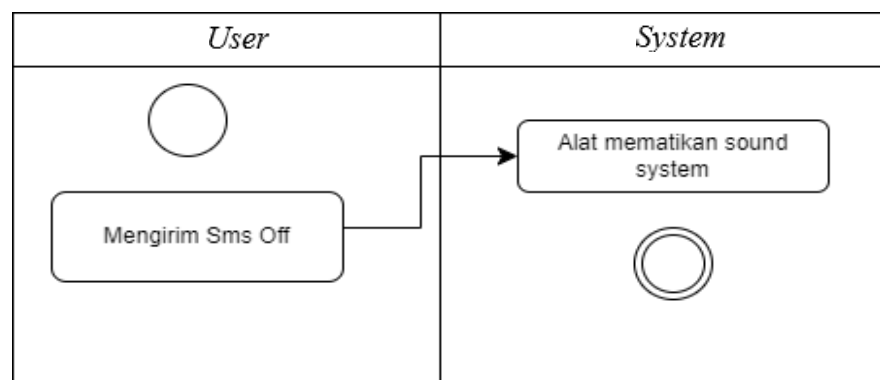
Activity Diagram Diagram Waktu sholat bertujuan untuk mendefinisikan bagaimana user Membuat *sound system* mengaji secara otomatis dengan perintah sms waktu sholat, selanjutnya *sound system* akan mengaji secara otomatis.



Gambar 4. 26 Activity Diagram Mengatur Waktu Sholat

e. *Activity diagram* Mematikan *sound system*

Activity Diagram Diagram Mematikan *sound system* bertujuan untuk mendefinisikan bagaimana *user* Mematikan *sound system* dengan perintah *sms off*, selanjutnya *sound system* akan mati melalui alat



Gambar 4. 27 Activity Diagram Mematikan Sound System

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Penulis telah merumuskan beberapa kesimpulan yang signifikan berdasarkan hasil penelitian yang mendalam mengenai rancang bangun alat sistem otomasi perangkat audio masjid. Penelitian ini bertujuan untuk memberikan solusi yang efektif dan inovatif bagi pengurus masjid dalam upaya mereka untuk lebih efisien dalam mengelola aktivitas masjid, khususnya yang berkaitan dengan perangkat audio. Kesimpulan-kesimpulan tersebut mencakup beragam aspek, termasuk keandalan dan fungsionalitas alat, kegunaan dalam mengatur jadwal sholat, serta dampak positif terhadap ketepatan waktu pelaksanaan ibadah di masjid tersebut. Dengan demikian alat ini diharapkan dapat memberikan manfaat yang signifikan bagi komunitas masjid dan mewujudkan upaya dalam menjalankan ibadah secara lebih teratur dan efisien. Selanjutnya, temuan ini mejadi dasar yang kuat untuk mengembangkan dan meningkatkan alat sistem otomasi perangkat audio masjid di masa depan.

1. Rancang bangun sistem alat ini terdiri dari beberapa komponen: Adaptor menjadi pusat sumber tegangan, *Arduino Mega 2560*, *stepdown*, *SIM800L*, modul *Real Time Clock*, *DFPlayer Mini*, *Relay*, *Sound System* dan *Handphone*.

2. Pengembangan aplikasi yang melibatkan pengendalian panggilan telepon, pemutaran audio, dan kontrol *relay* berdasarkan waktu sholat dengan modul *GPRS* memerlukan perhatian khusus terhadap konfigurasi modul, perintah *AT*, serta uji coba dan *debugging* untuk memastikan fungsionalitas yang diinginkan dapat tercapai.
3. Hasil perancangan perangkat Ini menunjukkan bagaimana *mikrokontroller Arduino* dapat berkomunikasi dengan perangkat *SIM800L* yang dihubungkan dengan *Handphone*.
4. Proses pengiriman dan penerimaan *SMS* sangat bergantung pada sinyal provider dan suara yang keluar ke *Handphone*
5. Pada saat melakukan operasional murotal kemudian terjadi pemadaman listrik maka alat akan *restart* sehingga mempengaruhi audio.

B. Saran

Rancang bangun sistem pengaturan otomatis perangkat audio masjid dengan *SMS* dan Telepon yang menggunakan *Handphone* untuk mengatur kegiatan masjid masih belum sempurna. Karena itu, beberapa saran diberikan kepada pengembang, seperti berikut:

1. Alat ini akan mencari kembali sinyal *GPRS* ketika tidak terhubung dengan . Anda harus menggunakan komponen yang lebih baik agar alat tidak restart ketika terputus dengan listrik.
2. Untuk mengatasi kesalahan kabelisasi pada pin speaker *SIM800L* yang tertukar polaritasnya, Anda dapat menukarnya dengan benar.

DAFTAR PUSTAKA

- Andhika Wahyu Putra, Rizki Nuryanto & Asni Tafrikhatin,(2020). Fitur Pengingat Kegiatan Masjid Dengan Kontrol WI-FI Berbasis ESP-32 Pada Jam Digital
- Adi, B., & Herlina, A. (2019). Smart Home With Smart Control, Berbasis Bluetooth Mikrokontroller. JEECOM: Journal of Electrical Engineering and Computer, 1(1), 1–11. Diperoleh dari:<https://doi.org/10.33650/jeecom.v1i1.883>
- Hamdani, R., Puspita, I. H., & Wildan, B. D. R. W. (2019). Pembuatan Sistem Pengamanan Kendaraan Bermotor Berbasis Radio Frequency Identification (Rfid). *Indept*, 8(2), 56–63.
<http://jurnal.unnur.ac.id/index.php/indept/article/download/290/278>
- Jabbar, M. J. A. A. (2015). Pendidikan Al-Qur ' an Masjid Arofah , Dusun Bandung. *Jurnal Inovasi Dan Kewirausahaan*, 4(3), 161–165.
- Jamila. Pengenalan Bahasa C.32 Di akses dari <http://www.jamilah.staff.gunadarma.ac.id/Downloads/files/33355/pengenalan-c-sdcc.pdf> pada tanggal 22 November 2018.
- Kallang, A. (2018). Konteks Ibadah Menurut Al-Quran. *Al-Din: Jurnal Dakwah Dan Sosial Keagamaan*, 4(2), 1–13.
- Kristanto, Andri. (2013). *Algoritma dan Pemrograman dengan C++ edisi ke-3*. Yogyakarta: Graha Ilmu.
- Menengah Mahir, P. (2011). *PENGENALAN ARDUINO* ✓ Oleh : Feri Djuandi. 1–24.
- MohammadRidwan. (2017). Optimalisasi Teknologi Informasi Berbasis SMS Gateway Memanfaatkan Perintah Dasar AT COMMAND. di akses dari: <https://ejournal.unis.ac.id/index.php/jutis/article/view/3/1>
- Muhammad Anton, Sulistiyanto, Muhammad Hasan Basri & Yuli, P. (2020). Perancangan Jam Istiwa Otomatis Menggunakan *Running Text* Dan *Speaker* Sebagai Alat Bantu Waktu Sholat Di Masjid Nurul Hidayah Al-Takwa
- Rosa dan Shalahuddin. 2017. *UML (Unified Modelling Language)*. Bandung.