



PROYEK AKHIR - VE180626

**PENUNJUK ARAH KIBLAT DAN WAKTU SHALAT
MENGUNAKAN *GLOBAL POSITIONING SYSTEM* (GPS)
BERBASIS WEMOS D1 MINI**

Rahmad Hervandana Djafaar
NRP 1031150000092

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

Halaman ini sengaja dikosongkan



FINAL PROJECT - VE180626

QIBLA DIRECTION AND PRAYER TIME USING GLOBAL POSITION SYSTEM (GPS) BASED ON WEMOS D1 MINI

Rahmad Hervandana Djafaar
NRP 103115000092

Supervisor
Ir. Josaphat Pramudijanto, M.Eng

DEPARTMENT OF ELECTRICAL AUTOMATION ENGINEERING
Faculty of Vocations
Institut Teknologi Sepuluh Nopember
Surabaya 2019

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul **“PENUNJUK ARAH KIBLAT DAN WAKTU SHALAT MENGGUNAKAN *GLOBAL POSITIONING SYSTEM (GPS) BERBASIS WEMOS D1 MINI*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 17 Januari 2019



Rahmad Hervandana Djafaar
1031150000092

Halaman ini sengaja dikosongkan

**PENUNJUK ARAH KIBLAT DAN WAKTU SHALAT
MENGUNAKAN GLOBAL POSITIONING SYSTEM (GPS)
BERBASIS WEMOS D1 MINI**

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing

Ir. Josaphat Pramudijanto, M.Eng
NIP. 19621005 199003 1 003

**SURABAYA
JANUARI 2019**

Teknologi
Sepuluh Nopember

Halaman ini sengaja dikosongkan

**PENUNJUK ARAH KIBLAT DAN WAKTU SHALAT
MENGUNAKAN *GLOBAL POSITIONING SYSTEM* (GPS)
BERBASIS WEMOS D1 MINI**

Nama : Rahmad Hervandana Djafaar
Pembimbing : Ir. Josaphat Pramudijanto, M.Eng.

ABSTRAK

Bagi orang muslim salat adalah ibadah yang wajib dikerjakan setiap hari, terutama untuk shalat fardhu, itu dilakukan sehari semalam 5 waktu, yaitu Shalat Subuh, Shalat Dzuhur, Shalat Ashar, Shalat Maghrib, dan Shalat Isya'. Untuk dapat melaksanakan ibadah ini, tentunya memiliki syarat-syarat sah shalat yang harus dipenuhi, diantaranya salah satu syarat sah shalat adalah menghadap kiblat. Menurut Imam Syafi'i dan sebagian Imamiyah berpendapat bahwa menghadap kiblat secara pasti merupakan kewajiban baik bagi itu bagi mereka yang dapat melihat fisik Ka'bah ataupun yang jauh. Jika dapat mengetahui arah Ka'bah secara pasti maka harus menghadap ke arah tersebut.

Pada proyek akhir ini akan dirancang sebuah alat yang dapat menentukan arah kiblat serta jadwal shalat fardhu dengan menggunakan sistem global dengan teknologi GPS dengan kontroler wemos d1 mini. Metode utama yang digunakan pada sistem ini adalah melakukan perhitungan matematik dari data GPS yang berupa posisi koordinat, kemudian membandingkan hasil perhitungannya dengan data dari kompas digital yang merepresentasikan kemana arah pengguna alat menghadap. Arah kiblat dari pengguna dapat diketahui dengan melihat tampilan pada LCD. Selain itu pada LCD juga akan ditampilkan jadwal shalat fardhu yang waktunya mengikuti perubahan sesuai dengan posisi pengguna.

Hasil pengujian pada sensor HMC5883L dengan perbandingan kompas saat kondisi di dalam ruangan menghasilkan nilai rata-rata sebesar $5,875^{\circ}$ sedangkan pada kondisi diluar ruangan (tempat lapang) menghasilkan nilai rata-rata sebesar $2,375^{\circ}$.

Kata Kunci : Arah kiblat, jadwal shalat, wemos d1 mini.

Halaman ini sengaja dikosongkan

QIBLA DIRECTION AND PRAYER TIME USING GLOBAL POSITION SYSTEM (GPS) BASED ON WEMOS D1 MINI

Name : Rahmad Hervandana Djafaar
Supervisor : Ir. Josaphat Pramudijanto, M.Eng

ABSTRACT

For Muslims, shalat is a service that must be done every day, especially for shalat fardhu, it is carried out 5 times a day, namely subuh, dzuhur, Ashar, Maghrib, and Isya'. To be able to carry out shalat, of course, have legal requirements for shalat that must be fulfilled, including one of the legitimate conditions of prayer is facing the Qibla. According to Imam Shafi'i and some Imamiyah argued that facing the Qibla is definitely an obligation both for those who can see the physical Ka'bah or distant. If you can know the direction of the Ka'bah with certainty, it must face in that direction.

In this final project, a tool that can determine the direction of Qibla and the schedule of shalat fardhu will be designed using a global system with GPS technology with the Wemos D1 mini controller. The main method used in this system is to perform mathematical calculations from GPS data in the form of coordinate positions, then compare the results of calculations with data from a digital compass that represents where the user is facing. The direction of the Qibla from the user can be seen by looking at the display on the LCD. In addition, on the LCD you will also be shown the schedule for the Fardhu prayer, which is the time to follow changes according to the user's position.

The test results on the HMC5883L sensor with a comparison of compasses when the conditions in the room produce an average value of 5.875° while those outside the room (field) produce an average value of 2.375°.

Keywords: *direction of qibla, schedule of shalat fardhu, wemos d1 mini*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Allah SWT, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Proyek Akhir dengan judul “**PENUNJUK ARAH KIBLAT DAN WAKTU SHALAT MENGGUNAKAN *GLOBAL POSITION SYSTEM (GPS) BERBASIS WEMOS D1 MINI***” untuk memenuhi syarat kelulusan pada Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Laporan Proyek Akhir ini dapat diselesaikan oleh penulis berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada orang tua dan keluarga tercinta yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing serta sebagai kepala laboratorium atas bimbingan dan arahnya dalam proses pengerjaan Proyek Akhir serta mendidik penulis menjadi lebih baik.. Bapak Ir. Joko Susila, M.T. selaku dosen wali yang banyak membantu penulis dalam menjalani perkuliahan selama ini. Dosen Departemen Teknik Elektro Otomasi atas pendidikan dan ilmunya. Teman-teman dan semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya. Laporan ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun.

Surabaya, 21 Januari 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN JUDUL	iii
PERNYATAAN KEASLIAN PROYEK AKHIR.....	v
LEMBAR PENGESAHAN	Error! Bookmark not defined.
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Sistematika Penulisan	3
BAB II TEORI DASAR	5
2.1 Wemos D1 Mini	5
2.1.1 IDE Arduino.....	7
2.2 Modul Kompas GY-273 HMC5883L.....	9
2.2.1 Deklinasi Magnetik	10
2.3 <i>Global Positioning System</i> (GPS).....	12
2.3.1 Sistem Kerja GPS.....	13
2.3.2 Menentukan Posisi dari <i>Receiver</i> ke Satelit GPS	13
2.3.3 Modul GPS (VK2828U7G5LF).....	14
2.3.4 NMEA-0813	15
BAB III PERANCANGAN DAN PEMBUATAN SISTEM.....	17
3.1 Diagram Fungsional.....	17
3.1.1 Jadwal Shalat dan Rumus Arah Kiblat.....	17
3.1.2 Mikrokontroler	19
3.1.3 GPS	19
3.1.4 HMC5883L	20
3.1.5 OLED 1.3”	21
3.1.6 <i>Buzzer</i>	21
3.2 Konfigurasi Elektronik	21
3.2.1 Konfigurasi Wemos D1 Mini dengan Modul GPS	21

3.2.2 Konfigurasi Wemos D1 Mini dengan HMC5883L	21
3.2.3 Konfigurasi Wemos D1 Mini dengan OLED 1.3''	22
3.2.4 Konfigurasi Wemos D1 Mini dengan <i>Push Button</i>	23
3.2.5 Konfigurasi Wemos D1 Mini dengan <i>Buzzer</i>	23
3.2.6 Konfigurasi Rangkaian Alat Secara Keseluruhan	24
3.3 Konfigurasi Mekanik	25
3.4 Cara Penggunaan Alat	27
3.5 Perbandingan Alat Sejenis dengan Alat Proyek Akhir	29
3.5.1 Kelebihan Alat Proyek Akhir dengan Alat Sejenis	30
BAB IV PENGUJIAN DAN ANALISA	31
4.1 Pengujian Resolusi Kompas	31
4.2 Pengujian Koordinat dan Akuisisi Modul GPS	35
4.2.1 Pengujian Koordinat	36
4.2.2 Pengujian Akuisisi GPS	36
4.3 Pengujian Seluruh Sistem	37
BAB V PENUTUP	43
5.1 Kesimpulan	43
5.2 Saran	43
DAFTAR PUSTAKA	45
LAMPIRAN	47
Pengujian Alat	47
Program <i>Address I2C</i>	48
Program Alat	50
RIWAYAT HIDUP.....	63

DAFTAR GAMBAR

Gambar 2. 1	Wemos D1 Mini	5
Gambar 2. 2	<i>Interface</i> IDE Arduino.....	8
Gambar 2. 3	Modul kompas GY-273 HMC5883L	9
Gambar 2. 4	Ilustrasi Deklinasi Magnetik.....	11
Gambar 2. 5	Deklinasi Magnetik Wilayah Jakarta.....	11
Gambar 2. 6	Deklinasi Magnetik Wilayah Surabaya	12
Gambar 2. 7	Simulasi Posisi Satelit	13
Gambar 2. 8	Pinout Modul GPS.....	14
Gambar 3. 1	Diagram Fungsional.....	17
Gambar 3. 2	Segitiga Bola	18
Gambar 3. 3	<i>Flowchart</i> Data GPS	20
Gambar 3. 4	Konfigurasi Wemos D1 Mini dengan Modul GPS	21
Gambar 3. 5	Konfigurasi Wemos D1 Mini dengan HMC5883L	22
Gambar 3. 6	Konfigurasi Wemos D1 Mini dengan OLED 1.3”	22
Gambar 3. 7	Konfigurasi Wemos D1 Mini dengan <i>Push Button</i>	23
Gambar 3. 8	Konfigurasi Wemos D1 Mini dengan Buzzer.....	24
Gambar 3. 9	Konfigurasi Rangkaian Alat Secara Keseluruhan	24
Gambar 3. 10	Hasil Dari Rangkaian yang Dibuat	25
Gambar 3. 11	Konfigurasi Mekanik.....	25
Gambar 3. 12	Rangkaian Elektronik yang Telah di <i>Packing</i> (Alat Jadi) ...	26
Gambar 3. 13	Tampilan Informasi Masing-masing <i>Display</i>	26
Gambar 3. 14	Tampilan Awal Alat Pada Saat Alat Dihidupkan	27
Gambar 3. 15	Tampilan Pada Saat Alat Telah Menerima Data GPS	27
Gambar 3. 16	Tampilan Jadwal Shalat.....	27
Gambar 3. 17	Tampilan Awal Arah kiblat	28
Gambar 3. 18	Posisi Depan Alat	28
Gambar 3. 19	Tampilan Arah Kiblat yang Telah Disesuaikan.....	29
Gambar 3. 20	Alat Sejenis.....	29
Gambar 4. 1	Perbandingan Resolusi Kompas di Dalam Ruangan.....	32
Gambar 4. 2	Perbandingan Resolusi Kompas di Luar Ruangan	33
Gambar 4. 3	Perbandingan Resolusi Kompas dengan Alat.....	33
Gambar 4. 4	Penggunaan Alat Pada Posisi Datar.....	34
Gambar 4. 5	Penggunaan Alat Pada Posisi yang Salah.....	35
Gambar 4. 6	Tampilan Jadwal Shalat.....	38
Gambar 4. 7	Tampilan Kompas Penunjuk Arah Kiblat.....	38

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2. 1	Spesifikasi Teknis Wemos D1 Mini.....	6
Tabel 2. 2	Pin Konfiguration Wemos D1 Mini	6
Tabel 2. 3	Keterangan Tombol pada Tampilan IDE Arduino	8
Tabel 2. 4	Data yang Menunjukkan Mata Angin	10
Tabel 2. 5	<i>Pinout</i> Modul GPS	14
Tabel 2. 6	Informasi NMEA	15
Tabel 4. 1	Resolusi Kompas di Dalam Ruangan.....	31
Tabel 4. 2	Resolusi Kompas di Luar Ruangan.....	32
Tabel 4. 3	Perbandingan Hasil Alat Pada Posisi Datar dan Salah.....	35
Tabel 4. 4	Pengujian Koordinat	36
Tabel 4. 5	Pengujian Akuisisi GPS	36
Tabel 4. 6	Pengujian Arah Kiblat.....	39
Tabel 4. 7	Perbandingan <i>Error</i> Arah Kiblat.....	39
Tabel 4. 8	Perbandingan Jadwal Shalat.....	41

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Shalat adalah rukun kedua dari rangkaian lima rukun-rukun Islam, dan shalat adalah rukun yang paling ditekankan setelah dua kalimat syahadat. Beberapa ulama berpendapat bahwa barangsiapa yang meninggalkan shalat karena menyepelekan atau malas, dan bukan karena mengingkari kewajibannya, ia telah kafir. Dalam melakukan kegiatan shalat, umat Islam harus memperhatikan syarat sahnya dalam shalat, agar ibadah menjadi sempurna dan diterima. Adapun yang dimaksud dengan syarat sahnya shalat disini ialah Islam, berakal, Tamyiz (dapat membedakan antara yang baik dan buruk), menghilangkan hadats, menghilangkan najis, menutup aurat, masuknya waktu, menghadap kiblat, dan niat. Semua syarat sahnya shalat merupakan ketentuan yang harus dilakukan supaya ibadah menjadi sempurna dan diterima oleh Allah SWT. Seringkali dalam pelaksanaan shalat kita tidak mengetahui arah kiblat, padahal dalam kondisi yang tidak terdesak misalnya. Dalam perjalanan/kendaraan dan sakit. Menghadap kiblat merupakan syarat sahnya shalat.

Arah kiblat sebagai arah menghadap ka'bah dalam melaksanakan shalat yang merupakan salah satu syarat sahnya dalam shalat. umumnya dimasyarakat dalam pembangunan masjid/Musholla menghadap hanya kebarat. Padahal untuk daerah indonesia untuk menghadap ke ka'bah menghadap kebarat agak serong ke utara beberapa derajat. Shalat fardhu merupakan kewajiban ibadah bagi seluruh umat muslim, tetapi terkadang kewajiban tersebut terabaikan salah satu penyebab nya bisa disebabkan karena kurangnya informasi waktu shalat yang disebabkan lokasi baru sehingga orang tersebut tidak tahu.

Keunggulan perangkat GPS adalah kemampuannya untuk melacak posisi objek dan memberitahukan arah atau tujuan yang hendak dituju. Demikian juga bagi umat Muslim yang selalu membutuhkan alat penunjuk arah kiblat dan jadwal shalat yang benar terutama bila sedang berada di suatu tempat yang belum pernah dikunjungi sebelumnya. Menghadap kearah kiblat merupakan suatu permasalahan yang sangat penting dalam syariat Islam. Selain itu dengan penggunaan wemos d1 mini merupakan salah satu perangkat mikrokontroler mini sehingga diharapkan alat mudah dibawa.

1.2 Perumusan Masalah

Berdasarkan dari uraian latar belakang, maka dapat dirumuskan suatu permasalahan. Bagaimana membuat sebuah alat yang menggunakan perangkat mikrokontroler wemos d1 mini yang dapat digunakan di berbagai tempat untuk menampilkan informasi arah kiblat dan waktu shalat serta secara otomatis .

1.3 Batasan Masalah

Agar dalam pengerjaan proyek akhir ini dapat lebih terarah, maka pembahasan penulisan ini dibatasi pada ruang lingkup pembahasan sebagai berikut:

1. Penunjukkan arah kiblat sesuai dengan arah kiblat dimana lokasi pengguna
2. Jadwal waktu shalat fardhu hanya bisa disesuaikan dimana lokasi pengguna
3. Sistem penunjuk arah kiblat dan jadwal shalat fardhu hanya dapat digunakan menggunakan mikrokontroler wemos d1 mini yang memanfaatkan *global position system* (GPS)

1.4 Tujuan

Tujuan dari pengerjaan proyek akhir ini adalah membuat sebuah alat penunjuk arah kiblat dan jadwal shalat fardhu agar dapat diimplementasi umat muslim dengan menggunakan mikrokontroler wemos d1 mini agar mudah dibawa serta dapat digunakan dimanapun dengan memanfaatkan *Global Positioning System* (GPS) .

1.5 Manfaat

Manfaat yang dapat diperoleh dari perancangan alat ini yaitu :

1. Membantu khususnya umat muslim agar dapat mengetahui arah kiblat serta jadwal shalat yang berada dilokasi yang baru dikunjungi
2. Membantu umat muslimm melaksanakan ibadah shalat dan menentukan arah kiblat yang benar
3. Meningkatkan pemahaman tentang struktur dan sistem kerja dalam pengembangan pengaplikasian pada mikrokontroler wemos d1 mini.

1.6 Sistematika Penulisan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, metodologi, serta sistematika laporan Proyek Akhir yang dibuat.

BAB II TEORI DASAR

Menjelaskan teori yang dapat dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

BAB III PERANCANGAN SISTEM

Membahas perencanaan dan pembuatan tentang *hardware* yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

BAB IV PENGUJIAN DAN ANALISA SISTEM

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

BAB V PENUTUP

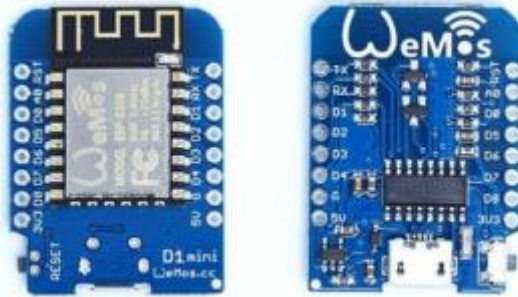
Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Proyek Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

Halaman ini sengaja dikosongkan

BAB II TEORI DASAR

2.1 Wemos D1 Mini [1]

Wemos D1 Mini adalah mikrokontroler dengan kemampuan wifi berbasis ESP-8266EX yang dapat diprogram dengan Arduino IDE. *Board* ini telah mendukung pemrograman serial dan OTA. Berikut merupakan tampak fisik dari wemos d1 mini yang ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Wemos D1 Mini

Board ini memiliki soket *micro-USB* untuk menghubungkan ke komputer. Di samping itu terdapat tombol untuk mengatur ulang papan(*reset*). Sepanjang sisi *board* terdapat dua baris pin, di mana kita akan menghubungkan kabel dengan *project* yang kita buat.

Di *board* terdapat simbol yang mempunyai makna sebagai berikut:

- 3,3V merupakan tegangan sebesar 3,3 V yang dijalankan *board* secara internal. Pin ini seperti kutub sisi positif dari sebuah baterai.
- G merupakan *ground*. Pin ini seperti kutub minus dari sebuah baterai.
- (rx,tx, do-d8) merupakan GPIO adalah singkatan dari *general purpose input output*. Pin ini digunakan untuk mengirim dan menerima sinyal ke berbagai perangkat yang akan kita sambungkan menggunakan kabel.
- a0 - ini adalah pin analog yang dapat mengukur tegangan yang tetapi hanya bisa menangani hingga 3,3V.
- 5V merupakan pin yang terhubung dengan 5V dari komputer. Selain itu pin ini dapat menggunakannya untuk memberi daya pada *board* dengan sebuah baterai saat tidak terhubung ke komputer.

Berikut merupakan tabel spesifikasi teknis serta pin konfigurasi yang masing-masing ditampilkan pada Tabel 2.1 dan 2.2

Tabel 2. 1 Spesifikasi Teknis Wemos D1 Mini

Microcontroller	ESP-8266EX
<i>Operating Voltage</i>	3,3V
<i>Digital I/O Pins</i>	11
<i>Analog Input Pins</i>	1(<i>Max input: 3,3V</i>)
<i>Clock Speed</i>	80MHz/160MHz
<i>Flash</i>	4M bytes
<i>Length</i>	34,2mm
<i>Width</i>	25,6mm
<i>Weight</i>	10g

Tabel 2. 2 Pin Konfiguration Wemos D1 Mini

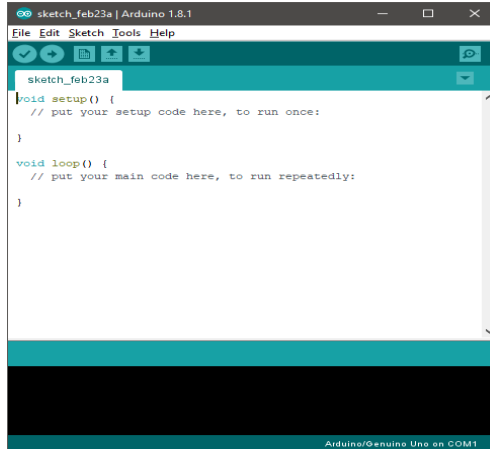
Pin	Fungsi	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	<i>Analog input, max 3,3V input</i>	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO, 10k <i>Pull-up</i>	GPIO0
D4	IO, 10k <i>Pull-up</i> , <i>BUILTIN_LED</i>	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO, 10k <i>Pull-down</i> , SS	GPIO15
G	<i>Ground</i>	GND

Pin	Fungsi	ESP-8266 Pin
5V	5V	–
3V3	3,3V	3,3V
RST	<i>Reset</i>	RST

2.1.1 IDE Arduino

IDE merupakan singkatan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Arduino IDE digunakan untuk membuat, membuka, dan mengubah *sketches* (Arduino menyebut program dengan "*sketches*") seperti yang terlihat pada Gambar 2.2. *Sketches* mendefinisikan apa yang *board* akan lakukan. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.





Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.





Gambar 2. 2 Interface IDE Arduino

IDE Arduino membutuhkan beberapa pengaturan yang digunakan untuk mendeteksi *board* Arduino yang sudah dihubungkan ke komputer. Beberapa pengaturan tersebut adalah mengatur jenis *board* yang digunakan sesuai dengan *board* yang terpasang, dan mengatur jalur komunikasi data melalui perintah *Serial Port*. Kedua pengaturan tersebut dapat ditemukan pada *pull down menu Tools*. Penjelasan tombol IDE Arduino dapat dilihat pada Tabel 2.3.

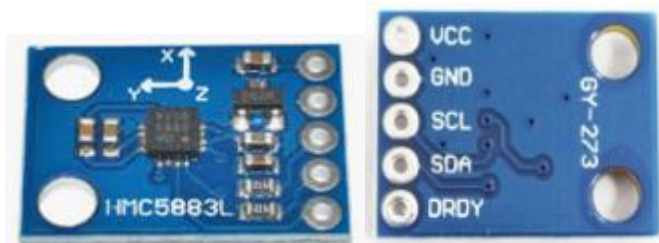
Tabel 2. 3 Keterangan Tombol pada Tampilan IDE Arduino

No	Tombol	Nama	Fungsi
1		<i>Verify</i>	Menguji apakah ada kesalahan pada program atau <i>sketch</i> . Apabila <i>sketch</i> sudah benar, maka <i>sketch</i> tersebut akan dikompilasi. Kompilasi adalah proses mengubah kode program ke dalam kode mesin
2		<i>Upload</i>	Menggunakan atau memasukkan kode mesin hasil kompilasi ke <i>board</i> Arduino
3		<i>New</i>	Membuat <i>sketch</i> yang baru
4		<i>Open</i>	Membuka <i>sketch</i> yang sudah ada atau tersimpan

No	Tombol	Nama	Fungsi
5		<i>Save</i>	Menyimpan <i>sketch</i>
6		<i>Serial Monitor</i>	Menampilkan data yang dikirim dan diterima melalui komunikasi serial

2.2 Modul Kompas GY-273 HMC5883L [2]

Modul Kompas GY-273 adalah sebuah modul yang digunakan untuk menunjukkan arah mata angin digital, atau juga disebut kompas digital. Modul ini menggunakan komponen utama berupa IC HMC5883 yang merupakan IC kompas digital 3 axis yang memiliki *interface* berupa 2 pin I2C. HMC5883 memiliki sensor *magneto-resistive* HMC118X series ber-resolusi tinggi, ditambah ASIC dengan konten *amplification*, *automatic degaussing strap driver*, *offset cancellation* dan 12 bit ADC yang memungkinkan keakuratan kompas mencapai 1 sampai 2 derajat. Modul ini biasa digunakan untuk keperluan sistem navigasi otomatis, *mobile phone*, *netbook* dan perangkat navigasi personal. Berikut merupakan bentuk fisik dari modul yang ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Modul kompas GY-273 HMC5883L

Konfigurasi pin sebagai berikut :

VCC : *Power Supply* (3-5V)

GND : GND

SCL : *I2C Clock Input*

SDA : *I2C Data IO*

DRDY: *Data Ready Output (not connected)*

Berikut adalah beberapa fitur dari Modul Kompas GY-273 :

- Berbasis sensor *magneto-resistive* 3 axis.

- 12-Bit ADC terkopling dengan *Low Noise* AMR Sensor yang memiliki *2 mili-gauss field* dengan resolusi.kurang lebih *8 gauss fields*.
- Tegangan kerja 5V DC.
- Menggunakan antarmuka I2C.
- Keluaran rata-rata maksimum 160 Hz.

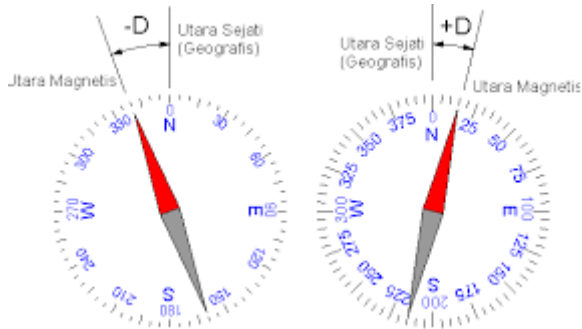
Sensor HMC5883L ini adalah sensor yang sangat sensitif sekali terhadap rotasi dan arah hadap sensor, dikarenakan sensor ini menggunakan medan magnet sebagai acuan dari pendeteksiannya. Data yang telah didapat untuk 4 buah mata angin yang ditunjukkan pada Tabel 2.4.

Tabel 2. 4 Data yang Menunjukkan Mata Angin

Arah Mata Angin	Data
Utara	$255 < X < 257$
Selatan	$79 < X < 81$
Timur	$16 < X < 18$
Barat	$148 < X < 150$

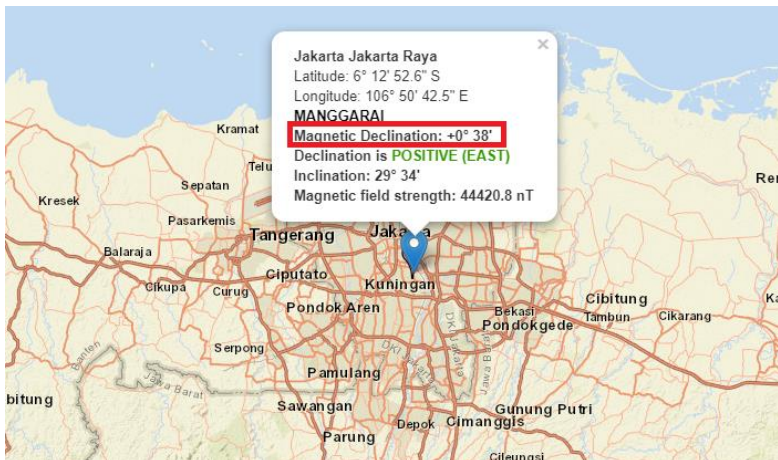
2.2.1 Deklinasi Magnetik

Untuk mendapatkan utara sejati (*true north*) ketika menggunakan kompas, dibutuhkan koreksi deklinasi magnetik terhadap arah jarum kompas. Hal ini karena kutub magnet utara memiliki selisih dengan utara sejati yang besarnya berubah-ubah. Selisih itu disebut variasi magnet atau juga disebut deklinasi magnetik. Nilai variasi magnet ini selalu berbeda di setiap. Pada Gambar 2.4 merupakan ilustrasi dari deklinasi magnetik.

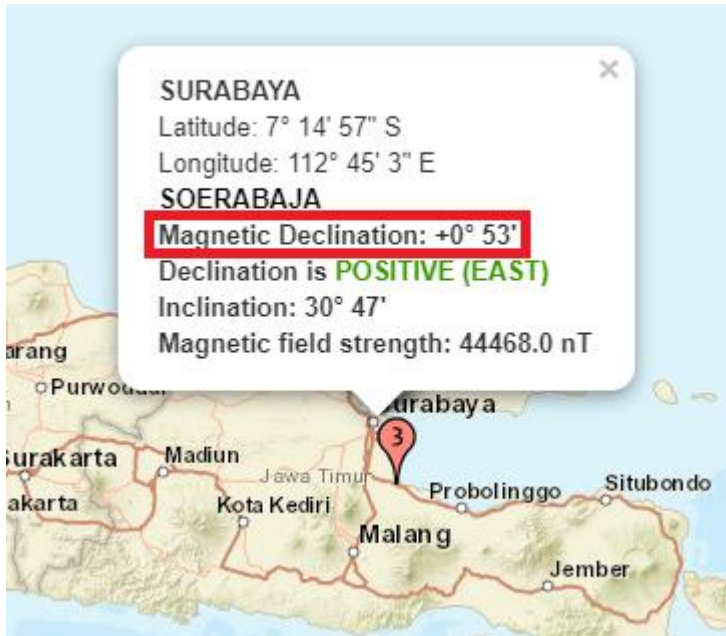


Gambar 2.4 Ilustrasi Deklinasi Magnetik

Sehingga bisa kita ambil kesimpulan bahwa deklinasi magnetik merupakan variabel terpenting dalam menentukan arah kompas yang sebenarnya kita peroleh. Untuk mengetahui nilai deklinasi magnetik pada setiap kota di seluruh dunia bisa dilihat pada web <http://www.magnetic-declination.com> dengan cara meng-*input*-kan kota dan negara yang ingin diketahui nilai dari deklinasi magnetiknya. Berikut merupakan contoh hasil dari kota Jakarta dan Surabaya yang diketahui nilai dari deklinasi magnetiknya yang ditunjukkan pada Gambar 2.5 dan 2.6.



Gambar 2.5 Deklinasi Magnetik Wilayah Jakarta



Gambar 2. 6 Deklinasi Magnetik Wilayah Surabaya

2.3 Global Positioning System (GPS) [3]

GPS merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana dia berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. GPS adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit.

GPS membutuhkan transmisi dari 3 satelit untuk mendapatkan informasi dua dimensi (lintang dan bujur), dan 4 satelit untuk tiga dimensi (lintang, bujur dan ketinggian). GPS terdiri dari 3 segmen yaitu, segmen angkasa, kontrol/pengendali, dan pengguna. Segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam).

Ada beberapa karakteristik yang menjadikan GPS menarik untuk digunakan yaitu dapat digunakan setiap saat tanpa tergantung waktu dan

cuaca, posisi yang dihasilkan mengacu pada suatu datum global, pengoperasian alat *receiver* relatif mudah, relatif tidak terpengaruh dengan kondisi topografis, dan ketelitian yang dihasilkan dapat diandalkan

2.3.1 Sistem Kerja GPS

Secara teoritis, GPS bekerja dengan cara mengumpulkan data dari satelit, masing-masing satelit akan memberikan informasi jarak antara lokasi satelit tersebut dengan sebuah titik di bumi (*GPS receiver*). Dari proses pengambilan lokasi-lokasi tersebut akan diperoleh koordinat-koordinat yang disebut waypoint. Untuk menginformasikan posisi user, 24 satelit GPS yang ada di orbit sekitar 12.000 mil di atas kita, bergerak konstan mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, mempunyai baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari.



Gambar 2. 7 Simulasi Posisi Satelit

2.3.2 Menentukan Posisi dari *Receiver* ke Satelit GPS

Sebuah *GPS receiver* mengetahui lokasi dari satelit dengan cara menghitung seberapa jauh jarak antara satelit dan *receiver* dengan menggunakan rumus sebagai berikut :

$$KECEPATAN \times WAKTU = JARAK \quad (2.1)$$

Keterangan:

- Kecepatan = kecepatan gelombang mikro yang dikirimkan dari satelit
- Waktu = waktu yang dibutuhkan dari satelit mengirimkan sinyal hingga diterima *GPS receiver*
- Jarak = jarak antara satelit dengan *GPS receive*

2.3.3 Modul GPS (VK2828U7G5LF)

VK2828U7G5LF merupakan modul GPS dan antena yang dapat digunakan oleh mikrokontroler untuk mengambil koordinat geografis (lintang dan bujur, ketinggian), kecepatan, dan waktu GMT. Berikut merupakan spesifikasi modul VK2828U7G5LF:

- 1Hz(*default*) ~ 10Hz *output*
- 9600bps(*default*) [support:4800 , 9600 , 19200 , 38400 , 57600 , 115200 , 230400 , 460800,921600] *serial interface*
- 5v @ 30mA (3,3-5v)
- 56-Channel receiver
- *Extremely high sensitivity* -161dBm
- Akurasi 2,5m
- *Operating temperature*: -40°C to 85°C
- *Hot Start* : 1s
- *Warm Start* : 28s
- *Cold Start* : 29s
- *Module Size* 28x28x8.6mm
- LED indikator
- *Optional output code* : NMEA 0183 V3.0(GGA, GSA, GSV, RMC, VTG, GLL)



Gambar 2. 8 Pinout Modul GPS

Tabel 2. 5 Pinout Modul GPS

<i>Name</i>	<i>Function Description</i>
PPS	<i>Output time pulses (pulses per second)</i>
VCC	3.3V~+5V
TX	UART/TTL interface

<i>Name</i>	<i>Function Description</i>
RX	UART/TTL interface
GND	Koneksi pin ground
EN	Power Enable

2.3.4 NMEA-0813

NMEA-0183 adalah standar kalimat laporan yang dikeluarkan oleh GPS receiver. Standar NMEA memiliki banyak jenis bentuk kalimat laporan, di antaranya :

Tabel 2. 6 Informasi NMEA

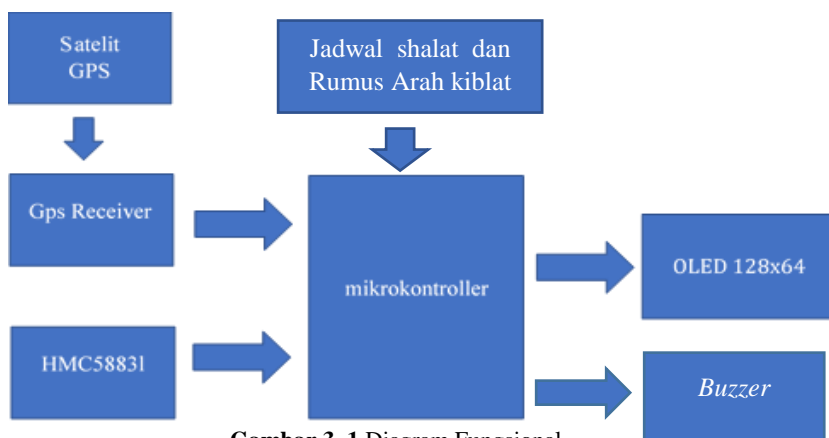
<i>Name</i>	<i>Function Description</i>
\$GPGGA	Global positioning system fixed data
\$GPGLL	Geographic position-latitude/longitude
\$GPGSA	GNSS DOP and active satellites
\$GPGSV	GNSS satellites in view
\$GPRMC	Recommended minimum specific GNSS data
\$GPVTG	Course over ground and ground speed

Halaman ini sengaja dikosongkan

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

3.1 Diagram Fungsional

Diagram Fungsional merupakan salah satu bagian terpenting dalam perancangan peralatan elektronika, karena dari diagram fungsional dapat diketahui prinsip kerja keseluruhan dari rangkaian elektronika yang dibuat. Sehingga keseluruhan dari alat yang dibuat dapat membentuk suatu sistem yang dapat bekerja sesuai dengan perencanaan. Berikut adalah diagram fungsional dari alat yang akan dibuat yang ditunjukkan pada Gambar 3.1.

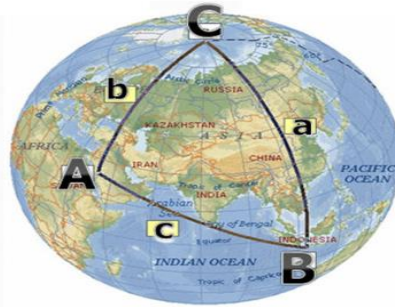


Gambar 3. 1 Diagram Fungsional

Dari Gambar 3.1 dapat dilihat komponen penyusun sistem ini terdiri dari :

3.1.1 Jadwal Shalat dan Rumus Arah Kiblat

Jadwal shalat dan rumus arah kiblat merupakan bagian yang penting dari sitem ini. Pada penentuan jadwal shalat didapatkan dari data GPS Berupa garis lintang dan garis bujur serta tanggal dan waktu dari data tersebut nantinya akan di-*input* pada *libray prayer times* yang dapat diunduh pada <https://github.com/asmaklad/Arduino-Prayer-Times>. Untuk rumus arah kiblat digunakan oleh mikrokontroler untuk menghasilkan arah kiblat berdasarkan data garis lintang dan garis bujur yang dikirim oleh GPS. Untuk penentuan arah kiblat didapatkan dari rumus segitiga bola yang ditunjukkan pada Gambar 3.2 berikut ini.



Gambar 3. 2 Segitiga Bola

Dari gambar di atas Segitiga bola ABC yang menghubungkan titik A (Ka'bah), titik B (lokasi) dan titik C (kutub Utara). Titik A (Ka'bah) memiliki koordinat bujur B_a dan lintang L_a . Titik B memiliki koordinat bujur B_b dan lintang L_b . Titik C memiliki lintang 90 derajat. Busur a adalah panjang busur yang menghubungkan titik B dan C. Busur b adalah panjang busur yang menghubungkan titik A dan C. Busur c adalah panjang busur yang menghubungkan titik A dan B. Sudut C tidak lain adalah selisih antara bujur B_a dan bujur B_b . Jadi sudut $C = B_a - B_b$. Sementara sudut B adalah arah menuju titik A (Ka'bah). Jadi arah kiblat dari titik B dapat diketahui dengan menentukan besar sudut B. Selanjutnya, jari-jari bumi dianggap sama dengan 1. Sudut yang menghubungkan titik di katulistiwa, pusat bumi dan kutub utara adalah 90 derajat. Karena lintang titik A adalah L_a , maka busur b sama dengan $90 - L_a$. Karena lintang titik B adalah L_b , maka busur a sama dengan $90 - L_b$. Dalam trigonometri bola, terdapat rumus-rumus standar sebagai berikut

$$\cos(b) = \cos(a) \cos(c) + \sin(a) \sin(c) \cos(B) \quad (3.1)$$

$$\cos(c) = \cos(a) \cos(b) + \sin(a) \sin(b) \cos(C) \quad (3.2)$$

$$\frac{\sin(A)}{\sin(a)} = \frac{\sin(B)}{\sin(b)} = \frac{\sin(C)}{\sin(c)} \quad (3.3)$$

Dengan menggabungkan ketiga rumus di atas, pada akhirnya akan diperoleh persamaan :

$$\tan(B) = \frac{\sin(C)}{\cos(a) \cdot \tan(b) - \sin(a) \cdot \cos(C)} \quad (3.4)$$

Karena $C = B_a - B_b$, $a = 90 - L_b$, $b = 90 - L_a$, serta mengingat $\cos(90 - x) = \sin(x)$, $\sin(90 - x) = \cos(x)$ dan $\cot(90 - x) = \tan(x)$, Persamaan 3.4 menjadi:

$$\tan(B) = \frac{\sin(Ba - Bb)}{\cos Lb \cdot \tan La - \sin Lb \cdot \cos(Ba - Bb)} \quad (3.5)$$

di mana:

- Ba = Bujur ka'bah (39,82627)
- La = Lintang ka'bah(21,42255)
- Bb = Bujur tempat
- Lb = Lintang tempat

Diketahui lokasi saat ini dengan koordinat dengan lintang -7,28551 dan bujur 112,77432 tentukan arah (*azimuth*) kiblat!

Penyelesaian :

$$a = \sin(112,77432 - 39,82627)$$

$$b = \cos(-7,28551)$$

$$c = \cos(-7,28551)$$

$$d = \sin(-7,28551)$$

$$e = \cos(112,77432 - 39,82627)$$

$$\tan(B) = \frac{a}{b \cdot c - d \cdot e}$$

$$\tan(B) = \frac{0,95}{0,044}$$

$$\tan(B) = 2,159$$

$$B = 65,147$$

$$\text{Azimuth arah kiblat} = 360 - 65,147 = 294,853$$

Jadi arah kiblat di lokasi tersebut adalah 294,853 derajat

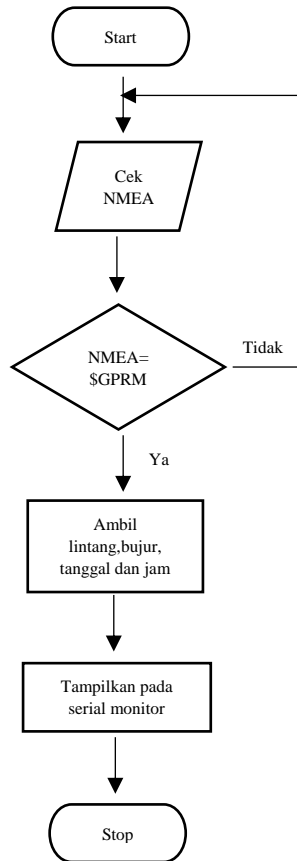
3.1.2 Mikrokontroler

Mikrokontroler yang digunakan pada alat ini adalah wemos d1 mini . wemos d1 mini berfungsi untuk memproses dan mengolah data GPS dan kompas digital sehingga menghasilkan arah kiblat serta jadwal shalat.

3.1.3 GPS

GPS ini berfungsi untuk melacak posisi koordinat lintang (*latitude*), bujur (*Longitude*), waktu (*time*) dan tanggal (*Date*) yang digunakan oleh mikrokontroler untuk memperoleh arah kiblat dari perhitungan rumus

kiblat dan jadwal shalat. Informasi data GPS tersebut diperoleh dari hasil data NMEA (Gambar 3.3) *Output* data GPS berupa text.



Gambar 3. 3 *Flowchart* Data GPS

3.1.4 HMC5883L

HMC5883L merupakan kompas digital pada proyek akhir ini digunakan sebagai pembanding arah dari perhitungan rumus kiblat. Dari perbandingan arah perhitungan dan data kompas inilah diperoleh arah kiblat dari posisi pengguna.

3.1.5 OLED 1.3"

OLED ini berfungsi sebagai tampilan *output* dari sistem, yaitu berupa data arah kompas, arah kiblat, dan jadwal shalat.

3.1.6 Buzzer

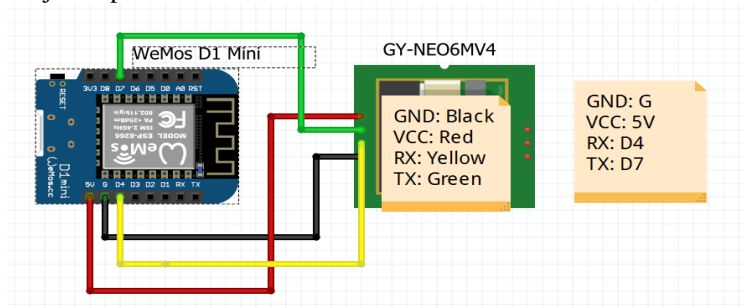
Buzzer berfungsi sebagai *output* pengingat dari jadwal shalat 5 waktu

3.2 Konfigurasi Elektronik

Konfigurasi Elektronik secara keseluruhan menggunakan komponen wemos d1 mini, HMC5883L, Modul GPS, OLED 1.3", *push button* dan battery 3,7V.

3.2.1 Konfigurasi Wemos D1 Mini dengan Modul GPS

Berikut adalah konfigurasi wemos d1 mini dengan modul GPS yang ditunjukkan pada Gambar 3.4.



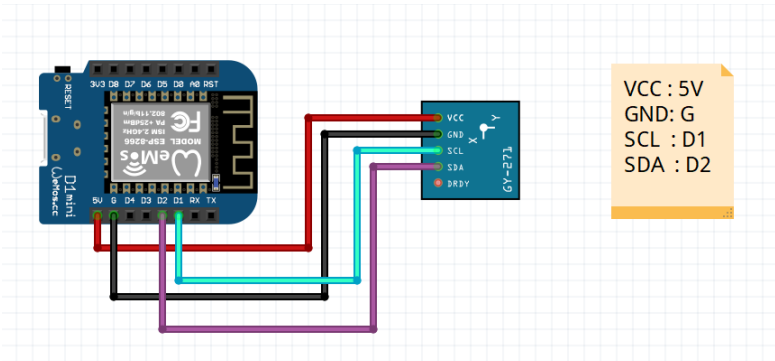
Gambar 3. 4 Konfigurasi Wemos D1 Mini dengan Modul GPS

Pin yang digunakan modul GPS pada wemos d1 mini adalah:

1. Pin VCC dihubungkan dengan 5 V
2. Pin GND dihubungkan dengan G
3. Pin RX dihubungkan dengan D4
4. Pin TX dihubungkan dengan D7

3.2.2 Konfigurasi Wemos D1 Mini dengan HMC5883L

Berikut adalah konfigurasi wemos d1 mini dengan HMC5883L yang ditunjukkan pada Gambar 3.5.



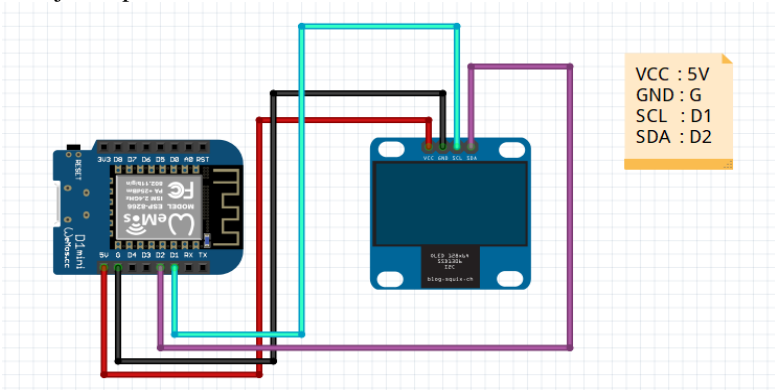
Gambar 3. 5 Konfigurasi Wemos D1 Mini dengan HMC5883L

Pin yang digunakan HMC5883L pada wemos d1 mini adalah:

1. Pin VCC dihubungkan dengan 5 V
2. Pin GND dihubungkan dengan G
3. Pin SCL dihubungkan dengan D1
4. Pin SDA dihubungkan dengan D2

3.2.3 Konfigurasi Wemos D1 Mini dengan OLED 1.3''

Berikut adalah konfigurasi wemos d1 mini dengan OLED 1.3'' yang ditunjukkan pada Gambar 3.6.



Gambar 3. 6 Konfigurasi Wemos D1 Mini dengan OLED 1.3''

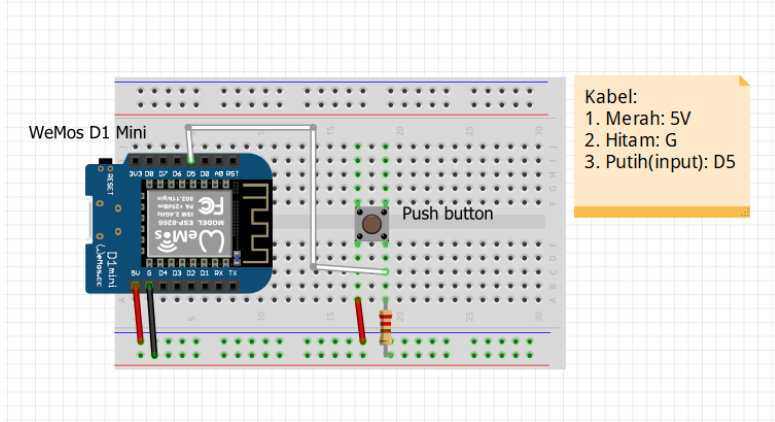
Pin yang digunakan OLED 1.3'' pada wemos d1 mini adalah:

1. Pin VCC dihubungkan dengan 5 V
2. Pin GND dihubungkan dengan G

3. Pin SCL dihubungkan dengan D1
4. Pin SDA dihubungkan dengan D2

3.2.4 Konfigurasi Wemos D1 Mini dengan *Push Button*

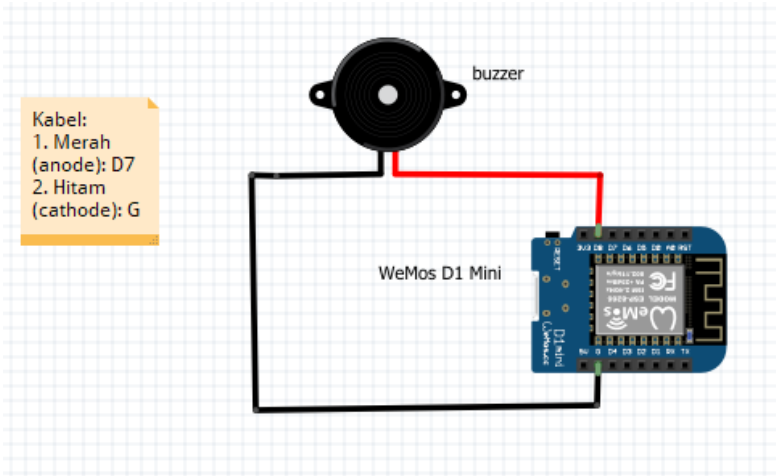
Pada perancangan ini *Push button* digunakan untuk mengganti tampilan informasi pada OLED, Pin yang digunakan pada *push button* adalah D5. Gambar berikut adalah gambar rangkaian *push button* yang dikoneksikan dengan wemos d1 mini yang ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Konfigurasi Wemos D1 Mini dengan *Push Button*

3.2.5 Konfigurasi Wemos D1 Mini dengan *Buzzer*

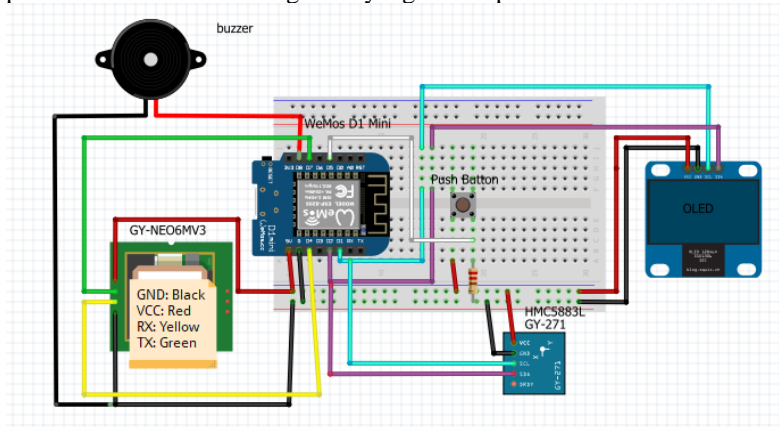
Pada perancangan ini *buzzer* digunakan untuk pengingat waktu shalat, Pin yang digunakan pada *push button* adalah D8. Gambar berikut adalah gambar rangkaian *buzzer* yang dikoneksikan dengan wemos d1 mini yang ditunjukkan pada Gambar 3.8.



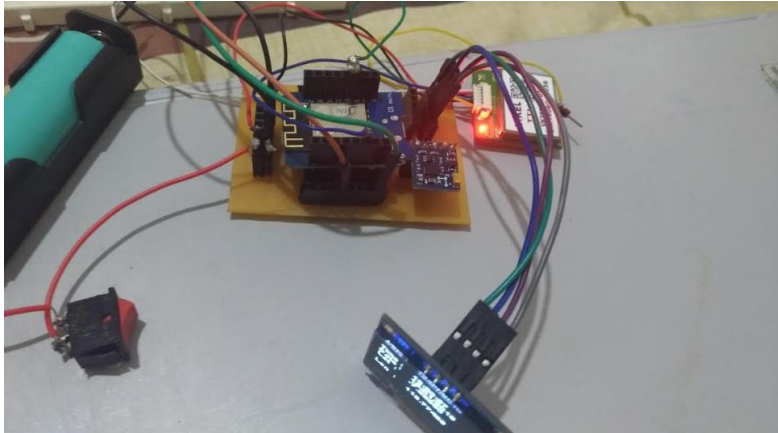
Gambar 3. 8 Konfigurasi Wemos D1 Mini dengan Buzzer

3.2.6 Konfigurasi Rangkaian Alat Secara Keseluruhan

Berikut adalah rangkaian alat secara keseluruhan yang ditunjukkan pada Gambar 3.9 serta rangkaian yang dibuat pada Gambar 3.10.



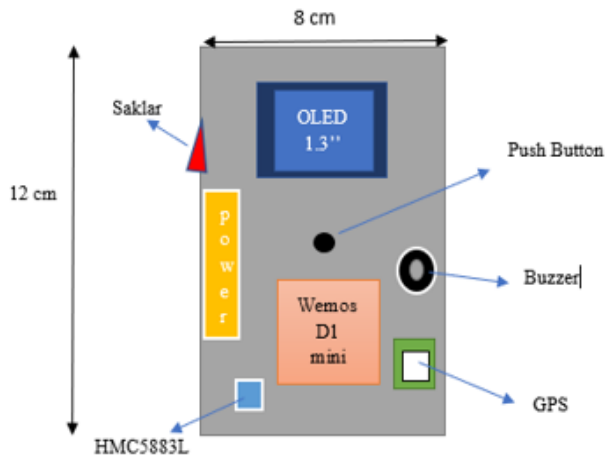
Gambar 3. 9 Konfigurasi Rangkaian Alat Secara Keseluruhan



Gambar 3. 10 Hasil Dari Rangkaian yang Dibuat

3.3 Konfigurasi Mekanik

Setelah konfigurasi rangkaian alat secara keseluruhan telah selesai selanjutnya rangkaian elektronik di *packing* dengan menggunakan *box project* elektronika. *Box project* elektronika menggunakan bahan plastik dengan dimensi panjang sebesar 12 cm, lebar 8 cm dan tinggi 5 cm. Berikut adalah gambar dari perancangan alat penunjuk arah kiblat dan jadwal shalat yang ditunjukkan pada Gambar 3.11 dan alat jadi yang ditunjukkan pada Gambar 3.12.



Gambar 3. 11 Konfigurasi Mekanik



Gambar 3. 12 Rangkaian Elektronik yang Telah di *Packing* (Alat Jadi)

Setelah semua rangkaian telah ter-*packing* alat dapat dicoba dengan menekan saklar untuk menyalakan alat, setelah itu muncul informasi yang ada pada OLED 1.3". Informasi yang ditunjukkan terdiri dari 3 *display*. Informasi *display* yang pertama adalah jam, tanggal serta bujur dan lintang berupa desimal serta *DMS (Degree Minute Second)*. Pada informasi *display* kedua adalah jadwal shalat. Pada *display* ketiga memberikan informasi penunjuka arah kiblat. Masing-masing tampilan *display* tersebut ditunjukkan pada Gambar 3.13.



Gambar 3. 13 Tampilan Informasi Masing-masing *Display*

3.4 Cara Penggunaan Alat

Untuk menggunakan alat ini mula-mula tekan saklar yang letaknya terdapat pada samping alat selanjutnya *display* akan menyala berikut tampilan awal alat yang ditunjukkan pada Gambar 3.14.



Gambar 3. 14 Tampilan Awal Alat Pada Saat Alat Dihidupkan

Selanjutnya setelah alat dinyalakan tunggu sampai GPS menerima data. Untuk mengecek GPS telah menerima data maka tampilan layar akan berubah mulai dari jam, tanggal, lintang dan bujur sesuai dengan Gambar 3.15.



Gambar 3. 15 Tampilan Pada Saat Alat Telah Menerima Data GPS

Setelah alat menerima data dari GPS maka alat telah bisa digunakan. Untuk mengganti tampilan layar tekan tombol yang terletak persis pada bagian bawah layar. Setelah menekan tombol didapati informasi jadwal shalat yang sesuai pada Gambar 3.16.



Gambar 3. 16 Tampilan Jadwal Shalat

Untuk mengetahui informasi arah kiblat tekan kembali tombol maka akan ditunjukkan tampilan arah kiblat, berikut adalah tampilan awal informasi arah kiblat yang ditunjukkan pada Gambar 3.17



Gambar 3. 17 Tampilan Awal Arah kiblat

Pada Gambar 3.17 merupakan contoh arah kiblat yang belum disesuaikan. Dimana jarum indikator belum menunjukkan tepat pada titik bantu serta nilai *azimuth* tidak sesuai dengan nilai arah kiblat. Untuk mendapatkan arah kiblat yaitu dengan cara memutar alat tersebut dan posisikan alat secara horizontal dengan posisi alat untuk bagian depan (indikator penunjuk arah) yang ditunjukkan pada Gambar 3.18, sampai didapati nilai *azimuth* kompas sesuai dengan nilai arah kiblat serta jarum indikator tepat pada titik bantu yang ditunjukkan pada Gambar 3.19



Gambar 3. 18 Posisi Depan Alat

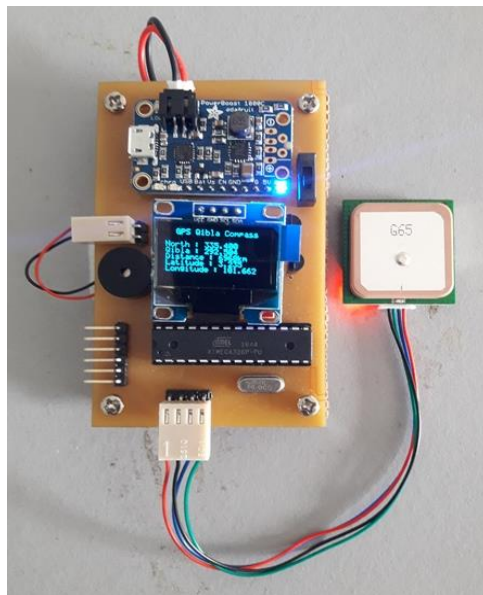


Gambar 3. 19 Tampilan Arah Kiblat yang Telah Disesuaikan

Untuk mengulang tampilan informasi tekan kembali tombol begitupun seterusnya.

3.5 Perbandingan Alat Sejenis dengan Alat Proyek Akhir

Berikut merupakan gambaran alat sejenis dengan alat proyek akhir yang telah dibuat yang ditunjukkan pada Gambar 3.20.



Gambar 3. 20 Alat Sejenis [6]

Pada alat tersebut cara penggunaan untuk menentukan arah kiblat sama yaitu dengan cara diputar dengan letak secara horizontal perbedaannya indikator pada saat sesuai dengan arah kiblat dengan menggunakan

buzzer serta tampilan informasi pada *display* yaitu nilai *azimuth* kompas, nilai *azimuth* kiblat, keinggian serta bujur dan lintang (desimal).

3.5.1 Kelebihan Alat Proyek Akhir dengan Alat Sejenis

Berikut merupakan kelebihan dari Alat proyek akhir yang dibuat dengan alat sejenis:

- Dalam bentuk koordinat (lintang dan bujur) pada alat sejenis hanya terdapat pada bentuk konversi desimal saja sedangkan pada alat proyek akhir dalam bentuk desimal serta *DMS (Degree Minute Second)*.
- Untuk Arah kiblat pada alat proyek akhir yang dibuat menggunakan indikator visual dalam bentuk kompas sehingga lebih mudah pengguna dalam menentukan arah kiblat.
- Pada alat proyek akhir informasi yang ditampilkan pada *display* lebih banyak yaitu berupa tanggal, jam serta terdapat jadwal shalat dengan pengingat berupa *buzzer*.

BAB IV PENGUJIAN DAN ANALISA

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah sesuai dengan yang diharapkan atau tidak, maka perlu dilakukan pengujian dan analisa berdasarkan perencanaan dari sistem yang telah dibuat.

Pengambilan data ini dilakukan untuk mengetahui seberapa efektif *software* dan *hardware* yang telah dibuat sehingga alat ini dapat bekerja sesuai dengan harapan.

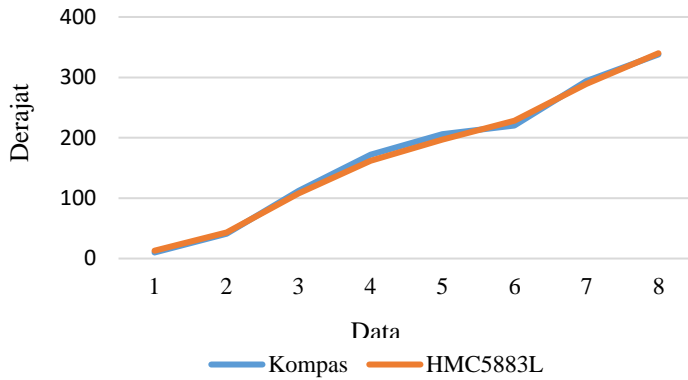
Metode pengujian alat Proyek Akhir yang dilakukan berupa Perbandingan data HMC5883L dengan kompas, perbandingan hasil data koordinat (bujur dan lintang) yang diperoleh oleh alat dengan aplikasi *google maps* serta perbandingan jadwal shalat yang dihasilkan alat dengan jadwal yang dikeluarkan oleh KEMENAG RI dan KEMENAG Surabaya.

4.1 Pengujian Resolusi Kompas

Pengujian resolusi kompas ini dilakukan untuk mengetahui kepresisian arah dan kompas HMC5883L dibandingkan dengan arah dari kompas dan data HMC5883L dengan membandingkan pada di dalam ruangan serta di luar ruangan ditunjukkan pada Tabel 4.1 dan 4.2 .

Tabel 4. 1 Resolusi Kompas di Dalam Ruangan

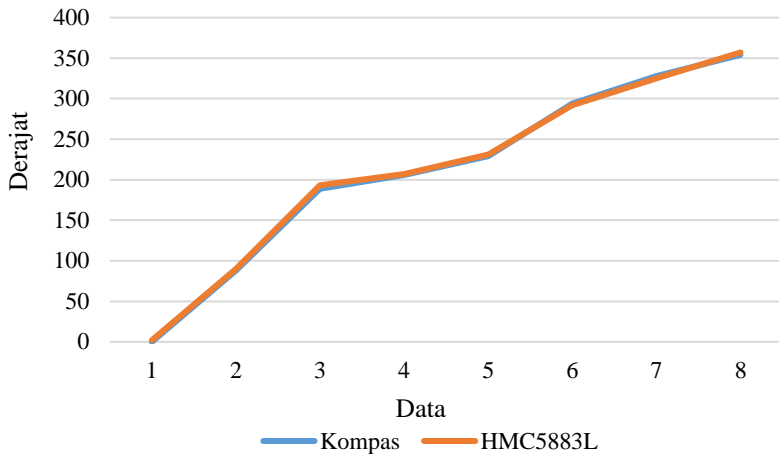
No	Kompas	HMC5833L	<i>Error</i>
1	10°	13°	3°
2	41°	43°	4°
3	112°	108°	4°
4	172°	162°	10°
5	206°	197°	11°
6	220°	228°	8°
7	294°	289°	5°
8	338°	340°	2°
	Rata-rata <i>error</i>		5,875°



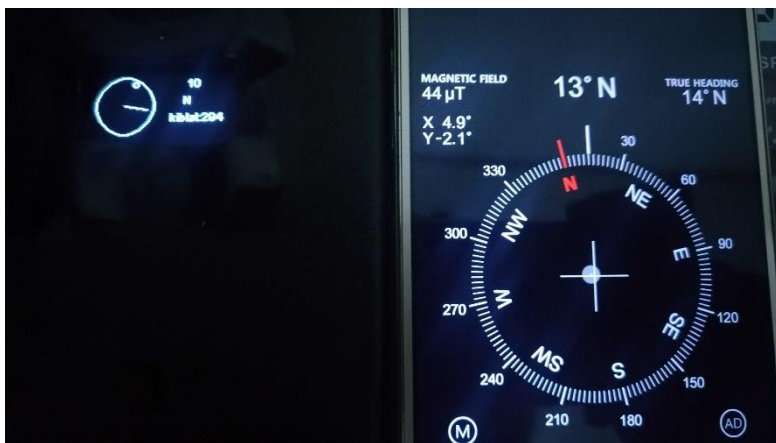
Gambar 4. 1 Perbandingan Resolusi Kompas di Dalam Ruangan

Tabel 4. 2 Resolusi Kompas di Luar Ruangan

No	Kompas	HMC5883L	<i>Error</i>
1	0°	2°	2°
2	88°	90°	2°
3	189°	193°	4°
4	206°	207°	1°
5	229°	231°	2°
6	294°	292°	2°
7	328°	325°	3°
8	354°	357°	3°
	Rata-rata <i>error</i>		2,375°



Gambar 4. 2 Perbandingan Resolusi Kompas di Luar Ruang



Gambar 4. 3 Perbandingan Resolusi Kompas dengan Alat

Dari Gambar 4.1 dan 4.2, dapat dianalisa linearitas data terhadap sudut arah. Untuk besarnya *error* data dari pengujian perbandingan kompas dengan kompas HMC5883L. untuk rata-rata *error* didapatkan dengan menggunakan rumus :

$$\text{Rata - rata error} = \left[\frac{\text{Jumlah selisih seluruh nilai}}{\text{jumlah data yang didapat}} \right] \quad (4.1)$$

Dari Tabel 4.1 dan 4.2 bisa dilihat nilai *error* rata-rata yang dihasilkan pada Tabel 4.2 lebih kecil daripada Tabel 4.1. Ini disebabkan karena sifat sensor HMC5883L sensitif dengan benda disekitar, sehingga bila terdapat benda yang menghalangi sensor HMC5883L hasil yang didapatkan akan tidak maksimal.

Perlu diketahui bahwa sensor HMC5883L bersifat magnetoresisive yang sangat sensitif sekali dengan medan magnet. Sehingga dalam penggunaannya harus dijauhkan dari benda-benda yang menimbulkan medan magnet, seperti kabel bertegangan tinggi, benda-benda dari bahan besi selain itu agar mendapatkan hasil yang baik sebaiknya dalam penggunaan sensor ini dilakukan di tempat terbuka (tempat yang lapang). Selain faktor lingkungan dalam penggunaan alat sebisa mungkin dalam keadaan datar karena mempengaruhi nilai *azimuth*. Berikut merupakan contoh penggunaan alat pada posisi datar yang ditunjukkan pada Gambar 4.4 dan penggunaan alat pada posisi yang salah yang ditunjukkan Pada Gambar 4.5 serta perbandingan hasil ujicoba dengan kompas serta kondisi alat saat digunakan posisi datar dan posisi yang salah.



Gambar 4.4 Penggunaan Alat Pada Posisi Datar



Gambar 4. 5 Penggunaan Alat Pada Posisi yang Salah

Tabel 4. 3 Perbandingan Hasil Alat Pada Posisi Datar dan Salah

No	Nilai Kompas	Alat Posisi Datar	Alat Posisi Salah	Selisih Posisi Datar	Selisih Posisi Salah
1	17°	14°	5°	3°	12°
2	47°	46°	32°	1°	15°
3	96°	93°	82°	3°	14°
4	132°	129°	106°	3°	26°
5	136°	133°	110°	3°	26°
6	201°	199°	215°	2°	14°
7	254°	250°	271°	4°	17°
8	310°	309°	314°	1°	4°
<i>Error rata-rata</i>				2,5°	16°

Dari *error* rata-rata pada Tabel 4.3 bisa dilihat bahwa nilai *error* posisi datar lebih kecil daripada posisi yang salah. Sehingga bisa disimpulkan bahwa agar mendapatkan nilai *azimuth* terbaik sebisa mungkin alat diletakan pada posisi datar.

4.2 Pengujian Koordinat dan Akuisisi Modul GPS

Pengujian koordinat GPS dilakukan untuk mengetahui keakuratan dari modul GPS itu sendiri serta waktu yang diperlukan modul GPS untuk mendapatkan informasi melalui satelit.

4.2.1 Pengujian Koordinat

Untuk pengujian koordinat adalah membandingkan data yang ada pada modul GPS dengan aplikasi *google maps* terdapat pada Tabel 4.4.

Tabel 4. 4 Pengujian Koordinat

No	Lokasi	lintang(Google Maps)	bujur(Google Maps)	lintang(modul GPS)	bujur(modul GPS)
1	Kenpark surabaya	-7,25167°	112,80356°	-7,25283°	112,79598°
2	rumah penulis	-7,28547°	112,77427°	-7,28551°	112,77433°
3	halaman gedung robotika ITS	-7,27790°	112,79753°	-7,27819°	112,79782°
4	lapangan taman alumni ITS	-7,28079°	122,79197°	-7,28102°	122,79201°
5	Gebang putih	-7,28310°	112,78540°	-7,28325°	112,78576°
6	Hutan bambu keputih	-7,29385°	112,80233°	-7,29403°	112,80244°
7	Taman Flora	-7,29492°	112,76214°	-7,29451°	112,76236°
8	Unair kampus C	-7,26999°	112,78267°	-7,27064°	112,78295°

Dari Tabel 4.4 di atas menunjukkan sedikit perbedaan data antara modul GPS dengan *google maps*. Ini disebabkan salah satunya adalah penerimaan lokasi pengguna dengan GPS. Walaupun begitu data tersebut masih batas toleransi sehingga modul GPS masih layak digunakan dalam menentukan arah kiblat dan jadwal shalat.

4.2.2 Pengujian Akuisisi GPS

Modul GPS pada pengujian ini akan diaktifkan dengan dua kondisi, yaitu *cold start* dan *hot start*. Pengujian dilakukan di beberapa lokasi dengan beberapa kondisi tempat yang berbeda, yang ditunjukkan pada Tabel 4.5.

Tabel 4. 5 Pengujian Akuisisi GPS

No	Tempat	<i>Cold Start</i> (dtk)	<i>Hot Start</i> (dtk)
1	Kenpark Surabaya	217	8
2	Rumah Penulis	486	7
3	Halaman Gedung Robotika ITS	234	7

No	Tempat	<i>Cold Start</i> (dtk)	<i>Hot Start</i> (dtk)
4	Lapangan Taman Alumni ITS	246	8
5	Gebang Putih	312	7
6	Hutan Bambu Keputih	306	7
7	Taman Flora	281	7
8	Unair Kampus C	251	8
	Rata-rata	291,625	7,375

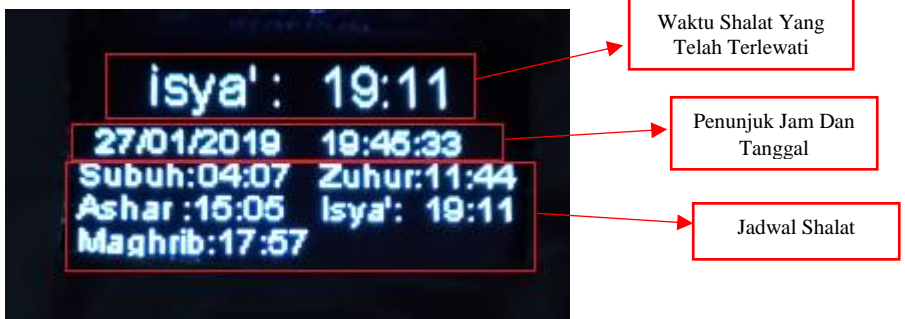
Dari Tabel 4.5 bisa dilihat bahwa GPS dalam keadaan *cold start* memerlukan waktu yang lebih lama daripada *hot start*. Ini disebabkan oleh GPS yang *start* mulai dari keadaan mati yang lama akan memerlukan waktu lagi untuk mendapatkan sinyal dari satelit daripada keadaan *hot start* yang telah load data dari satelit. Selain itu, jika diperhatikan pengujian di atas terdapat perbedaan waktu yang signifikan, misalnya antara di area rumah penulis. Ini disebabkan oleh faktor lingkungan tempat tinggal dari penulis dimana terdapat rumah-rumah berjejeran dan atap bangunan akan menghalangi GPS untuk mendapatkan transmisi satelit, sehingga mengurangi akurasi dan kecepatan penentuan lokasi. Dan juga bisa dilihat untuk taman alumni ITS didapati lebih cepat daripada wilayah yang lain, ini disebabkan karena pada lapangan taman alumni ITS transmisi dari satelit ke modul GPS tidak ada suatu halangan. Dari pengujian ini dapat diketahui tentang akuisisi GPS, diantaranya:

- penggunaan GPS pada kondisi *cold start* memerlukan waktu yang lebih lama dalam mendapatkan sinyal dari satelit.
- penggunaan GPS di area perumahan akan membutuhkan waktu yang lebih lama dari pada tempat yang lapang, ini disebabkan dalam mendapatkan sinyal GPS karena terdapat rumah-rumah yang berjejeran sehingga menghalangi GPS untuk mendapatkan transmisi satelit.

4.3 Pengujian Seluruh Sistem

Pengujian ini dilakukan untuk mengetahui kestabilan dari alat yang telah dibuat. Pengujian ini melibatkan seluruh komponen yang telah dilakukan pengujian sebelumnya. Pengujian ini dilakukan pada tempat yang sama dari uji coba alat sebelumnya. Berikut Gambar 4.4 tampilan

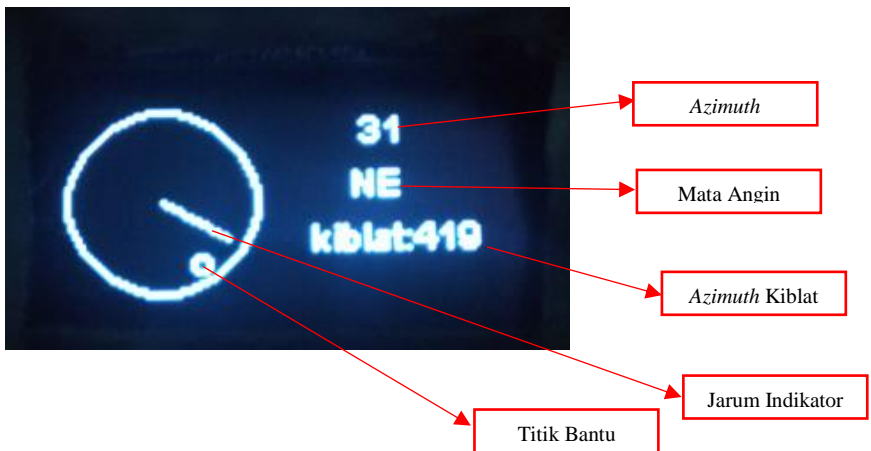
jadwal shalat dan Gambar 4.5 tampilan kompas penunjuk kiblat pada OLED



Gambar 4. 6 Tampilan Jadwal Shalat

Keterangan Gambar:

- Waktu shalat yang telah terlewati bertujuan untuk pengingat pengguna sudah atau belum melaksanakan shalat tersebut.
- Penunjuk jam dan tanggal berfungsi mencocokkan waktu dengan jadwal shalat.
- Jadwal shalat merupakan penunjuk waktu shalat 5 waktu



Gambar 4. 7 Tampilan Kompas Penunjuk Arah Kiblat

Keterangan Gambar:

- *Azimuth* arah kompas merupakan hasil *azimuth* dari kompas digital HMC5883L.
- *Azimuth* kiblat merupakan data hasil dari perhitungan rumus kiblat dari data lintang dan bujur yang diterima GPS
- Indikator arah mata angin merupakan mata angin yang ditunjukkan pada jarum indikator kompas.
- Jarum indikator kompas merupakan jarum penunjuk mata angin serta data *azimuth*
- Titik bantu berfungsi sebagai pembantu pengguna untuk menentukan arah kiblat

Berikut adalah pengujian arah kiblat alat pada kondisi dalam ruangan serta luar ruangan yang ditunjukkan pada Tabel 4.6 serta rata-rata *error* yang ditunjukkan pada Tabel 4.7.

Tabel 4. 6 Pengujian Arah Kiblat

No	Lokasi	Qibla Map(a)	Hasil Rumus Alat(b)	Alat di Dalam Ruangan(c)	Alat di Luar Ruangan(d)
1	Kenpark Surabaya	293,0221°	294,0241°	289°	292°
2	Rumah Penulis	293,9139°	294,0356°	289°	292°
3	Halaman Gedung Robotika ITS	293,9072°	294,0289°	289°	292°
4	Lapangan Taman Alumni ITS	293,9091°	294,0307°	289°	292°
5	Gebang Putih	293,9105°	294,0385°	289°	292°
6	Hutan Bambu Keputih	293,9096°	294,0311°	289°	292°
7	Taman Flora	293,9185°	294,0401°	289°	292°
8	Unair Kampus C	293,9088°	294,0305°	289°	292°

Tabel 4. 7 Perbandingan *Error* Arah Kiblat

No	Lokasi	<i>Error</i> di Dalam Ruangan (%)	<i>Error</i> di Luar Ruangan(%)
1	Kenpark Surabaya	1,3726	0,3488
2	Rumah Penulis	1,6719	0,6512
3	Halaman Gedung Robotika ITS	1,6696	0,6489

No	Lokasi	<i>Error</i> di Dalam Ruang (%)	<i>Error</i> di Luar Ruang (%)
4	Lapangan Taman Alumni ITS	1,6703	0,6496
5	Gebang Putih	1,6707	0,6500
6	Hutan Bambu Keputih	1,6704	0,6497
7	Taman Flora	1,6734	0,6527
8	Unair Kampus C	1,6702	0,6495
	Rata-rata <i>error</i> (%)	1,6337	0,6125

Untuk nilai *error* di dalam dan di luar ruangan didapatkan dengan menggunakan rumus :

$$\text{error di dalam ruangan}(\%) = \frac{(a - c)}{a} \times 100 \quad (4.2)$$

$$\text{error di luar ruangan}(\%) = \frac{(a - d)}{a} \times 100 \quad (4.3)$$

Dari Tabel 4.7 bahwa nilai *error* yang dihasilkan oleh alat dan Qibla Map untuk pengujian didalam ruangan dihasilkan rata-rata *error* sebesar 1,6337% sedangkan bila dilakukan pengujian diluar ruangan dihasilkan rata-rata *error* sebesar 0,6125%. Untuk selisih nilai alat dengan Qibla Map yang ditunjukkan pada Tabel 4.6 yang dihasilkan di luar ruangan sebesar kurang dari 2° dianggap wajar, hal ini disebabkan pada *datasheet* HMC5883L toleransinya pada batas 1-2°. Sedangkan selisih nilai yang dihasilkan di dalam ruangan didapati lebih dari batas toleransi yaitu sebesar kurang dari 5°, sehingga bisa disimpulkan dalam penggunaan sensor ini sebisa mungkin dilakukan diruangan terbuka, agar didapati nilai yang baik, karena sensor ini sensitif terhadap kondisi di lingkungan sekitar, sehingga bila terdapat benda yang menghalangi contohnya seperti tembok akan berpengaruh terhadap nilai yang didapati sensor HMC5883L.

Untuk pengujian selanjutnya yaitu perbandingan jadwal shalat Antara alat dengan jadwal shalat yang dikeluarkan oleh KEMENAG RI dan KEMENAG Surabaya pada tanggal 13 Januari 2019 yang ditunjukkan pada Tabel 4.8.

Tabel 4. 8 Perbandingan Jadwal Shalat

No	sholat	alat	KEMENAG RI	KEMENAG Surabaya
1	subuh	4:02	4:02	4:03
2	zuhur	11:42	11:42	11:43
3	ashar	15:05	15:05	15:06
4	maghrib	17:56	17:56	17:56
5	isya'	19:11	19:11	19:11

Dari Tabel 4.8 bisa dilihat hasil yang didapat dari jadwal shalat untuk wilayah surabaya sama dengan jadwal shalat wilayah surabaya yang dikeluarkan oleh KEMENAG RI. Sedangkan perbandingan pada jadwal shalat untuk wilayah daerah surabaya yang dikeluarkan oleh KEMENAG Surabaya memiliki perbedaan 1 menit untuk jadwal shalat subuh, zuhur dan ashar. Kemungkinan disebabkan penerimaan data koordinat yang diterima oleh KEMENAG Surabaya dengan lokasi penulis berbeda sehingga mempengaruhi perhitungan.

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

Bab penutup berisi tentang kesimpulan yang didapatkan selama proses pembuatan Proyek Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa yang dilakukan pada Proyek Akhir Penunjuk Arah Kiblat Dan Waktu Shalat Menggunakan *Global Positioning System* (GPS) Berbasis Wemos D1 Mini diperoleh beberapa kesimpulan antara lain:

- a. Berdasarkan pengujian alat mengenai resolusi kompas bisa disimpulkan bahwa penggunaan alat sebaik mungkin dilakukan di luar ruangan agar didapati nilai *azimuth* yang mendekati nilai dari kompas, karena pada saat didalam ruangan kompas terhalangi oleh benda disekitar contohnya tembok, serta dalam penggunaan alat bisa disimpulkan bahwa pada penggunaan HMC5883L yang tepat adalah dengan posisi datar agar didapati nilai *azimuth* yang mendekati dengan nilai kompas.
- b. Berdasarkan pengujian akuisisi GPS penggunaan GPS pada kondisi area gedung-gedung tinggi, terdapat pepohonan disekitar pengguna serta di dalam ruangan membutuhkan waktu yang lebih lama dari pada tempat yang lapang dalam mendapatkan sinyal satelit, ini disebabkan pada lingkungan tersebut menghalangi transmisi satelit ke modul GPS, sehingga dalam mendapatkan data GPS akan lebih lama.
- c. Berdasarkan pengujian perbandingan antara arah kiblat dari alat dan arah kiblat dari *software* Qibla Map, perbedaan yang dihasilkan untuk pengujian di dalam ruangan dihasilkan nilai rata-rata *error* sebesar 1,6337% sedangkan pada pengujian di luar ruangan dihasilkan nilai rata-rata *error* sebesar 0,6125%.

5.2 Saran

Berikut ini adalah saran-saran yang dapat diterapkan untuk pengembangan dan aplikasi lebih lanjut dari Proyek Akhir ini:

- a. Sebaiknya modul GPS di lakukan di tempat terbuka (tempat yang lapang) agar lebih cepat mendapatkan transmisi data dari satelit ke GPS *receiver*.

- b. Pada saat memakai sensor HMC5883L sebaiknya pada tempat yang bebas dari benda-benda yang menimbulkan medan magnet serta untuk mendapatkan hasil yang maksimal dilakukan di tempat yang terbuka.

DAFTAR PUSTAKA

- [1] Andrianto, Heri dan Aan Darmawan. **Arduino Belajar Cepat dan Pemograman**, Bandung :Informatika Bandung, 2016
- [2] Henry's Bench, **Arduino GY-273 HMC5883L Magnetometer Compass**[internet], Tersedia:<http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-gy-273-hmc5883l-magnetometer-compass>[diakses 8 desember 2018]
- [3] Abidin, Zainal Hasanuddin, **Penentuan Posisi dengan GPS dan Aplikasinya**, Jakarta: Pradnya Paramita, 2002
- [4] J. Iswanto, **Penggunaan Segitiga Bola Dalam Menentukan Rumus Arah Kiblat (Pada Pendalaman Materi Diklat Hisab Rukyat)**, Surabaya: Balai Diklat Keagamaan, 2010.
- [5] Paulus Andi Nalwan, **Panduan Praktis Penggunaan dan Antarmuka Modul LCD**, Jakarta: PT Elex Media Komputindo, 2014
- [6] Nizam, shahrul, **Arduino Project: GPS Kompas Kiblat (QiblaCompass)**[internet], Tersedia:<http://shahrulnizam.com/arduino-project-gps-kompas-kiblat/>[diakses 26 Januari 2019]
- [7] Era Muslim, **Segitiga Bola dan Arah Kiblat**[internet], Tersedia:<https://www.eramuslim.com/peradaban/ilmu-hisab/segitiga-bola-dan-arah-kiblat/>[diakses 26 Januari 2019]
- [8] Bagus Hari Sasongko, **Pemrograman Mikrokontroler dengan Bahasa C**, Yogyakarta: Andi, 2012

Halaman ini sengaja dikosongkan

LAMPIRAN

Pengujian Alat





Program Address I2C

```
#include <Wire.h>  
const int sclPin = D1;  
const int sdaPin = D2;
```

```

void setup()
{
  Wire.begin(sdaPin, sclPin);

  Serial.begin(9600);
  Serial.println("I2C Scanner");
}
void loop()
{
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for (address = 1; address < 127; address++)
  {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address < 16) {
        Serial.print("0");
      }
      Serial.print(address, HEX);
      Serial.println(" !");
      nDevices++;
    }
    else if (error == 4)
    {
      Serial.print("Unknown error at address 0x");
      if (address < 16) {
        Serial.print("0");
      }
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0) {

```

```

    Serial.println("No I2C devices found\n");
}
else {
    Serial.println("Done.\n");
}
delay(2000);
}

```

Program Alat

```

#include <TimeLib.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include "PrayerTimes.h"
#include <Wire.h>
#include "SH1106Wire.h",
#include "SH1106.h"
#include<math.h>
#include <MechaQMC5883.h>
MechaQMC5883 qmc;
const int UTC_offset = +7;
SH1106 display(0x3c, D2, D1);
SoftwareSerial Serial_GPS(D7, D4);
TinyGPSPlus gps;
String twoDigits(int digits){if(digits < 10) {String i = '0'+String(digits);
return i;} else {return String(digits);}}
static void smartDelay(unsigned long ms){unsigned long start = millis();
do {while (Serial_GPS.available()) gps.encode(Serial_GPS.read());}
while (millis() - start < ms);}
double times[sizeof(TimeName)/sizeof(char*)];
int dst=7;
int jam, menit, detik, tahun, bulan, tanggal;
String cetak_jam;
String cetak_menit;
String cetak_detik;
String jam_sekarang;
String jam_sholat;
int a = 0; int b = 0; int c = 0;
int stringStart, stringStop = 0;
String wkt_sbh;
String wkt_zhr;

```

```

String wkt_asr;
String wkt_mgr;
String wkt_isy;
String wkt_sbhjam;
String wkt_sbhmenit;
String wkt_zhrjam;
String wkt_zhrmenit;
String wkt_asrjam;
String wkt_asrmenit;
String wkt_mgrjam;
String wkt_mgrmenit;
String wkt_isyjam;
String wkt_isyamenit;
int ambil_jam_sholat;
int ambil_menit_sholat;
int satuan_jam_sholat;
int puluhan_jam_sholat;
int satuan_menit_sholat;
int puluhan_menit_sholat;
int i = 0;
float u,LK,BK;
int angle;
int WhichScreen =1; // This variable stores the current Screen number
boolean hasChanged = true;
const int buttonPin = D5;
int buttonState;
int lastButtonState = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50; //
int mod =1;
const int buzzer = D8;
char degSymbol[3] = { 0xC2, 0xB0, 0x00 };
void setup() {
  Serial.begin(9600);
  Serial_GPS.begin(9600);
  display.init();
  display.flipScreenVertically();
  pinMode(buttonPin, INPUT);
  pinMode(buzzer, OUTPUT);
Wire.begin();
qmc.init();

```

```

digitalWrite(buttonPin, HIGH);
}
void loop() {
  alarm();
  smartDelay(1000);
  jam = gps.time.hour();
  menit = gps.time.minute();
  detik = gps.time.second();
  tahun = gps.date.year();
  bulan = gps.date.month();
  tanggal = gps.date.day();
  set_calc_method(MWL);
  set_asr_method(Shafii);
  set_high_lats_adjust_method(AngleBased);
  set_fajr_angle(20);
  set_isha_angle(18);
  get_prayer_times(gps.date.year(),
gps.date.month(),gps.date.day(),gps.location.lat(), gps.location.lng(), dst,
times);
ambil_jadwal_sholat(0);
wkt_sbh = String(puluhan_jam_sholat) + String(satuan_jam_sholat) + ":"
+ String(puluhan_menit_sholat) + String(satuan_menit_sholat);
wkt_sbhjam=String(puluhan_jam_sholat) + String(satuan_jam_sholat);
wkt_sbhmenit=String(puluhan_menit_sholat)
+
String(satuan_menit_sholat);
ambil_jadwal_sholat(2);
wkt_zhr = String(puluhan_jam_sholat) + String(satuan_jam_sholat) + ":"
+ String(puluhan_menit_sholat) + String(satuan_menit_sholat);
wkt_zhrjam=String(puluhan_jam_sholat) + String(satuan_jam_sholat);
wkt_zhrmenit=String(puluhan_menit_sholat)+
String(satuan_menit_sholat);
ambil_jadwal_sholat(3);
wkt_asr = String(puluhan_jam_sholat) + String(satuan_jam_sholat) + ":"
+ String(puluhan_menit_sholat) + String(satuan_menit_sholat);
wkt_asrjam=String(puluhan_jam_sholat) + String(satuan_jam_sholat);
wkt_asrmenit=String(puluhan_menit_sholat)
+
String(satuan_menit_sholat);
ambil_jadwal_sholat(5);
wkt_mgr = String(puluhan_jam_sholat) + String(satuan_jam_sholat) +
":" + String(puluhan_menit_sholat) + String(satuan_menit_sholat);
wkt_mgrjam=String(puluhan_jam_sholat) + String(satuan_jam_sholat);

```



```

wkt_mgrmenit=String(puluhan_menit_sholat)          +
String(satuan_menit_sholat);
ambil_jadwal_sholat(6);
wkt_isy = String(puluhan_jam_sholat) + String(satuan_jam_sholat) + ":"
+ String(puluhan_menit_sholat) + String(satuan_menit_sholat);
wkt_isyjam=String(puluhan_jam_sholat) + String(satuan_jam_sholat);
wkt_isymenit=String(puluhan_menit_sholat)          +
String(satuan_menit_sholat);
if(jam<10){ cetak_jam = "0" + String(jam); }
else { cetak_jam = String(jam); }
if(menit<10){ cetak_menit = "0" + String(menit); }
else { cetak_menit = String(menit); }
if(detik<10){ cetak_detik = "0" + String(detik); }
else { cetak_detik = String(detik); }
jam_sholat = "Subuh : " + wkt_sbh + " - Dzuhur : " + wkt_zhr + " - Ashar
: " + wkt_asr + " - Maghrib : " + wkt_mgr + " - Isya : " + wkt_isy + " ";
i++;
if (i>=1 && i<20){
stringStart=0;
stringStop=13;
}
if (i>=20 && i<40){
stringStart=16;
stringStop=30;
}
if (i>=40 && i<60){
stringStart=33;
stringStop=46;
}
if (i>=60 && i<80){
stringStart=49;
stringStop=64;
}
if (i>=80 && i<100){
stringStart=67;
stringStop=79;
}
if (i>=100 && i<120){
i=0;
}
switch(mod) {

```

```

case 1:
{
  display.clear();
  draw2();
  display.display();
}
break
case 2:
{ display.clear();
draw3();
display.display();
}
break;

case 3:
{ display.clear();
draw1();
display.display();
}
break;
}
if(digitalRead(buttonPin) == HIGH){
  button1();
}
}
void button1(){
  mod++;
  if (mod == 4){
    mod = 1;
  }
}
delay(1000);
}
void draw2()
{float lin = gps.location.lat();
float buj = gps.location.lng();
int derajatlin = (int) lin;
float komadesimallin=lin-derajatlin;
int derajatbuj = (int) buj;
float komadesimalbuj=buj-derajatbuj;
float menitlin= komadesimallin*60;
int menitlin1= (int) menitlin;

```

```

float menitbuj= komadesimalbuj*60;
int menitbuj1= (int) menitbuj;
float sisamenitlin= menitlin-menitlin1;
float sisamenitbuj= menitbuj-menitbuj1;
int detiklin = sisamenitlin*60;
int detikbuj = sisamenitbuj*60;
display.setFont(ArialMT_Plain_10);
display.drawString(0, 0, "jam  :");
display.drawString(40, 0, twoDigits(gps.time.hour()+7) + ":" +
twoDigits(gps.time.minute()) + ":" + twoDigits(gps.time.second()));
display.drawString(0, 10, "tanggal:");
display.drawString(40, 10, twoDigits(gps.date.day()) + "/" +
twoDigits(gps.date.month()) + "/" + String(gps.date.year()));
display.drawString(0, 20, "Lintang:");
display.drawString(40, 20, String(gps.location.lat(),6));
display.drawString(40,30, String(derajatlin));
display.drawString(55,30, degSymbol);
display.drawString(61,30, String(menitlin1));
display.drawString(76,30, "");
display.drawString(78,30, String(detiklin));
display.drawString(92,30, "");
display.drawString(98,30, "LU");
display.drawString(0, 40, "Bujur  :");
display.drawString(38,40,String(gps.location.lng(),6));
display.drawString(38,50, String(derajatbuj));
display.drawString(55,50, degSymbol);
display.drawString(61,50, String(menitbuj1));
display.drawString(74,50, "");
display.drawString(78,50, String(detikbuj));
display.drawString(92,50, "");
display.drawString(98,50, "BT");
}
void ambil_jadwal_sholat(int waktu)
{
get_float_time_parts(times[waktu],ambil_jam_sholat,ambil_menit_sholat);
if (ambil_menit_sholat < 58)
{
ambil_menit_sholat = ambil_menit_sholat + 2;
ambil_jam_sholat = ambil_jam_sholat;
}
}

```

```

else if (ambil_menit_sholat == 58)
{
ambil_menit_sholat = 0;
ambil_jam_sholat = ambil_jam_sholat + 1;
}
else if (ambil_menit_sholat == 59)
{
ambil_menit_sholat = 1;
ambil_jam_sholat = ambil_jam_sholat + 1;
}
satuan_jam_sholat = ambil_jam_sholat % 10;
puluhan_jam_sholat = ambil_jam_sholat / 10;
satuan_menit_sholat = ambil_menit_sholat % 10;
puluhan_menit_sholat = ambil_menit_sholat / 10;
}
void GPS_Timezone_Adjust()
{
while (Serial_GPS.available())
{
if (gps.encode(Serial_GPS.read()))
{
int Year = gps.date.year();
byte Month = gps.date.month();
byte Day = gps.date.day();
byte Hour = gps.time.hour();
byte Minute = gps.time.minute();
byte Second = gps.time.second();
setTime(Hour, Minute, Second, Day, Month, Year);
adjustTime(UTC_offset * SECS_PER_HOUR);
}
}
}
void draw3()
{
int jamnow= (gps.time.hour()+7)*60;
int menitnow= gps.time.minute();
int waktunow= jamnow + menitnow;
int sbhjam=wkt_sbhjam.toInt();
int sbhmenit=wkt_sbhmenit.toInt();
int zhrjam=wkt_zhrjam.toInt();
int zhrmenit=wkt_zhrmenit.toInt();

```

```

int asrjam=wkt_asrjam.toInt();
int asrmenit=wkt_asrmenit.toInt();
int mgrjam=wkt_mgrjam.toInt();
int mgrmenit=wkt_mgrmenit.toInt();
int isyjam=wkt_isyjam.toInt();
int isymenit=wkt_isymenit.toInt();
if (jamnow+menitnow >= (sbhjam*60)+ sbhmenit    &&
jamnow+menitnow < (zhrjam*60)+ zhrmenit ) {
display.setFont(ArialMT_Plain_16);
display.drawString(20, 0, "Subuh :");
display.drawString(70, 0, wkt_sbh);}
else if
(jamnow+menitnow >= (zhrjam*60)+ zhrmenit    && jamnow+menitnow
< (asrjam*60)+ asrmenit ) {
display.setFont(ArialMT_Plain_16);
display.drawString(20, 0, "zuhur :");
display.drawString(70, 0, wkt_zhr );
}
else if
(jamnow+menitnow >= (asrjam*60)+ asrmenit    && jamnow+menitnow
< (mgrjam*60)+ mgrmenit ) {
display.setFont(ArialMT_Plain_10);
display.drawString(20, 0, "ashar :");
display.drawString(70, 0, wkt_asr);}
else if
(jamnow+menitnow >= (mgrjam*60)+ mgrmenit    &&
jamnow+menitnow < (isyjam*60)+ isymenit ) {
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "maghrib :");
display.drawString(80, 0, wkt_mgr);}
else if
(jamnow+menitnow >= (isyjam*60)+ isymenit || jamnow+menitnow <
(sbhjam*60)+ sbhmenit ) {
display.setFont(ArialMT_Plain_16);
display.drawString(20, 0, "isya' :");
display.drawString(70, 0, wkt_isy);}
display.setFont(ArialMT_Plain_10);
display.drawString(5, 20, twoDigits(gps.date.day()) + "/" +
twoDigits(gps.date.month()) + "/" + String(gps.date.year()));
display.drawString(70, 20, twoDigits(gps.time.hour()+7) + ":" +
twoDigits(gps.time.minute()) + ":" + twoDigits(gps.time.second()));

```

```

display.drawString(0, 30, "Subuh:");
display.drawString(35, 30, wkt_sbh);
display.drawString(70, 30, "Zuhur:");
display.drawString(100, 30, wkt_zhr);
display.drawString(0, 40, "Ashar :");
display.drawString(33, 40, wkt_asr);
display.drawString(0, 50, "Maghrib:");
display.drawString(40,50, wkt_mgr);
display.drawString(70, 40, "Isya:");
display.drawString(100,40, wkt_isy);
}
void draw1() {
LK=21.422547*(0.0174532925195433);//Lintang (Latitude)Ka'bah
BK=39.82626667*(0.0174532925195433);
float lati= (gps.location.lat()*(0.0174532925195433));
float lngi= (gps.location.lng()*(0.0174532925195433));
double c=sin(lngi-BK*(0.0174532925195433));//sin(Ba-Bb)
double c=sin(lngi-BK);
double d=cos(lati);//cos(Lb)
double e=tan(LK)//tan(21,25)
double f=sin(lati);//sin(Lb)
double g=cos(lngi-BK);//cos(Ba-39,50)
float dd = (d*e)-(f*g);
u= c/dd;
double gg = atan(u)*(180/PI);
float kiblat= 360-gg;
int kiblah = 360-gg;
int x, y, z;
int azimuth;
float azimuth; //is supporting float too
qmc.read(&x, &y, &z,&azimuth);
azimuth = qmc.azimuth(&y,&x);//you can get custom azimuth
Serial.print("x: ");
Serial.print(x);
Serial.print(" y: ");
Serial.print(y);
Serial.print(" z: ");
Serial.print(z);
Serial.print(" a: ");
Serial.print(azimuth);
Serial.println();

```

```

delay(100);
angle = atan2(-(double) y, -(double) x) * (180 / 3.14159265) + 180;
angle += (19+1/ 60);
float angle_rad = 3.14159265 * (float) angle / 180.0;
float angle_rad1 = 3.14159265 * kiblat / 180.0;
char angle_array [] = " "; //
dtostrf(angle, 3, 0, angle_array);
angle_array [3] = char(176);
String corner;
if (azimuth > 337 || azimuth < 23) corner = "N ";
else if (azimuth > 22 && azimuth < 68) corner = "NE";
else if (azimuth > 67 && azimuth < 113) corner = "E ";
else if (azimuth > 112 && azimuth < 158) corner = "SE";
else if (azimuth > 157 && azimuth < 203) corner = "S ";
else if (azimuth > 202 && azimuth < 248) corner = "SW";
else if (azimuth > 247 && azimuth < 293) corner = "W ";
else if (azimuth > 292 && azimuth < 338) corner = "NW";
char corner_array [3];
corner.toCharArray(corner_array, 3);
Serial.print("*****");
Serial.print(angle_array);
Serial.print("*****");
int x0 = 30 + cos(angle_rad) * 20; // tip of the arrow on circle
int y0 = 30 + sin(angle_rad) * 20;
int x1 = 30 + cos(angle_rad1) * 20; // triangle point
int y1 = 30 + sin(angle_rad1) * 20;
display.setFont(ArialMT_Plain_10);
display.drawString(70, 34, "kiblat:   ");
display.drawString(97, 34, String(kiblat, 0));
display.drawString(80, 4, String(angle_array));
display.drawString(80, 19, corner_array);
display.drawCircle(30, 30, 25);
display.drawLine(x0, y0, 30, 30);
display.drawCircle(x1, y1, 2);
}
void alarm()
{int jamnow= gps.time.hour()+7;
int menitnow= gps.time.minute();
int detiknow= gps.time.second();
if (twoDigits(gps.time.hour()+7) == wkt_sbhjam &&
twoDigits(gps.time.minute()) == wkt_sbhmenit && detiknow<=11 ) {

```

```

display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat subuh");
display.drawString(40, 30, wkt_sbh);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
else if
(twoDigits(gps.time.hour()+7) == wkt_zhrjam &&
twoDigits(gps.time.minute()) == wkt_zhrmenit&& detiknow<=11 ) {
display.clear();
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat zuhur");
display.drawString(40, 30, wkt_zhr);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
else if
(twoDigits(gps.time.hour()+7) == wkt_asrjam &&
twoDigits(gps.time.minute()) == wkt_asrmenit&& detiknow<=11 ) {
display.clear();
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat ashar");
display.drawString(40, 30, wkt_asr);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
else if
(twoDigits(gps.time.hour()+7) == wkt_mgrjam &&
twoDigits(gps.time.minute()) == wkt_mgrmenit&& detiknow<=11 ) {
display.clear();
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat maghrib");

```



```

display.drawString(40, 30, wkt_mgr);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
else if
(twoDigits(gps.time.hour()+7) == wkt_isyjam &&
twoDigits(gps.time.minute()) == wkt_isy menit && detiknow<=11 ) {
display.clear();
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat isya");
display.drawString(40, 30, wkt_isy);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
else if
(jamnow == 11 && menitnow == 22 && detiknow<=11) {
display.clear();
display.setFont(ArialMT_Plain_16);
display.drawString(10, 0, "sholat subuh");
display.drawString(40, 30, wkt_sbh);
display.display();
digitalWrite(buzzer, HIGH);
delay(3000);
digitalWrite(buzzer, LOW);
delay(1000);
}
}
}

```

Halaman ini sengaja dikosongkan

DAFTAR RIWAYAT HIDUP



Nama : Rahmad Hervandana Djafaar
TTL : Tulungagung, 30 April 1995
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Ds. Bungur RT: 04 RW:02
Kec. Karangrejo Kab.
Tulungagung
Telp/HP : 082213821586
Email : rahmad.djafaar@gmail.com

RIWAYAT PENDIDIKAN

1. 2002-2008 : SDN 1 BUNGUR
2. 2008-2011 : SMP 1 KAUMAN
3. 2011-2004 : SMA 1 BOYOLANGU
4. 2015-2019 : D3 Teknik Elektro Otomasi Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN ORGANISASI

1. Staff Data Center IARC 2016
2. Staff kesma HIMADETEKTRO 2016/2017
3. Staff Ahli Data Center IARC 2017