

**SISTEM KEAMANAN PADA KOTAK AMAL MASJID
BERBASIS MIKROKONTROLER**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Kevin Francisko NIM 0031912

Deta Khashanrusnuria Anggelia NIM 0031935

**POLITEKNIK MANUFaktur NEGERI
BANGKA BELITUNG
TAHUN 2022**

LEMBAR PENGESAHAN

**SISTEM KEAMANAN PADA KOTAK AMAL MASJID BERBASIS
MIKROKONTROLER**

Oleh:

Kevin Francisko / 0031912

Deta Khashanrusnuria Anggelia / 0031935

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

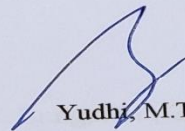
Menyetujui,

Pembimbing 1



Indra Dwisaputra, M.T

Pembimbing 2



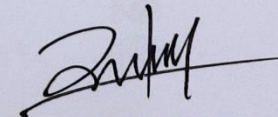
Yudhi, M.T

Penguji 1



M. Iqbal Nugraha, M.Eng

Penguji 2



Ahmat Josi, M.Kom

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Kevin Francisko

NIM : 0031912

Nama Mahasiswa 2 : Deta Khashanrusnuria Anggelia

NIM : 0031935

Dengan Judul : Sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila di kemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

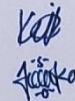
Sungailiat, 22 Agustus 2022

Nama Mahasiswa

Tanda Tangan

1. Kevin Francisko

2. Deta Khashanrusnuria Anggelia



ABSTRAK

Kotak amal umumnya hanya menggunakan pengaman biasa berupa kunci konvensional seperti hanya digembok saja. Jenis pengaman seperti ini masih memungkinkan terjadinya pembobolan kotak amal masjid karena gembok tidak begitu sulit untuk dihancurkan. Dengan menggunakan komponen-komponen tambahan yang efektif dalam menjaga keamanan kotak amal sehingga dapat meminimalisir atau bahkan dapat menghentikan tingkat kriminalitas dalam mencuri kotak amal masjid tanpa perlu mengganggu orang-orang yang sedang beribadah di masjid. Proyek akhir ini telah mengumpulkan berbagai informasi-informasi baik dari buku-buku, literature maupun jurnal yang bersumber pada internet. Apabila sidik jari yang dimasukkan merupakan sidik jari yang tidak terdaftar, maka kunci selenoid tidak akan terbuka secara otomatis sehingga kotak amal akan terkunci. Dan apabila kotak amal dibuka secara paksa bahkan tanpa menggunakan sidik jari yang terdaftar, maka alarm akan berbunyi. Meskipun kotak amal dibawa lari sejauh apapun, lokasi kotak amal tersebut dapat dilacak dengan menggunakan pesan lokasi yang telah terkirim pada telepon seluler pengguna.

Kata kunci: Kunci Selenoid, Sidik Jari, Sistem Keamanan, Kotak Amal, GPS, Pesan lokasi



ABSTRACT

Generally, charity boxes only has conventional security features in place, such as being padlocked and using conventional keys. Because of how simple it is to break the padlock on the mosque charity box, this security level still allows thievery. By using additional components that are effective in maintaining the security of the charity box, it can minimize or even stop the crime rate of stealing the mosque's charity box without the need to disturb people who are worshiping in the mosque. This final project has collected various pieces of information, both from literature books and journals sourced on the internet. If the fingerprint entered is an unregistered fingerprint, the solenoid lock will not open automatically, so the charity box will be locked. And if the charity box is forcibly opened without even using a registered fingerprint, an alarm will sound. Even if the box is carried away, no matter how far, the location of the charity box can still be tracked by getting the messages that have been sent on the user's cell phone.

Keywords : Solenoid Lock, Fingerprint, Security System, Charity Box, GPS, Location messages



KATA PENGANTAR

“Assalamu’alaikum Warahmatullahi Wabarakatuh”

Alhamdulillah, Puji dan syukur kami panjatkan kehadirat Allah SWT, karena berkat, rahmat dan karunia-Nyalah kami dapat menyelesaikan Proyek Akhir ini yang berjudul ”Sistem Keamanan pada Kotak Amal Berbasis Mikrokontroler” dengan tepat pada waktunya. Proyek Akhir ini dibuat guna memenuhi salah satu persyaratan atau kewajiban mahasiswa dalam menyelesaikan kurikulum program Diploma III Teknik Elektronika di Politeknik Manufaktur Negeri Bangka Belitung. Pembuatan proyek akhir ini sesuai dengan instruksi dan arahan dari Pembimbing yang dilakukan oleh penulis selama membuat tugas akhir ini.

Dalam pembuatan Proyek Akhir ini penulis tidak sedikit mendapat bimbingan dan bantuan dari berbagai pihak. Pada kesempatan ini perkenankanlah penulis mengucapkan banyak terimakasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang terus memotivasi, memberikan dukungan dan terus mendoakan penulis dimanapun penulis berada.
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Iqbal Nugraha, M.Eng selaku Ketua Jurusan Teknik Elektronika dan Informatika Politeknik Manufaktur Negeri Bangka Belitung
4. Bapak Indra Dwisaputra, M. T. selaku pembimbing ke I yang telah meluangkan banyak waktu, tenaga serta pikiran dalam memberikan pengarahan pada penulisan Karya Tulis Proyek Akhir ini.
5. Bapak Yudhi, M. T. selaku pembimbing ke II yang telah meluangkan banyak waktu dan telah banyak pula memberi saran-saran dan solusi dari masalah-masalah yang penulis hadapi selama proses penyusunan Karya Tulis Proyek Akhir ini.

6. Bapak Ocsirendi, M.T selaku Ka. Prodi D-III Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh dosen-dosen Politeknik Manufaktur Negeri Bangka Belitung, yang telah mengajarkan banyak hal sehingga penulis menjadi seorang yang mempunyai wawasan dan ilmu serta kepada seluruh staf administrasi Politeknik Manufaktur Negeri Bangka Belitung.
8. Seluruh teman-teman Mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan proyek akhir.

Penulis menyadari bahwa Karya Tulis Proyek Akhir ini jauh dari sempurna dikarenakan keterbatasan pengetahuan dan ilmu penulis, maka dari itu saran serta kritik yang bersifat membangun sangat diharapkan oleh penulis. Akhir kata penulis berharap agar Karya Tulis Proyek Akhir ini dapat berguna bagi penulis dan dapat dipergunakan dikemudian harinya.

“Wassalamua’alaikum Warahmatullahi Wabarakatuh”

Sungailiat, 22 Agustus 2022

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN.....	xi
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
BAB II.....	5
LANDASAN TEORI.....	5
2.1 Sistem Keamanan Kotak Amal	5
2.2 Mikrokontroler	5
2.3 <i>Global Positioning System</i> (GPS).....	7
2.4 Modul NodeMCU ESP8266.....	8
2.5 Sensor Magnet	Error! Bookmark not defined.

2.6	<i>Buzzer</i>	10
2.7	<i>Finger Print</i>	10
2.8	<i>Solenoid Lock</i>	11
BAB III.....		12
METODE PELAKSANAAN.....		12
3.1	Studi Literatur.....	13
3.2	Merancang Sistem <i>Hardware</i> dan <i>Software</i>	13
3.3	Membuat Sistem <i>Hardware</i> dan <i>Software</i>	14
3.3.1	Sistem <i>Hardware</i>	14
3.3.2	Sistem <i>Software</i>	14
3.4	Menguji Coba Sistem Secara Keseluruhan	14
BAB IV		15
PEMBAHASAN.....		15
4.1	Deskripsi Alat.....	15
4.2	Perancangan <i>Hardware</i> dan <i>Software</i> Alat	16
BAB V.....		31
PENUTUP.....		31
5.1	Kesimpulan.....	31
5.2	Saran	31
DAFTAR PUSTAKA		32
LAMPIRAN		35

DAFTAR GAMBAR

Gambar 2. 1 Arduino Uno.....	6
Gambar 2.2 Modul GPS.....	7
Gambar 2.3 NodeMCU ESP8266.....	9
Gambar 2.4 Sensor Magnet.....	Error! Bookmark not defined.
Gambar 2.5 <i>Buzzer</i>	10
Gambar 2.6 <i>Finger Print</i>	10
Gambar 2.7 <i>Solenoid Lock</i>	11
Gambar 3.1 Flowchart Metode Pelaksanaan.....	12
Gambar 3.2 Blok Diagram Rancangan Alat.....	14
Gambar 4.1 Konstruksi box kotak amal.....	17
Gambar 4.2 Box penyimpanan semua komponen	18
Gambar 4.3 Gambar rangkaian keseluruhan alat.....	18
Gambar 4.4 Data yang ditampilkan oleh NodeMCU ESP8266	19
Gambar 4.5 Lokasi Kotak Amal.....	20
Gambar 4.6 Sidik jari yang terdaftar	21
Gambar 4.7 Sidik jari yang belum terdaftar	21
Gambar 4.8 Data sidik jari yang telah didaftarkan	22
Gambar 4.9 Lokasi kota k amal ditampilkan oleh modul NodeMCU ESP8266... 23	
Gambar 4.10 Lokasi kotak amal yang ditampilkan oleh <i>Google Maps</i>	24
Gambar 4.11 Lokasi kedua yang ditampilkan NodeMCU ESP8266	25
Gambar 4.12 Lokasi kedua yang ditampilkan pada <i>google maps</i>	26
Gambar 4.13 Lokasi ketiga yang ditampilkan NodeMCU.....	27
Gambar 4.14 Lokasi ketiga yang ditampilkan di <i>google maps</i>	27
Gambar 4.15 Lokasi keempat yang ditampilkan oleh NodeMCU	29
Gambar 4.16 Lokasi keempat yang ditampilkan <i>google maps</i>	29

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino Uno.....	6
---	---



DAFTAR LAMPIRAN

Lampiran 1 DAFTAR RIWAYAT HIDUP	35
Lampiran 2 PROGRAM KESELURUHAN	38



BAB I

PENDAHULUAN

1.1 Latar Belakang

Umumnya disetiap masjid mempunyai kotak amal yang berfungsi sebagai tempat menyimpan uang yang berasal dari sumbangan para jamaah dimana uang tersebut digunakan untuk kebutuhan fisik bangunan masjid dan kegiatan lainnya seperti dana untuk membantu anak yatim, bencana alam dan lain-lain. Kotak-kotak amal tersebut umumnya hanya menggunakan pengaman biasa yang berupa kunci konvensional seperti hanya digembok saja. Jenis pengaman ini masih memungkinkan terjadinya pembobolan kotak amal masjid karena kunci gembok tidak begitu sulit untuk dihancurkan. Hal ini kerap terjadi karena mudahnya akses masuk masjid dan minimnya penjagaan area masjid. Sehingga pelaku dengan mudahnya melakukan pencurian dengan langsung membawa kabur kotak amal tersebut tanpa harus membobol dan menghancurkan kunci kotak amal terlebih dahulu. Kotak amal yang dicuri inilah menjadi penyebab sulitnya pencarian karena tidak ada petunjuk dan sangat sulit untuk menemukan lokasi kotak amal yang sudah dicuri. Sehingga membutuhkan sebuah sistem keamanan yang tinggi untuk mencegah terjadinya hal itu dan juga agar dapat melacak di mana posisi kotak amal tersebut.

Peneliti sebelumnya telah melakukan penelitian mengenai sistem keamanan kotak amal. Nugroho & Almasri, (2021) telah meneliti mengenai alat keamanan kotak amal mendapatkan hasil bahwa jika kotak amal sudah aktif maka akan menelpon 15 detik dan dalam waktu 3 menit akan menelpon kembali tanpa mereset dan akan berulang hingga kotak amal diriset. Nursaleh, (2019) telah merancang kotak amal anti maling dan mendapatkan hasil bahwa masing-masing komponen saling terkait satu sama lain dalam hal fungsinya, sehingga jika terdapat masalah atau kegagalan pada salah satu komponen, alat kotak anti maling yang menggunakan SMS Gateway tidak akan berfungsi dengan baik. Tullah, Mustofa, & Nugraha, (2019) meneliti sistem keamanan rumah menggunakan

teknologi RFID dengan SMS *Gateway* mampu menjadi ide baru dalam dunia keamanan karena memberikan kenyamanan kepada pemilik agar tidak cemas ketika ada yang menerobos masuk ke dalam rumah, maka akan ada pemberitahuan melalui SMS. Wishnuadji, Narendro, & Peristiwa, (2022) meneliti tentang kontak penyimpanan dengan arduino sebagai pusat kontrol dan kendali yang mampu mengendalikan sistem keamanan dengan baik, sistem keamanannya membutuhkan pergantian kata sandi secara berkala demi keamanan. (Saleh & Haryanti, 2017) merancang sistem keamanan rumah menggunakan *relay* sebagai alat peringatan dengan kondisi apabila jendela atau pintu dibuka 25-90 derajat yang membuat alarm menyala sebagai sumber peringatan kepada pemilik rumah. (Ramadhan & Handoko, 2016) merancang sistem keamanan rumah yang kinerja sensor PIR dan sensor *magnetic switch* yang diletakkan pada tempat yang pas dapat bekerja dengan baik disaat sensor mendeteksi sebuah gerakan. (Qalbi, et al., 2020) merancang kotak amal cerdas yang memiliki fitur sensor *ultrasonic HC-SR04*, *infrared* yang mendeteksi orang, RFID-RC522 sebagai sensor yang tidak sembarangan orang yang bisa membuka akses kotak amal, LCD yang memperlihatkan jumlah para penyumbang. (Kurniawan, Sunarya, & Tulloh, 2018) meneliti sistem keamanan rumah dengan sensor PIR mendeteksi jarak maksimal sejauh 6 meter, sudut horizontal 90-135 derajat, vertikal 60-120 derajat, suhu dibawah dari suhu ruang tidak terlalu berpengaruh, dan seluruh sistem terbukti dapat berfungsi dengan baik. (Dita, Fahrezi, Prasetyawan, & Amarudin, 2021) merancang sistem keamanan pintu yang proses percobaannya berjalan dengan baik dan dapat dikatakan berhasil akan tetapi terdapat kesalahan pada sistem karena rancangan dan program yang belum sempurna. (Akil, Muchtar, & Fitriati, 2020) mendesain kotak amal mesjid tanpa sentuh dalam menghadapi era normal baru di tempat ibadah menggunakan sistem penggerak yang mirip dengan yang penggerak robot pada umumnya yang menggunakan sensor ultrasonik sebagai pembaca isyarat tangan ke bagian depan robot .

Oleh karena itu maka dirancang sistem keamanan kotak amal masjid berbasis mikrokontroler untuk menjaga kotak amal dari tindakan pencurian. Sistem tersebut dirancang ketika arduino dinyalakan yang kemudian akan

melakukan pemeriksaan ke seluruh komponen. Ketika tutup kotak amal dibuka dengan sidik jari yang sudah terdaftar dan sensor magnet pada penutup kotak amal akan memberikan informasi ke Arduino dan kemudian ke *buzzer*, sehingga *buzzer* akan nonaktif dan tidak akan berbunyi. Apabila kotak amal di buka dengan sidik jari yang belum terdaftar, maka sensor magnet yang terdapat pada penutup kotak amal akan memberikan informasi ke Arduino dan kemudian ke *buzzer*, sehingga *buzzer* diaktifkan dan berbunyi, dan ketika kotak amal dipindahkan ke tempat lain, maka GPS yang aktif akan memberikan informasi ke smartphone dan menunjukkan posisi kotak amal tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dibahas sebelumnya, adapun rumusan masalahnya sebagai berikut:

1. Bagaimana cara merancang sistem keamanan pada kotak amal?
2. Bagaimana cara untuk mengetahui letak posisi kotak amal?

1.3 Tujuan

Tujuan dari proyek akhir yang berjudul Sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler berdasarkan perumusan masalah di atas adalah mendesain dan membuat sistem pada kotak amal menggunakan *fingerprint* sebagai akses buka tutup pintu kotak amal dan modul GPS sebagai akses pencari lokasi kotak amal.

1.4 Batasan Masalah

Agar pembahasan masalahnya dapat diterapkan menjadi objektif dan tidak menyimpang dari rumusan masalah, maka batasan masalah pada proyek akhir ini adalah berikut :

1. Konstruksi yang dibuat bersifat *prototype*.
2. Daya baterai yang digunakan terbatas karena harus diisi ulang.
3. *Internet Access Point* sangat dibutuhkan agar dapat melacak lokasi kotak amal.



BAB II

LANDASAN TEORI

2.1 Sistem Keamanan Kotak Amal

Keamanan merupakan upaya untuk menghindari munculnya atau ancaman kejahatan yang akan mengganggu. Keamanan harus mencakup faktor-faktor seperti perlindungan data, integritas, keaslian, dan memiliki akses (Somba, 2015). Kotak amal masjid adalah tempat uang yang berasal dari sedekah para jamaah yang dikelola oleh pengurus masjid di mana uang hasil sedekah ini akan digunakan untuk keperluan pembangunan masjid ataupun untuk sedekah kepada fakir miskin dan anak yatim.

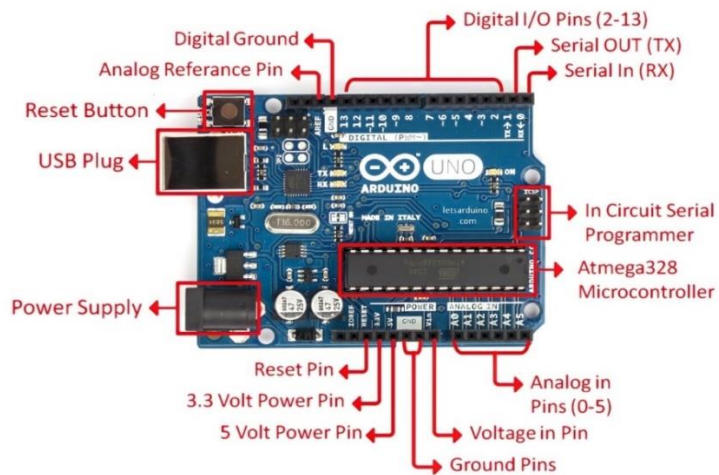
Istilah keamanan dapat digunakan merujuk pada hal kejahatan, semua jenis kecelakaan, dan lain-lain. Topik mengenai keamanan sangat luas dimana mencakup keamanan dari serang para teroris, keamanan dari para peretas computer, keamanan sebuah rumah dari pencuri ataupun penyusup, keamanan dari kehancuran ekonomi, serta banyak situasi yang lainnya. Konsep dasar dari keamanan yakni mengacu pada kemampuan untuk menghindari bahaya, sebagaimana yang sudah ditentukan oleh pemahaman, persepsi, dan inspirasi individu dalam melakukan tindakan untuk pencegahan. Ada tiga faktor yang signifikan yang terkait dengan keamanan, khususnya tingkat kesadaran hati nurani individu, kapasitas mental dan aktual untuk mengambil tindakan pencegahan, dan lingkungan fisik yang tidak aman atau kemungkinan timbulnya potensi bahaya (Akbar, 2020).

2.2 Mikrokontroler

Mikrokontroler adalah elektronik digital berupa *input* dan *output* yang dapat ditempatkan dan dihapus dari kontrol program dengan cara yang tidak biasa. Pengoperasian mikrokontroler adalah untuk membaca dengan teliti dan menyusun informasi data. Mikrokontroler umumnya digunakan dalam perangkat yang dikontrol secara otomatis, misalnya sistem kontrol mesin, *remote control*, mesin

kantor dan peralatan rumah tangga. Kehadiran mikrokontroler membuat kontrol listrik lebih membantu untuk dimanfaatkan

Arduino Uno merupakan model perangkat lunak yang bersifat terbuka berbasis mikrokontroler Amel AVR ATmega328. Arduino dirancang untuk membuat *prototype* perangkat keras elektronik dengan mudah. Arduino Uno mempunyai osilator Kristal 16 MHz, terkoneksi USB, steker, dan *header* ICSP (Pambudi, 2020) .Gambar Arduino Uno dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Arduino Uno

Berikut adalah spesifikasi lengkap Arduino Uno :

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
<i>Operating voltage</i>	5V
<i>Input Voltage</i>	7-9V
<i>Input Voltage (batas)</i>	6-20V
Digital I/O Pins	14 pin dengan 6 pin memberikan <i>output</i> PWM
PWM Digital I/O Pins	6
Pin masukan analog	6
Arus DC per I/O Pin	20Ma
Arus Untuk DC 3.3V Pin	50Ma
<i>Flash Memory</i>	32KB (ATmega328P) dengan 0.5KB digunakan untuk <i>bootloader</i>
SRAM	2KB (ATmega328P)
EEPROM	1KB (ATmega328P)
Clock Speed	16 MHz

2.3 *Global Positioning System (GPS)*

GPS merupakan sistem navigasi dan penentu penempatan dalam satelit yang dikelola dan merupakan kepemilikan AS. Sistem GPS dirancang untuk melengkapi sebuah informasi waktu, kecepatan, dan posisi tiga dimensi secara terus-menerus di seluruh dunia, terlepas dari waktu dan cuaca kepada banyak orang secara bersamaan. Saat ini, GPS sudah banyak dipakai dalam penjuru dunia. Di Indonesia banyak diadopsi terkhusus yang terlibat dalam aplikasi yang membutuhkan data posisi ataupun lokasi. Jika dipandang dari metode atau sistem penentu lokasi lainnya, sistem ini memiliki banyak keunggulan dan memberikan banyak tambahan, baik dari segi kinerja ataupun status lokasi yang disediakan (Pengenalan GPS, 2014). Gambar modul GPS dapat dilihat pada Gambar 2.2.



Gambar 2.2 Modul GPS

Sebenarnya GPS memiliki istilah yakni NAVSTAR GPS (*Global Positioning Satellite and Global Positioning System*). GPS memiliki 3 elemen penting, yaitu: satelit, pengontrol (*controller*), dan penerima (*receiver*). Di bawah ini adalah fungsi dan deskripsi dari tiga elemen penting GPS:

1. Satelit

Satelit memiliki efek untuk melegalkan dan merapikan data yang dikirimkan oleh stasiun pemantau. Membenahi serta memelihara data waktu yang sangat akurat (sebagaimana ditentukan oleh jam atom satelit) dan mengirimkan alamat dan data secara terus-menerus ke penyambut dari pemakai.

2. Pengontrol (*Controller*)

Controller atau pengontrol memiliki fungsi untuk memonitor dan mengendalikan satelit dari bumi, guna memeriksa status satelit, menentukan dan memprediksi waktu lintasan, penyelarasan waktu antara satelit dan pengiriman data ke satelit.

3. Penerima (*receiver*)

Receiver bekerja dalam mengolah data dari satelit untuk memastikan lokasi (lokasi dengan gambaran yang lengkap, yakni titik suatu lokasi di suatu tempat ketinggian), petunjuk, langkah serta periode yang dibutuhkan pemakai satelit. Terdapat dua versi *receiver* yaitu versi *NAVIGATION* dan versi *GEODETIC*. Jenis penerima *NAVIGATION* yakni : *Trimble Ensign, Trimble Pathfinder, Garmin, Sony* dan lain-lain. Untuk tipe *GEODETIC* yaitu : *Topcon, Leica, Astech, Trimble 4000 series* dan lain sebagainya.

Penentuan posisi dengan GPS adalah metode umum mengukur jarak secara bersamaan ke beberapa satelit (dengan koordinat yang diketahui). Penentuan posisi GPS secara umum dibagi menjadi dua metode yakni absolut dan relatif (Riadi, 2017)

1. Metode Absolut

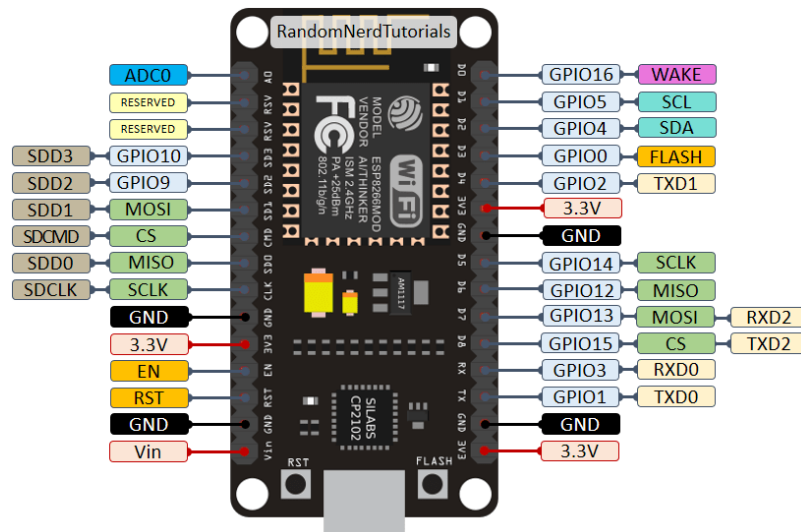
Metode absolut atau juga disebut penentuan posisi titik, menentukan posisi berdasarkan penerima tunggal. Akurasi posisi hanya beberapa meter (tidak terlalu akurat) dan biasanya hanya digunakan untuk tujuan navigasi.

2. Metode Relatif

Metode relatif atau lebih dikenal *differential positioning*, menggunakan beberapa receiver untuk menentukan posisi. GPS ditempatkan di lokasi tertentu di bumi dan menerima sinyal dari satelit secara terus menerus untuk jangka waktu tertentu dan menggunakannya sebagai referensi lain. Metode ini menghasilkan posisi yang sangat akurat dan dapat diterapkan pada pengukuran atau pemetaan *geodetic* yang membutuhkan akurasi tinggi.

2.4 Modul NodeMCU ESP8266

NodeMCU ialah papan elektronik yang berbasis chip ESP8266 yang menerapkan fungsi mikrokontroler serta dapat terhubung ke internet (WiFi). Dengan banyak pin I/O, dapat digunakan untuk memantau dan mengontrol aplikasi untuk proyek IOT. NodeMCU ESP8266 adalah bentuk fisik dengan konektor USB (mini USB) dan dapat diprogram dengan *compiler* yang menggunakan Arduino IDE. Gambar modul NodeMCU ESP8266 ditunjukkan pada Gambar 2.3 di bawah ini.



Gambar 2.3 NodeMCU ESP8266

Spesifikasi NodeMCU adalah sebagai berikut :

1. Papan ini didasarkan pada ESP8266 (*Single on Chip*) WiFi Serial Soc dengan built-in USB ke TTL. *Wireless* yang dipakai adalah IEEE 802.11b/g/n.
2. 2 kapasitor tantalum 100 mikro farad & 10 mikro farad.
3. Regulator LDO 3.3 V
4. Sebagai *indicator* dengan led biru.
5. Cp2102 usb ke jembatan UART.
6. Tombol reset, *port* USB, serta tombol *flash*.
7. Ada 9 GPIO termasuk 3 pin PWM, 1 saluran ADC, dan pin RX TX.
8. 3 pin tanah.
9. S3 dan S2 adalah pin GPIO 4
10. S1 MOSI (*Master Output Slave Input*) adalah jalur data dari master dan masuk ke dalam *slave*, sc cmd/sc.
11. S0 MISO (*Master Input Slave Input*) adalah jalur data keluar dari *slave* dan masuk ke dalam master.
12. SK adalah SCLK dari master ke *slave* yang berfungsi sebagai *clock*.
13. Pin Vin sebagai tegangan *input*.
14. Mikrokontroler 32-bit tertanam.

2.5 Buzzer

Buzzer adalah komponen elektronik dimana dapat mengubah getaran listrik ke getaran yang dapat didengar. Prinsip kerja *buzzer* hampir sama dengan prinsip speaker, yaitu sebuah kumparan dipasang ke diafragma, serta kumparan yang menerima energi listrik menjadi elektromagnet, kumparan tersebut menyala dan mati tergantung pada arah hisap. Karena kumparan menempel pada diafragma, pergerakan kumparan menyebabkan diafragma bergerak maju mundur, yang menggetarkan udara dan menimbulkan suara yang berisik (Sulistyowati & Febriantoro, 2012). Gambar buzzer ditunjukkan pada Gambar 2.4 di bawah ini.



2.6 Finger Print

Finger print merupakan fitur elektronik yang menggunakan sensor pemindai untuk mengenali sidik jari seseorang untuk mengotentikasi informasi pribadi. Sensor ini digunakan di banyak perangkat elektronik seperti smartphone, pintu, sensor kehadiran staf atau karyawan dan berbagai perangkat elektronik yang membutuhkan tingkatan keamanan tinggi dan hanya tersedia untuk individu tertentu. Gambar *finger print* ditunjukkan pada Gambar 2.5 dibawah ini.



Gambar 2.5 Finger Print

2.8 *Solenoid Lock*

Solenoid lock ialah kunci dengan pintu listrik berbasis solenoid yang bisa digunakan dalam pembuatan sistem keamanan. *Solenoid lock* tersebut berada pada tegangan 12V dengan desain lubang pemasangan yang mempermudah pemasangan sekrup ke bagian pintu. Dapat diintegrasikan dengan mikrokontroler untuk aplikasi proyek keamanan pada rumah. Gambar *solenoid lock* ditunjukkan pada Gambar 2.6 dibawah ini.

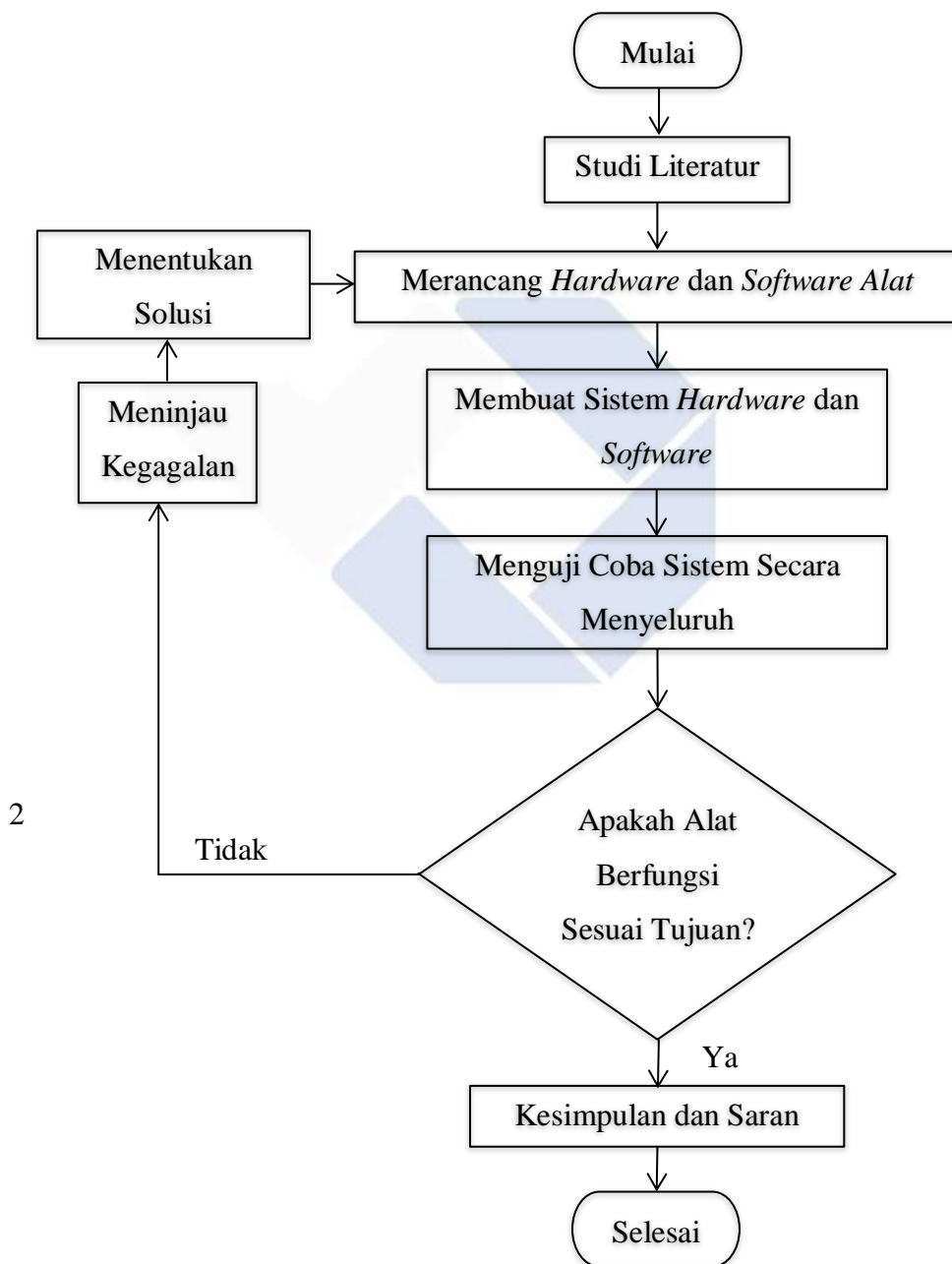


Gambar 2.6 *Solenoid Lock*

BAB III

METODE PELAKSANAAN

Prosedur pelaksanaan yang digunakan untuk memecahkan permasalahan dalam menuntaskan tugas akhir ditunjukkan pada Gambar 3.1 dibawah ini.



Gambar 3.1 Flowchart Metode Pelaksanaan

3.1 Studi Literatur

Penelitian atau studi literature yang dilakukan guna memperoleh referensi mengenai materi serta metode yang mendukung sebuah riset. Langkah ini dilakukan dengan mencari referensi dari sebagian buku, jurnal, ataupun postingan mengenai sistem keamanan yang berbasis mikrokontroler dan pengetahuan yang terikat dengan materi proyek akhir.

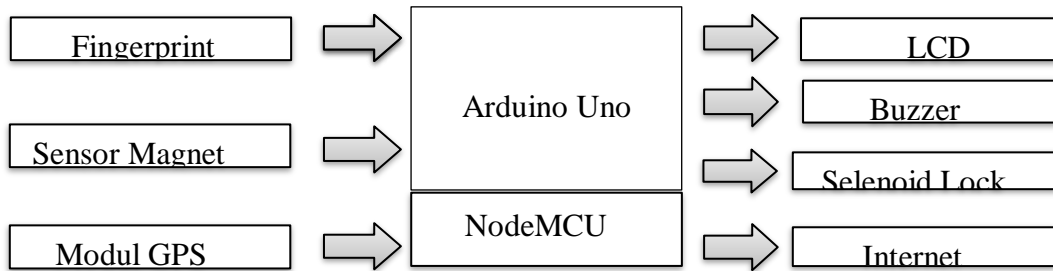
Penentuan komponen yang digunakan ini didapat berdasarkan studi literatur yang sudah dilakukan. Adapun komponen-komponen yang akan dipakai pada proyek akhir sebagai berikut :

1. Arduino Uno
2. Modul GPS
3. Modul NodeMCU ESP8266
4. Sensor Magnet
5. *Buzzer*
6. LCD
7. *Fingerprint*
8. *Relay*
9. *Solenoid Lock*

3.2 Merancang Sistem *Hardware* dan *Software*

Merancang sistem *hardware* dan *software* ialah langkah yang dilakukan bersamaan dalam mengidentifikasi serta merancang peralatan yang akan digunakan untuk proyek akhir ini. Perangkat *hardware* yang dipakai meliputi *input*, prosesor, dan *output*. Perangkat *input* adalah sumber baterai 5V, sensor *magnetic door* MC-38, modul GPS, dan modul NodeMCU ESP8266. Perangkat pengolah terdiri dari *fingerprint*. Kemudian untuk perangkat *ouput* meliputi LCD, *buzzer*, dan *relay*. Merancang sebuah *software* yang dapat digunakan dalam mengontrol keseluruhan sistem menggunakan *software* arduino uno hingga dapat memeriksa keamanan kotak amal dengan layar LCD.

Adapun blok diagram rancangan alat seperti Gambar 3.2 berikut.



Gambar 3.2 Blok Diagram Rancangan Alat

3.3 Membuat Sistem *Hardware* dan *Software*

3.3.1 Sistem *Hardware*

Proses pembuatan *hardware* sistem keamanan kotak amal masjid berbasis mikrokontroler yaitu membuat sebuah konstruksi pada alat sistem keamanan pada kotak amal, prosesnya yang terdiri dari pembuatan box untuk kotak amal yang dilakukan dengan menggunakan metode manual menggunakan papan kayu dengan ketebalan kira-kira 1.5 cm yang kemudian dirakit menggunakan palu dan paku.

3.3.2 Sistem *Software*

Proses pembuatan perangkat *software* pada proyek akhir ini yakni dengan menggunakan metode pengkodean pada program yaitu Arduino Uno yang akan digunakan dalamqk mengontrol sistem secara keseluruhan dengan alat “Sistem Keamanan Kotak Amal Masjid Berbasis Mikrokontroler”. Apa yang akan dilakukan pada proyek akhir ini. Proses pengkodean dengan program Arduino Uno berbeda untuk setiap komponen yang ditampilkan layar LCD.

3.4 Menguji Coba Sistem Secara Keseluruhan

Proses pengujian sistem pada proyek akhir ini umumnya dilakukan ketika semua komponen dan perangkat yang tersusun secara baik yang sesuai dengan rancangan yang sudah dibuat, setelah itu prosedur pengujian menentukan apakah alat yang dibuat dapat bekerja dengan baik atau dalam keadaan sesuai dengan tujuan yang diinginkan atau tidak.

BAB IV

PEMBAHASAN

Pada bab pembahasan ini menyampaikan proses penyelesaian tugas akhir ini berdasarkan metode-metode yang sudah diuraikan di bab sebelumnya. Adapun penjelasan prosesnya yang diuraikan di bawah ini:

1. Deskripsi tentang Alat
2. Perancangan *Hardware* dan *Software* Alat Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler
3. Pembuatan Konstruksi Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler
4. Perakitan Alat Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler
5. Pengujian *Hardware* Elektrik setiap Komponen
6. Perancangan Tampilan *Software*
7. Pengujian Alat

4.1 Deskripsi Alat

Proyek akhir yang dibuat mengenai sistem sebuah alat keamanan berbasis mikrokontroler yang dirancang untuk membantu masyarakat khususnya pengurus masjid dalam menjaga kotak amal masjid sehingga lokasinya dapat di monitoring langsung karena menggunakan modul GPS. Alat ini menggunakan tegangan sumber baterai 5V. Alat ini akan beroperasi apabila arduino dinyalakan di mana akan melakukan pengecekan terhadap semua komponen.

Pada saat penutup kotak amal dibuka menggunakan sidik jari yang telah didaftar, maka sensor magnet pada penutup kotak amal akan memberikan informasi ke Arduino dan kemudian ke *buzzer*, sehingga *buzzer* akan dalam keadaan nonaktif dan tidak akan berbunyi. Apabila kotak amal di buka dengan sidik jari yang belum terdaftar, maka sensor magnet yang terdapat pada penutup kotak amal akan memberikan informasi ke Arduino dan kemudian ke *buzzer*, sehingga *buzzer* diaktifkan dan berbunyi. Ketika kotak amal dipindahkan ke

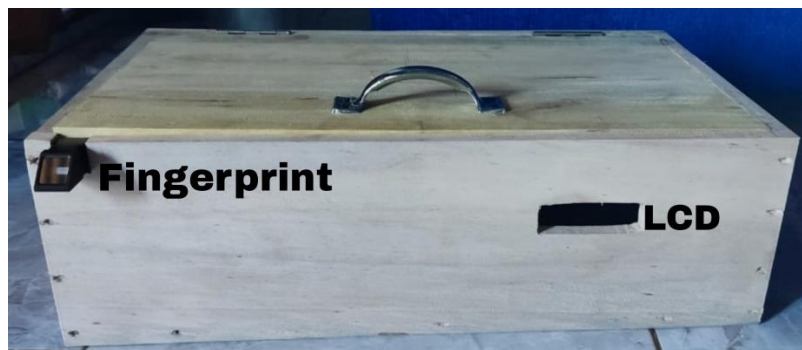
tempat lain, maka modul NodeMCU ESP8266 dan modul GPS akan terlihat atau bisa melakukan *monitoring* melalui internet sehingga dapat memberikan informasi ke *smartphone* dan menunjukkan posisi kotak amal tersebut.

4.2 Perancangan *Hardware* dan *Software* Alat

Untuk rancangan *hardware* Rangkaian kontrol sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler ini dirangkai di dalam sebuah box yang ukurannya sudah disesuaikan dengan ukuran kotak amal masjid pada umumnya. Ukuran dan bentuk box pada proyek akhir ini dibuat dengan bahan papan kayu yang memiliki ukuran lebar sebesar 30 cm, dan panjangnya sekitar 50 cm, serta memiliki tinggi 15 cm. Pembuatan box dengan bentuk kotak bertujuan untuk memudahkan peletakan komponen dan rangkaian kontrol. Di dalam rangkaian ini terdapat rancangan *software* alat yaitu sebuah NodeMCU ESP8266 digunakan sebagai perantara komunikasi perangkat keras dengan *smartphone* berbasis internet. Modul NodeMCU ESP8266 juga digunakan untuk menampilkan data pada *smartphone* yang diterima dari modul GPS melalui komunikasi serial. NodeMCU ESP8266 memiliki fungsi proses *monitoring* data dapat dilakukan secara *real time* tidak lambat. Sensor magnet digunakan untuk mendeteksi pintu kotak amal apakah dalam keadaan terbuka atau tertutup. Alasan menggunakan sensor magnet karena sensor ini sesuai dengan proyek akhir ini karena cara kerja dari sensor magnet adalah saat pintu terbuka, dua bagian utama sensor secara tidak langsung awalnya melekat menjadi terlepas. Hal ini menyebabkan magnet sensor bereaksi terhadap pintu yang terbuka dan menganggap bahwa pintu dalam kondisi terbuka. LED sebagai indikator jika keadaan aman hijau, keadaan waspada kuning, dan keadaan merah bahaya. *Buzzer* sebagai indikator jika sensor magnet dibuka tanpa melalui *fingerprint*. *Display Battery* digunakan untuk menampilkan hasil pada battery 5V dalam kondisi berkurang ataupun penuh. Pada proyek akhir ini menggunakan Baterai 5V sebagai sumbernya. Komponen tersebut disambungkan NodeMCU ESP8266 sebagai pengontrol alat ini. Sehingga setiap komponen memiliki pin tersendiri dan harus sama pada saat program di Arduino R3.

4.3 Dasar Pertimbangan Desain

Pada tahap pembuatan konstruksi Sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler ini proses yang akan dilakukan yaitu membuat box menggunakan bahan papan kayu yang memiliki ukuran panjang 50cm, lebar 30cm dan tinggi 15cm serta dengan ketebalan 1.5cm. Box kotak amal masjid ini dirakit secara manual dengan menggunakan bahan dari papan kayu. Papan kayu dipilih sebagai bahan dalam pembuatan konstruksi kotak amal karena mudah didapatkan dan dibentuk sehingga dapat mempermudah dalam membuat konstruksi kotak amal tersebut. Adapun konstruksi box seperti Gambar 4.1 berikut.



Gambar 4.1 Konstruksi box kotak amal

Konstruksi kotak amal masjid yang terbuat dari papan kayu ini di dalamnya terbagi atas 2 bagian yaitu bagian atas untuk meletakkan uang hasil sumbangan para jamaah di masjid dan bagian bawah untuk meletakkan komponen-komponen yang digunakan. Bagian atas dan bagian bawah dipisah menggunakan sepotong papan kayu agar antara uang hasil sumbangan para jamaah dan komponen-komponen memiliki ruang tersendiri serta agar komponen-komponen dapat berfungsi dengan baik tanpa ada gangguan dari tumpukan uang hasil sumbangan para jamaah.

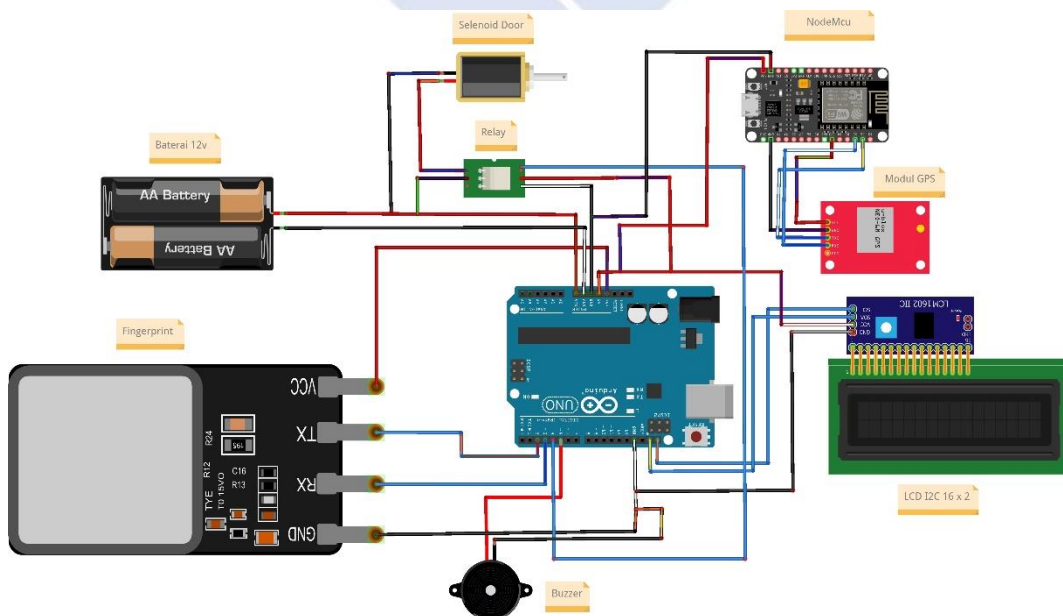
4.4 Perakitan *Hardware* Alat

Sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler dengan menyatukan semua komponen pada proyek akhir yang akan dimasukkan ke dalam box. Adapun box penyimpanan semua komponen-komponen yang akan digunakan seperti Gambar 4.2 berikut.



Gambar 4.2 Box penyimpanan semua komponen

Gambar 4.2 menunjukkan box penyimpanan untuk semua komponen. Terdiri dari modul NodeMCU ESP8266, modul GPS, *relay*, *solenoid*, *buzzer*, dan *Arduino* rangkaian. Untuk gambaran rangkaian keseluruhan alat ada pada Gambar 4.3 berikut.



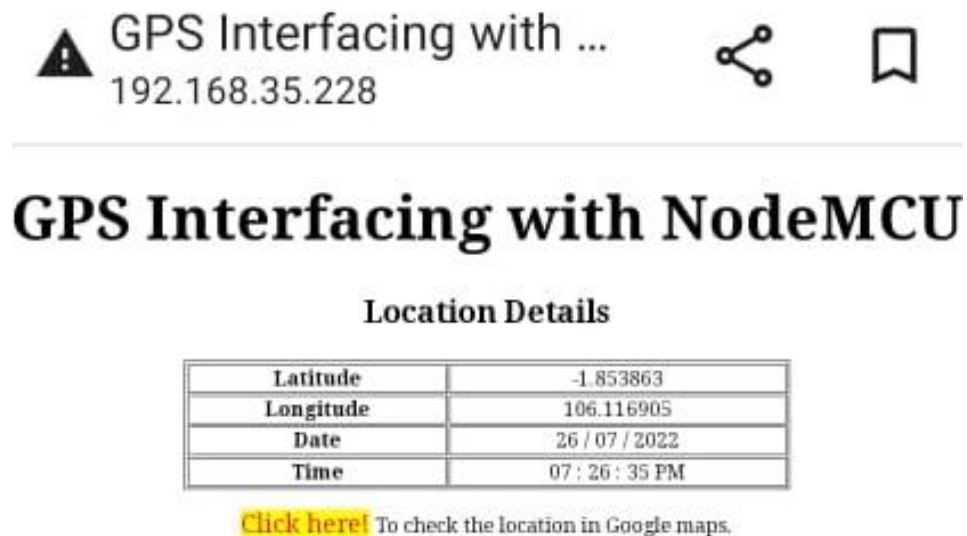
Gambar 4.3 Gambar rangkaian keseluruhan alat

4.5 Pembuatan *Software* Alat

Pembuatan *software* Sistem Keamanan pada Kotak Amal Masjid Berbasis Mikrokontroler dilakukan dengan melakukan pemrograman menggunakan software Arduino IDE dengan *monitoring* pada *smartphone* (*google maps*) karena menggunakan modul GPS dan modul NodeMCU ESP8266.

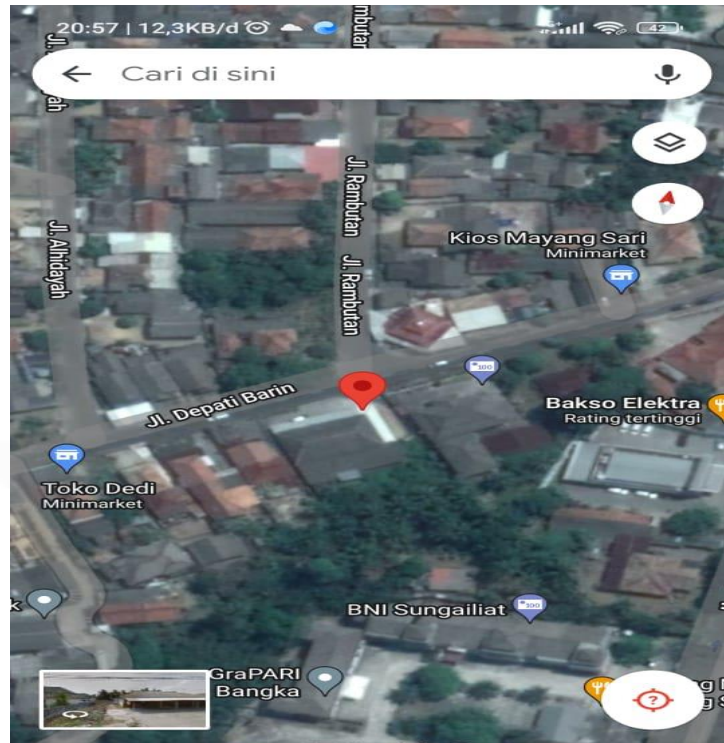
4.5.1 Menampilkan Data pada Smartphone di Google Maps

Menampilkan informasi lokasi kotak amal masjid menggunakan NodeMCU ESP8266 dalam *me-monitoring* data yang diperoleh dari modul GPS dengan menampilkan server IP *wi-fi* melalui komunikasi serial pada aplikasi Arduino IDE yang menampilkan titik lintang (*latitude*) dan titik bujur (*longitude*) yang merupakan posisi atau lokasi kotak amal dan kemudian akan ditampilkan pada *google maps* secara *real time*. Tujuan penggunaan modul NodeMCU ESP8266 adalah menampilkan data yang dikirim oleh modul GPS yang kemudian lokasi kotak amal masjid akan ditampilkan pada *google maps*. Untuk contoh data yang ditampilkan pada modul NodeMCU ESP 8266 ada pada gambar 4.4 di bawah ini.



Gambar 4.4 Data yang ditampilkan oleh NodeMCU ESP8266

Pada gambar 4.4 di atas merupakan data lokasi atau posisi kotak amal masjid dengan mengakses alamat IP *wi-fi* dengan titik koordinat yang tercantum pada data yang telah ditampilkan pada internet. Jika tulisan *Click Here!* diklik maka akan menampilkan lokasi kotak amal masjid tersebut. Adapun lokasi kotak amal masjid tertera pada Gambar 4.5 di bawah ini.



Gambar 4.5 Lokasi Kotak Amal

3.6. Pengujian *Finger Print*

Pengujian finger print dilakukan untuk memasukkan mendaftarkan data-data sidik jari yang akan di *input* ke Sistem Keamanan yang kami buat Berbasis Mikrokontroler ini. Apabila sidik jari yang terdaftar mengakses fingerprint maka solenoid lock akan tertutup seperti pada Gambar 4.6, sedangkan apabila sidik jari yang tidak terdaftar mengakses fingerprint maka solenoid lock akan terbuka dan alarm buzzer pada kotak amal akan berbunyi seperti pada Gambar 4.7. Adapun pendaftaran sidik jari pada sistem keamanan adalah seperti Gambar 4.6 dan Gambar 4.7 adalah sebagai berikut.

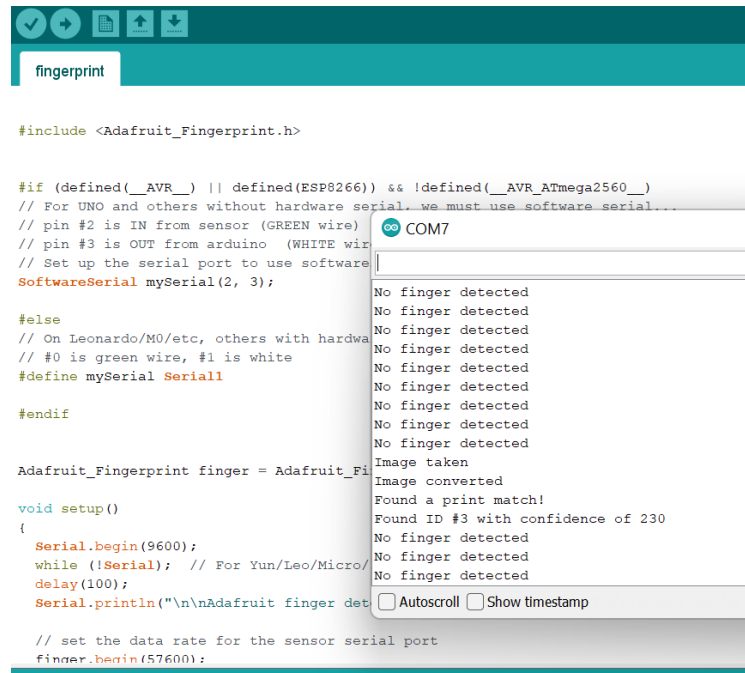


Gambar 4.6 Sidik jari yang terdaftar



Gambar 4.7 Sidik jari yang belum terdaftar

Untuk tampilan data sidik jari yang telah terdaftar dapat dilihat pada Gambar 4.8 sebagai berikut.



Gambar 4.8 Data sidik jari yang telah didaftarkan

Dapat dilihat pada Gambar 4.8 bahwa saat sidik jari yang tidak terdaftar sedang mengakses *fingerprint* maka serial pada arduino diatas akan menampilkan *No finger detected*. Sedangkan apabila sidik jari yang telah terdaftar mengakses *fingerprint* maka serial pada arduino akan menampilkan *Image converted*.

Pengujian *fingerprint* dilakukan menggunakan pemrograman arduino IDE dengan list sebagai berikut:

```
#include "Adafruit_Fingerprint.h"
#include "SoftwareSerial.h"
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
uint8_t id;

void setup() {
  Serial.begin(9600);
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("FingerPrint Sensor Ditemukan!");
  } else {
    Serial.println("FingerPrint Sensor Tidak Ditemukan! :(");
    while (1) {
      delay(1);
    }
  }
}
```

4.6.2 Pengujian Modul GPS

Pengujian modul GPS dilaksanakan guna mengecek lokasi terbaru kotak amal masjid. Data-data lokasi kotak amal masjid yang ditampilkan pada Modul NodeMCU ESP8266 berupa tanggal dan waktu saat NodeMCU ini diakses serta dapat dilihat pada Gambar 4.9 di bawah ini.



GPS Interfacing with ...
192.168.35.228

GPS Interfacing with NodeMCU

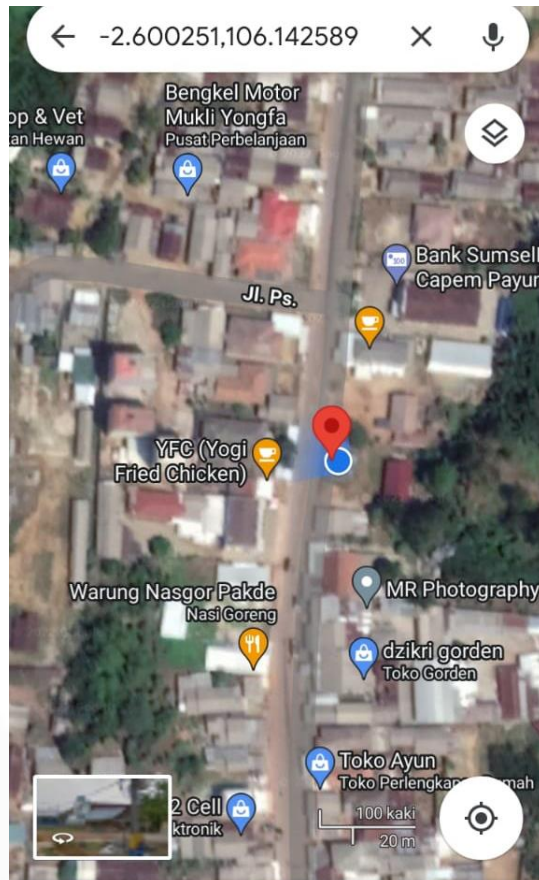
Location Details

Latitude	-2.600310
Longitude	106.142609
Date	29 / 07 / 2022
Time	09 : 22 : 28 AM

[Click here!](#) To check the location in Google maps.

Gambar 4.9 Lokasi kota k amal ditampilkan oleh modul NodeMCU ESP8266

Untuk lokasi pertama kotak amal yang ditampilkan oleh modul GPS dapat diakses menggunakan titik koordinat yang muncul pada Modul NodeMCU pada *google maps*. Di bawah ini merupakan lokasi yang ditampilkan oleh *google maps* serta dapat dilihat pada Gambar 4.10 berikut ini.



Gambar 4.10 Lokasi kotak amal yang ditampilkan oleh *Google Maps*

Dapat dilihat bahwa titik koordinat yang ditampilkan pada data *smartphone* dengan di internet berbeda, akan tetapi selisih jarak titik koordinat data pada *smartphone* dan internet dapat dihitung menggunakan metode berikut :

Dik : Garis lintang pada internet = -2.600310

Garis lintang pada *smartphone* = -2.600251

Garis bujur pada internet = 106. 142609

Garis bujur pada *smartphone* = 106.142589

$1^\circ = 111,322 \text{ km} = 111322 \text{ meter}$

Dit : Selisih jarak pada *smartphone* dan internet

Jawab : - Selisih garis lintang

$$= -2.600310 - (-2.600251)$$

$$= 0.000059$$

$$= 0.000059 \times 111322$$

$$= 6,5 \text{ meter}$$

- Selisih garis bujur

$$= 106.142609 - 106.142589$$

$$= 0.00002$$

$$= 0.00002 \times 111322$$


$$= 2,2 \text{ meter}$$

- Selisih Jarak sebenarnya

$$\begin{aligned} &= \sqrt{6.5^2 + 2.2^2} \\ &= \sqrt{42.25 + 4.84} \\ &= \sqrt{47.09} \\ &= 6.8 \text{ m} \end{aligned}$$

Pada hitungan diatas, maka dapat disimpulkan bahwa selisih jarak titik koordinat pada *smartphone* dan internet pada lokasi pertama hanya berjarak 6.8 m

Untuk lokasi kedua yang akan ditampilkan oleh modul NodeMCU berupa *latitude*, *longitude*, tanggal dan waktu pengaksesan dilakukan serta dapat dilihat pada Gambar 4.11 berikut.



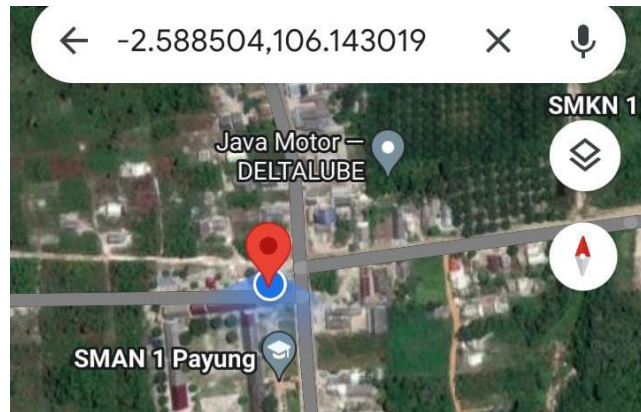
The screenshot shows a web browser interface for a NodeMCU project. At the top, there is a title bar with a warning icon, the text "GPS Interfacing with ...", and the IP address "192.168.35.228". To the right of the title bar are share and bookmark icons. Below the title bar is a large heading "GPS Interfacing with NodeMCU". Underneath the heading is a sub-heading "Location Details". Below this is a table with four rows: Latitude, Longitude, Date, and Time. Below the table is a yellow button labeled "Click here!" followed by the text "To check the location in Google maps."

Location Details	
Latitude	-2.588466
Longitude	106.143044
Date	29 / 07 / 2022
Time	09 : 39 : 04 AM

[Click here!](#) To check the location in Google maps.

Gambar 4.11 Lokasi kedua yang ditampilkan NodeMCU ESP8266

Titik koordinat yang ditampilkan NodeMCU diatas akan diakses dengan mengklik *Click here!* yang akan ditunjukan ke google maps serta dapat dilihat pada Gambar 4.12 di bawah ini.



Gambar 4.12 Lokasi kedua yang ditampilkan pada *google maps*

Dapat dilihat bahwa titik koordinat yang ditampilkan pada data *smartphone* dengan di internet berbeda, akan tetapi selisih jarak titik koordinat data pada *smartphone* dan internet dapat dihitung menggunakan metode berikut :

Dik : Garis lintang pada internet = -2.588466

Garis lintang pada *smartphone* = -2.888504

Garis bujur pada internet = 106. 143044

Garis bujur pada *smartphone* = 106.143019

1° = 111,322 km = 111322 meter

Dit : Selisih jarak pada *smartphone* dan internet

Jawab : - Selisih garis lintang

$$= -2.588508 - (-2.588466)$$

$$= 0.000042$$

$$= 0.000042 \times 111322$$

$$= 4,6 \text{ meter}$$

- Selisih garis bujur

$$= 106.143044 - 106.143019$$

$$= 0.000025$$

$$= 0.000025 \times 111322$$

$$= 2,7 \text{ meter}$$

- Selisih Jarak sebenarnya

$$= \sqrt{4.6^2 + 2.7^2}$$

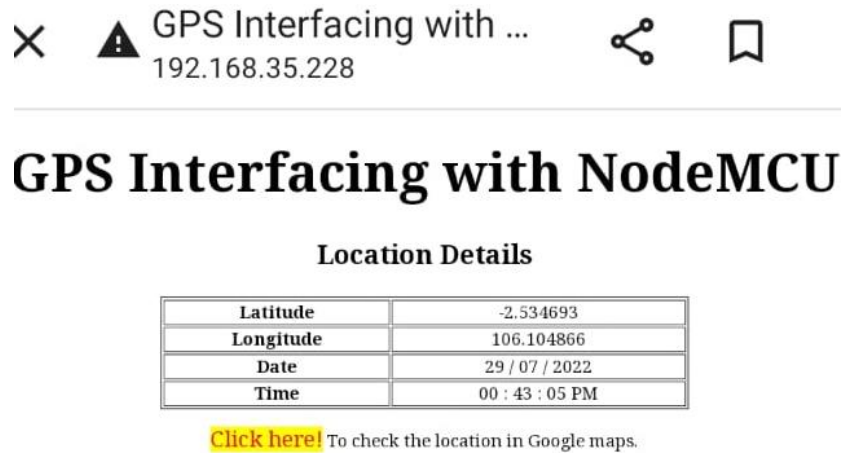
$$= \sqrt{21.16 + 7.29}$$

$$= \sqrt{28.45}$$

$$= 5,3 \text{ m}$$

Pada hitungan diatas, maka dapat disimpulkan bahwa selisih jarak titik koordinat pada *smartphone* dan internet pada lokasi kedua hanya berjarak 5.3 m.

Untuk lokasi kotak amal yang ketiga dapat dilihat pada NodeMCU berupa *latitude*, *longitude*, tanggal dan waktu pengaksesan tersebut serta dapat dilihat pada Gambar 4.13 berikut.



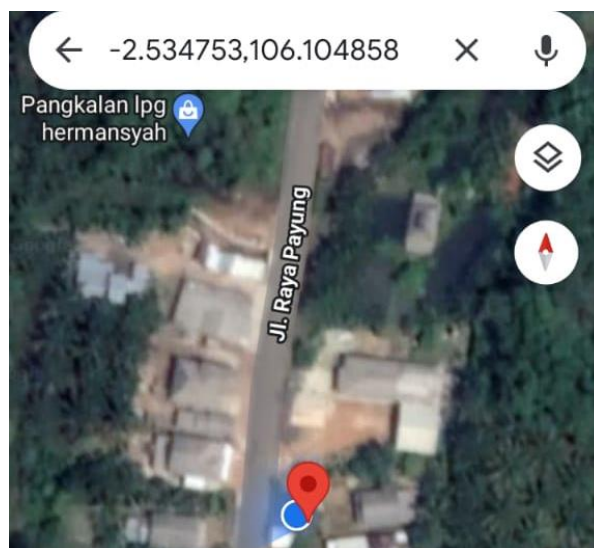
The screenshot shows a web browser window with the title "GPS Interfacing with ..." and the IP address "192.168.35.228". Below the title is a large heading "GPS Interfacing with NodeMCU". Underneath is a section titled "Location Details" containing a table with the following data:

Latitude	-2.534693
Longitude	106.104866
Date	29 / 07 / 2022
Time	00 : 43 : 05 PM

Below the table is a yellow button labeled "Click here!" followed by the text "To check the location in Google maps."

Gambar 4.13 Lokasi ketiga yang ditampilkan NodeMCU

Akurasi antara data yang ditampilkan oleh NodeMCU dengan yang ditampilkan oleh *google maps* yaitu 4 angka pertama *latitude* pada NodeMCU sama dengan 4 angka pertama yang ditampilkan *google maps* dan 6 angka pertama *longitude* yang ditampilkan oleh NodeMCU sama dengan 6 angka pertama yang ada di *google maps* serta untuk lokasi yang ditampilkan *google maps* dapat dilihat pada Gambar 4.14 di bawah ini.



Gambar 4.14 Lokasi ketiga yang ditampilkan di *google maps*

Dapat dilihat bahwa titik koordinat yang ditampilkan pada data *smartphone* dengan di internet berbeda, akan tetapi selisih jarak titik koordinat data pada *smartphone* dan internet dapat dihitung menggunakan metode berikut :

Dik : Garis lintang pada internet = -2.534693

Garis lintang pada *smartphone* = -2.534753

Garis bujur pada internet = 106. 104866

Garis bujur pada *smartphone* = 106.104858

$1^\circ = 111,322 \text{ km} = 111322 \text{ meter}$

Dit : Selisih jarak pada *smartphone* dan internet

Jawab : - Selisih garis lintang

$$= -2.534693 - (-2.534753)$$

$$= 0.00006$$

$$= 0.00006 \times 111322$$

$$= 6,6 \text{ meter}$$

- Selisih Jarak sebenarnya

$$= \sqrt{6.6^2 + 0.8^2}$$

$$= \sqrt{43.56 + 0.64}$$

$$= \sqrt{44.2}$$

$$= 6.6 \text{ m}$$

- Selisih garis bujur

$$= 106.104866 - 106.104858$$

$$= 0.000008$$

$$= 0.000008 \times 111322$$

$$= 0,8 \text{ meter}$$

Pada hitungan diatas, maka dapat disimpulkan bahwa selisih jarak titik koordinat pada *smartphone* dan internet pada lokasi ketiga hanya berjarak 6.6 m.

Untuk lokasi keempat kotak amal akan ditampilkan oleh NodeMCU berupa *latitude*, *longitude*, tanggal dan waktu pengaksesan tersebut serta dapat dilihat pada Gambar 4.15 di bawah ini.



GPS Interfacing with NodeMCU

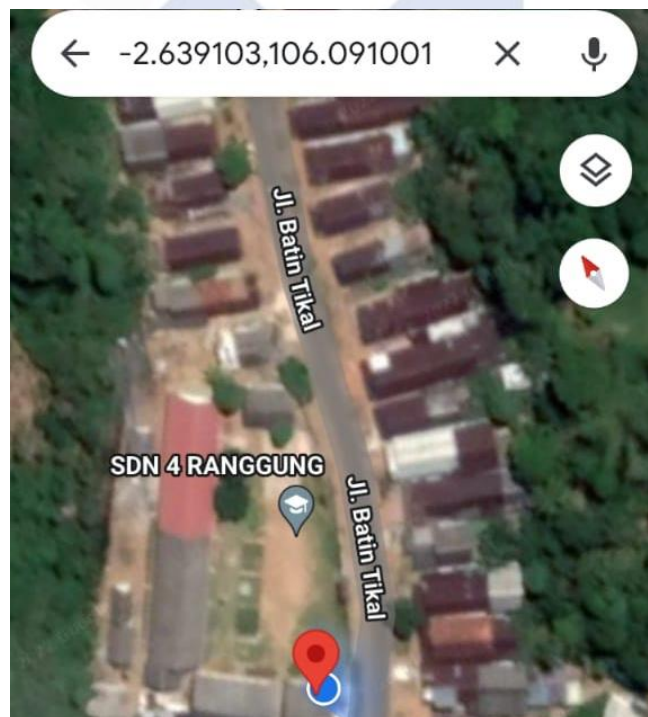
Location Details

Latitude	-2.639058
Longitude	106.091003
Date	29 / 07 / 2022
Time	01 : 19 : 03 PM

[Click here!](#) To check the location in Google maps.

Gambar 4.15 Lokasi keempat yang ditampilkan oleh NodeMCU

Akurasi data lokasi yang ditampilkan antara NodeMCU dan lokasi yang ditampilkan oleh *google maps* dengan 4 angka pertama *latitude* yang sama serta 6 angka pertama *longitude* juga sama serta untuk lokasi yang ditampilkan oleh *google maps* dapat dilihat pada Gambar 4.16 di bawah ini.



Gambar 4.16 Lokasi keempat yang ditampilkan *google maps*.

Dapat dilihat bahwa titik koordinat yang ditampilkan pada data *smartphone* dengan di internet berbeda, akan tetapi selisih jarak titik koordinat data pada *smartphone* dan internet dapat dihitung menggunakan metode berikut :

Dik : Garis lintang pada internet = -2.639058

Garis lintang pada *smartphone* = -2.639103

Garis bujur pada internet = 106.091003

Garis bujur pada *smartphone* = 106.091001

$1^\circ = 111,322 \text{ km} = 111322 \text{ meter}$

Dit : Selisih jarak pada *smartphone* dan internet

Jawab : - Selisih garis lintang

$$= -2.639103 - (-2.639058)$$

$$= 0.000045$$

$$= 0.000045 \times 111322$$

$$= 5 \text{ meter}$$

- Selisih Jarak sebenarnya

$$= \sqrt{5^2 + 0.2^2}$$

$$= \sqrt{25 + 0.4}$$

$$= \sqrt{25.4}$$

$$= 5.03 \text{ m}$$

- Selisih garis bujur

$$= 106.091003 - 106.091001$$

$$= 0.000002$$

$$= 0.000002 \times 111322$$

$$= 0,2 \text{ meter}$$

Pada hitungan diatas, maka dapat disimpulkan bahwa selisih jarak titik koordinat *smartphone* dan internet pada lokasi keempat hanya berjarak 5.03 m.

Pengujian GPS dilakukan menggunakan pemrograman arduino IDE dengan list berikut ini:

```
while (ss.available() > 0)
  if (gps.encode(ss.read()))
  {
    if (gps.location.isValid())
    {
      latitude = gps.location.lat();
      lat_str = String(latitude , 6);
      longitude = gps.location.lng();
      lng_str = String(longitude , 6);
```

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan data hasil rancang sistem keamanan kotak amal masjid berbasis mikrokontroler yang menggunakan *fingerprint* sebagai akses keamanan dan modul NodeMCU ESP8266 sebagai akses pencari lokasi kotak amal. Maka dapat disimpulkan bahwa :

1. Kami melakukan ujicoba pada *fingerprint* menggunakan sidik jari yang dimasukkan ke dalam daftar dan tidak dimasukkan ke dalam daftar. Hasil pengujian yang dilakukan menunjukkan bahwa *fingerprint* dalam kondisi baik dan bisa mendeteksi sidik jari.
2. Penampilan koordinat menggunakan modul NodeMCU ESP8266 dapat terbaca pada *google maps* berupa lokasi terbaru pengaksesan koordinat kotak amal masjid dengan *latitude* dan *longitude* yang sama.

5.2 Saran

Saran-saran yang dapat diajukan untuk perancangan proyek akhir adalah sebagai berikut :

1. Untuk bahan yang digunakan dapat diubah menggunakan bahan yang lebih baik guna menjamin tingkat keamanan kotak amal.
2. Dapat menambahkan sensor getar sehingga pada saat kotak amal dirubah posisinya dapat mengeluarkan suara peringatan.
3. Volume kotak amal dapat diubah sesuai dengan volume kotak amal pada umumnya.

DAFTAR PUSTAKA

- Pengenalan GPS*. (2014, April 6). Retrieved July 2022, 14, from bpsdm.pu.go.id:https://bpsdm.pu.go.id/center/pelatihan/uploads/edok/2018/01/8733f_pengenalan_GPS.pdf#:~:text=PENGENALAN%20GPS%20GPS%20%28Global%20Positioning%20System%29%20adalah%20sistem,posisi%20yang%20dimiliki%20dan%20dikelola%20oleh%20Amerika%20Serikat.
- Akbar, K. (2020). *Sistem Keamanan Kotak Amal Anti Maling Berbasis Arduino*. Sumatera Utara: STMIK ROYAL.
- Akil, M., Muchtar, A., & Fitriati, A. (2020). Desain Kotak Amal Mesjid Tanpa Sentuh Dalam Upaya Menghadapi Tatanan Normal Baru Pada Tempat Ibadah. *Jurnal MediaTIK : Jurnal Media Pendidikan Teknik Informatika dan Komputer*, 57-63.
- Dita, P. E., Fahrezi, A. A., Prasetyawan, P., & Amarudin. (2021). Sistem Keamanan Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik dan Sistem Komputer (JTikom)*, 121-135.
- Kurniawan, M. I., Sunarya, U., & Tulloh, R. (2018). Internet of Things : Sistem Keamanan Rumah Berbasis Raspberry Pi dan Telegram Messenger. *ELKOMIKA*, 1-15.
- Nugroho, A., & Almasri. (2021). Alat Keamanan Kotak Amal Untuk Mengatasi Pencurian Berbasis GSM. *Jurnal Vocational Teknik Elektronika dan Informatika*, 52-60.
- Nursaleh, F. (2019). Rancang Bangun Kotak Amal Anti Maling Menggunakan SMS Gateway Berbasis Mikrokontroler. *Fakultas Sains dan Teknologi UIN Alauidin MAkaasar*.
- Pambudi, B. C. (2020). Pengaman Kotak Amal Masjid Dilengkapi GPS dan SMS Gateway. *Program Studi Teknik Elektronika Fakultas Teknik Universitas Muhammadiyah Diponogoro*, 7-9.

- Putra Perwira, P. M. (2018). Redesain Masjid Kompleks Besar Jatinom Dengan Pendekatan Infill Desain Untuk Fasilitas Pendukung Masjid. *Universitas Negeri Yogyakarta*, 15-62.
- Qalbi, N. I., Rasyid, C. W., Nurdinah, N. I., Muhira, AR, W. A., Kaswar, A. B., et al. (2020). Rancang Bangun Kotak Amal Cerdas Sebagai Solusi Ketidak Efisienan Pendistribusi Kotak Amal Di Masjid. *Jurnal MEDIA ELEKTRIK*, 25-32.
- Ramadhan, A. S., & Handoko, L. B. (2016). Rancang Bangun Sistem Keamanan Rumah Berbasis Arduino Mega 2560. *Techo.COM*, 117-124.
- Riadi, M. (2017). *GPS (Global Positioning System)*. Retrieved Juni 28, 2022, from Kajian Pustaka: <https://www.kajianpustaka.com/2017/09/gps-global-positioning-system.html>
- Saleh, M., & Haryanti, M. (2017). Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay. *Teknologi Elektro Universitas Mercu Buana*, 181-186.
- Somba, I. A. (2015, Februari 12). *Keamanan*. Retrieved Juli 14, 2022, from Universitas Pasundan: <https://www.unpas.ac.id/keamanan/>
- Sulistiyowati, R., & Febriantoro, D. D. (2012). Perancangan Prototype Sistem Kontrol dan Monitoring Pembatas Daya Listrik Berbasis Mikrokontroler. *Jurnal IPTEK*, 24-32.
- Tullah, R., Mustofa, S. M., & Nugraha, D. E. (2019). Sistem Keamanan Rumah Berbasis Mikrokontroler Arduino dan SMS Gateway. *Academic Jurnal Of Computer Sciece Research*, 7-12.
- Wishnuadji, T., Narendro, A., & Peristiwa, H. (2022). Kontak Penyimpanan Dengan Sistem Kemananan Berbasis Arduino. *Seminar Nasional Riset dan Inovasi Teknologi*, 947-952.



LAMPIRAN

Lampiran 1 DAFTAR RIWAYAT HIDUP



DAFTAR RIWAYAT HIDUP

1. *Data Pribadi*

Nama Lengkap : Kevin Francisko
Tempat Tanggal Lahir : Pangkal Pinang, 27
Agustus 2001
Alamat Rumah : RT 14, RW 2 Desa Payung
Kec. Payung
Kab Bangka Selatan
No Handphone : 082169278422
Email : charlessholihah45@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. *Riwayat Pendidikan*

SDN 1 PAYUNG	Lulus 2013
SMPN 1 PAYUNG	Lulus 2016
SMA NEGERI 1 PAYUNG	Lulus 2019

3. *Pengalaman Kerja*

Praktik kerja lapangan di PT. Citra Plastik Makmur	Tahun 2021
--	------------

4. *Pengetahuan Bahasa*

Bahasa Indonesia dan Bahasa Inggris

Sungailiat, ... Agustus 2022

Kevin Francisko

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Deta
Khashanrusnuria
Anggelia

Tempat Tanggal Lahir : Cit, 24 September
2001

Alamat Rumah : Air Tenggiling

No Handphone : 083160073981

Email : detaangelia933@gmail.com

Jenis Kelamin : Perempuan

Agama : Islam



2. Riwayat Pendidikan

SDN 9 RIAU SILIP Lulus 2013

SMPN 2 SUNGAILIAT Lulus 2016

SMA NEGERI 1 SUNGAILIAT Lulus 2019

3. Pengamalan Kerja

Praktik kerja lapangan di PT. Garba General Kontraktor Tahun 2021

4. Pengetahuan Bahasa

Bahasa Indonesia dan Bahasa Inggris

Sungailiat, ... Agustus 2022

Deta Khashanrusnuria Anggelia

LAMPIRAN 2

Lampiran 2 PROGRAM KESELURUHAN



PENGUJIAN *FINGER PRINT*

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  finger.getTemplateCount();
  Serial.print("Sensor          contains          ");
  Serial.print(finger.templateCount); Serial.println(" templates");
  Serial.println("Waiting for valid finger...");
}

void loop()          // run over and over again
{
  getFingerprintIDez();
  delay(50);          //don't need to run this at full speed.
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }
}
```

```

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_NOTFOUND) {
  Serial.println("Did not find a match");
  return p;
} else {
  Serial.println("Unknown error");
  return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print("      with      confidence      of      ");
Serial.println(finger.confidence);

  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.fingerFastSearch();

```

```
if (p != FINGERPRINT_OK) return -1;

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print("      with      confidence      of      ");
Serial.println(finger.confidence);
return finger.fingerID;
}
```



PENGUJIAN *FINGER PRINT ENROLL*

```
#include "Adafruit_Fingerprint.h"
#include "SoftwareSerial.h"
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
uint8_t id;

void setup() {
  Serial.begin(9600);
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("FingerPrint Sensor Ditemukan!");
  } else {
    Serial.println("FingerPrint Sensor Tidak Ditemukan! :(");
    while (1) {
      delay(1);
    }
  }
}

void loop() {
  ENROLL();
}

//-----ENROLL-----
//-----//
void ENROLL() {
  Serial.println("Ready to enroll a fingerprint!");
  Serial.println("Please type in the ID # (from 1 to 127) you want
to save this finger as...");
  id = readnumber();
  if (id == 0) { // ID #0 not allowed, try again!
    return;
  }
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (! getFingerprintEnroll() );
}

//-----PROSES ENROLL-----
//-----//

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}

//----RETURN NUM-----//
```

```

uint8_t getFingerprintEnroll() {
    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #");
    Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }

    // OK success!

    p = finger.image2Tz(1);
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:
            Serial.println("Image too messy");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            return p;
        case FINGERPRINT_FEATUREFAIL:
            Serial.println("Could not find fingerprint features");
            return p;
        case FINGERPRINT_INVALIDIMAGE:
            Serial.println("Could not find fingerprint features");
            return p;
        default:
            Serial.println("Unknown error");
            return p;
    }

    Serial.println("Remove finger");
    delay(2000);
    p = 0;
    while (p != FINGERPRINT_NOFINGER) {
        p = finger.getImage();
    }
}

```

```

Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.print(".");
      break;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      break;
    default:
      Serial.println("Unknown error");
      break;
  }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
  Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
}

```



```
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}
}

//-----ENROLL END-----
```

PENGUJIAN *FINGER PRINT LOCK DOOR*

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
int state = 0;

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

#define hidup HIGH
#define mati LOW
int relay = 4;
int buzzer = 5;
int bacaSensor = 8;

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  finger.begin(57600);

  pinMode(buzzer, OUTPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(buzzer, LOW);
  digitalWrite(relay, mati);
  delay(5);

  if (finger.verifyPassword()) {
    lcd.init();
    lcd.backlight();
    lcd.setCursor(3,0);
    lcd.print("verifikasi");
    //Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Tidak ditemukan Sensor, Periksa kembali!");
    while (1) {
      delay(1);
    }
  }

  finger.getTemplateCount();
}

void loop(){ // run over and over again

  int bacaSensor = digitalRead(8);
  if (bacaSensor == HIGH && state == 0)
  {
    digitalWrite(buzzer, HIGH);
    state = 1;
    lcd.init();
  }
}
```

```

lcd.backlight();
lcd.setCursor (4,0);
lcd.print("waspada");
  lcd.setCursor (3,1);
lcd.print("pencurian");
delay (100);
}
else if (bacaSensor == LOW)
{
  state = 0;
  digitalWrite (buzzer, LOW);
  lcd.init();
lcd.backlight();
lcd.setCursor (1,0);
lcd.print("Kotak Terbuka");
delay (100);
}

getFingerprintID();
delay(50);          //don't ned to run this at full speed.
}

uint8_t getFingerprintID() {
uint8_t p = finger.getImage();
switch (p) {
  case FINGERPRINT_OK:
    //Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    //Serial.println("No finger detected");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unkown error");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    //Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");

```

```

    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
    digitalWrite(relay, HIGH);
    digitalWrite (bacaSensor , HIGH);
    lcd.init();
    lcd.backlight();
    lcd.setCursor (1,0);
    lcd.print("Akses Diterima");
    digitalWrite(relay, LOW);
    lcd.init();
    lcd.backlight();
    lcd.setCursor (1,0);
    lcd.print("verifikasi Finger");

} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;

} else if (p == FINGERPRINT_NOTFOUND) {
    digitalWrite(buzzer, HIGH);
    delay(250);
    digitalWrite(buzzer, LOW);
    delay(175);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    delay(500);
    lcd.init();
    lcd.backlight();
    lcd.setCursor (1,0);
    lcd.print("Akses Ditolak");
    delay (2500);
    lcd.setCursor (0,0);
    lcd.print("Verifikasi Finger");
    delay (5000);

    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!

```

```

Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print("          with          confidence          of          ");
Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print("          with          confidence          of          ");
Serial.println(finger.confidence);
return finger.fingerID;
}

```



PENGUJIAN GPS

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>

TinyGPSPlus gps; // The TinyGPS++ object

SoftwareSerial ss(4, 5); // The serial connection to the GPS
device

const char* ssid = "Sukjai";
const char* password = "oil30723072#";

float latitude , longitude;
int year , month , date, hour , minute , second;
String date_str , time_str , lat_str , lng_str;
int pm;

WiFiServer server(80);
void setup()
{
  Serial.begin(115200);
  ss.begin(9600);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop()
{
  while (ss.available() > 0)
  if (gps.encode(ss.read()))
  {
    if (gps.location.isValid())
    {
      latitude = gps.location.lat();
      lat_str = String(latitude , 6);
      longitude = gps.location.lng();
```

```

    lng_str = String(longitude , 6);
}

if (gps.date.isValid())
{
    date_str = "";
    date = gps.date.day();
    month = gps.date.month();
    year = gps.date.year();

    if (date < 10)
        date_str = '0';
    date_str += String(date);

    date_str += " / ";

    if (month < 10)
        date_str += '0';
    date_str += String(month);

    date_str += " / ";

    if (year < 10)
        date_str += '0';
    date_str += String(year);
}

if (gps.time.isValid())
{
    time_str = "";
    hour = gps.time.hour();
    minute = gps.time.minute();
    second = gps.time.second();

    minute = (minute + 30);
    if (minute > 59)
    {
        minute = minute - 60;
        hour = hour + 1;
    }
    hour = (hour + 5);
    if (hour > 23)
        hour = hour - 24;

    if (hour >= 12)
        pm = 1;
    else
        pm = 0;

    hour = hour % 12;

    if (hour < 10)
        time_str = '0';
    time_str += String(hour);

    time_str += " : ";
}

```

```

    if (minute < 10)
        time_str += '0';
    time_str += String(minute);

    time_str += " : ";

    if (second < 10)
        time_str += '0';
    time_str += String(second);

    if (pm == 1)
        time_str += " PM ";
    else
        time_str += " AM ";

}

}
// Check if a client has connected
WiFiClient client = server.available();
if (!client)
{
    return;
}

// Prepare the response
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n
<!DOCTYPE html> <html> <head> <title>GPS Interfacing with
NodeMCU</title> <style>";
s += "a:link {background-color: YELLOW;text-decoration: none;}";
s += "table, th, td {border: 1px solid black;} </style> </head>
<body> <h1 style="";
s += "font-size:300%;";
s += " ALIGN=CENTER> GPS Interfacing with NodeMCU</h1>";
s += "<p ALIGN=CENTER style="";
s += "> <b>Location Details</b></p> <table ALIGN=CENTER style="";
s += "width:50%";
s += "> <tr> <th>Latitude</th>";
s += "<td ALIGN=CENTER >";
s += lat_str;
s += "</td> </tr> <tr> <th>Longitude</th> <td ALIGN=CENTER >";
s += lng_str;
s += "</td> </tr> <tr> <th>Date</th> <td ALIGN=CENTER >";
s += date_str;
s += "</td></tr> <tr> <th>Time</th> <td ALIGN=CENTER >";
s += time_str;
s += "</td> </tr> </table> ";

if (gps.location.isValid())
{
    s += "<p align=center><a style="";
s += "color:RED;font-size:125%;""
href="";
s += "http://maps.google.com/maps?&z=15&mrt=yp&t=k&q="";
s += "+";

```



```
s += lng_str;
s += "" target=""_top"">Click here!</a> To check the location
in Google maps.</p>";
}

s += "</body> </html> \n";

client.print(s);
delay(100);
}
```

