



TUGAS AKHIR - EE 184801

SISTEM PEMANTAUAN GUNUNG BERAPI BERBASIS IOT MENGGUNAKAN NODEMCU DAN LORA

Hendrik Hermawan
NRP 0711154000011

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE 184801

**SISTEM PEMANTAUAN GUNUNG BERAPI BERBASIS
IOT MENGGUNAKAN NODEMCU DAN LORA**

Hendrik Hermawan
NRP 0711154000011

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EE 184801

***VOLCANO MONITORING SYSTEM BASE ON IOT USING
NODEMCU AND LORA***

Hendrik Hermawan
NRP 0711154000011

Supervisor
Dr. Muhammad Rivai, ST., MT.

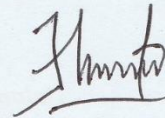
ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Sistem Pemantauan Gunung Berapi Berbasis IoT Menggunakan NodeMCU dan LoRa” 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, 1 Juli 2019



Hendrik Hermawan
NRP 0711 15 40000 011

.....*Halaman ini sengaja dikosongkan*.....

**SISTEM PEMANTAUAN GUNUNG BERAPI
BERBASIS IOT MENGGUNAKAN NODEMCU DAN
LORA**

TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro**

Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Dr. Muhammad Rivai, S.T., M.T.
NIP. 196904261994031003



.....*Halaman ini sengaja dikosongkan*.....

SISTEM PEMANTAUAN GUNUNG BERAPI BERBASIS IOT MENGGUNAKAN NODEMCU DAN LORA

Nama : Hendrik Hermawan
Pembimbing : Dr. Muhammad Rivai, ST., MT.

ABSTRAK

Sistem Pemantauan Aktivitas Gunung berapi berbasis IoT menggunakan NodeMCU dan LoRa merupakan sistem yang dapat memantau aktivitas gunung berapi dengan menempatkan beberapa sensor dan mengirimkan data sensor yang didapat menggunakan LoRa. Pembuatan alat ini didasarkan oleh posisi Indonesia yang berada pada cincin api sehingga membuat Indonesia mempunyai banyak gunung berapi sehingga rawan terjadinya letusan, baik letusan yang mengeluarkan material vulkanik, maupun gas vulkanik yang berbahaya. Seperti pada kawah Sileri di Dieng yang mengeluarkan gas beracun yang memakan beberapa korban. Pemantauan aktivitas seismik tidak efektif jika terjadi letusan yang mengeluarkan gas tanpa ada aktivitas seismik. Oleh karena itu pemantauan aktivitas gunung berapi juga dapat dipantau dari peningkatan konsentrasi gas. Sensor yang digunakan adalah sensor gas jenis MQ135 dan MQ136. Dalam tugas akhir ini, akan mengimplementasikan sistem IoT untuk pemantauan aktivitas gunung berapi. Dalam pembuatan alatnya menggunakan modul Arduino. Sensor akan dipasang di puncak gunung, sensor terdiri dari dua sensor gas. Untuk pengiriman datanya menggunakan modul LoRa. Setelah data diterima, maka akan diolah, kemudian diteruskan ke NodeMCU dan diteruskan lagi ke aplikasi yang ada pada *smartphone*. pengujian sensor gas dilakukan beberapa bahan seperti asap dari kertas, asap kayu, dan uap larutan CaC_2 . Hasil yang didapatkan antara asap kayu dan asap kertas bedanya tidak begitu terlihat. Pada wilayah Surabaya jarak yang bisa di jangkau adalah sekitar 600 meter. Pada pengujian keseluruhan di wilayah gunung Bromo alat dapat melakukan komunikasi sejauh sekitar 3 Km tanpa adanya halangan. Sedangankan konsentrasi gas yang terukur pada saat pengujian relatif naik turun. Hal ini bisa disebabkan oleh faktor alam terutama faktor tiupan angin. Respon dari sensor gas cenderung lama dikarenakan prinsip kerja dari sensor itu sendiri.

Kata Kunci: LoRa, Konsentrasi Gas, Pemantauan Aktivitas Gunung Berapi

.....*Halaman ini sengaja dikosongkan*.....

VOLCANO MONITORING SYSTEM BASE ON IOT USING NODEMCU AND LORA

Name : Hendrik Hermawan
Supervisor : Dr. Muhammad Rivai, ST., MT.

ABSTRACT

Volcanic monitoring system base-on IoT using NodeMCU and LoRa is a system that can monitor volcanic activity by placing several sensors and sending sensor data obtained using LoRa. Making of this system is based on the position of Indonesia in the ring of fire, which makes Indonesia has many volcanoes which are prone to eruptions, both eruptions that emit volcanic material or dangerous volcanic gases. Like in the crater of Sileri in Dieng, which emits toxic gases that consume several victims. Seismic activity monitoring is not effective if there is an eruption that emits gas without seismic activity. Therefore, monitoring volcanic activity can also be monitored from increasing gas concentration. The sensors are used for sensing gas is MQ135 gas sensor and MQ136 gas sensor. In this final project, it will implement an IoT system for monitoring volcanic activity. This device is made using the Arduino module. The sensor will be installed on the mountaintop, the sensor consists of two gas sensors. LoRa module is used for data transmission. After the data is received, data will be processed with Arduino module, and then forwarded to the NodeMCU and then forwarded again to the application on the smartphone. testing gas sensors are carried out by several materials such as smoke from paper, wood smoke, and steam from CaC₂ solvent. The results obtained between wood smoke and paper smoke are not so obvious. In the Surabaya area, the distance that can be covered by LoRa module is around 600 meters. In the overall testing in the Bromo mountain area, communication devices of LoRa module can do communication about 3 kilometers without any obstacles. While the gas concentration measured at the time of testing is relatively up and down. This can be caused by natural factors, especially the wind blowing factor. The response of the gas sensor tends to be slow because of the working principle of the sensor itself.

Keywords: Gas Concentration, LoRa, Volcanic Activity Monitoring

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Segala puji syukur kepada Allah SWT yang telah memberikan nikmat dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul “**Sistem Pemantauan Gunung Berapi Berbasis IoT Menggunakan NodeMCU dan LoRa**”, sebagai salah satu persyaratan dalam menyelesaikan pendidikan program studi S1 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dan memberikan dukungan dalam penulisan dan penyusunan laporan tugas akhir ini. Terutama kepada semua dosen dan teman-teman yang telah membantu penulis menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih yang tulus dan sebesar-besarnya kepada:

1. Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing yang telah membimbing dan memberikan saran selama pengerjaan dan penulisan laporan tugas akhir.
2. Astria Nur Irfansyah, ST., M.Eng., Ph.D., Fajar Budiman, ST., M.Sc., Ir. Harris Pirngadi, MT., Ir. Tasripan, MT. sebagai dosen penguji yang telah menyetujui dan memberikan masukan selama melakukan revisi.
3. Kepala Departemen Teknik Elektro ITS, Dr. Eng. Ardyono Priyadi, ST., M.Eng. atas izin dan kesempatan yang diberikan kepada penulis untuk melaksanakan tugas akhir ini.
4. Orang tua yang selalu memberikan dukungan dan doa kepada penulis.
5. Teman-teman yang tidak dapat penulis sebutkan namanya satu-persatu.

Penulis berharap agar tugas akhir ini dapat memberikan manfaat kepada siapapun yang membacanya. Penulis juga menyadari bahwa masih banyak kekurangan dalam penulisan laporan tugas akhir ini. Oleh karena itu penulis menerima setiap kritik dan saran yang diberikan. Akhir kata penulis mengucapkan terima kasih yang sebesar-besarnya.

Surabaya, 1 Mei 2019

Penulis

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	i
ABSTRAK	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II TINJAUAN PUSTAKA	7
2.1 Kandungan Gas Pada Gunung Berapi	7
2.2 Sensor Gas <i>H2S</i>	8
2.3 Sensor Gas <i>CO2</i>	10
2.4 LoRa RF96	11
2.5 <i>Microcontroller</i> Arduino Uno	13
2.6 NodeMCU ESP6288	16
2.7 Sel Surya	17
2.8 LCD I2C 16X2	17
2.9 Komunikasi SPI	19
2.10 Komunikasi I2C	20
2.11 Komunikasi Serial	21
2.11.1 TX - UART Transmitter	22
2.11.2 RX – UART Receiver	23
2.12 Aplikasi Blynk	24
2.12.1 Blynk Apps	24
2.12.2 Blynk Server	25
2.12.3 Blynk Library	25
BAB III PERANCANGAN SISTEM	27
3.1 Gambaran Umum Sistem	27
3.2 Diagram Blok Sistem	27
3.3 Perancangan Perangkat Keras	28

3.3.1	Sensor Gas <i>H2S</i>	28
3.3.2	Sensor Gas <i>CO2</i>	29
3.3.3	LoRa RF96.....	30
3.3.4	LCD 16X2 I2C	32
3.3.5	NodeMCU.....	33
3.3.6	<i>Board Microcontroller Transmitter</i>	34
3.3.7	<i>Board Microcontroller Receiver</i>	36
3.4	Perancangan Perangkat Lunak.....	37
3.4.1	Pembacaan Sensor MQ136	37
3.4.2	Pembacaan Sensor MQ135	39
3.4.3	Pengiriman Data dengan Lora	42
3.4.4	Penerimaan Data dengan LoRa.....	43
3.4.5	Pengiriman Data ke NodeMCU	43
3.4.6	Pengolahan Data di NodeMCU	44
3.4.7	Pengiriman Data ke Aplikasi Blynk	44
3.5	Perancangan Kotak Packaging.....	45
3.5.1	Kotak Packaging Transmitter	45
3.5.2	Kotak Packaging Receiver.....	46
3.5.3	Rangka Penopang Sel Surya dan Kotak Transmitter	47
BAB IV PENGUJIAN DAN ANALISIS		49
4.1	Pengujian Sensor Gas	49
4.1.1	Pengujian Sensor Terhadap Udara Bersih	49
4.1.2	Pengujian Sensor Terhadap Asap Kertas	50
4.1.3	Pengujian Sensor Terhadap Asap Kayu	52
4.1.4	Pengujian Sensor Terhadap Larutan <i>CaCl2</i>	54
4.2	Pengujian Sensor per Volume.....	55
4.2.1	Pengujian Sensor MQ135	55
4.2.2	Pengujian Sensor MQ136	57
4.3	Pengujian LoRa	58
4.3.1	Pengujian LoRa dengan Posisi Receiver pada Gedung B lantai 4	58
4.3.2	Pengujian LoRa dengan Posisi Receiver di Jl. Teknik Perkapalan ITS.....	60
4.3.3	Metode Pengukuran Jarak.....	60
4.4	Pengujian Alat Secara Keseluruhan.....	61
4.4.1	Pengambilan Data Pertama	61
4.4.2	Pengambilan Data Kedua.....	63
4.4.3	Pengambilan Data Ketiga	64
4.4.4	Pengambilan Data Keempat.....	66
BAB V PENUTUP		69
5.1	Kesimpulan	69

5.2	Saran	69
	DAFTAR PUSTAKA	71
	LAMPIRAN A Program Arduino	73
	LAMPIRAN B Data Percobaan	85
	LAMPIRAN C Dokumentasi Data Percobaan	99
	LAMPIRAN D Dokumentasi Kegiatan	105
	LAMPIRAN E Datasheet.....	107
	BIODATA PENULIS	123

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR GAMBAR

Gambar 2.1	Rangkaian uji dasar MQ136	9
Gambar 2.2	Kurva sensitivitas sensor MQ136	9
Gambar 2.3	Kurva sensitivitas konsentrasi gas	10
Gambar 2.4	Kurva sensitivitas sensor MQ135	11
Gambar 2.5	Skematik blok diagram	12
Gambar 2.6	Pin diagram RF96	12
Gambar 2.7	Board Arduino Uno	15
Gambar 2.8	NodeMCU dan pinout.....	16
Gambar 2.9	Konstruksi sel surya.....	17
Gambar 2.10	Pin diagram LCD 16X2	18
Gambar 2.11	Modul I2C untuk LCD.....	18
Gambar 2.12	Blok diagram komunikasi SPI	20
Gambar 2.13	Perangkat dengan berbagai tegangan suplai dan berbagi bus yang sama.....	21
Gambar 2.14	Transfer data pada bus I2C	21
Gambar 2.15	Blok diagram UART.....	23
Gambar 2.16	Diagram RX timing	24
Gambar 2.17	Diagram TX timing.....	24
Gambar 2.18	Ilustrasi hubungan antara BLYNK dan perangkat IoT ..	25
Gambar 3.1	Diagram Blok Sistem.....	27
Gambar 3.2	Wiring diagram sensor gas <i>H2S</i> MQ136 dengan Arduino	29
Gambar 3.3	Wiring diagram sensor gas <i>CO2</i> MQ135 dengan Arduino	30
Gambar 3.4	Wiring diagram RF96 ke Arduino Uno	31
Gambar 3.5	Dragino LoRa Shield	32
Gambar 3.6	Wiring diagram LCD 16X2 I2C dengan Arduino Uno....	33
Gambar 3.7	Wiring diagram NodeMCU to Arduino Uno	34
Gambar 3.8	Desain board microcontroller transmitter pada EAGLE..	35
Gambar 3.9	Desain schematic microcontroller transmitter pada EAGLE.....	35
Gambar 3.10	Desain board microcontroller receiver pada EAGLE	36
Gambar 3.11	Desain schematic microcontroller receiver pada EAGLE	37
Gambar 3.12	Kotak transmitter	46
Gambar 3.13	Kotak receiver.....	46
Gambar 3.14	Rangka penopang sel surya.....	47

Gambar 4.1	Grafik tegangan terhadap waktu pada udara bersih.....	50
Gambar 4.2	Grafik konsentrasi terhadap waktu pada udara bersih	50
Gambar 4.3	Grafik tegangan terhadap waktu pada asap kertas	51
Gambar 4.4	Grafik konsentrasi terhadap waktu pada asap kertas	52
Gambar 4.5	Grafik tegangan terhadap waktu pada asap kayu	53
Gambar 4.6	Grafik konsentrasi terhadap waktu pada asap kayu	53
Gambar 4.7	Grafik tegangan terhadap waktu pada larutan $CaCl_2$	55
Gambar 4.8	Grafik konsentrasi terhadap waktu pada larutan $CaCl_2$	55
Gambar 4.9	Grafik hubungan antara kenaikan volume dan konsentrasi	56
Gambar 4.10	Grafik hubungan antara kenaikan volume dan konsentrasi	58
Gambar 4.11	Penerimaan data jarak 50 meter	59
Gambar 4.12	Pengiriman data jarak 50 meter	59
Gambar 4.13	Penerimaan data jarak 200 meter	59
Gambar 4.14	Pengiriman data jarak 200 meter	60
Gambar 4.15	Metode pengukuran jarak	61
Gambar 4.16	Grafik pengambilan data pertama.....	62
Gambar 4.17	Tampilan pengambilan data pertama pada Blynk.....	62
Gambar 4.18	Grafik pengambilan data kedua	63
Gambar 4.19	Tampilan pengambilan data kedua pada Blynk	64
Gambar 4.20	Grafik pengambilan data ketiga.....	65
Gambar 4.21	Tampilan pengambilan data ketiga pada Blynk.....	65
Gambar 4.22	Grafik pengambilan data keempat	66
Gambar 4.23	Tampilan pengambilan data keempat pada Blynk	67

DAFTAR TABEL

Tabel 2.1 Komposisi udara bersih di atmosfer	7
Tabel 2.2 Nilai ambang batas pada udara bebas.....	7
Tabel 2.3 Nomor pin, nama pin, tipe pin dan diskripsi pin	13
Tabel 3.1 Wiring pin antara pin Arduino dan sensor MQ136	29
Tabel 3.2 Wiring pin antara pin Arduino dan sensor MQ135	30
Tabel 3.3 Wiring pin standar antara pin Arduino dan RF96	31
Tabel 3.4 Wiring pin antara Arduino Uno dengan LCD I2C	32
Tabel 3.5 Wiring pin antara Arduino Uno dengan NodeMCU.....	34
Tabel 4.1 Pengujian sensor terhadap udara bersih.....	49
Tabel 4.2 Pengujian sensor terhadap asap kertas.....	51
Tabel 4.3 Pengujian sensor terhadap asap kayu.	52
Tabel 4.4 Pengujian sensor terhadap larutan <i>CaCl2</i>	54
Tabel 4.5 Pengujian sensor terhadap asap Kertas.....	56
Tabel 4.6 Pengujian sensor terhadap asap rokok.....	57
Tabel 4.7 Pengujian pertama LoRa.	58
Tabel 4.8 Pengujian kedua LoRa.....	60

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

1.1 Latar Belakang

Letusan gunung berapi adalah bencana alami yang terjadi pada alam. Bencana ini dapat menimbulkan dampak parah pada ekologi dan ekonomi, juga sering mengancam kelangsungan hidup seluruh komunitas [1]. Posisi Indonesia yang berada pada tiga titik pertemuan lempeng tektonik membuat Indonesia mempunyai banyak gunung berapi dan dikenal sebagai cincin api. Efek dari posisi itu, Indonesia rawan akan meletusnya gunung berapi [2]. Sisi positif dari banyaknya gunung berapi di Indonesia adalah tanah di sekitarnya menjadi subur [3]. Selain letusan yang mengeluarkan material vulkanik, keluarnya gas vulkanik juga berbahaya bagi manusia. Seperti pada kawah Sileri di Dieng yang mengeluarkan gas beracun yang memakan beberapa korban[4]. Pasca letusan freatik kawah Sileri di pegunungan Dieng dari Pusat Vulkanologi dan Mitigasi Bencana Geologi (PVMBG) memberikan pengumuman pada masyarakat dan wisatawan di Kawasan Dieng tentang bahaya gas CO₂ dan H₂S. Walaupun tingkat aktivitas pegunungan Dieng masih Normal, berikut rekomendasi dari Pusat Vulkanologi dan Mitigasi Bencana Geologi (PVMBG). Seperti masyarakat diharapkan tidak melakukan aktivitas di Kawah Timbang, karena adanya ancaman bahaya gas CO₂ dan H₂S yang berbahaya bagi kehidupan. Masyarakat agar waspada jika melakukan penggalian tanah di sekitar Kawah Timbang dengan kedalaman lebih dari 1 (satu) meter karena dari tempat tersebut masih berpotensi terancam bahaya gas CO₂ dan H₂S. Masyarakat agar tetap tenang, tidak terpancing isu-isu terkait dengan aktivitas G. Dieng. Pusat Vulkanologi dan Mitigasi Bencana Geologi selalu berkoordinasi dengan Pemerintah Provinsi Jawa Tengah dan Pemkab Banjarnegara, Pemkab Wonosobo serta Pemkab Batang tentang aktivitas G. Dieng. Masyarakat agar selalu mengikuti arahan dari SATLAK PB dan Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Jawa Tengah. Bagi wisatawan yang mengunjungi kawasan wisata kawah disarankan tidak terlalu mendekat. Sehubungan dengan adanya peningkatan aktivitas vulkanik di kawah Sileri, maka masyarakat dan pengunjung diharapkan untuk meningkatkan kewaspadaan, dengan tidak mendekati kawah Sileri pada jarak 100 meter dari bibir kawah [5]. Oleh karena itu pemantauan aktivitas gunung berapi secara berkala penting untuk dilakukan[6]. Pemantauan aktivitas gunung berapi yang paling umum di Indonesia adalah dengan memantau aktivitas seismik yang ada pada gunung berapi.

Karena anomali seismik yang terjadi pada gunung berapi terkait dengan meletusnya gunung berapi merupakan merupakan hasil pertukaran energi antara gunung berapi dengan lingkungan [7]. Pengamatan pada aktivitas seismik ini yang diamati adalah frekuensi atau seberapa sering terjadinya gempa lokal pada gunung berapi. Semakin sering terjadi gempa lokal pada gunung berapi menandakan kalau aktivitas gunung berapi juga meningkat. Dan aktivitas ini dipantau pada stasiun pemantau yang biasanya terletak di kaki gunung. Jika ada peningkatan aktivitas gunung berapi maka petugas yang berjaga akan memberi peringatan kepada penduduk sekitar. Pengamatan aktivitas seismik ini tidak efektif jika terjadi letusan yang mengeluarkan gas vulkanik berbahaya seperti yang terjadi di dieng, di mana saat terjadi letusan gas yang keluar tidak terdeteksi dan menyebar keluar.

Pada tugas akhir ini, pemantauan gunung berapi dilakukan dengan cara yang berbeda. Jika pada umumnya memantau aktivitas seismik, maka kali ini pengamatan dilakukan dengan mengamati gas terutama gas yang berbahaya bagi manusia. Selain mengamati konsentrasi gas, dalam tugas akhir ini juga mengamati tingkat temperatur yang ada pada gunung berapi. Pengamatan pada suhu dilakukan dengan mengamati kenaikan temperaturnya, sedangkan pengamatan gas dilakukan dengan mengamati kenaikan konsentrasi gas. Data dari sensor-sensor tersebut akan diolah oleh Arduino dan akan ditransmisikan menggunakan modul pengirim. Modul pengirim yang digunakan adalah LoRa. Setelah diolah diterima oleh modul penerima. Kemudian sinyal akan diolah dan dikirim ke aplikasi smartphone.

1.2 Perumusan Masalah

Berdasar latar belakang di atas, dapat dirumuskan masalah sebagai berikut.

1. Bagaimana mendapatkan informasi kandungan gas yang ada di gunung berapi.
2. Bagaimana mendapatkan informasi perubahan suhu yang ada di gunung berapi.
3. Bagaimana mengirim data yang terukur dengan jarak yang jauh.

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut.

1. Pemanfaatan jenis sensor gas terutama untuk gas H_2S .
2. Pemanfaatan jenis sensor gas terutama untuk gas CO_2 .
3. Pemanfaatan jenis sensor temperatur terutama untuk suhu tinggi.
4. Mengirim data jarak jauh agar dapat diterima pada stasiun

penerima menggunakan sistem *LoRa*.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut.

1. Luas area yang dijadikan sampel pada puncak gunung.
2. Parameter yang digunakan adalah kandungan gas dan perubahan temperatur.
3. Analisis yang dilakukan tidak mempertimbangkan aspek eksternal yang mungkin ditimbulkan.
4. Pembahasan yang dilakukan dengan melibatkan dua *node* sensor saja.

1.5 Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. Studi literatur kandungan gas gunung berapi

Studi literatur ini akan berisi pengumpulan serta pengkajian teori, data dan penelitian yang dianggap relevan dan terpercaya untuk mendukung keabsahan tugas akhir ini, terutama literatur tentang kandungan gas yang ada pada gunung berapi. Literatur yang digunakan akan memiliki batasan-batasan tertentu. Yaitu, literatur yang digunakan harus bersumber dari *paper*, jurnal, buku, maupun artikel yang berasal dari badan pemerintahan atau institusi akademik terpercaya.

2. Perancangan keseluruhan sistem monitoring

Perancangan rangkaian sensor dibutuhkan untuk menentukan agar bisa didapatkan data konsentrasi kandungan gas dan kenaikan suhu pada gunung berapi yang optimal. Akan diperlukan pengambilan data-data, agar didapatkan karakteristik sensor gas dan sensor suhu. *Microcontroller* digunakan untuk mengolah data yang terbaca oleh sensor. Hal ini sangat dibutuhkan dikarenakan sinyal data dari sensor berupa sinyal analog dan akan diolah menjadi sinyal digital. Setelah itu dilakukan pembuatan *packaging* yang tepat.

3. Perancangan dan Pengujian sensor gas dengan tabel

Perancangan sensor bisa dengan menggunakan pcb yang sudah dibuat dan didalamnya terdapat modul Arduino, dimana sensor akan disesuaikan tempatnya agar mendapatkan data yang tepat. Pengujian dilakukan dengan gas buatan yang dibuat sedemikian rupa sehingga mirip dengan gas yang ada di gunung berapi.

4. Perancangan dan Pengujian sensor suhu

Perancangan sensor suhu juga bisa dengan menggunakan pcb yang sudah dibuat dan didalamnya terdapat modul Arduino, dimana sensor

akan disesuaikan tempatnya agar mendapatkan data yang tepat. Pengujian dengan membuat suhu buatan yang dibuat sedemikian rupa sehingga mirip dengan suhu yang ada pada gunung berapi.

5. Perancangan dan Pengujian LoRa

Perancangan LoRa juga bisa dengan menggunakan pcb yang sama dengan pcb yang digunakan untuk sensor yang sudah dibuat dan didalamnya terdapat modul Arduino, dimana sistem LoRa akan ditempatkan dengan tepat agar dapat mengirimkan data yang sesuai dengan yang diinginkan. Pada pengujian LoRa akan didapatkan nilai yang dapat dilihat dari jauhnya jarak antara *transmitter* dan *receiver*.

6. Perancangan system di smarthphone

Perancangan system yang ada di smartphone, yaitu merancang sebuah aplikasi berbasis *smarthphone* dari data yang telah diterima dan diolah oleh modul arduino. Perancangan sistem ini membutuhkan suatu server internet pada suatu *mikrokontroler* untuk menyimpan dan memperbarui data konsentrasi gas dan kenaikan suhu. *Software* antarmuka untuk pemantauan aktivitas gunung berapi ini akan dirancang dengan memanfaatkan komponen-komponen yang tersedia pada aplikasi Blynk.

7. Pengujian secara keseluruhan

Pengujian secara keseluruhan dilakukan ketika semua system mulai dari system sensor, system pengirim dan system yang ada pada smartphone telah dibuat. Pengujian dilakukan untuk mengetahui apakah system yang dibuat telah sesuai dengan apa yang diinginkan atau belum.

8. Penyusunan Laporan Tugas Akhir

Tahap penyusunan laporan merupakan tahap terakhir dari proses pengerjaan tugas akhir ini. Laporan berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- BAB I: Pendahuluan
Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.
- BAB II: Tinjauan Pustaka

Bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya teori yang berhubungan dengan komponen yang akan digunakan.

- **BAB III: Perancangan Sistem**
Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk pembuatan alat pada tugas akhir ini.
- **BAB IV: Pengujian Dan Analisis**
Bab ini berisi tentang pengujian alat pada sistem yang telah dibuat dan analisa hasil dari pengujian yang telah didapat.
- **BAB V: Penutup**
Bab ini berisi tentang kesimpulan yang diperoleh dari alat yang telah dibuat serta saran untuk pengembangan lebih lanjut.

1.7 Relevansi

Dengan dibuatnya tugas akhir ini diharapkan akan membantu pemantauan aktivitas gunung berapi, terutama aktivitas gas yang berbahaya bagi manusia. Pemantauan dilakukan dengan memasang beberapa sensor yang diletakkan pada sumber gas pada gunung berapi.

Data sensor akan diolah dan dikirim secara nirkabel dengan menggunakan radio frekuensi. Data yang telah diterima akan dikirimkan ke sebuah aplikasi *smartphone*

.....*Halaman ini sengaja dikosongkan*.....

BAB II TINJAUAN PUSTAKA

2.1 Kandungan Gas Pada Gunung Berapi

Material yang ada pada gunung berapi yang masih aktif, mempunyai tiga jenis material. Material itu adalah lahar (*lava*), kepingan batu (*tephra*), dan gas (*gases*). Tipe dan jumlah material yang dikeluarkan dari gunung berapi aktif berbeda-beda, tergantung dari komposisi magma didalam gunung api tersebut. Komponen penyangga paling penting dari gas vulkanik adalah yang melibatkan H_2S dan SO_2 dari fase penguapan. Konsentrasi gas yang bereaksi cepat, seperti H_2 dan CO , mencerminkan kontrol redoks oleh “buffer gas” ini, gas dengan respon kinetik yang lebih lambat, seperti CH_4 , sebagian besar dihasilkan dalam sistem hidrotermal yang bergerak lambat terkait dengan struktur vulkanik yang paling aktif. Sensor yang dipakai adalah jenis MQ136 untuk gas H_2S dan sensor MQ7 untuk gas CO , karena memiliki sensitivitas yang tinggi [8]. Pada tingkat yang dangkal, kandungan H_2S dan SO_2 dipengaruhi oleh deposisi dan reevaporasi unsur S dan disproporsi SO_2 menjadi H_2S dan sulfat dalam lingkungan hidrotermal[9].

Tabel 2.1 Komposisi udara bersih di atmosfer

Gases	Konsentrasi	
	Volume (%)	PPM
Nitrogen	78,08	780,840
Oksigen	20,95	209,460
Argon	0,934	9,340
Karbondioksida	0,033	330
Neon	0,00180	18
Helium	0,00050	5
Metana	0,00020	2
Kripton	0,00010	1

Tabel 2.2 Nilai ambang batas pada udara bebas [6]

Gas	Ambang Batas pada Udara Bebas (PPM)
SO_2	2
H_2S	10
CO_2	5000
CO	50
CH_4	100000

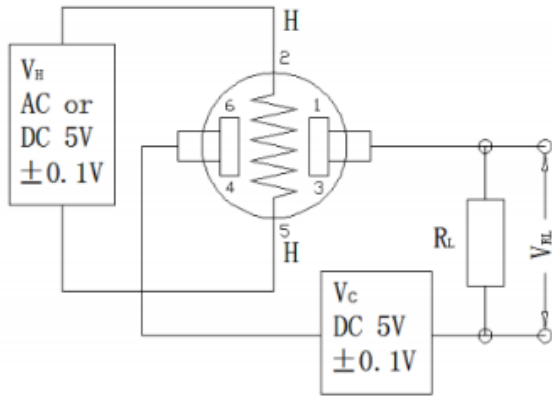
2.2 Sensor Gas H_2S

Material dari sensor gas MQ136 terbuat dari SnO_2 . Ketika sensor berada di udara bersih, sensor akan memiliki konduktivitas yang rendah. Ketika gas H_2S ada, konduktivitas sensor semakin meningkat seiring dengan meningkatnya konsentrasi gas. Pengguna juga dapat mengkonversi perubahan konduktivitas untuk menyesuaikan keluaran sinyal konsentrasi gas melalui sirkuit yang sederhana. Sensor gas MQ136 memiliki sensitivitas yang tinggi terhadap gas H_2S , juga dapat memonitor uap organik termasuk sulfur dengan baik. Sensor ini memiliki jangkauan yang luas dan umur yang relatif Panjang, serta rangkaian pengontrol yang sederhana.

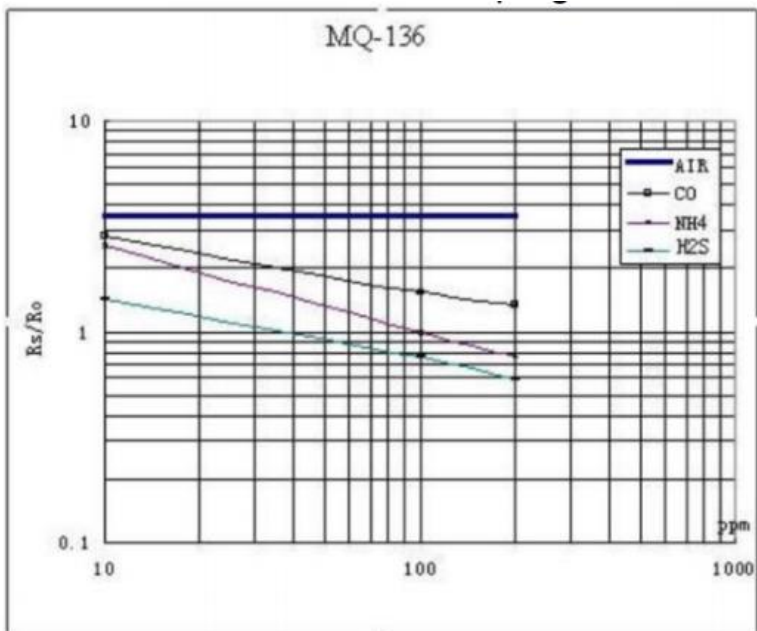
Gambar 2.1. merupakan gambar rangkaian uji dasar MQ136. Sensor memerlukan dua input tegangan, yaitu tegangan pemanas (VH) dan tegangan sirkuit (VC). VH digunakan untuk memasok suhu kerja standar ke sensor, VH dapat disuplai oleh daya DC ataupun AC, sedangkan VRL adalah tegangan resistansi beban RL yang dipasang seri dengan sensor. VC memasok tegangan deteksi untuk resistensi RL yang dimuat dan VC harus memakai daya DC.

Gambar 2.2. merupakan gambar kurva sensitivitas sensor dengan garis ordinatnya adalah rasio resistansi dari sensor (R_s / R_0), sedangkan garis absis adalah konsentrasi gas. R_s merupakan resistansi udara bersih ditambah gas dengan konsentrasi berbeda, sedangkan R_0 resistansi sensor di keadaan udara bersih.

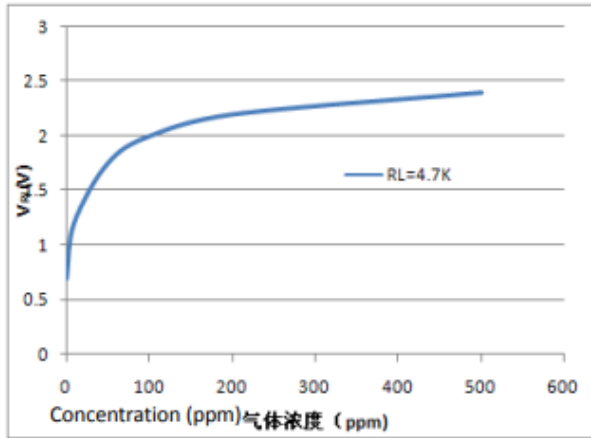
Gambar 2.3. menunjukkan hubungan antara VRL dalam H_2S dengan konsentrasi berbeda. Beban yang dipakai (RL) memiliki resistansi $4,7K\Omega$ dan tes selesai dilakukan di kondisi uji standar. Prinsip kerja dari sensor ini adalah Ketika timah dioksida (partikel semikonduktor) dipanaskan di udara pada suhu tinggi, oksigen diserap di permukaan. Di udara bersih, elektron donor dalam timah dioksida tertarik ke arah oksigen yang diserap pada permukaan bahan sensor. Ini mencegah aliran arus listrik. Dengan adanya gas pereduksi, kerapatan permukaan oksigen yang teradsorpsi berkurang karena bereaksi dengan gas pereduksi. Elektron kemudian dilepaskan ke dalam timah dioksida, yang memungkinkan arus mengalir bebas melalui sensor.



Gambar 2.1 Rangkaian uji dasar MQ136



Gambar 2.2 Kurva sensitivitas sensor MQ136

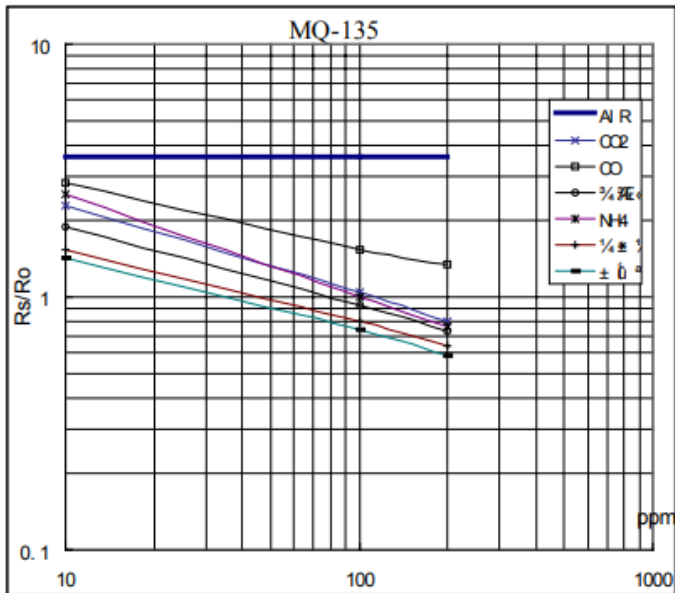


Gambar 2.3 Kurva sensitivitas konsentrasi gas

2.3 Sensor Gas CO_2

Sensor MQ-135 merupakan sebuah modul sensor yang dapat digunakan untuk menentukan kadar konsentrasi gas-gas berbahaya dalam udara. Modul ini berbasiskan sensor MQ-135, yaitu sensor yang dapat mendeteksi gas CO_2 , gas amonia, bensol, alkohol, serta gas berbahaya lainnya. Modul ini cocok digunakan pada proses penentuan kualitas udara (*air quality control*). Sensor yang biasanyadipakai adalah sensor sensor dengan seri MQ. Sensor dengan seri MQ digunakan karena sensor jenis ini hemat biaya dan mudah tersedia dipasaran. Sensor MQ-135 membutuhkan waktu pemanasan lebih sedikit, memberikan respon cepat dan sensitivitasnya baik [10].

Prinsip kerja dari sensor ini adalah Ketika timah dioksida (partikel semikonduktor) dipanaskan di udara pada suhu tinggi, oksigen diserap di permukaan. Di udara bersih, elektron donor dalam timah dioksida tertarik ke arah oksigen yang diserap pada permukaan bahan sensor. Ini mencegah aliran arus listrik. Dengan adanya gas pereduksi, kerapatan permukaan oksigen yang teradsorpsi berkurang karena bereaksi dengan gas pereduksi. Elektron kemudian dilepaskan ke dalam timah dioksida, yang memungkinkan arus mengalir bebas melalui sensor. Gambar 2.4 merupakan gambar kurva sensitivitas sensor terhadap konsentasi berbagai jenis gas.



Gambar 2.4 Kurva sensitivitas sensor MQ135

2.4 LoRa RF96

RF96 / 97/98 menggabungkan modem loRa *spread spectrum* yang mana mampu mencapai jangkauan yang jauh secara signifikan dari sistem yang ada berdasarkan modulasi FSK atau OOK. Dengan skema modulasi baru ini sistem memiliki sensitivitas 8 dB lebih baik daripada FSK yang dapat dicapai dengan referensi kristal berbiaya rendah dan toleransi yang rendah. Peningkatan sensitivitas ini memberikan banyak hal seperti, jangkauan yang lebih panjang dan ketahanan terhadap gangguan tanpa perlu penguatan eksternal. LoRa juga memberikan keuntungan yang signifikan pada selektivitas dan kinerja pemblokiran gangguan, lebih lanjut dapat meningkatkan keandalan dalam komunikasi tanpa kabel. Untuk fleksibilitas maksimum, pengguna dapat menentukan penyebaran spectrum pada modulasi *bandwidth* (BW), faktor penyebaran (SF) dan tingkat koreksi kesalahan (CR). Keuntungan lainnya dari *spread modulation* adalah bahwa setiap faktor penyebaran bersifat ortogonal - sehingga beberapa sinyal yang ditransmisikan dapat menempati saluran yang sama tanpa terinterferensi. Hal ini juga memungkinkan koeksistensi sederhana dengan sistem berbasis FSK yang ada. Standar modulasi

Tabel 2.3 Nomor pin, nama pin, tipe pin dan diskripsi pin

Nomor Pin	Nama Pin	Tipe Pin	Diskripsi Pin
1	RFI_LF	I	RF input for lower bands
2	VR_ANA	-	Regulated supply voltage for analogue circuitry
3	VBAT_ANA	-	Supply voltage for analogue circuitry
4	VR_DIG	-	Regulated supply voltage for digital blocks
5	XTA	I/O	XTAL connection or TCXO input
6	XTB	I/O	XTAL connection
7	NRESET	I/O	Reset trigger input
8	DIO0	I/O	Digital I/O, software configured.
9	DIO1/DCLK	I/O	Digital I/O, software configured.
10	DIO2/DATA	I/O	Digital I/O, software configured.
11	DIO3	I/O	Digital I/O, software configured.
12	DIO4	I/O	Digital I/O, software configured.
13	DIO5	I/O	Digital I/O, software configured.
14	VBAT_DIG	-	Supply voltage for digital blocks
15	GND	-	Ground
16	SCK	I	SPI Clock input
17	MISO	O	SPI Data output
18	MOSI	I	SPI Data input
19	NSS	I	SPI Chip select input
20	RXTX/RF_MOD	O	Rx/Tx switch control: high in Tx
21	RFI_HF	I	RF input for upper bands
22	RFO_HF	O	RF output for upper bands
23	GND	-	Ground
24	VBAT_RF	-	Supply voltage for RF blocks
25	VR_PA	-	Regulated supply for the PA
26	GND	-	Ground
27	PA_BOOST	O	Optional high-power PA output, lower or upper bands
28	RFO_LF	O	RF output for lower bands

2.5 Microcontroller Arduino Uno

Arduino UNO merupakan sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO sendiri memiliki 14 pin digital yang bias digunakan sebagai input atau output (6 pin di antaranya dapat digunakan sebagai output PWM), selain itu juga memiliki 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP *header*, dan sebuah tombol reset. Arduino Uno

berbeda dari semua board Arduino sebelumnya, Arduino UNO tidak menggunakan chip driver FTDI USB-to-serial.

Arduino UNO dapat disuplai melalui koneksi USB atau dengan sebuah power suplai eksternal. Suplai eksternal (non-USB) untuk Arduino dapat diperoleh dari sebuah adaptor AC ke DC atau dari sebuah battery. Pemasangan adaptor dapat dilakukan dengan mencolokkan sebuah center-positive plug yang panjangnya 2,1 mm ke power jack dari board. Sedangkan untuk pemasangan baterai, kabel negatif dari sebuah baterai dapat dimasukkan ke dalam header / kepala pin Ground (Gnd) dan kabel positif dari baterai dapat dimasukkan ke dalam header / kepala pin Vin pada Arduino.

Board Arduino UNO beroperasi pada sebuah suplai tegangan eksternal 6 sampai 20 Volt. Jika disuplai dengan tegangan yang lebih kecil dari 7 V, contohnya ketika diberi tegangan 5 Volt mungkin akan mensuplai lebih kecil dari 5 Volt dan board Arduino UNO bisa jadi tidak stabil. Jika suplai yang digunakan lebih besar dari 12 Volt, voltage regulator yang ada di dalam Arduino bisa kelebihan panas dan dapat membahayakan board Arduino UNO. Range tegangan suplai eksternal yang direkomendasikan adalah 7 sampai 12 Volt.

Berikut adalah *power pin* yang dimiliki Arduino Uno:

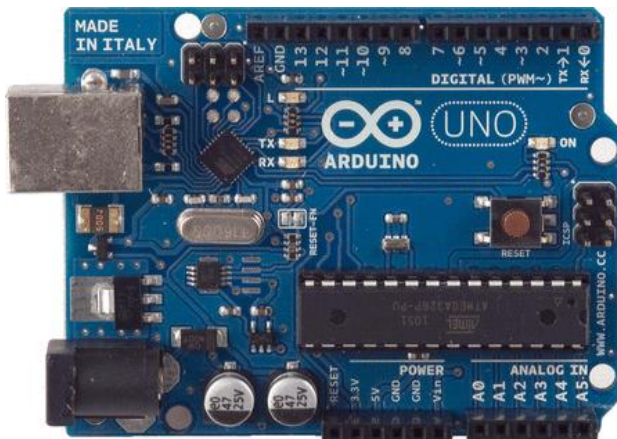
1. **VIN** : Pin ini dapat digunakan sebagai sumber *external (unregulated)* untuk daya pada Arduino yang dapat diberi *input* sebesar 7-12V.
2. **5V** : Pin ini merupakan pin 5V hasil dari tegangan yang telah melewati *regulator*.
3. **3V3** : Pin ini merupakan pin 3,3V hasil dari tegangan yang telah melewati *regulator* dengan arus maksimal 50mA.
4. **GND** : *Ground pin*.

Arduino Uno memiliki 6 pin analog *input* atau biasa disebut ADC (*Analog to Digital Converter*) dengan resolusi 8 - 10 bit, dengan label A0 sampai A5. Selain itu Arduino unojuga memiliki 14 pin digital yang 6 diantaranya dapat digunakan sebagai PWM, dengan label angka 0 sampai 13. pada Arduino Mega juga terdapat juga pin yang memiliki beberapa fungsi khusus yaitu *serial ports*, *external interrupts*, *PWM (Pulse Width Modulation)*, *SPI (Serial Peripheral Interface)*, *LED pin 13*, *TWI (Two Wire Interface)*, *AREF (Analog Reference)*, dan *Reset*. Berikut ini untuk lebih jelasnya:

1. **Serial: 0 (RX) dan 1 (TX)**. Pin ini digunakan untuk berkomunikasi dengan protokol *serial* yang menggunakan dua pin TX (*transmit*) dan RX (*receive*). Setiap TX dan RX berpasangan sebagai berikut; Serial : 0 (RX) and 1 (TX).

2. **External Interrupts:** Pin ini digunakan untuk memicu masuknya program ke program *interrupt*. Pin ini terdiri dari: 2 (interrupt 0), 3 (interrupt 1).
3. **PWM:** pin ini Memberikan 8-bit PWM output dengan fungsi *analogWrite()*. Pin ini terdiri dari pin 3 - 11
4. **SPI:** Pin ini dapat mensupport komunikasi SPI menggunakan *SPI library*. Pin SPI terdiri dari pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK).
5. **LED:** Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai HIGH LED menyala, ketika pin bernilai LOW LED mati.
6. **TWI:** Pin ini dapat mensupport komunikasi TWI dengan menggunakan *Wirelibrary*. Pin itu adalah pin A4 atau SDA dan pin A5 atau SCL.
7. **AREF:** Pin ini digunakan oleh Arduino sebagai tegangan referensi *analog input*.
8. **Reset.** Membawa saluran ini LOW untuk mereset mikrokontroler. Secara khusus, digunakan untuk menambahkan sebuah tombol reset untuk melindungi yang memblock sesuatu pada board.

Memori yang ada pada Arduino Uno berbasis pada ATmega328 yang mempunyai memori sebesar 32 KB (dengan 0,5 KB digunakan untuk bootloader). ATmega 328 juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis (RW/read and written) dengan EEPROM library). Untuk melihat bentuk fisik dari Arduino uno dapat dilihat pada gambar 2.7.

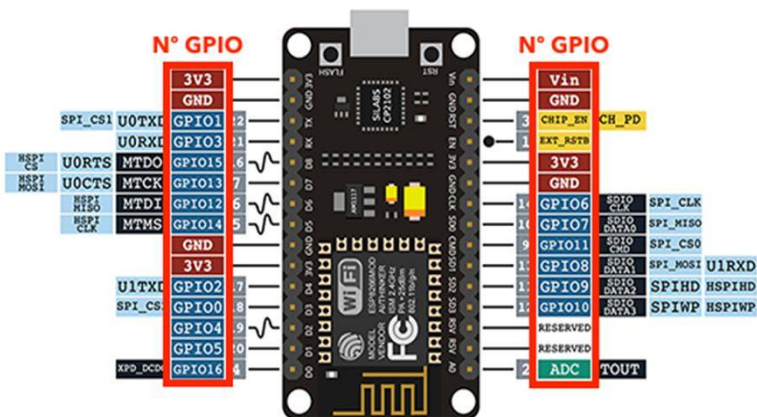


Gambar 2.7 Board Arduino Uno

2.6 NodeMCU ESP6288

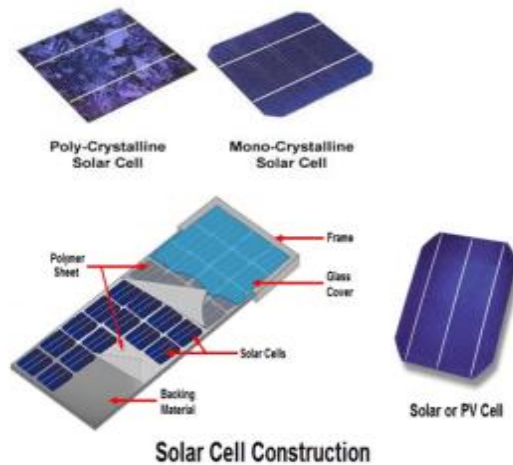
NodeMCU merupakan sebuah Firmware Open-Source dan *Development Kit* yang dapat membantu membangun produk berbasis IoT. NodeMCU dikembangkan untuk mempermudah Penggunaan API (Application Programming Interface) yang canggih untuk perangkat keras IO. API dapat mengurangi pekerjaan berlebih untuk mengkonfigurasi dan memanipulasi perangkat keras. NodeMCU dirancang memiliki *Input* dan *Output* seperti perangkat keras Arduino (IO). NodeMCU menggunakan MCU Wi-Fi dengan biaya terendah yaitu ESP 8266. ESP8266 Merupakan Chip Wi-Fi paling terintegrasi. Ukuran Chip adalah 5mm x 5mm. ESP8266EX minimal membutuhkan rangkaian eksternal dan pengintegrasian 32-bit Tensilica MCU, standar antarmuka digital perifer, sakelar antena, balun R, *Power Amplifier*, penguat penerima dengan *noise* rendah, filter dan modul manajemen daya. Semua diintegrasikan dalam satu paket kecil.

ESP8266EX mengintegrasikan Tensilica L106 *Micro* 32-bit *Controller* (MCU) dengan fitur ekstra konsumsi daya yang rendah dan 16-bit RSIC, dapat mencapai kecepatan clock maksimum 160 MHz. Dengan Real Time Operation System (RTOS) yang diaktifkan dan WiFi *stack* yang berfungsi, sekitar 80% pemrosesan daya masih tersedia untuk pemrograman dan pengembangan aplikasi pengguna. NodeMCU memiliki Pin tegangan *input* dan *output* sebesar 3.3 V pada setiap GPIO. Menyediakan pin tegangan 3.3 Volt sebanyak 3 buah. Mempunyai 16 buah GPIO [11]. Bentuk fisik dari NodeMCU dan *pinout* dapat dilihat pada gambar 2.8.



2.7 Sel Surya

Sel surya atau yang biasa disebut fotovoltaik merupakan perangkat yang mampu mengkonversi langsung sumber cahaya matahari menjadi sumber listrik. Rintangan utama untuk memasarkan sel surya atau fotovoltaik (PV) ke pasar listrik secara luas adalah tingginya biaya bahan yang dibutuhkan dalam pembuatan sel surya, yaitu semikonduktor yang berasal dari kristal yang sangat murni. Sel surya dituntut memiliki efisiensi yang tinggi, memiliki ketahanan tinggi terhadap kondisi lingkungan, dan memiliki bobot yang ringan. Efisiensi pada sel surya dapat ditingkatkan dengan cara mengubah struktur perangkat yang berbeda (setruktur dari kristalnya) [12]–[14]. Gambar 2.9 menunjukkan konstruksi dari sel surya.



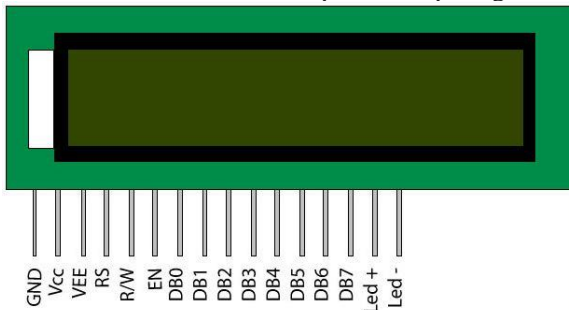
Gambar 2.9 Konstruksi sel surya

2.8 LCD I2C 16X2

Layar LCD (Liquid Crystal Display) adalah modul penampil elektronik dan dapat digunakan dalam berbagai aplikasi. Layar LCD 16x2 adalah modul yang sangat dasar dan sangat umum digunakan di berbagai perangkat dan sirkuit. Alasannya adalah LCD memiliki harga yang murah, mudah diprogram, tidak memiliki batasan untuk menampilkan karakter & bahkan untuk karakter khusus (tidak seperti pada *Seven-Segment*), animasi, dan sebagainya. LCD 16x2 berarti dapat menampilkan 16 karakter per baris dan ada 2 baris pada layar. Dalam LCD ini setiap karakter ditampilkan dalam matriks 5x7 piksel. LCD ini memiliki dua register, yaitu *Command* dan *Data*. Register *Command* menyimpan

instruksi perintah yang diberikan kepada LCD. Perintah adalah instruksi yang diberikan kepada LCD untuk melakukan tugas yang telah ditentukan seperti menginisialisasi, membersihkan layarnya, mengatur posisi kursor, mengontrol tampilan dll. Register *Data* menyimpan data yang akan ditampilkan pada LCD. Data merupakan kode ASCII dari karakter yang akan ditampilkan pada LCD. Gambar pin yan ada pada modul LCD 16X2 dapat dilihat pada gambar 2.10.

Modul LCD I2C merupakan modul untuk antarmuka LCD dengan mikrokontroler. Biasanya, untuk mengkoneksikan layar LCD ke mikokontroler akan menghabiskan pin yang ada pada mikrokontroler dengan mudah, terutama ketika mengkoneksikannya dengan Arduino Uno. Selain itu akan sangat rumit untuk membuat jalur. Setelah LCD dikoneksikan modul LCD I2C 16x2 dan selanjutnya dikoneksikan dengan Arduino, Itu berarti hanya akan membutuhkan 4 pin untuk layar LCD yaitu VCC, GND, SDA, SCL. Ini akan menghemat setidaknya 4 pin digital / analog pada Arduino. Untuk menghindari pertentangan alamat I2C dengan perangkat I2C lainnya, seperti sensor ultrasonik, IMU, akselerometer dan giroskop, alamat I2C modul dapat dikonfigurasi dari 0x20-0x27. Dan kontrasnya dapat disesuaikan secara manual. Gambar pin yan ada pada modul LCD I2C 16X2 dapat dilihat pada gambar 2.11.



Gambar 2.10 Pin diagram LCD 16X2



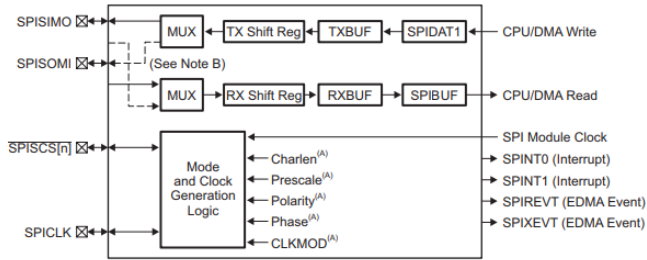
Gambar 2.11 Modul I2C untuk LCD

2.9 Komunikasi SPI

SPI merupakan port input / output serial sinkron berkecepatan tinggi yang memungkinkan aliran bit serial dari bit panjang terprogram (2 hingga 16 bit) untuk digeser masuk dan keluar dari perangkat pada laju transfer bit terprogram. SPI biasanya digunakan untuk komunikasi antar perangkat dan periferal eksternal. Aplikasi yang umum termasuk antarmuka ke eksternal I / O atau ekspansi periferal melalui perangkat seperti register geser, driver layar, SPI EPROMS, dan konverter analog-ke-digital. Komunikasi SPI memiliki beberapa fitur seperti:

1. 16-bit shift register
2. 16-bit Receive buffer register (SPIBUF) and 16-bit Receive buffer emulation alias register (SPIEMU)
3. 16-bit Transmit data register (SPIDAT0) and 16-bit Transmit data and format selection register (SPIDAT1)
4. 8-bit baud clock generator
5. Serial clock (SPICLK) I/O pin
6. Slave in, master out (SPISIMO) I/O pin
7. Slave out, master in (SPISOMI) I/O pin
8. Multiple slave chip select (SPISCS[n]) I/O pins (4 pin mode only)
9. Programmable SPI clock frequency range
10. Programmable character length (2 to 16 bits)
11. Programmable clock phase (delay or no delay)
12. Programmable clock polarity (high or low)
13. Interrupt capability
14. DMA support (read/write synchronization events)
15. Up to 66 MHz operation

Komunikasi SPI memungkinkan perangkat lunak untuk memprogram dengan beberapa pilihan seperti, Frekuensi SPICLK (modul SPI Clock / 2 hingga modul SPI Clock / 256). Sedangkan untuk penggunaan pin dapat digunakan 3-pin (MISO, MOSI, dan SCK) atau 4-pin (MISO, MOSI, SCK, dan SS atau Chip Select). Panjang karakter yang dikirim mulai dari 2 bit hingga 16 bit dan arah gesernya MSB/LSB terlebih dahulu. Fase clock bisa diberi Penundaan atau tanpa penundaan dan polaritas dapat belogika tinggi atau rendah. Dalam mode master memungkinkan keterlambatan antar transmisi. Dalam mode master terdapat pengaturan *chip select*. Fungsi *Chip Select* akan ditahan dalam mode master. Gambar 2.12 merupakan gambar yang menunjukkan blok diagram dari komunikasi SPI.



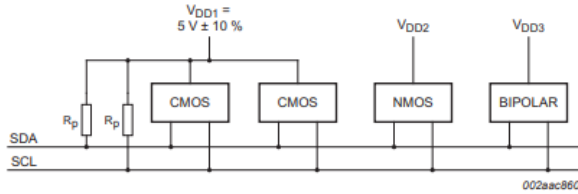
(A) Indicates the log controlled by SPI register bits.
 (B) Solid line represents data flow for SPI master mode.

Gambar 2.12 Blok diagram komunikasi SPI

2.10 Komunikasi I2C

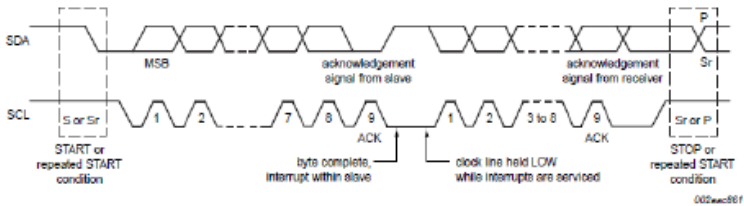
Komunikasi I2C merupakan komunikasi dengan tipe serial *synchronous* yaitu komunikasi dengan menyamakan ketukan transfer data, dengan model *two wire interface* (dua kabel antarmuka) yang dapat mengirimkan data digital melalui jalur bus. Komunikasi I2C pertama kali diperkenalkan oleh Philips Semiconductor. Alamat perangkat yang menggunakan komunikasi I2C terdiri dari 7-bit data alamat dan 1-bit konfigurasi R/W (*read* atau *write*). I2C memiliki protokol komunikasi yang terdiri dari *start condition*, MSB data, LSB data, ACK, dan *stop condition*. I2C memiliki 2 jalur data, yaitu SDA dan SCL. Jalur data ini membawa informasi antara perangkat yang terhubung ke bus. Setiap perangkat dikenali oleh alamat unik (perangkat tersebut seperti mikrokontroler, *driver LCD*, memori atau antarmuka *keyboard*) dan setiap perangkat dapat beroperasi sebagai pemancar atau penerima, tergantung pada fungsi perangkat. Driver LCD mungkin hanya berfungsi sebagai penerima data, sedangkan memori dapat menerima dan mengirim data. Baik SDA dan SCL merupakan saluran dua arah, kedua jalur tersebut terhubung ke tegangan suplai positif melalui sumber arus atau resistor *pull-up* seperti pada gambar. Ketika bus tidak tersambung apapun, kedua jalur itu belogika tinggi. Pada bagian output perangkat yang terhubung ke bus harus memiliki jalur *open-drain* atau *open-collector* untuk mendukung kerja fungsi *wired-AND*. Pada kedua jalur tersebut perlu diberikan resistor *pull-up* karena bersifat *open drain*. Pemilihan resistor *pull-up* dapat menggunakan persamaan (2.1) dan gambar 2.13 menunjukkan bahwa komunikasi I2C dapat digunakan untuk lebih dari satu komunikasi, sedangkan protokol komunikasi I2C dapat dilihat pada gambar 2.14 [15].

$$R_{min} = \frac{V_{cc}-V_{OL}}{I_{OL}} \text{ dan } R_{max} = \frac{T_{SMBus}}{C_{Bus}} \quad (2.1)$$



V_{DD2} , V_{DD3} are device-dependent (for example, 12 V).

Gambar 2.13 Perangkat dengan berbagai tegangan suplai dan berbagi bus yang sama



Gambar 2.14 Transfer data pada bus I2C

2.11 Komunikasi Serial

Universal Asynchronous Receiver / Transmitter (UART) merupakan protokol komunikasi serial standar untuk bertukar data antara dua perangkat. Dalam protokol komunikasi ini, data ditransfer secara berurutan, satu bit dalam satu waktu. Implementasi ini menggunakan bingkai, dalam satu bingkai terdiri dari 8 bit data, satu bit mulai, satu opsional bit paritas, dan satu atau beberapa bit *stop*. Bit paritas merupakan 1 bit dengan nilai 0 atau 1, yang ditambahkan ke blok data untuk tujuan pendeteksian kesalahan. Bit ini bersifat opsional, itu mungkin atau tidak mungkin ditambahkan ke muatan data. Juga dapat diatur menjadi ganjil atau genap. Bit-bit ini sering digunakan dalam transmisi data untuk memastikan data tersebut tidak rusak selama proses transfer. Jika protokol transmisi data diatur ke paritas ganjil, masing-masing paket data harus memiliki paritas ganjil. Jika disetel genap, setiap paket harus memiliki paritas genap. Jika paket diterima dengan paritas yang salah, akan muncul kesalahan dan data harus dikirim ulang. Bit paritas untuk setiap paket data dihitung sebelum data dikirimkan. Karena protokol UART bersifat tidak sinkron, protokol ini tidak memerlukan sinyal *clock*. Dalam komunikasi UART, kecepatan ditentukan oleh *baud rate*. *Baud rate* didefinisikan

sama dengan jumlah bit yang ditransmisikan per detik termasuk bit mulai dan berhenti. Untuk pengiriman dan penerimaan data yang andal dan tanpa adanya bit yang hilang, baik pemancar dan penerima harus memiliki *baud rate* yang sama. Ketidakcocokan *baud rate* antara pemancar dan penerima biasanya menghasilkan kesalahan *framing*. *Baud rate* yang umum digunakan adalah 4800, 9600, 19200, 38400, 57600, dan 115200. Tetapi *rate* lainnya mungkin juga bisa digunakan.

Modul Pengguna UART mengimplementasikan pemancar dan penerima serial. Peta UART menjadi dua PSoC Blok Komunikasi Digital yang didesain menjadi TX dan RX. TX Blok PSoC memberikan fungsionalitas pemancar dan RX blok PSoC memberikan fungsionalitas penerima. RX dan TX beroperasi secara independen. Masing-masing memiliki daftar Kontrol dan Status mereka sendiri, dapat diberikan program interupsi, *Input / Output*, register *Buffer*, dan register *Shift*. Baik TX dan RX berbagi format *enable*, *clock*, dan *data* yang sama. Mengatur bit *enable* pada register Kontrol RX dan Kontrol TX memungkinkan UART untuk beroperasi. Mengaktifkan dan menonaktifkan dilakukan menggunakan fungsi yang telah disediakan API (*Application Programming Interface*). Modul *clock* Pengguna UART dibagikan oleh komponen RX dan TX. Frekuensi *clock* yang dipilih harus delapan kali frekuensi dari laju bit data yang diperlukan. Setiap bit data yang diterima atau dikirim membutuhkan delapan siklus *clock input*. *Clock* ini dikonfigurasi menggunakan Editor Perangkat PSoC Designer. Data yang diterima dan dikirim adalah bit *stream* yang terdiri dari bit awal, delapan bit data, bit *opsional parity*, dan bit *stop*. Paritas dapat diatur menjadi tidak ada, genap, atau ganjil, dan diatur menggunakan PSoC Designer Editor atau menggunakan API UART. Baik RX dan TX diatur ke konfigurasi paritas yang sama. Blok diagram dari komunikasi serial dapat dilihat pada gambar 2.15.

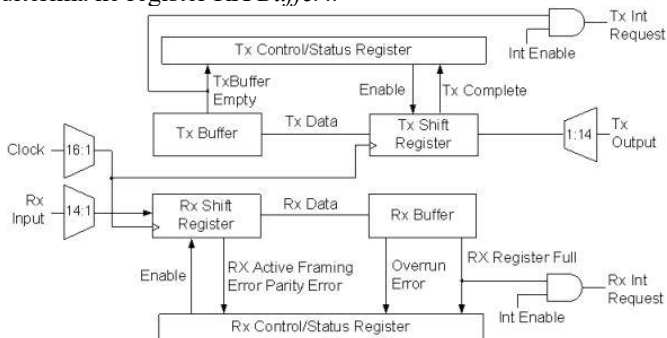
2.11.1 TX - UART Transmitter

Transmitter menggunakan register TX *Buffer*, TX *Shift*, dan TX *Control* dari tipe *Digital Communications Blok PSoC*. Register TX *control* diinisialisasi dan dikonfigurasi menggunakan UART *User Module firmware API routines*. Ketika bit *enable* di register Kontrol TX diatur, bit *clock* internal dibagi delapan akan dihasilkan. Sebuah *byte* data untuk pengiriman ditulis oleh API *routine* ke dalam register *Buffer* TX, pembersihan TX *Buffer Empty* bit status di register *control* TX. Bit status ini dapat digunakan untuk mendeteksi dan mencegah *overrun* pengiriman *error*. Kenaikan dari bit *clock* berikutnya akan mentransfer data ke register *Shift* dan mengatur bit TX *Buffer Empty* di register *control* TX. Jika *interrupt*

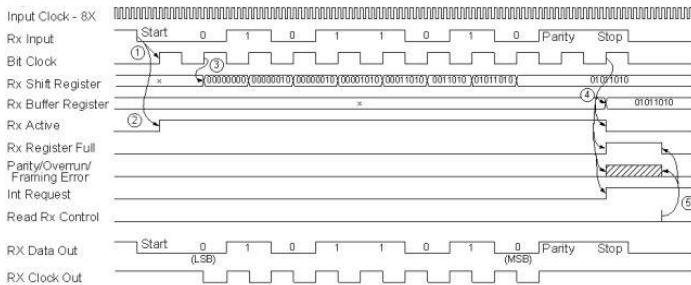
enable mask diaktifkan, perintah interupsi akan dipicu. Interupsi ini memungkinkan antrian *byte* berikutnya untuk dikirim. Jadi ketika *byte* data saat ini sepenuhnya ditransmisikan, *byte* baru ditransmisikan pada transmisi *clock* berikutnya yang tersedia. Bit mulai ditransmisikan pada saat yang sama dengan *byte* data ditransfer dari register *TX Buffer* ke register *TX Shift*. Bit *Clock* berturut-turut menggeser bit *Stream* serial ke *output*. Bit *stream* tersusun dari masing-masing bit *byte* data, bit *LSB (Least Significant Bit)*, bit *paritas opsional*, dan bit *stop*. Ketika bit berhenti sepenuhnya ditransmisikan, bit *TX Complete Status* pada register *TX Control* diatur. Bit ini tetap ada dan valid hingga dibaca. Jika *byte* data baru telah ditulis ke register *TX Buffer*, *byte* data ditransfer ke register *TX Shift* dan transmisi data dimulai pada kenaikan berikutnya dari bit *Clock*.

2.11.2 RX – UART Receiver

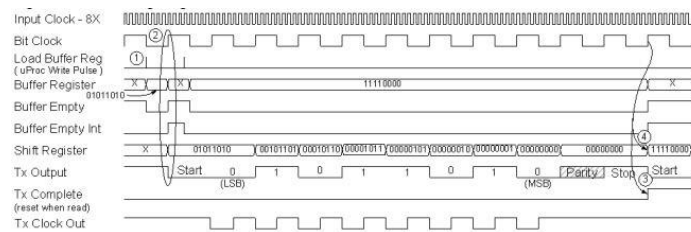
Receiver menggunakan register *RX Buffer*, *RX Shift*, dan *RX Control* dari tipe *Digital Communications Blok PSoc*. Register Kontrol RX diinisialisasi dan dikonfigurasi menggunakan *UART User Module firmware API routines*. Inisialisasi RX terdiri dari pengaturan paritas UART, secara opsional memungkinkan terjadinya interupsi pada register *RX Full Condition*, kemudian aktifkan UART. Ketika bit awal terdeteksi pada *input* RX, bit *clock* yang terbagi delapan dimulai dan disinkronkan ke sampel data di tengah bit yang diterima. Pada kenaikan bit *clock* yang terbagi delapan berikutnya, *input* data diambil sampelnya dan dialihkan ke register *RX Shift*. Jika paritas diaktifkan, bit *clock* berikutnya akan mengambil sampel bit paritas. Pengambilan sampel dari bit *stop*, pada bit *clock* berikutnya, menghasilkan transfer *byte* data yang diterima ke register *RX Buffer*..



Gambar 2.15 Blok diagram UART



Gambar 2.16 Diagram RX *timing*



Gambar 2.17 Diagram TX *timing*

2.12 Aplikasi Blynk

Blynk merupakan platform sebuah layanan *server* yang digunakan untuk mendukung project *Internet of Things* dengan aplikasi berbasis iOS dan Android yang dapat dikontrol dengan Arduino atau Raspberry Pi dengan menggunakan akses Internet. Blynk merupakan dasbor digital di mana aplikasi ini dapat membangun antarmuka grafis bagi pengguna dengan tombol hanya drag dan drop [16]. Blynk diciptakan dengan tujuan untuk *control* dan *monitoring hardware* secara jarak jauh menggunakan komunikasi data internet ataupun intranet (jaringan LAN). Kemampuan menyimpan data dan menampilkan data secara visual baik menggunakan angka, warna ataupun grafis semakin memudahkan dalam pembuatan *project* di bidang *Internet of Things*. Gamba 2.18 merupakan ilustrasi hubungan antara Blynk dengan perangkat IoT. Terdapat 3 komponen utama Blynk, yaitu:

2.12.1 Blynk Apps

Blynk Apps memungkinkan untuk membuat proyek antarmuka dengan berbagai macam komponen input dan output. Blynk mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data tersebut sesuai dengan komponen yang

dipilih. Representasi data dapat berbentuk visual angka maupun grafik. Terdapat 4 kategori komponen pada aplikasi Blynk, yaitu:

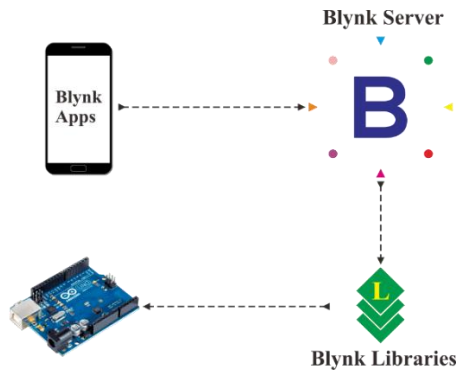
1. Controller: berfungsi untuk mengirim data atau perintah ke hardware.
2. Display: berfungsi untuk menampilkan data yang berasal dari hardware ke smartphone.
3. Notification: berfungsi untuk mengirim pesan notifikasi.
4. Interface: berfungsi untuk mengatur tampilan pada aplikasi Blynk, dapat berupa menu ataupun tab.
5. Others: komponen tambahan yang tidak termasuk pada 3 kategori sebelumnya seperti Bridge, RTC, dan Bluetooth.

2.12.2 Blynk Server

Blynk *server* merupakan fasilitas *back-end service* berbasis *cloud* yang bertanggung jawab untuk mengatur komunikasi antara aplikasi *smartphone* dengan lingkungan hardware. Kemampun untuk menangani hardware yang banyak pada saat yang bersamaan semakin memudahkan untuk membuat proyek dengan sistem IoT. Blynk *server* juga tersedia dalam bentuk server lokal apabila digunakan pada lingkungan tanpa internet. Blynk server lokal bersifat terbuka dan dapat diimplementasikan pada sederhana sekalipun, contohnya *Raspberry Pi* atau Arduino.

2.12.3 Blynk Library

Blynk *Library* berfungsi untuk membantu pengembangan code. Blynk library tersedia pada banyak platform perangkat keras sehingga semakin mempermudah dalam pembuatan proyek berbasis IoT dengan fleksibilitas hardware yang didukung oleh lingkungan Blynk.



Gambar 2.18 Ilustrasi hubungan antara BLYNK dan perangkat IoT

.....*Halaman ini sengaja dikosongkan*.....

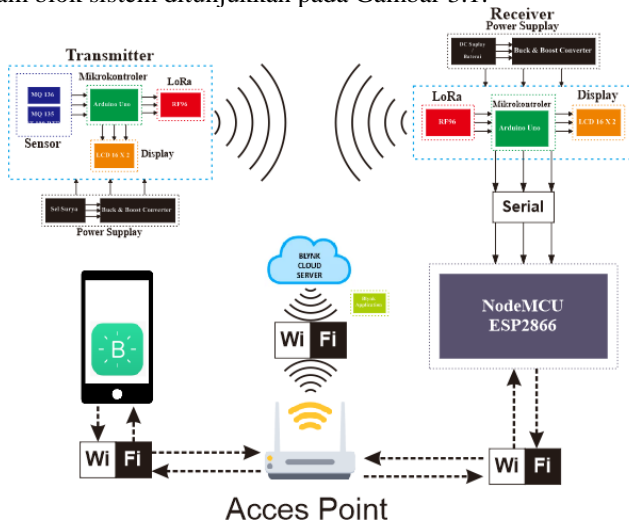
BAB III PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Pada bab ini dijelaskan perancangan sistem secara keseluruhan. Alat yang dirancang bertujuan untuk memonitoring aktifitas gunung berapi terutama konsentrasi gas karbon dioksida (CO_2), gas hidrogen sulfida (H_2S), dan suhu pada suatu wilayah gunung berapi. Perancangan Alat berupa sebuah *board* yang berisi mikrokontroler Arduino Uno dilengkapi dengan dua sensor gas, satu sensor suhu, LoRa (RF96), LCD *display*, dan terminal catu daya. Sensor gas dan sensor suhu digunakan untuk mengetahui konsentrasi gas dan kenaikan suhu pada wilayah tersebut. LCD digunakan untuk menampilkan data sensor yang didapat. Mikrokontroler akan membaca data dari sensor-sensor yang terhubung dengannya kemudian data tersebut akan dikirim dengan modul LoRa. Alat dilengkapi dengan dengan sel surya sebagai sumber dayapada bagian kotak *transmitter*. Data yang dikirim akan diterima oleh mikrokontroler pada kotak *receiver* melalui modul LoRa. Selanjutnya data diolah dan diteruskan ke mikrokontroler dan lcd *display*. Setelah data diterima oleh NodeMCU selanjutnya akan dikirim ke aplikasi *smartphone* berupa Blynk.

3.2 Diagram Blok Sistem

Diagram blok sistem ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem

Pada tugas akhir ini, mikrokontroler yang digunakan sebagai unit pengolah data yaitu Arduino Uno.

Sensor yang digunakan adalah sensor gas MQ136 dan sensor gas MQ135. Sensor MQ136 dan MQ135 diletakkan pada kotak *transmitter* bersama dengan LCD dan LoRa serta terminal catu daya. Baik sensor MQ136, MQ135, dan LCD diletakkan terpisah dengan board dan dihubungkan dengan kabel agar mudah dipasang atau dilepas. Sensor MQ136 digunakan untuk mengetahui konsentrasi gas H_2S . Sensor MQ135 digunakan untuk mengetahui konsentrasi gas CO_2 . Sedangkan LoRa digunakan untuk mengirim data sensor yang telah didapat. Sumber tegangan dari mikrokontroler dan sensor didapat dari sel surya yang telah diturunkan tegangannya terlebih dahulu menggunakan *buck converter*.

Sensor gas MQ136, sensor gas MQ135, dan sensor suhu PT100 dibaca oleh mikrokontroler Arduino Uno setiap 5 detik sekali. Hasil pembacaan dari sensor ini akan dikirimkan dengan menggunakan Modul LoRa. Selain itu akan ditampilkan pada LCD *display*.

Data yang dikirim akan diterima dengan LoRa pada kotak *Receiver*. Pada kotak receiver juga terdapat mikrokontroler Arduino Uno untuk mengolah data sensor yang diterima dari LoRa. Kemudian juga ada Mikrokontroler NodeMCU untuk mengirimkan data yang diterima ke sebuah aplikasi di *smartphone* yang bernama Blynk.

3.3 Perancangan Perangkat Keras

Perangkat keras pada tugas akhir yaitu desain elektronik. Desain elektronik meliputi pembuatan *board* microcontroller *Transmitter* terpadu yang berisi Arduino Uno, LoRa Shield + RF96, Sensor, LCD dan terminal catu daya. Selain itu juga pembuatan *board microcontroller Receiver* terpadu yang berisi Arduino Uno, LoRa Shield + RF96, LCD dan terminal catu daya.

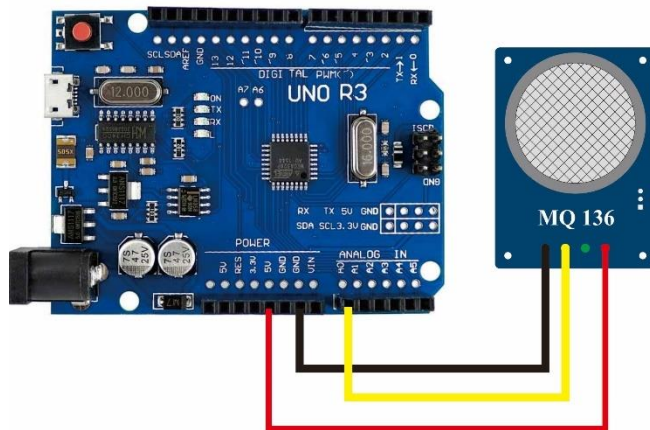
3.3.1 Sensor Gas H_2S

Pada perancangan sensor gas H_2S dilakukan dengan menggunakan sensor berjenis MQ136. Sensor MQ136 ini dihubungkan dengan Arduino Uno. Ada 4 pin yang ada pada sensor MQ136, yaitu pin VCC, Ground, Digital Output, dan Analog Output. Pin yang digunakan hanya tiga, yaitu pin suplai (VCC dan Ground) dan pin untuk membaca nilai tegangan sensor yang bisa dipilih antara pin analog output atau digital output. Pada perancangan ini digunakan pin analog output. Tegangan yang keluar dari analog output akan dibaca dengan pin analog input Arduino Uno, sedangkan untuk suplai sensor bisa didapat dari tegangan

suplai 5 Volt. Untuk hubungan antara sensor dan Arduino dapat dilihat pada tabel 3.1. Untuk ilustrasi gambar hubungan antara sensor MQ136 dengan Arduino Uno dapat dilihat pada gambar 3.2.

Tabel 3.1 Wiring pin antara pin Arduino dan sensor MQ136

Arduino Uno	Sensor MQ136
Pin 5Volt	Pin VCC
Pin Ground	Pin Ground
Pin Analog A0	Pin Analog Output (AOUT)



Gambar 3.2 Wiring diagram sensor gas H_2S MQ136 dengan Arduino

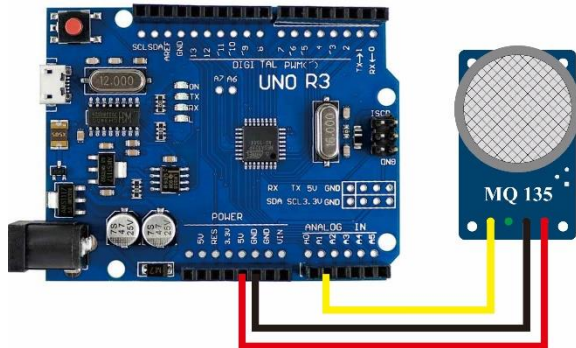
3.3.2 Sensor Gas CO_2

Pada perancangan sensor gas CO_2 dilakukan dengan menggunakan sensor berjenis MQ135. Sensor MQ135 ini dihubungkan dengan Arduino Uno. Ada 4 pin yang ada pada sensor MQ135, yaitu pin VCC, Ground, Digital Output, dan Analog Output. Pin yang digunakan hanya tiga, yaitu pin suplai (VCC dan Ground) dan pin untuk membaca nilai tegangan sensor yang bisa dipilih antara pin analog output atau digital output. Pada perancangan ini digunakan pin analog output. Tegangan yang keluar dari analog output akan dibaca dengan pin analog input Arduino Uno, sedangkan untuk suplai sensor bisa didapat dari tegangan suplai 5 Volt. Untuk hubungan antara sensor dan Arduino dapat

dilihat pada tabel 3.2. Untuk ilustrasi gambar hubungan antara sensor MQ136 dengan Arduino Uno dapat dilihat pada gambar 3.3.

Tabel 3.2 *Wiring* pin antara pin Arduino dan sensor MQ135

Arduino Uno	Sensor MQ135
Pin 5Volt	Pin VCC
Pin Ground	Pin Ground
Pin Analog A1	Pin Analog Output (AOUT)



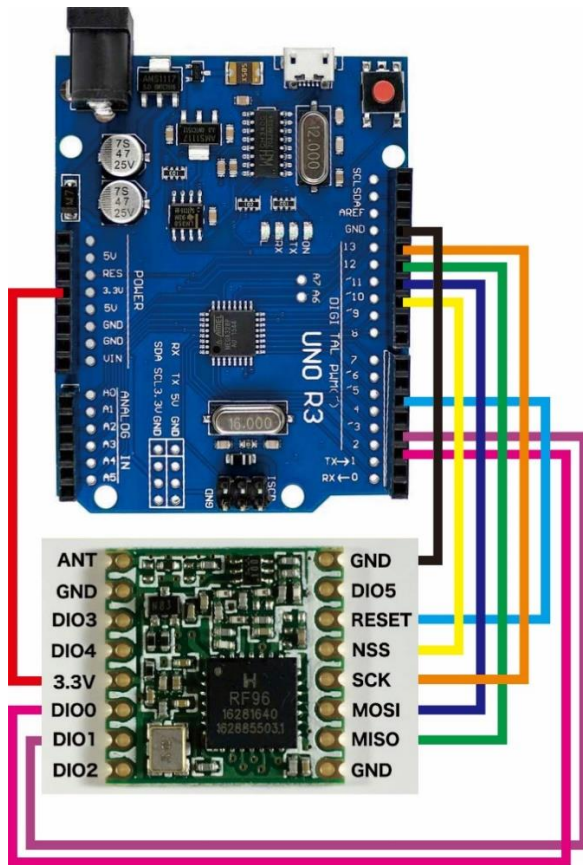
Gambar 3.3 *Wiring diagram* sensor gas CO₂ MQ135 dengan Arduino

3.3.3 LoRa RF96

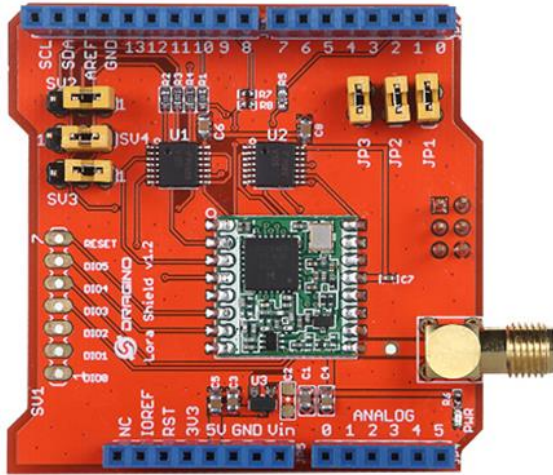
Pada perancangan perangkat keras RF96 dilakukan dengan menghubungkan beberapa pin yang ada pada RF96. Pinyang dipakai adalah pin 3.3 Volt, pin Ground, pin MISO, pin MOSI, Pin SCK, Pin SS, pin RESET, pin DIO0, dan pin DIO1. Pin 3.3 Volt di hubungkan dengan pin 3.3 Volt yang ada pada Arduino, pin Ground pada RF96 dihubungkan dengan pin Ground Arduino. Pin MISO, MOSI, SCK dan SS dihubungkan dengan Pin 11, 12, 13 dan 10 pada Arduino. Pin RESET dihubungkan dengan pin 5, sedangkan pin DIO0 dan DIO1 dihubungkan dengan pin 2 dan pin 3 pada Arduino. Untuk lebih jelasnya bisa dilihat pada gambar 3.4. Tetapi disini digunakan LoRa Shield yang langsung kompatibel dengan Arduino Uno. Untuk hubungan antara LoRa dan Arduino dapat dilihat pada tabel 3.3. Untuk ilustrasi gambar hubungan standar antara RF96 dengan Arduino Uno dapat dilihat pada gambar 3.4. sedangkan gambar 3.5 merupakan gambar *shield* Arduino Uno untuk RF96.

Tabel 3.3 Wiring pin standar antara pin Arduino dan RF96

Arduino Uno	RF96
Pin 3.3 Volt	Pin 3.3 Volt
Pin Ground	Pin Ground
Pin Digital 5 (5)	Pin Reset
Pin NSS (10)	Pin NSS
Pin SCK (13)	Pin SCK
Pin MISO (12)	Pin MISO
Pin MOSI (11)	Pin MOSI
Pin Digital 2 (2)	Pin DIO0
Pin Digital 3 (3)	Pin DIO1



Gambar 3.4 Wiring diagram RF96 ke Arduino Uno



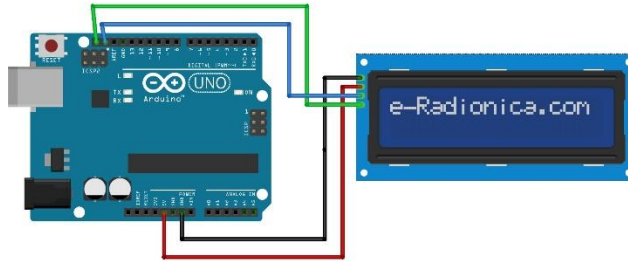
Gambar 3.5 Dragino LoRa Shield

3.3.4 LCD 16X2 I2C

Perancangan LCD pada Arduino dilakukan dengan menggabungkan LCD 16X2 dengan modul LCD I2C. Modul LCD I2C sendiri memiliki 4 pin yaitu VCC, Ground, SDA, dan SCL. Tujuan digunakannya modul LCD I2C adalah agar tidak menghabiskan banyak pin, cukup memakai 4 pin. Jika tidak memakai modul LCD I2C pengkoneksian antara LCD dan Arduino membutuhkan 16 pin dengan konfigurasi jalur yang tidak efisien. 4 pin yang digunakan dihubungkan pada pin Arduino dengan konfigurasi dapat dilihat pada tabel dan gambar 3.3 untuk visualisasi kabel. Untuk hubungan antara LCD I2C dan Arduino dapat dilihat pada tabel 3.4. Untuk ilustrasi hubungan antara LCD I2C dengan Arduino uno dapat dilihat pada gambar 3.6.

Tabel 3.4 Wiring pin antara Arduino Uno dengan LCD I2C

Arduino Uno	LCD I2C
5 V	VCC
Ground	Ground
SDA	SDA
SCL	SCL



fritzing

Gambar 3.6 Wiring diagram LCD 16X2 I2C dengan Arduino Uno

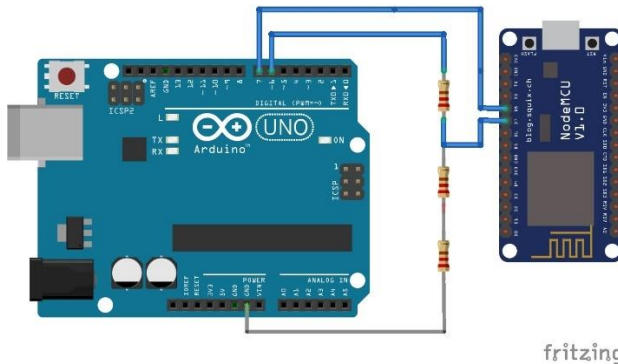
3.3.5 NodeMCU

Perancangan komunikasi antara NodeMCU dan Arduino dilakukan dengan menggunakan model komunikasi Serial. Untuk pin yang digunakan untuk berkomunikasi serial, tidak langsung memakai pin serial yang ada pada Arduino Uno maupun yang ada pada NodeMCU. Tetapi menggunakan pin lain, lebih tepatnya menggunakan pin digital yang ada pada masing-masing *microcontroller*. Karena ada pin (pin digital) yang pada masing-masing *microcontroller* yang bisa difungsikan sebagai pin serial atau lebih tepatnya sebagai pin RX dan TX. Penggunaan pin ini harus diatur menjadi pin RX dan TX pada program Arduino dengan menggunakan *library Software Serial*. Pada perancangan ini digunakan pin Digital 6 sebagai pin TX dan pin Digital 7 sebagai pin RX. Sedangkan pada NodeMCU digunakan pin D7 sebagai pin RX dan pin D8 sebagai pin TX. Hubungan kabel TX dan RX antara Arduino Uno dengan NodeMCU harus disilang, pin TX pada Arduino Uno dihubungkan dengan pin RX pada NodeMCU, begitu juga sebaliknya. Karena *input* dan *output* dari NodeMCU adalah tegangan 3.3 Volt dan *input* dan *output* Arduino Uno adalah 5 Volt, ketika hubungan antara pin TX NodeMCU dengan pin RX Arduino Uno bisa langsung dihubungkan. Sedangkan jika pin TX yang ada pada Arduino Uno dihubungkan dengan pin RX NodeMCU harus menggunakan pembagian rangkaian pembagi tegangan agar yang masuk pada NodeMCU adalah 3.3 Volt, karena keluaran dari pin TX Arduino Uno adalah 5 Volt. Rangkaian pembagi tegangan dapat digunakan rangkaian sederhana yaitu dengan menggunakan 3 buah resistor bernilai 1 KOhm yang dihubungkan seri, untuk rumus dapat dilihat pada rumus 3.1. Untuk visualisasi gambar dapat dilihat pada gambar 3.7. Untuk hubungan antara NodeMCU dan Arduino dapat dilihat pada tabel 3.5.

$$V_{Out} = \frac{2}{3} V_{Tx} \quad (3.1)$$

Tabel 3.5 Wiring pin antara Arduino Uno dengan NodeMCU

Arduino Uno	NodeMCU
Pin 6 (TX)	Pin D7 (RX)
Pin 7 (RX)	Pin D8 (TX)



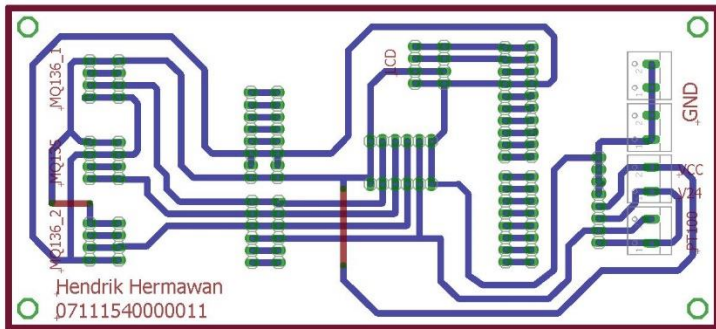
Gambar 3.7 Wiring diagram NodeMCU to Arduino Uno

3.3.6 Board Microcontroller Transmitter

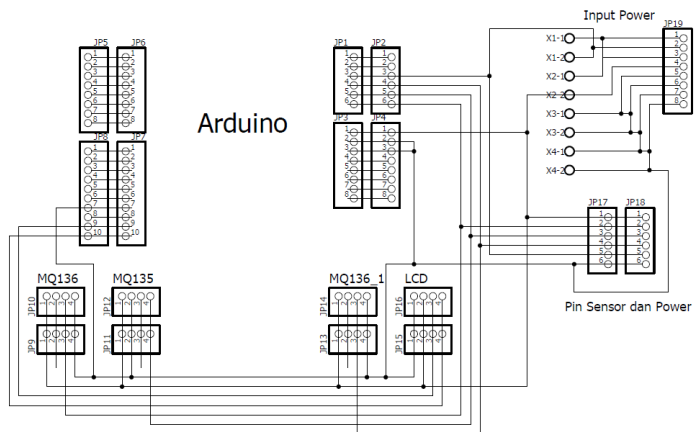
Board microcontroller transmitter terpadu merupakan gabungan dari komponen-komponen yang sudah dijelaskan sebelumnya. Komponen utama dari board ini adalah Arduino Uno dan juga terdapat 5 komponen eksternal yang dibutuhkan pada *Board microcontroller transmitter* terpadu ini. 5 komponen tersebut adalah sensor MQ 136, sensor MQ135, sensor RTD PT100, LCD I2C, dan terminal catu daya. Untuk terminal catu daya diletakkan menjadi satu *board* dengan Arduino Uno. Sedangkan untuk 3 sensor dan LCD *display* terletak terpisah atau dapat dipasang maupun dilepas. *Board* ini dilengkapi dengan terminal catu daya 5 V dan 24 V.

Skematik *board* ini dibuat dengan menggunakan *software* EAGLE. Petama dibuat skematik dari komponen-komponen yang dijelaskan sebelumnya, setelah itu adalah menata letak dari komponen-komponen tersebut agar didapatkan desain yang efisien dengan dimensi yang kecil. Selain itu tata letak yang baik akan

memudahkan dalam membuat jalur. Komponen yang digunakan dalam mendesain *board* ini adalah komponen berjenis DIP (*Dual In-line Package*). Penggunaan komponen DIP dalam mendesain *board* pada alat ini karena *board* yang dibuat tidak terlalu membutuhkan dimensi yang kecil, sehingga tidak haus menggunakan komponen berjenis SMD (*Surface-Mount Device*) cukup dengan komponen berjenis DIP. Hasil desain *board microcontroller transmitter* terpadu untuk alat Sistem pemantauan Gunung Berapi Bebas IoT Menggunakan LoRa ditunjukkan pada Gambar 3.8 dan Gambar 3.9.



Gambar 3.8 Desain *board microcontroller transmitter* pada *EAGLE*

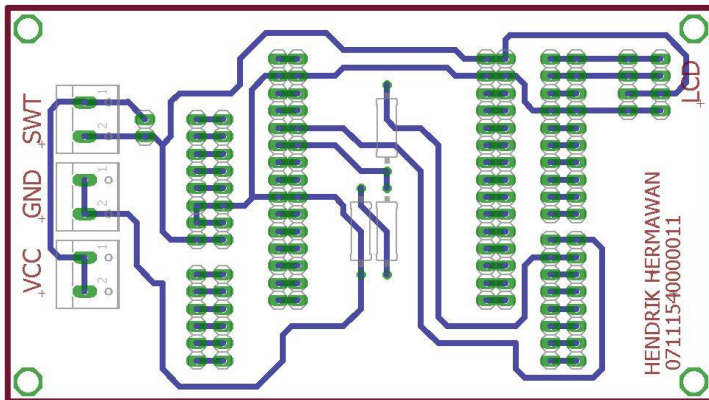


Gambar 3.9 Desain *schematic microcontroller transmitter* pada *EAGLE*

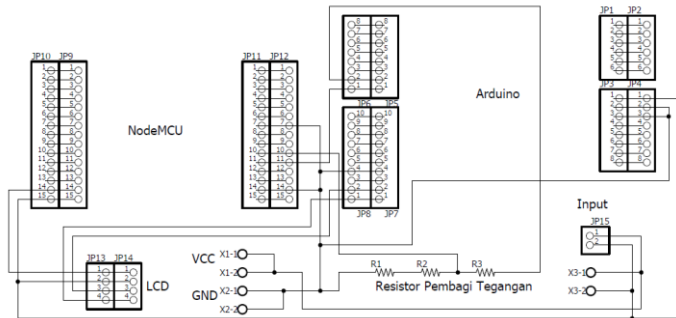
3.3.7 Board Microcontroller Receiver

Board microcontroller receiver terpadu merupakan gabungan dari komponen-komponen yang sudah dijelaskan sebelumnya. Komponen utama dari board ini adalah Arduino Uno dan juga terdapat 3 komponen eksternal yang dibutuhkan pada *Board microcontroller receiver* terpadu ini. 3 komponen tersebut adalah NodeMCU, LCD I2C, dan terminal catu daya. Untuk terminal catu daya diletakkan menjadi satu *board* dengan Aduino Uno. Sedangkan untuk NodeMCU dan LCD *display* terletak terpisah atau dapat dipasang maupun dilepas. *Board* ini dilengkapi dengan terminal catu daya 5 V.

Skematik *board* ini dibuat dengan menggunakan *software* EAGLE. Petama dibuat skematik dari komponen-komponen yang dijelaskan sebelumnya, setelah itu adalah menata letak dari komponen-komponen tersebut agar didapatkan desain yang efisien dengan dimensi yang kecil. Selain itu tata letak yang baik akan memudahkan dalam membuat jalur. Komponen yang digunakan dalam mendesain *board* ini adalah komponen berjenis DIP (*Dual In-line Package*). Penggunaan komponen DIP dalam mendesain *board* pada alat ini karena *board* yang dibuat tidak terlalu membutuhkan dimensi yang kecil, sehingga tidak haus menggunakan komponen berjenis SMD (*Surface-Mount Device*) cukup dengan komponen berjenis DIP. Hasil desain *board microcontroler receiver* terpadu untuk alat Sistem pemantauan Gunung Berapi Bebas IoT Menggunakan LoRa ditunjukkan pada Gambar 3.10 dan Gambar 3.11.



Gambar 3.10 Desain *board microcontroller receiver* pada EAGLE



Gambar 3.11 Desain *schematic microcontroller receiver* pada EAGLE

3.4 Perancangan Perangkat Lunak

3.4.1 Pembacaan Sensor MQ136

Perancangan perangkat lunak pada sensor MQ136 dimulai dengan pembacaan Tegangan ADC yang masuk pada Arduino Uno dengan menggunakan program sebagai berikut:

```
float MQResistanceCalculation(int raw_adc, float r1_value)
{
    return
    (long)((long)(1024* (long)r1_value) / raw_adc -
    (long)r1_value);
}
```

$r1_value$ merupakan nilai $r1$ yang terukur pada sensor, sedangkan raw_adc merupakan tegangan yang keluar dari analog *output* sensor dan dibaca oleh Arduino Uno. Selanjutnya akan dicari nilai r_s dengan menggunakan program seperti berikut:

```
float MQRead(int mq_pin, float r1_value)
{
    int i;
    float rs = 0;
    for (i = 0; i < Read_Sample_Times; i++) {
        rs += MQResistanceCalculation(analogRead(mq_pin),
        r1_value);
        delay(Read_Sample_Interval);
    }
    rs = rs / Read_Sample_Times;
    return rs;}
}
```

Read_Sample_Times merupakan banyaknya sample yang akan diambil. Fungsi analogRead() berfungsi mengambil data analog yang terbaca. Sedangkan Read_Sample_Interval merupakan delay waktu pengambilan satu sample data. Setelah didapat rs, selanjutnya mencari ro, dalam mencari ro dapat digunakan program sebagai berikut:

```
float MQCalibration(int mq_pin, double ppm, double rl_value,
float *pcurve )
{
    int i;
    float val = 0;
    for ( i = 0; i < Calibration_Sample_Times; i++) { //take
multiple samples
        val += MQResistanceCalculation(analogRead(mq_pin),
rl_value);
        delay(Calibration_Sample_Interval);
    }
    val = val / Calibration_Sample_Times;
//calculate the average value
//Ro = Rs * sqrt(a/ppm, b) = Rs * exp( ln(a/ppm) / b )
    return //val;
    (float)val * exp((log(pcurve[0] / ppm) / pcurve[1]));
}

```

Didalam program MQCalibration juga memakai Program MQResistanceCalculation yang telah dijelaskan diatas. Nilai PPM meupakan nilai PPM gas yang ada pada udara bersih. Calibration_Sample_Times merupakan pengambilan multiple sample. Sedangkan Calibration_Sample_Interval adalah lamanya jeda pengambilan nilai val. Val sendiri meupakan nilai yang dihasilkan dari program MQCalibration. Val disini juga bisa di definisikan sebagai ro. Setelah didapat ro selanjutnya akan dicari nilai persentasi konsentrasi gas. Dalam mencari persentasi gas dapat digunakan program sebagai berikut:

```
float MQGetGasPercentage(float rs_ro_ratio, float ro, int
gas_id, int sensor_id)
{
    if (sensor_id == MQ136) {
        if ( gas_id == GAS_SO2 ) {
            return MQGetPercentage(rs_ro_ratio, ro,SO2_Curve);
        } else if ( gas_id == GAS_NH4 ) {
            return MQGetPercentage(rs_ro_ratio, ro,NH4_secCurve);
        } else if ( gas_id == GAS_CO_sec ) {
            return MQGetPercentage(rs_ro_ratio, ro,CO_terCurve);
        }
    }
}

```

```

    } else if ( gas_id == GAS_H2S ) {
        return MQGetPercentage(rs_ro_ratio, ro,H2S_Curve);
    }
}
return 0;
}

```

Karena dalam satu sensor dapat membaca beberapa gas maka diunakan fungsi if untuk memilih gas mana yang akan dideteksi. Fungsi diatas merupakan fungsi MQGetPesentage dengan isi rasio rs terhadap ro, nilai dan kurva gas yang dibaca, nilainya didapat dari kurva karakteistik yang ada pada datasheet. Caa mencainya dapat menggunakan plot nilai PPM dan nilai perbandingan rs banding ro. Funsu diatas hanya menginisialisasi dan memisahkan gas apa yang dibaca. Setelah tahu gas apa yang dibaca maka akan masuk ke program berikut untuk mendapatkan nilai PPM:

```

float MQGetPercentage(float rs_ro_ratio, float ro, float
*pcurve)
{
    return (float)(pcurve[0] * pow(((float)rs_ro_ratio /
ro),pcurve[1]));
}

```

Didalam program diatas tedapat fungsi pow yang digunakan untuk meringkas perkalian suatu bilangan sebanyak pangkat bilangan tersebut. Dalam fungsi diatas berarti perbandingan rs banding ro dibagi dengan ro pangkat nilai curva pada array kedua. Setelah didapatkan hasilnya akan dikalikan dengan nilai curva yang pertama.

3.4.2 Pembacaan Sensor MQ135

Perancangan perangkat lunak pada sensor MQ135 hampir sama dengan perancangan sensor MQ136 dimulai dengan pembacaan Tegangan ADC yang masuk pada Aduino Uno dengan menggunakan program sebagai berikut:

```

float MQResistanceCalculation(int raw_adc, float r1_value)
{
    return
    (long)((long)(1024* (long)r1_value) / raw_adc -
(long)r1_value);}

```

r1_value merupakan nilai r1 yang terukur pada sensor, sedangkan raw_adc merupakan tegangan yang keluar dari analog

output sensor dan dibaca oleh Arduino Uno. Selanjutnya akan dicari nilai rs dengan menggunakan program seperti berikut:

```
float MQRead(int mq_pin, float r1_value)
{
    int i;
    float rs = 0;
    for (i = 0; i < Read_Sample_Times; i++) {
        rs += MQResistanceCalculation(analogRead(mq_pin),
r1_value);
        delay(Read_Sample_Interval);
    }
    rs = rs / Read_Sample_Times;
    return rs;
}
```

Read_Sample_Times merupakan banyaknya sample yang akan diambil. Fungsi analogRead() berfungsi mengambil data analog yang terbaca. Sedangkan Read_Sample_Interval merupakan delay waktu pengambilan satu sample data. Setelah didapat rs, selanjutnya mencari ro, dalam mencari ro dapat digunakan program sebagai berikut:

```
float MQCalibration(int mq_pin, double ppm, double r1_value,
float *pcurve )
{
    int i;
    float val = 0;
    for (i = 0; i < Calibration_Sample_Times; i++) {
        val += MQResistanceCalculation(analogRead(mq_pin),
r1_value);
        delay(Calibration_Sample_Interval);
    }
    val = val / Calibration_Sample_Times;
    return //val;
    (float)val * exp((log(pcurve[0] / ppm) / pcurve[1]));
}
```

Didalam program MQCalibration juga memakai Program MQResistanceCalculation yang telah dijelaskan diatas. Nilai PPM merupakan nilai PPM gas yang ada pada udara bersih. Calibration_Sample_Times merupakan pengambilan multiple sample. Sedangkan Calibration_Sample_Interval adalah lamanya jeda pengambilan nilai val. Val sendiri merupakan nilai yang dihasilkan dari program MQCalibration. Val disini juga bisa di definisikan sebagai ro. Setelah didapat ro selanjutnya akan dicari

nilai persentasi konsentrasi gas. Dalam mencari persentasi gas dapat digunakan program sebagai berikut:

```
float MQGetGasPercentage(float rs_ro_ratio, float ro, int
gas_id, int sensor_id)
{
    if (sensor_id == MQ135) {
        if ( gas_id == Gas_CO2 ) {
            return MQGetPercentage(rs_ro_ratio, ro, CO2_Curve);
        } else if ( gas_id == Gas_C2H5OH ) {
            return MQGetPercentage(rs_ro_ratio, ro, C2H5OH_Curve);
        } else if ( gas_id == Gas_CH3 ) {
            return MQGetPercentage(rs_ro_ratio, ro,CH3_Curve);
        } else if ( gas_id == Gas_CH3_2CO ) {
            return MQGetPercentage(rs_ro_ratio, ro,CH3_2CO_Curve);
        } else if ( gas_id == Gas_CO ) {
            return MQGetPercentage(rs_ro_ratio, ro,CO_Curve);
        }
    }
    return 0;
}
```

Karena dalam satu sensor dapat membaca beberapa gas maka diunakan fungsi if untuk memilih gas mana yang akan dideteksi. Fungsi diatas merupakan fungsi MQGetPesentage dengan isi rasio rs terhadap ro, nilai dan kurva gas yang dibaca, nilanya didapat dari kurva karakteistik yang ada pada datasheet. Caa mencainya dapat menggunakan plot nilai PPM dan nilai perbandingan rs banding ro. Funsu diatas hanya menginisialisasi dan memisahkan gas apa yang dibaca. Setelah tahu gas apa yang dibaca maka akan masuk ke program berikut untuk mendapatkan nilai PPM:

```
Float MQGetPercentage(float rs_ro_ratio, float ro, float
*pcurve)
{
    return (float)(pcurve[0] * pow(((float)rs_ro_ratio /
ro),pcurve[1]));
}
```

Didalam program diatas tedapat fungsi pow yang digunakan untuk meringkas perkalian suatu bilangan sebanyak pangkat bilangan tesebut. Dalam fungsi diatas berarti perbandingan rs banding ro dibagi dengan ro pangkat nilai curva pada array kedua. Setelah didapatkan hasilnya akan dikalikan dengan nilai curva yang pertama.

3.4.3 Pengiriman Data dengan Lora

Pada perancangan perangkat lunak untuk pengiriman data Menggunakan LoRa diawali dengan melakukan inisialisasi dengan mengatur *baudrate* dan inisialisasi LoRa library dengan fungsi `rf95.setFrequency`(frekuensi yang digunakan). Untuk lebih jelasnya dapat dilihat pada program berikut:

```
Serial.begin(9600);
```

```
while (!Serial) ; // Wait for serial port to be available
  if (!rf95.init())
    Serial.println("init failed");
  rf95.setFrequency(frequency);
  rf95.setTxPower(13);
```

Sebelum data dikirim dengan LoRa. Data yang bertipe *Float* diubah menjadi tipe *String*, setelah dirubah kemudian data digabung karena data yang dikirim terdapat lebih dari satu data. Untuk merubah tipe data dari tipe *Float* menjadi tipe *String* digunakan fungsi `dtostrf()`. Dan untuk menggabungkan data digunakan fungsi `strcat()` dan `strcpy()`. Untuk lebih jelasnya dapat dilihat pada program berikut:

```
CO2 = MQGetGasPercentage(MQRead(MQ135, RL0), Ro0, GAS_CO2,
MQ135);
H2S = MQGetGasPercentage(MQRead(MQ136, RL1), Ro1, GAS_H2S,
MQ136);
char data_sensor[100] = {"\0"};
dtostrf(CO2, 5, 2, data_1);
dtostrf(H2S, 5, 2, data_2);
strcat(data_sensor, data_1);
strcat(data_sensor, "\n");
strcat(data_sensor, data_2);
strcat(data_sensor, "\n");
strcpy((char*)data, data_sensor);
Serial.println((char*)data);
```

Setelah data diolah, kemudian data akan dikirim. Dalam library LoRa sudah terdapat fungsi untuk memaketkan data dan sekaligus mengirimnya untuk lebih jelas dapat dilihat pada program berikut:

```
rf95.send(data, sizeof(data));
rf95.waitPacketSent();
```

Dalam program diatas `rf95.send(data, sizeof(data));` merupakan fungsi digunakan untuk mengirimkan data. Didalam fungsi tersebut didisi dengan data yang akan dikirim dan ukuran data yang akan dikirim. Sedangkan fungsi `rf95.waitPacketSent();` digunakan untuk menunggu data terkirim ke penerima.

3.4.4 Penerimaan Data dengan LoRa

Pada perancangan perangkat lunak untuk penerimaan data Menggunakan LoRa diawali dengan melakukan inisialisasi dengan mengatur *baudrate* dan inisialisasi LoRa library dengan fungsi `rf95.setFrequency(frekuensi yang digunakan)`. Untuk lebih jelasnya dapat dilihat pada program berikut:

```
Serial.begin(9600);
while (!Serial) ; // Wait for serial port to be available
if (!rf95.init())
  Serial.println("init failed");
rf95.setFrequency(frequency);
rf95.setTxPower(13);
```

Setelah diinisialisasi, selanjutnya harus ditentukan banyaknya paket yang akan diterima. Ukuran paket yang diterima harus sama dengan ukuran paket yang dikirim. Untuk menentukan ukuran paket dapat digunakan program sebagai berikut:

```
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
```

setelah ukuran paket ditentukan, selanjutnya maka dilanjutkan penerimaan data yang akan disimpan kedalam variable data menggunakan program sebagai berikut:

```
rf95.recv(buf, &len);
```

3.4.5 Pengiriman Data ke NodeMCU

Perancangan perangkat lunak untuk pengiriman data dari Arduino Uno dilakukan dengan melakukan inisialisasi. Karena tidak menggunakan pin RX dan TX maka digunakanlah *library* SoftwareSerial. Setelah itu menentukan data awal, data akhir, dan data yang akan dikirim, juga menentukan baudrate pada SoftwareSerial seperti pada program berikut:

```
SoftwareSerial s(7,6);
char buff[3][10];
char initial;
```

```

char ln = '\n';
char start = '<';
char stopped = '>';

s.begin(9600);

```

Setelah itu data akan dikirim dengan suatu fungsi yang ada pada library SoftwareSerial. Fungsi yang dipakai adalah fungsi `name_Serial.write()`. Data yang dikirim adalah data start, data yang diterima dari LoRa yang disimpan pada variabel `data`, dan yang terakhir adalah data `stopped`. Untuk lebih jelasnya bisa dilihat pada program berikut:

```

s.write(start);
s.write(data,sizeof(data));
s.write(stopped);

```

3.4.6 Pengolahan Data di NodeMCU

Setelah data di terima di NodeMCU, dua data sensor yang masih menjadi satu kemudian dipisah menjadi 2. Program yang digunakan untuk emisah data seperti berikut:

```

for(i = 0; i <= numChars ; i++){
    if(receivedChars[i] != line){
        dataSensor[data_ke][index] = receivedChars[i];
        index++;
    }
    else{
        data_ke++;
        index = 0;
    }
}

```

3.4.7 Pengiriman Data ke Aplikasi Blynk

Setelah data dipisah kemudian untuk mengirim data sensor dan label data ke peta yang ada di Blynk data dan label yang berupa string di gabung menjadi satu dan diberi pemisah berupa karakter spasi. Fungsi yang digunakan seperti fungsi `strcat` dan `strcpy`. Untuk lebih jelasnya dapat dilihat pada program berikut:

```

char dataSenMap[50] = {"\0"};
strcat(dataSenMap, "CO2 =");
strcat(dataSenMap, dataSensor[0]);
strcat(dataSenMap, " ");
strcat(dataSenMap, "H2S =");
strcat(dataSenMap, dataSensor[1]);
strcpy(dataMap, dataSenMap);

```

Setelah data dipisah kemudian dilakukan pengiriman data ke aplikasi Blynk. Fungsi yang dipakai adalah `Blynk.virtualWrite()`. Pada fungsi ini berisi pin yang akan digunakan pada aplikasi Blynk yang kita tentukan sendiri dan data yang akan dikirim. Sedangkan untuk menampilkan data ke peta isi dari fungsi tersebut terdapat latitude dan longitude yang merupakan koordinat yang kita berikan.

```
Blynk.virtualWrite(V1, dataSensor[0]);  
Blynk.virtualWrite(V2, dataSensor[1]);  
Blynk.virtualWrite(V4, index_1, lat, lon, dataMap);
```

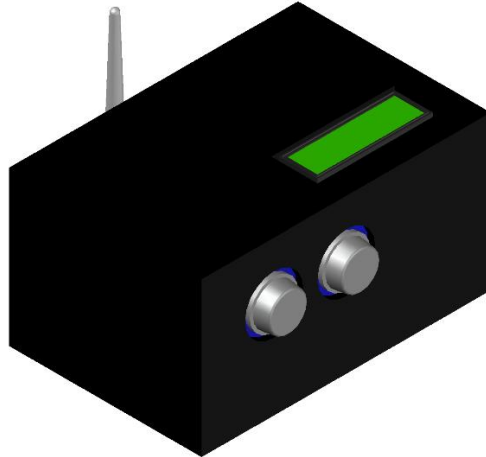
Untuk menghubungkan NodeMCU ke aplikasi Blynk pertama harus diinisiasi menggunakan fungsi `Blynk.begin()`. Fungsi ini berisi `auth`, `ssid`, `pass`. `Auth` merupakan nomor autentifikasi yang didapat dari aplikasi Blynk. `ssid` merupakan nama WiFi atau jaringan yang digunakan untuk menghubungkan NodeMCU dan aplikasi Blynk. `pass` merupakan password dari WiFi atau jaringan yang digunakan. Dan untuk menjalankan digunakan fungsi `Blynk.run()`. Fungsi diatas sudah tersedia pada *library* Blynk yang dapat diunduh secara gratis.

```
Blynk.begin(auth, ssid, pass);  
Blynk.run();
```

3.5 Perancangan Kotak Packaging

3.5.1 Kotak Packaging Transmitter

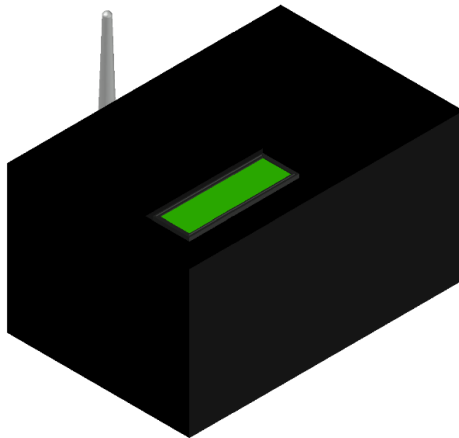
Pada box transmitter ini berisi Shield board PCB terpadu, Sensor MQ136, MQ135, LoRa Shield RF96, Arduino Uno, dan LCD I2C. Suplai tegangan VCC menggunakan kabel serabut yang dihubungkan ke *pin screw*. Ada dua jenis Suplai yaitu suplai 5 Volt dan suplai 24 Volt yang ditempatkan pada *pin screw*. Sedangkan untuk kabel terminal sensor dan LCD menggunakan *pin Molex*. Box sensor berfungsi sebagai wadah komponen. Bentuk desain dari kotak *transmitter* dapat dilihat pada gambar 3.12.



Gambar 3.12 Kotak *transmitter*

3.5.2 Kotak Packaging Receiver

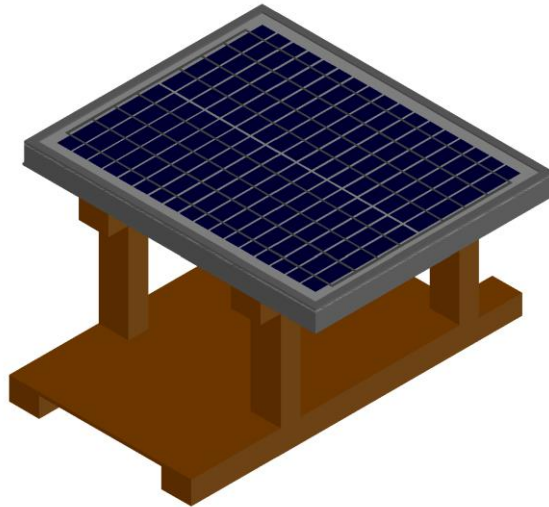
Pada box receiver ini berisi Shield board PCB terpadu receiver, LoRa Shield RF96, Arduino Arduino Uno, dan LCD I2C. Suplai tegangan Vcc menggunakan kabel serabut yang dibuhungkan ke *pin screw*. Suplai yang digunakan yaitu suplai 5 Volt yang ditempatkan pada *pin screw*. Sedangkan untuk kabel terminal LCD menggunakan *pin Molex*. Box sensor berfungsi sebagai wadah komponen. Bentuk desain dari kotak *transmitter* dapat dilihat pada gamba 3.13.



Gambar 3.13 Kotak *receiver*

3.5.3 Rangka Penopang Sel Surya dan Kotak Transmitter

Perancangan Rangka Penopang terbuat dari kayu yang dibentuk sedemikian rupa. Karena suplai daya berasal dari sel surya, pada rangka ini juga sebagai tempat komponen seperti baterai UPS atau ACCU yang dapat di *charge* menggunakan sel surya melalui sebuah *controller*. Selain itu ada juga sebuah *Solar Charge Controller*. *Solar Charge Controller* ini memiliki 3 terminal utama yaitu terminal *input* untuk input tegangan dari sel surya, terminal baterai untuk menghubungkan baterai dengan *Solar Charge Controller* baik digunakan untuk mengisi daya baterai atau menyalurkan daya yang tersimpan pada baterai ke beban, dan yang terakhir ada terminal beban untuk menghubungkan *Solar Charge Controller* dengan beban. Dan ada *Buck & Boost Converter* yang merupkan penurun atau penaik tegangan dari *Solar Charge Controller* sebelum masuk ke beban. Juga sebagai tempat meletakkan kotak *transmitter* yang berisi sensor. Untuk gambar desain rangka penopang sel surya dapat dilihat pada gambar 3.14.



Gambar 3.14 Rangka penopang sel surya

.....*Halaman ini sengaja dikosongkan*.....

BAB IV PENGUJIAN DAN ANALISIS

4.1 Pengujian Sensor Gas

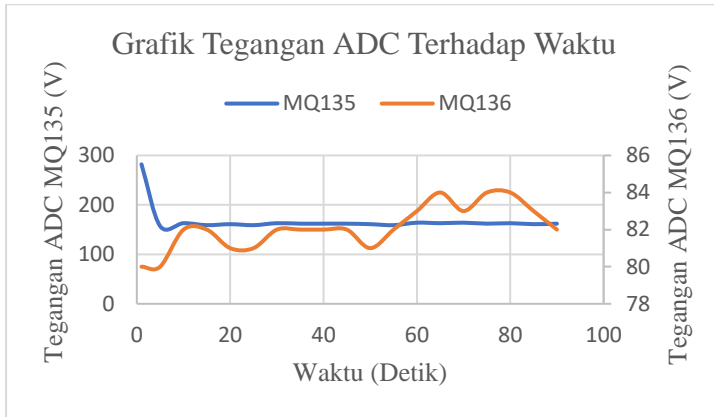
Pengujian sensor gas, baik sensor gas MQ136 maupun MQ135 yang sudah terpasang pada *board* mikrokontroler dilakukan secara bersama-sama yang bertempat di gedung B lantai 4 Teknik Elektro ITS. Pengujian ini dilakukan untuk mengetahui apakah sensor gas MQ136 dan sensor gas MQ135 yang terpasang pada *board* sudah bekerja dan terkalibrasi dengan baik. Pengujian dilakukan dengan empat kondisi, yaitu kondisi awal merupakan kondisi ketika udara bersih, kondisi kedua ketika ada asap kertas, kondisi ketiga ketika ada asap kayu, dan kondisi keempat ketika ada uap larutan $CaCl_2$.

4.1.1 Pengujian Sensor Terhadap Udara Bersih

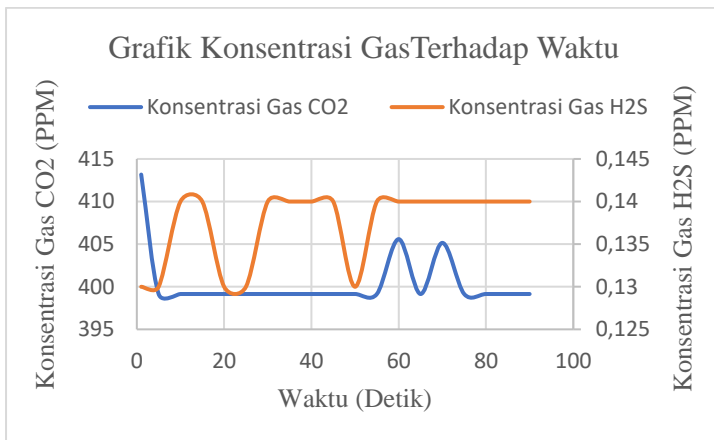
Pada pengujian ini dilakukan dengan udara yang ada pada sekitar lokasi uji, yaitu gedung B lantai 4. Pengujian ini dilakukan dengan tidak memperhitungkan adanya polusi yang ada pada lokasi uji. Sehingga data yang didapat sedikit sedikit melenceng. Seperti misalnya pada udara bersih konsentrasi gas CO_2 adalah mulai dari 385 PPM sampai 400 PPM.

Tabel 4.1 Pengujian sensor terhadap udara bersih.

Waktu (s)	Tegangan Terbaca		PPM	
	MQ136	MQ135	MQ136	MQ135
1	80	282	0,13	413,17
5	80	157	0,13	399,15
10	82	163	0,14	399,15
15	82	159	0,14	399,15
20	81	161	0,13	399,15
25	81	159	0,13	399,15
30	82	163	0,14	399,15
35	82	162	0,14	399,15
40	82	162	0,14	399,15
45	82	162	0,14	399,15
50	81	161	0,13	399,15
55	82	159	0,14	399,15
60	83	164	0,14	405,59
65	84	163	0,14	399,15
70	83	164	0,14	405,15
75	84	162	0,14	399,15
80	84	163	0,14	399,15
85	83	161	0,14	399,15
90	82	162	0,14	399,15



Gambar 4.1 Grafik tegangan terhadap waktu pada udara bersih



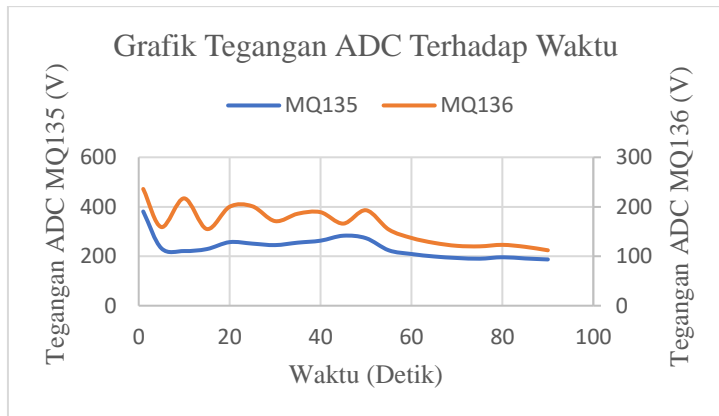
Gambar 4.2 Grafik konsentrasi terhadap waktu pada udara bersih

4.1.2 Pengujian Sensor Terhadap Asap Kertas

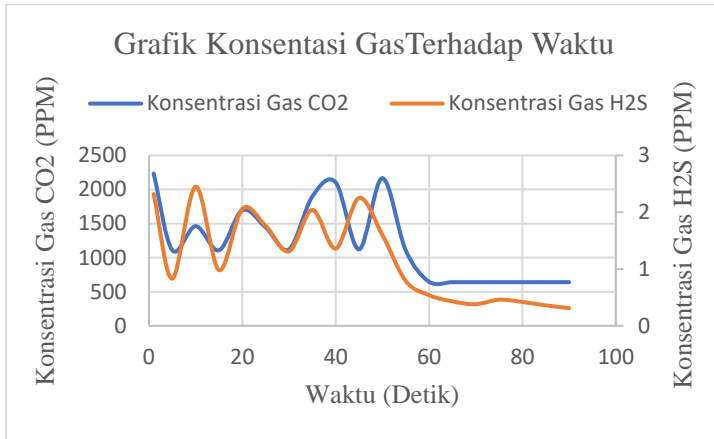
Pada pengujian terhadap asap kertas dilakukan dengan membakar kertas sehingga dapat menghasilkan asap. Pengujian ini tidak memperhitungkan adanya tiupan angin maupun jarak antara asap dengan sensor. Karena faktor tersebut data yang terbaca berubah ubah tidak mengikuti perubahan volume asap. Juga ada factor tidak stabilnya asap.

Tabel 4.2 Pengujian sensor terhadap asap kertas.

Waktu (s)	Tegangan Terbaca		PPM	
	MQ136	MQ135	MQ136	MQ135
1	236	381	2,32	2234,02
5	159	232	0,83	1108,25
10	217	221	2,45	1461,97
15	155	229	0,98	1108,25
20	200	257	2,06	1696,28
25	201	251	1,76	1442,86
30	171	245	1,31	1121,69
35	186	255	2,04	1899,02
40	189	263	1,36	2103,72
45	166	283	2,25	1121,69
50	193	273	1,60	2167,57
55	154	224	0,79	1108,25
60	137	209	0,54	646,80
65	127	199	0,43	640,33
70	121	193	0,38	640,33
75	120	190	0,46	640,33
80	123	196	0,42	640,33
85	119	191	0,36	640,33
90	112	187	0,31	640,33



Gambar 4.3 Grafik tegangan terhadap waktu pada asap kertas



Gambar 4.4 Grafik konsentrasi terhadap waktu pada asap kertas

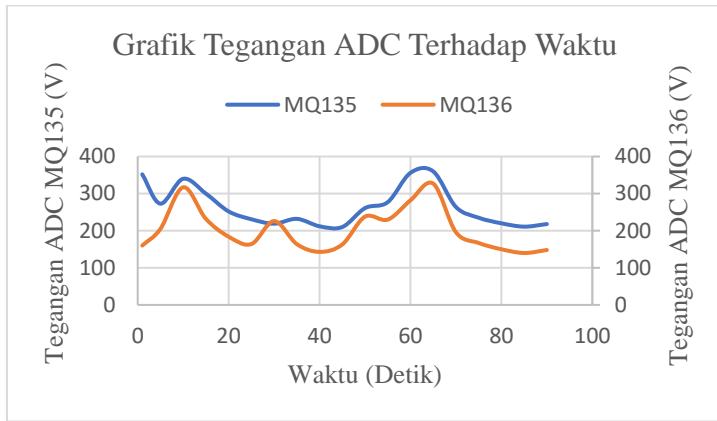
4.1.3 Pengujian Sensor Terhadap Asap Kayu

Pada pengujian terhadap asap kayu dilakukan dengan membakar kayu sehingga dapat menghasilkan asap. Pengujian ini tidak memperhitungkan adanya tiupan angin maupun jarak antara asap dengan sensor. Karena faktor tersebut data yang terbaca berubah ubah tidak mengikuti perubahan volume asap. Juga ada faktor tidak stabilnya asap.

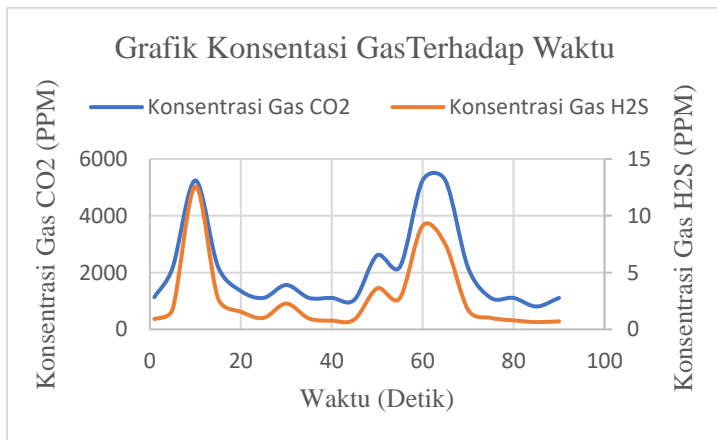
Tabel 4.3 Pengujian sensor terhadap asap kayu.

Waktu (s)	Tegangan Terbaca		PPM	
	MQ136	MQ135	MQ136	MQ135
1	160	352	0,91	1135,35
5	205	273	1,76	2167,57
10	317	340	12,58	5245,58
15	232	300	2,72	2200,46
20	184	252	1,56	1352,13
25	164	231	1,01	1108,25
30	226	219	2,27	1562,75
35	164	232	0,96	1108,25
40	143	212	0,77	1108,25
45	163	210	0,87	1044,17
50	238	261	3,61	2610,38
55	230	277	2,76	2200,46
60	282	356	9,12	5245,58

65	327	360	7,48	5245,58
70	196	264	1,72	2167,57
75	167	235	1,01	1108,25
80	150	220	0,79	1108,25
85	140	211	0,64	805,27
90	148	218	0,71	1108,25



Gambar 4.5 Grafik tegangan terhadap waktu pada asap kayu



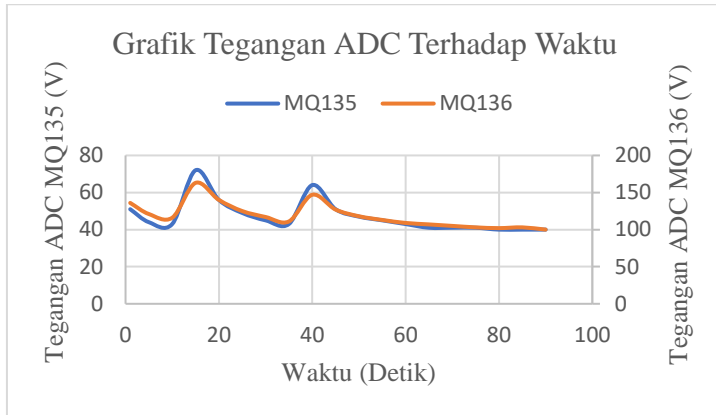
Gambar 4.6 Grafik konsentrasi terhadap waktu pada asap kayu

4.1.4 Pengujian Sensor Terhadap Larutan CaC_2

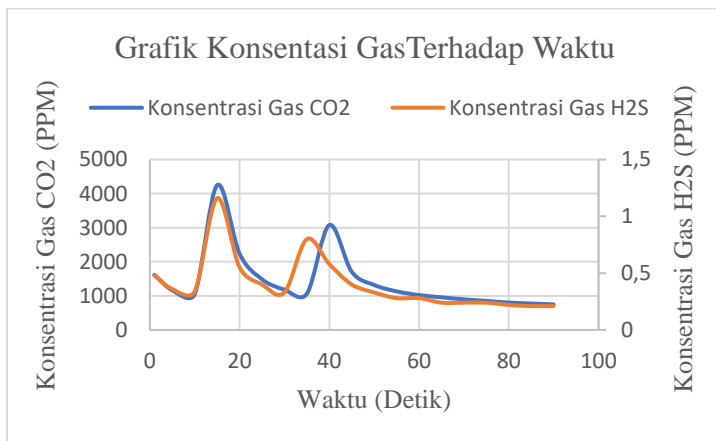
Pada pengujian terhadap larutan CaC_2 dilakukan dengan memanaskan larutan CaC_2 dengan api sehingga menghasilkan uap. Pengujian ini tidak memperhitungkan adanya tiupan angin maupun jarak antara uap larutan dengan sensor. Karena faktor tersebut data yang terbaca berubah ubah tidak mengikuti perubahan banyaknya uap. Juga ada faktor tidak stabilnya api sebagai sumber uap.

Tabel 4.4 Pengujian sensor terhadap larutan CaC_2

Waktu (s)	Tegangan Terbaca		PPM	
	MQ136	MQ135	MQ136	MQ135
1	136	51	0,48	1619,42
5	121	44	0,36	1165,53
10	116	43	0,34	1054,28
15	163	72	1,16	4249,61
20	140	56	0,55	2234,55
25	125	49	0,40	1485,94
30	117	45	0,33	1181,97
35	111	43	0,8	1062,88
40	147	64	0,58	3076,34
45	127	51	0,40	1705,63
50	118	47	0,33	1319,52
55	113	45	0,28	1131,99
60	109	43	0,28	1023,54
65	107	41	0,24	956,72
70	105	41	0,24	897,89
75	103	41	0,24	851,24
80	102	40	0,22	802,78
85	103	40	0,21	775,64
90	100	40	0,21	748,81



Gambar 4.7 Grafik tegangan terhadap waktu pada larutan CaC_2



Gambar 4.8 Grafik konsentrasi terhadap waktu pada larutan CaC_2

4.2 Pengujian Sensor per Volume

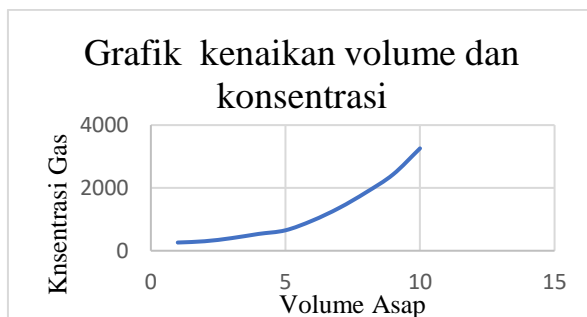
4.2.1 Pengujian Sensor MQ135

Pengujian ini dilakukan untuk mengetahui respon sensor terhadap volume asap yang diberikan. Asap yang digunakan merupakan asap yang berasal dari pembakaran kertas. Metode pengujian ini dilakukan dengan membakar kertas pada sebuah kaleng tertutup yang sudah diberi lubang kecil untuk mengambil asapnya. Untuk mengambil asap digunakan tabung suntik yang ada ukuran volumenya, satuan ukuran volumenya adalah mililiter.

Pengujian dilakukan dengan memberikan sensor asap yang diambil dengan tabung suntik. Sensornya sendiri berada didalam tabung plastik yang sudah diberi lubang untuk memasukkan asap. Hasil yang didapatkan setiap kenaikan satu mililiter volume nilai PPM yang terbaca juga naik. Tetapi kenaikan nilai PPM tidak naik secara linier sempurna. Hal ini disebabkan karena ketika mengambil sasap dari kaleng dan memindahkan ke tabung yang berisi sensor terdapat asap yang hilang, karena sumber yang digunakan adalah asap yang dapat keluar dari lubang yang kecil sekalipun.

Tabel 4.5 Pengujian sensor terhadap asap Kertas

Volume (mL)	MQ135 (PPM)
1	261,75
2	302,69
3	401,78
4	534,93
5	651,09
6	958,8
7	1363,65
8	1860,16
9	2430,19
10	3258,48



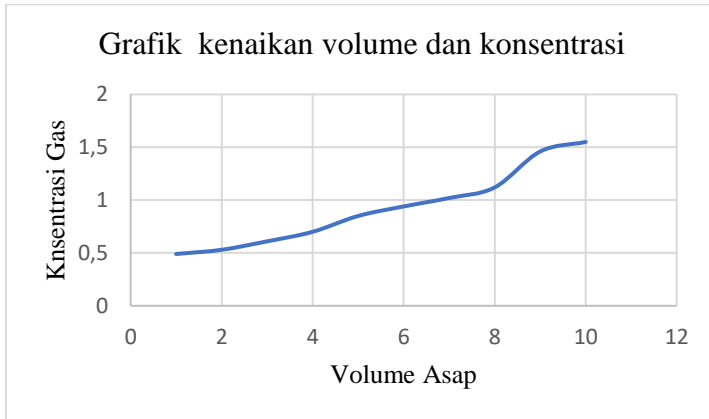
Gambar 4.9 Grafik hubungan antara kenaikan volume dan konsentrasi

4.2.2 Pengujian Sensor MQ136

Pengujian ini dilakukan untuk mengetahui respon sensor terhadap volume asap yang diberikan. Asap yang digunakan merupakan asap yang berasal dari pembakaran rokok. Metode pengujian ini dilakukan dengan membakar rokok pada sebuah kaleng tertutup yang sudah diberi lubang kecil untuk mengambil asapnya. Untuk mengambil asap digunakan tabung suntik yang ada ukuran volumenya, satuan ukuran volumenya adalah mililiter. Pengujian dilakukan dengan memberikan sensor asap yang diambil dengan tabung suntik. Sensornya sendiri berada didalam tabung plastik yang sudah diberi lubang untuk memasukkan asap. Hasil yang didapatkan setiap kenaikan satu mililiter volume nilai PPM yang terbaca juga naik. Tetapi kenaikan nilai PPM tidak naik secara linier sempurna. Hal ini disebabkan karena ketika mengambil asap dari kaleng dan memindahkan ke tabung yang berisi sensor terdapat asap yang hilang, karena sumber yang digunakan adalah asap yang dapat keluar dari lubang yang kecil sekalipun.

Tabel 4.6 Pengujian sensor terhadap asap rokok

Volume (mL)	MQ136 (PPM)
1	0,49
2	0,53
3	0,61
4	0,7
5	0,85
6	0,94
7	1,02
8	1,12
9	1,46
10	1,55



Gambar 4.10 Grafik hubungan antara kenaikan volume dan konsentrasi

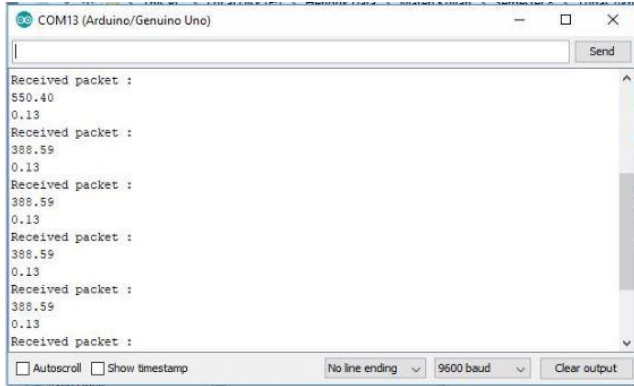
4.3 Pengujian LoRa

4.3.1 Pengujian LoRa dengan Posisi Receiver pada Gedung B lantai 4

Pengujian pengiriman data menggunakan LoRa yang pertama dilakukan dengan meletakkan LoRa Receiver pada gedung B lantai 4 Teknik Elektro ITS. Jarak yang bisa dicapai pada saat kondisi ini maksimal hanya sejauh 200 m. setelah jarak lebih dari dua meter LoRa *Transmitter* masih mengirimkan data tetapi LoRa Receiver tidak dapat menerima. Ada beberapa fakto yan menyebabkannya seperti posisi *receiver* dan *transmitter* yang berada diantara banyak gedung tinggi sehingga dapat menghalangi sinyal pengiriman.

Tabel 4.7 Pengujian pertama LoRa.

Jarak (m)	Data Tekirim		Data Diterima	
	MQ136	MQ135	MQ136	MQ135
50	0.13	388.59	0.13	388.59
100	0.13	354.25	0.13	354.25
200	0.13	282.69	0.13	282.69



Gambar 4.11 Penerimaan data jarak 50 meter



Gambar 4.12 Pengiriman data jarak 50 meter



Gambar 4.13 Penerimaan data jarak 200 meter



Gambar 4.14 Pengiriman data jarak 200 meter

4.3.2 Pengujian LoRa dengan Posisi Receiver di Jl. Teknik Perkapalan ITS

Pengujian pengiriman data menggunakan LoRa yang kedua dilakukan dengan meletakkan LoRa *receiver* pada sudut utara l. Teknik Pekapalan ITS, sedangkan *Transmitter* diletakkan pada jarak 620 m dari receiver pada garis lurus tanpa adanya ketinggian. Jarak yang bisa dicapai pada saat kondisi ini maksimal hanya sejauh 600 m. pada awalnya diletakkan sejauh 620 LoRa *Receiver* tidak dapat menerima data sama sekali. setelah mendekatkan jarak LoRa sekitar kurang lebih 600 meter, LoRa baru dapat menerima data dengan baik lebih dari 600 meter, LoRa *Transmitter* masih mengirimkan data tetapi LoRa *Receiver* tidak dapat menerima. Ada beberapa faktor yang memengaruhinya seperti interferensi atau gangguan frekuensi dai sinyal yang ada pada daerah perkotaan.

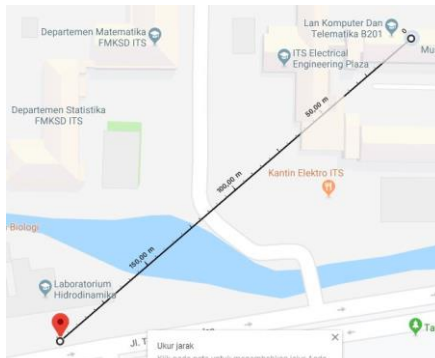
Tabel 4.8 Pengujian kedua LoRa.

Jarak (m)	Data Terkirim		Data diterima	
	MQ136	MQ135	MQ136	MQ135
100	0.13	550.40	0.13	550.40
200	0.13	480.47	0.13	480.47
300	0.13	220.35	0.13	220.35
400	0.22	1322.00	0.22	1322.00
500	0.13	388.59	0.13	388.59
600	0.13	128.01	0.13	128.01

4.3.3 Metode Pengukuran Jarak

Dalam penentuan jarak dilakukan dengan menentukan koordinat awal menggunakan aplikasi google map. Setelah

didapatkan koordinat awal maka selanjutnya dilakukan pengukuran jarak sesuai dengan yang akan diuji, dengan titik awal adalah koordinat awal. Setelah didapatkan jarak yang sesuai, kemudian catat koordinatnya.

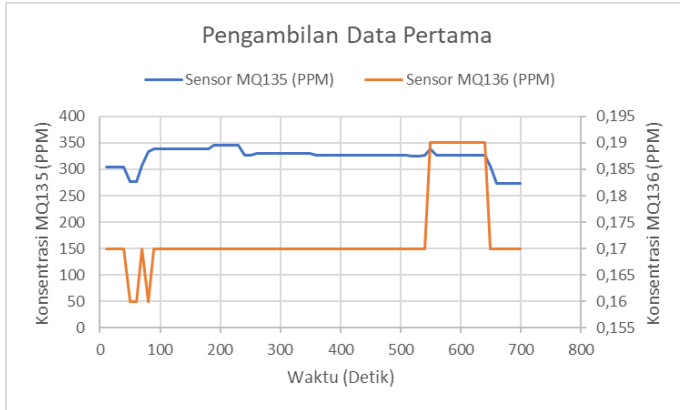


Gambar 4.15 Metode pengukuran jarak

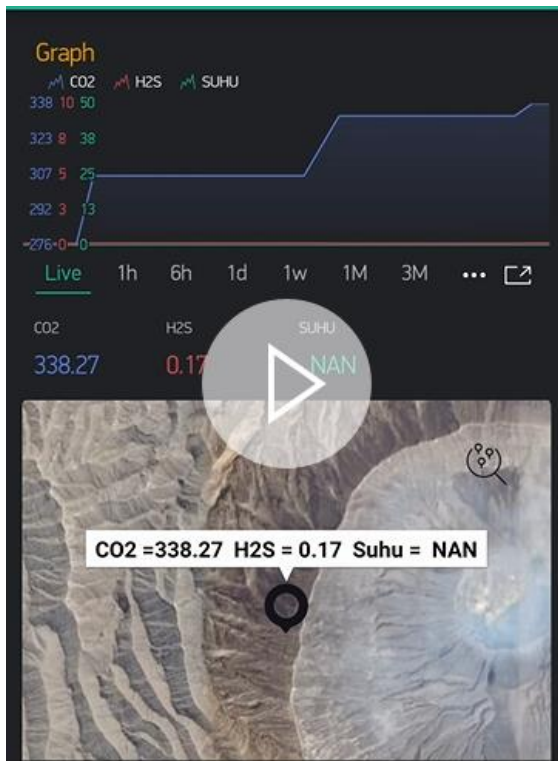
4.1 Pengujian Alat Secara Keseluruhan

4.4.1 Pengambilan Data Pertama

Pengambilan data pertama dilakukan di wilayah gunung Bromo. Pengujian dilakukan dengan meletakkan sensor di puncak dekat kawah gunung Bromo. Sedangkan penerima diletakkan pada jarak sekitar 3 Km. Pengamatan dilakukan selama 700 detik pengujian juga dilakukan tidak berdiam di tempat tetapi sambil berjalan kembali menuju puncak. Hasil yang didapatkan adalah konsentrasi gas CO_2 dan konsentrasi gas H_2S berubah-ubah. Perubahan konsentrasi gas CO_2 lebih terlihat dibandingkan konsentrasi gas H_2S yang perubahannya sangat kecil. Nilai konsentrasi yang berubah-ubah disebabkan oleh gas yang dibawa oleh asap yang keluar dari kawah gunung Bromo dipegaruhi oleh angin yang bertiup di puncak gunung Bromo. Angin ini menentukan arah asapnya, sehingga asap yang membawa gas tidak selalu teraca oleh sensor. Dengan jarak sejauh sekitar 3 Km dari posisi sensor, penerima masih menerima data dengan baik. Pada Jarak sekitar 3 Km itu merupakan posisi ujung dekat dengan tebing yang mengelilingi gunung Bromo, sehingga untuk pengujian dengan yang jauh tidak bisa dilakukan. Respon yang didapatkan dari sensor cukup lambat dikarenakan prinsip kerja sensor tersebut.



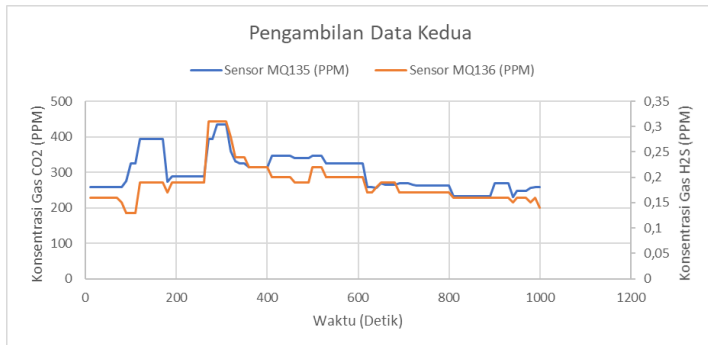
Gambar 4.16 Grafik pengambilan data pertama



Gambar 4.17 Tampilan pengambilan data pertama pada Blynk

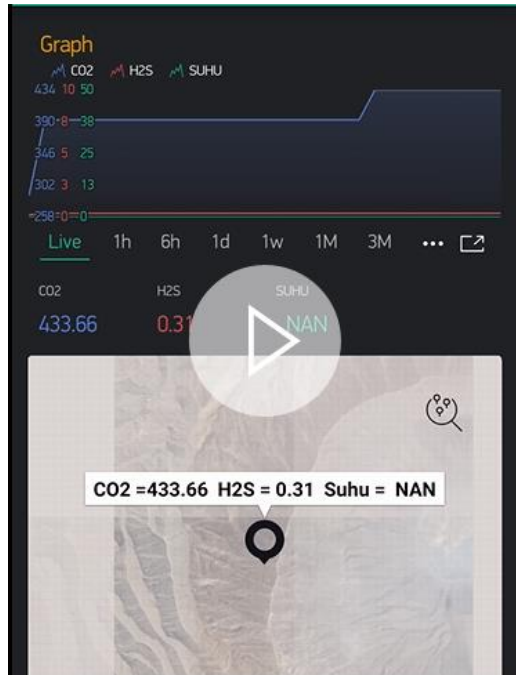
4.4.2 Pengambilan Data Kedua

Pengambilan data kedua juga dilakukan di wilayah gunung Bromo, pengujian ini merupakan lanjutan dari pengujian pertama. Pengujian dilakukan dengan meletakkan sensor di puncak dekat kawah gunung Bromo. Sedangkan penerima diletakkan pada jarak sekitar 2,5 Km. Pengamatan dilakukan selama 1000 detik pengujian juga dilakukan tidak berdiam di tempat tetapi sambil berjalan kembali menuju puncak. Hasil yang didapatkan adalah konsentrasi gas CO_2 dan konsentrasi gas H_2S berubah-ubah. Perubahan konsentrasi gas CO_2 lebih terlihat dibandingkan konsentrasi gas H_2S yang perubahannya sangat kecil. Nilai konsentrasi yang berubah-ubah disebabkan oleh gas yang dibawa oleh asap yang keluar dari kawah gunung Bromo dipegaruhi oleh angin yang bertiup di puncak gunung Bromo. Angin ini menentukan arah asapnya, sehingga asap yang membawa gas tidak selalu tebaca oleh sensor. Dengan jarak sejauh sekitar 2,5 Km dari posisi sensor, penerima masih menerima data dengan baik. Respon yang didapatkan dari sensor cukup lambat dikarenakan prinsip kerja sensor tersebut.



Gambar 4.18 Grafik pengambilan data kedua

Tampilan dari aplikasi Blynk berisi grafik nilai sensor, nilai angka sensor dan tampilan di google map yang menunjukkan posisi dari sensor yang diletakkan juga nilai sensor yang terbaca. Untuk contoh visualisasi tampilan data yang ada pada aplikasi Blynk yang ada di *smatphone* kuran lebihnya dapat dilihat pada gambar 4.19.

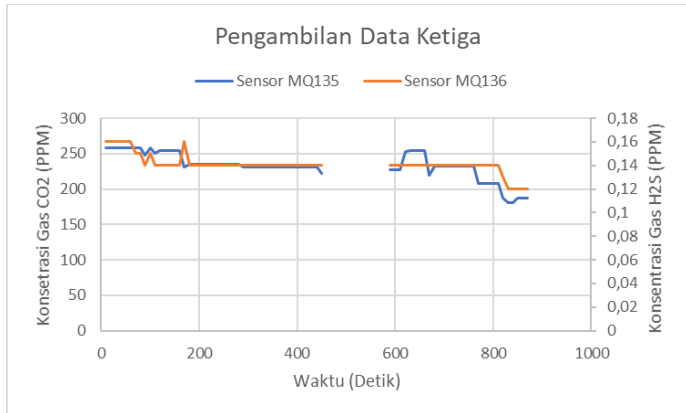


Gambar 4.19 Tampilan pengambilan data kedua pada Blynk

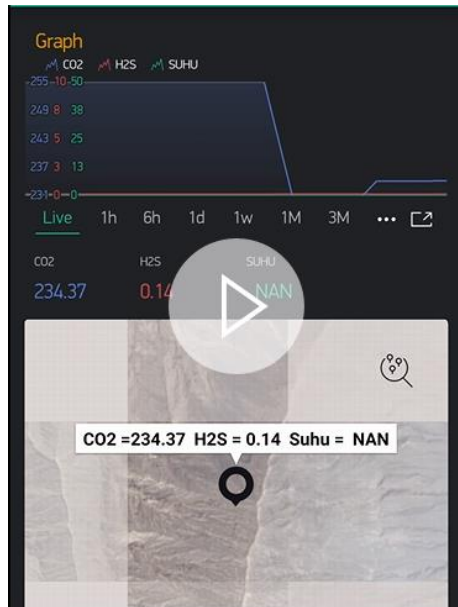
4.4.3 Pengambilan Data Ketiga

Pengambilan data ketiga juga dilakukan di wilayah gunung Bromo. Pengujian dilakukan dengan meletakkan sensor di puncak dekat kawah gunung Bromo. Sedangkan penerima diletakkan pada jarak sekitar 1,5 Km. Pengamatan dilakukan selama 870 detik pengujian juga dilakukan tidak berdiam di tempat tetapi sambil berjalan kembali menuju puncak. Hasil yang didapatkan adalah konsentrasi gas CO_2 dan konsentrasi gas H_2S berubah-ubah. Perubahan konsentrasi gas CO_2 lebih terlihat dibandingkan konsentrasi gas H_2S yang perubahannya sangat kecil. Nilai konsentrasi yang berubah-ubah disebabkan oleh gas yang dibawa oleh asap yang keluar dari kawah gunung Bromo dipegaruhi oleh angin yang bertiup di puncak gunung Bromo. Angin ini menentukan arah asapnya, sehingga asap yang membawa gas tidak selalu teBaca oleh sensor. Pada pengujian ini terdapat data yang kosong dikaenakan aplikasi Blynk yang terdapat pada *smartphone* memulai koneksi ulang. Dengan jarak sejauh 1,5 Km dari posisi sensor,

penerima masih menerima data dengan baik. Respon yang didapatkan dari sensor cukup lambat dikarenakan prinsip kerja sensor tersebut.



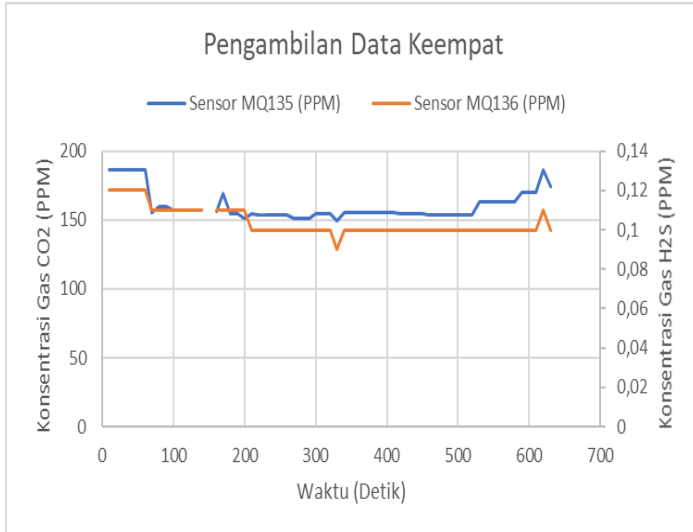
Gambar 4.20 Grafik pengambilan data ketiga



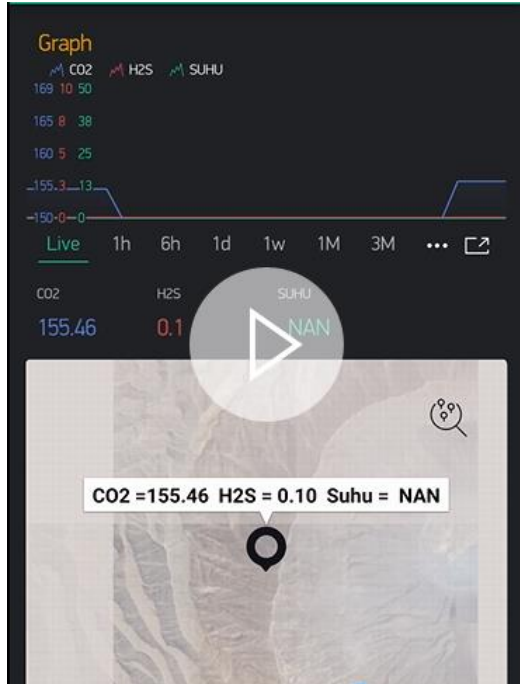
Gambar 4.21 Tampilan pengambilan data ketiga pada Blynk

4.4.4 Pengambilan Data Keempat

Pengambilan data ketiga juga dilakukan di wilayah gunung Bromo. Pengujian dilakukan dengan meletakkan sensor di puncak dekat kawah gunung Bromo. Sedangkan penerima diletakkan pada jarak sekitar 500 m. Pengamatan dilakukan selama 630 detik pengujian juga dilakukan tidak berdiam di tempat tetapi sambil berjalan kembali menuju puncak. Hasil yang didapatkan adalah konsentrasi gas CO_2 dan konsentrasi gas H_2S berubah-ubah. Perubahan konsentrasi gas CO_2 lebih terlihat dibandingkan konsentrasi gas H_2S yang perubahannya sangat kecil. Nilai konsentrasi yang berubah-ubah disebabkan oleh gas yang dibawa oleh asap yang keluar dari kawah gunung Bromo dipegaruhi oleh angin yang bertiup di puncak gunung Bromo. Angin ini menentukan arah asapnya, sehingga asap yang membawa gas tidak selalu teraca oleh sensor. Pada pengujian ini terdapat data yang kosong dikaenakan aplikasi Blynk yang terdapat pada *smartphone* memulai koneksi ulang, tetapi saat emulai ulang tidak selama yang ada pada percobaan ketiga. Dengan jarak sejauh 500 m dari posisi sensor, penerima masih menerima data dengan baik. Respon yang didapatkan dari sensor cukup lambat dikarenakan prinsip kerja sensor tersebut.



Gambar 4.22 Grafik pengambilan data keempat



Gambar 4.23 Tampilan pengambilan data keempat pada Blynk

.....*Halaman ini sengaja dikosongkan*.....

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan yaitu, Pembacaan sensor masih belum sempurna karena adanya beberapa faktor, terutama faktor dari alam, faktor itu diantaranya adalah tiupan angin yang menentukan sumber gas itu sampai ke sensor dengan baik atau tidak, juga posisi peletakan dari sensor terhadap sumber jauh atau tidak. Kalibrasi sensor gas harus sering diperhatikan, karena kadang berubah-ubah, tergantung pada kondisi tertentu seperti pengaruh dari tegangan input yang digunakan. Divais LoRa dapat mengirimkan data dengan baik pada jarak kurang lebih 600meter jika dicoba pada tempat yang memiliki halangan yang banyak seperti memiliki banyak gedung sehingga posisi antena yang tidak bisa saling terlihat. Pada saat pengujian keseluruhan di wilayah gunung Bromo yang tidak ada halangan alat ini dapat menjangkau jarak jauh sekitar 3 Km.

5.2 Saran

Sebagai sarana pengembangan Alat Pemantauan Aktivitas Gunung Berapi ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan seperti, pembacaan dari sensor gas yang berubah ubah tergantung pada seberapa dekat sensor tersebut dengan sumber gas dan kuatnya tiupan diperlukan divais tambahan berupa kipas yang dapat menyedot udara. Pengiriman data secara nirkabel memerlukan jenis divais untuk pengiriman data yang andal yang mampu menembus halangan dengan baik. Penelitian selanjutnya dapat memperbanyak jumlah sensor gas lain sehingga pemantauan lebih variatif dan dapat mengetahui apa saja gas yang berbahaya bagi manusia. Penelitian selanjutnya disarankan untuk menjalankan penelitian secara tim tidak secara individu, karena untuk mengamati komunikasi secara nirkabel susah jika diamati seorang diri.

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR PUSTAKA

- [1] R. P. W. Duin, M. Orozco-Alzate, and J. M. Londono-Bonilla, "Classification of Volcano Events Observed by Multiple Seismic Stations," in *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, 2010, pp. 1052–1055.
- [2] S. S. Hidayat, A. Suharjono, M. Anif, and S. Koesuma, "Design of Real-Time Gas Monitoring System Based-on Wireless Sensor Network(s for Merapi Volcano," *st International Conference on Information Technology*, p. 5, 2014.
- [3] I. Pratomo, "Klasifikasi gunung api aktif Indonesia, studi kasus dari beberapa letusan gunung api dalam sejarah," *Indonesian Journal on Geoscience*, 2006.
- [4] P. Gumilang, "BMKG: Gas Beracun Ada di Sekitar Letusan Kawah Dieng," *nasional*. [Online]. Available: <https://www.cnnindonesia.com/nasional/20170702150926-20-225222/bmkg-gas-beracun-ada-di-sekitar-letusan-kawah-dieng>. [Accessed: 14-Nov-2018].
- [5] "Pusat Vulkanologi dan Mitigasi Bencana Geologi - Badan Geologi." [Online]. Available: <http://pvmbg.geologi.esdm.go.id/index.php/gunungapi/aktivitas-gunungapi/1607-kronologis-letusan-freatik-kawah-sileri-2-juli-2017>. [Accessed: 26-Jun-2019].
- [6] J. Pramono and T. Susila, "Perancangan Alat Telemetri Temperatur dan Gas Pada Gunung Berapi Secara Wireless," p. 16.
- [7] M. Titos, A. Bueno, L. Garcia, and C. Benitez, "A Deep Neural Networks Approach to Automatic Recognition Systems for Volcano-Seismic Events," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1533–1544, May 2018.
- [8] H. A. Sujono, M. Rivai, and M. Amin, "Asthma Identification Using Gas Sensors and Support Vector Machine," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 4, p. 1468, Aug. 2018.
- [9] W. F. Giggenbach, "Chemical Composition of Volcanic Gases," in *Monitoring and Mitigation of Volcano Hazards*, R. Scarpa and R. I. Tilling, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 221–256.
- [10] V. Ravindra and S. M. Rajbhoj, "Identification of Toxic Gases using Electronic Nose," in *2017 International Conference on*

- Computing, Communication, Control and Automation (ICCUBEA)*, PUNE, India, 2017, pp. 1–5.
- [11] L. K. P. Saputra and Y. Lukito, “Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266,” in *2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, Yogyakarta, Indonesia, 2017, pp. 126–130.
- [12] E. Barrigon, E. Garcia-Tabares, I. Garcia, I. Rey-Stolle, and C. Algora, “Triple-junction solar cells for ultra-high concentrator applications,” in *Proceedings of the 8th Spanish Conference on Electron Devices, CDE’2011*, Palma de Mallorca, Spain, 2011, pp. 1–4.
- [13] S. Kawakita *et al.*, “High efficiency and radiation resistant InGaP/GaAs//CIGS stacked solar cells for space applications,” in *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Portland, OR, USA, 2016, pp. 2574–2577.
- [14] A. Banerjee *et al.*, “Advances in cell efficiency of a-Si:H and nc-Si:H-based multi-junction solar cells for space and near-space applications,” in *2010 35th IEEE Photovoltaic Specialists Conference*, Honolulu, HI, USA, 2010, pp. 002651–002655.
- [15] “UM10204 I2C-bus specification and user manual,” vol. 2014, p. 64, 2014.
- [16] N. A. Z. M. Noar and M. M. Kamal, “The development of smart flood monitoring system using ultrasonic sensor with blynk applications,” in *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, 2017, pp. 1–6.

LAMPIRAN A

Program Arduino

1. Kode Program Pemacaan dan Pengiriman Data

```
#include <Wire.h>
#include <SPI.h>
#include <RH_RF95.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
/*****Define Sensor*****/
#define MQ136 (A0)
#define MQ135 (A1)
/*****Sample Times And Interval*****/
#define Calibration_Sample_Times (500)
#define Calibration_Sample_Interval (250)
#define Read_Sample_Interval (50)
#define Read_Sample_Times (50)
/*****Gas Sensor*****/
#define GAS_CL2 (0)
#define GAS_O3 (1)
#define GAS_CO2 (2)
#define GAS_CO (3)
#define GAS_NH4 (4)
#define GAS_CH3 (6)
#define GAS_CH3_2CO (7)
#define GAS_H2 (8)
#define GAS_C2H5OH (9)
#define GAS_C4H10 (10)
#define GAS_LPG (11)
#define GAS_Smoke (12)
#define GAS_CO_sec (13)
#define GAS_LPG_sec (14)
#define GAS_CH4 (15)
#define GAS_NO2 (16)
#define GAS_SO2 (17)
#define GAS_C7H8 (18)
#define GAS_H2S (19)
#define GAS_NH3 (20)
#define GAS_C6H6 (21)
#define GAS_C3H8 (22)
#define GAS_NHEX (23)
#define GAS_HCHO (24)
```

```

/*****Curva Gas MQ136*****/
float      SO2_Curve[2]  = {40.44109566, -1.085728557};
float      CH4_secCurve[2] = {57.82777729, -1.187494933};
float      CO_terCurve[2] = {2142.297846, -2.751369226};
float      H2S_Curve[2]  = {27.55, -2.5};
float      NH4_secCurve[2] = {285.662, -3.375};
/*****Curva Gas MQ135*****/
float      CO2_Curve[2]  = {113.7105289, -3.019713765};
float      CO_Curve[2]   = {726.7809737, -4.040111669};
float      NH4_Curve[2]  = {84.07117895, -4.41107687};
float      C2H5OH_Curve[2] = {74.77989144, 3.010328075};
float      CH3_Curve[2]  = {47.01770503, -3.281901967};
float      CH3_2CO_Curve[2] = {7.010800878, -2.122018939};
/*****Inisialisasi Calibration*****/
unsigned long SLEEP_TIME = 600;
//float      Ro1;
//float      Ro0;
float      Ro1 = 0.58;
float      Ro0 = 13.65;
float      RL0 = 2.497;
float      RL1 = 2.57;
/*****Deteksi Gas*****/
float      CO2;
float      H2S;
/*****

char data_1[10] = {"\0"};
char data_2[10] = {"\0"};
char data_3[10] = {"\0"};
uint8_t data[100];

// Singleton instance of the radio driver
float frequency = 915.0;
RH_RF95 rf95;
//#define Serial SerialUSB

void setup()
{
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  // Serial.print("Calibrating.....\n");

```

```

// Ro0 = MQCalibration(MQ135, 400, RL0, CO2_Curve);
// Ro1 = MQCalibration(MQ136, 0.126, RL1, H2S_Curve);
// Serial.print("Ro0 = "); Serial.print(Ro0); Serial.print(" KOhm");
// Serial.print(" Ro1="); Serial.print(Ro1); Serial.println(" kohm");
// Serial.println("\n");
// lcd.setCursor(1,0);
// lcd.print("Hello Hendrik");
while (!Serial) ; // Wait for serial port to be available
if (!rf95.init())
Serial.println("init failed");
rf95.setFrequency(frequency);
rf95.setTxPower(13);
}

void loop()
{
Serial.println("Sending to rf95_server");
data_sender();
lcdDisplay();
rf95.send(data, sizeof(data));

rf95.waitPacketSent();
// Now wait for a reply
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);

if (rf95.waitAvailableTimeout(3000))
{
// Should be a reply message for us now
if (rf95.recv(buf, &len))
{
Serial.print(" got reply: ");
Serial.println((char*)buf);
}
else
{
Serial.println("recv failed");
}
}
else
{

```

```

Serial.println("No reply, is rf95_server running?");
}
delay(400);
}

```

```

void lcdDisplay(){
lcd.setCursor(0, 0);
lcd.print("CO2:");
lcd.setCursor(4, 0);
lcd.print(CO2);
lcd.setCursor(0, 1);
lcd.print("H2S:");
lcd.setCursor(4, 1);
lcd.print(H2S);
lcd.setCursor(10, 1);
lcd.print(Suhu);
lcd.setCursor(14, 1);
lcd.print(" C");
}

```

```

void data_sender(){
CO2 = MQGetGasPercentage(MQRead(MQ135, RL0), Ro0,
GAS_CO2, MQ135);
H2S = MQGetGasPercentage(MQRead(MQ136, RL1), Ro1, GAS_H2S,
MQ136);
char data_sensor[100] = {"\0"};
dtostrf(CO2, 5, 2, data_1);
dtostrf(H2S, 5, 2, data_2);
strcat(data_sensor, data_1);
strcat(data_sensor, "\n");
strcat(data_sensor, data_2);
strcat(data_sensor, "\n");
strcat(data_sensor, data_3);
strcat(data_sensor, "\n");
strcpy((char*)data, data_sensor);
Serial.println((char*)data);
}

```

/******MQResistanceCalculation*****

Input: raw_adc - raw value read from adc, which represents the voltage

Output: the calculated sensor resistance*****

```

float MQResistanceCalculation(int raw_adc, float rl_value)
{
return
(long)((long)(1024* (long)rl_value) / raw_adc - (long)rl_value);
}

/***** MQCalibration *****/
Input: mq_pin - analog channel
Output: Ro of the sensor *****/
float MQCalibration(int mq_pin, double ppm, double rl_value, float
*pcurve )
{
int i;
float val = 0;

for (i = 0; i < Calibration_Sample_Times; i++) { //take multiple
samples
val += MQResistanceCalculation(analogRead(mq_pin), rl_value);
delay(Calibration_Sample_Interval);
}
val = val / Calibration_Sample_Times; //calculate the average
value

return //val;
(float)val * exp((log(pcurve[0] / ppm) / pcurve[1]));

}

/***** MQRead *****/
Input: mq_pin - analog channel
Output: Rs of the sensor
The Rs changes as the sensor is in the different concentration of the
target gas. The sample times and the time interval between samples
could be configured by changing the definition of the macros.
*****/
float MQRead(int mq_pin, float rl_value)
{
int i;
float rs = 0;

for (i = 0; i < Read_Sample_Times; i++) {

```

```

rs += MQResistanceCalculation(analogRead(mq_pin), rl_value);
delay(Read_Sample_Interval);}
rs = rs / Read_Sample_Times;
return rs;}

```

/****** MQGetGasPercentage *****

Input: rs_ro_ratio - Rs divided by Ro

gas_id - target gas type

Output: ppm of the target gas

Remarks: This function passes different curves to the MQGetPercentage function which calculates the ppm (parts per million) of the target gas.

```

float MQGetGasPercentage(float rs_ro_ratio, float ro, int gas_id, int
sensor_id)

```

```

{
if (sensor_id == MQ136) {
if ( gas_id == GAS_SO2 ) {
return MQGetPercentage(rs_ro_ratio, ro,SO2_Curve); //MQ136
} else if ( gas_id == GAS_NH4 ) {
return MQGetPercentage(rs_ro_ratio, ro,NH4_secCurve); //MQ136
} else if ( gas_id == GAS_CO_sec ) {
return MQGetPercentage(rs_ro_ratio, ro,CO_terCurve); //MQ136
} else if ( gas_id == GAS_H2S ) {
return MQGetPercentage(rs_ro_ratio, ro,H2S_Curve); //TGS2602
}

}

else if (sensor_id == MQ135) {
if ( gas_id == GAS_CO2 ) {
return MQGetPercentage(rs_ro_ratio, ro, CO2_Curve); //MQ135
} else if ( gas_id == GAS_C2H5OH ) {
return MQGetPercentage(rs_ro_ratio, ro, C2H5OH_Curve); //MQ135
} else if ( gas_id == GAS_CH3 ) {
return MQGetPercentage(rs_ro_ratio, ro,CH3_Curve); //MQ135
} else if ( gas_id == GAS_CH3_2CO ) {
return MQGetPercentage(rs_ro_ratio, ro,CH3_2CO_Curve); //MQ135
} else if ( gas_id == GAS_CO ) {
return MQGetPercentage(rs_ro_ratio, ro,CO_Curve); //MQ135
}
}
return 0;
}

```

```

/***** MQGetPercentage *****/
Input: rs_ro_ratio - Rs divided by Ro
pcurve - pointer to the curve of the target gas
Output: ppm of the target gas
of the line could be derived if y(rs_ro_ratio) is provided. As it is a
logarithmic coordinate, power of 10 is used to convert the result to non-
logarithmic value.
*****/
float MQGetPercentage(float rs_ro_ratio, float ro, float *pcurve)
{
return (float)(pcurve[0] * pow(((float)rs_ro_ratio / ro),pcurve[1]));
}

```

2. Kode Program Penerimaan dan Pengolahan Data

```

#include <SPI.h>
#include <RH_RF95.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial kirimSerial(7,6);
/*****Inisial Pengiriman Serial*****/
char start = '<';
char stopped = '>';
/*****Inisial Pemisah Data*****/
char nilai[5][10];
char line = '\n';
const byte numChars = 50;
char dataTerima[RH_RF95_MAX_MESSAGE_LEN];
/*****Inisial RH_RF95*****/
float frequency = 915.0;
RH_RF95 rf95;
/*****

void setup()
{
Serial.begin(9600);
kirimSerial.begin(9600);
lcd.init();
lcd.backlight();

```

```

while (!Serial) ; // Wait for serial port to be available
if (!rf95.init())
  Serial.println("init failed");
  rf95.setFrequency(frequency);
  rf95.setTxPower(13);
}

void loop()
{
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf95.waitAvailableTimeout(10000))
  {
    if (rf95.recv(buf, &len))
    {
      String dataSensor = (char*)buf;
      dataSensor.toCharArray(dataTerima,sizeof(dataTerima));
      Serial.println(dataTerima);
      kirimSerial.write(start);
      kirimSerial.write(dataTerima,sizeof(dataTerima));
      kirimSerial.write(stopped);
      dataPerSensor();
      Serial.println("got request: ");
      lcdDisplay();

      // Send a reply
      uint8_t data[] = "I'm Back";
      rf95.send(data, sizeof(data));
      rf95.waitPacketSent();
      Serial.println("Sent a reply");
    }
    else
    {
      Serial.println("recv failed");
    }
  }
  else
  {
    Serial.println("No reply, is rf95_server running?");
    lcd.clear();
    lcd.setCursor(1, 0);
  }
}

```



```

        lcd.print("failed");
    }
    delay(400);
}

void dataPerSensor(){
    int data_ke = 0;
    int index = 0;
    int i;
    for(i = 0; i <= numChars ; i++){
        if(dataTerima[i] != line){
            nilai[data_ke][index] = dataTerima[i];
            index++;
        }
        else
        {
            data_ke++;
            index = 0;
        }
    }
}

void lcdDisplay(){
    lcd.setCursor(0, 0);
    lcd.print("CO2:");
    lcd.setCursor(4, 0);
    lcd.print(nilai[0]);
    lcd.setCursor(0, 1);
    lcd.print("H2S:");
    lcd.setCursor(4, 1);
    lcd.print(nilai[1]);
    lcd.setCursor(10, 1);
    lcd.print(nilai[2]);
    lcd.setCursor(14, 1);
    lcd.print(" C");
}

```

3. Kode Program Pengolahan dan Pengiriman Data Sensor ke Aplikasi

```

#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

```

```

#define BLYNK_PRINT Serial
SoftwareSerial s(D7,D8);

char auth[] = "2c81a2057f6e49419ae7e959ada46266";
char ssid[] = "Hendrik";
char pass[] = "hendrik13";

char buff[3][10];
char line = '\n';
const byte numChars = 32;
char receivedChars[numChars];
boolean newData = false;

void setup() {
  Serial.begin(9600);
  s.begin(9600);
  Blynk.begin(auth, ssid, pass);
  Serial.println("<Arduino is ready>");
}

void loop() {
  recvWithStartEndMarkers();
  showNewData();
  dataSet();
  dataBlynk();
  Blynk.run();
  Serial.println("DATA");
  Serial.println(buff[0]);
  Serial.println(buff[1]);
}

void recvWithStartEndMarkers() {
  static boolean recvInProgress = false;
  static byte ndx = 0;
  char startMarker = '<';
  char endMarker = '>';
  char rc;

  while (s.available() > 0 && newData == false) {
    rc = s.read();

```

```

if (recvInProgress == true) {
    if (rc != endMarker) {
        receivedChars[ndx] = rc;
        ndx++;
        if (ndx >= numChars) {
            ndx = numChars - 1;
        }
    }
    else {
        receivedChars[ndx] = '\0'; // terminate the string
        recvInProgress = false;
        ndx = 0;
        newData = true;
    }
}

else if (rc == startMarker) {
    recvInProgress = true;
}
}

void showNewData() {
    if (newData == true) {
        newData = false;
    }
}

void dataSet(){
    int data_ke = 0;
    int index = 0;
    int i;
    for(i = 0; i <= numChars ; i++){
        if(receivedChars[i] != line){
            buff[data_ke][index] = receivedChars[i];
            index++;
        }
        else
        {
            data_ke++;
        }
    }
}

```

```
    index = 0;
  }
}
}

void dataBlynk(){
  dataSet();
  Blynk.virtualWrite(V1, buff[0]);
  Blynk.virtualWrite(V2, buff[1]);
}
```

LAMPIRAN B

Data Percobaan

1. Pengambilan Pertama pada Jarak 3 Km

Pengambilan Pertama		
Waktu (S)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
10	303,96	0,17
20	303,96	0,17
30	303,96	0,17
40	303,96	0,17
50	276,34	0,16
60	276,34	0,16
70	306,85	0,17
80	332,94	0,16
90	338,27	0,17
100	338,27	0,17
110	338,27	0,17
120	338,27	0,17
130	338,27	0,17
140	338,27	0,17
150	338,27	0,17
160	338,27	0,17
170	338,27	0,17
180	338,27	0,17
190	346,25	0,17
200	346,25	0,17
210	346,25	0,17
220	346,25	0,17
230	345,82	0,17
240	326,94	0,17
250	326,94	0,17

Pengambilan Pertama		
Waktu (S)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
260	330,12	0,17
270	330,12	0,17
280	330,12	0,17
290	330,12	0,17
300	330,12	0,17
310	330,12	0,17
320	330,12	0,17
330	330,12	0,17
340	330,12	0,17
350	330,12	0,17
360	325,76	0,17
370	325,76	0,17
380	325,76	0,17
390	325,76	0,17
400	325,76	0,17
410	325,76	0,17
420	325,76	0,17
430	325,76	0,17
440	325,76	0,17
450	325,76	0,17
460	325,76	0,17
470	325,76	0,17
480	325,76	0,17
490	325,76	0,17
500	325,76	0,17
510	325,76	0,17
520	324,19	0,17

Pengambilan Pertama		
Waktu (S)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
530	324,19	0,17
540	325,76	0,17
550	338,27	0,19
560	325,76	0,19
570	325,76	0,19
580	325,76	0,19
590	325,76	0,19
600	325,76	0,19
610	325,76	0,19
620	325,76	0,19
630	325,76	0,19
640	325,76	0,19
650	306,12	0,17
660	273,2	0,17
670	273,2	0,17
680	273,2	0,17
690	273,2	0,17
700	273,2	0,17

2. Pengambilan Kedua pada Jarak 2,5 Km

Pengambilan Kedua		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
10	258,2	0,16
20	258,2	0,16
30	258,2	0,16
40	258,2	0,16
50	258,2	0,16

Pengambilan Kedua		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
60	258,2	0,16
70	258,49	0,16
80	258,2	0,15
90	276,34	0,13
100	324,97	0,13
110	324,97	0,13
120	392,68	0,19
130	392,68	0,19
140	392,68	0,19
150	392,68	0,19
160	392,68	0,19
170	392,68	0,19
180	273,2	0,17
190	288,03	0,19
200	288,03	0,19
210	288,03	0,19
220	288,03	0,19
230	288,03	0,19
240	288,03	0,19
250	288,03	0,19
260	288,03	0,19
270	392,68	0,31
280	392,68	0,31
290	433,66	0,31
300	433,66	0,31
310	433,66	0,31
320	358,48	0,28

Pengambilan Kedua		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
330	331,33	0,24
340	325,76	0,24
350	325,76	0,24
360	314,98	0,22
370	314,98	0,22
380	314,98	0,22
390	314,98	0,22
400	314,98	0,22
410	345,4	0,2
420	345,4	0,2
430	345,4	0,2
440	345,4	0,2
450	345,4	0,2
460	339,93	0,19
470	339,93	0,19
480	339,93	0,19
490	339,93	0,19
500	345,4	0,22
510	345,4	0,22
520	345,4	0,22
530	325,76	0,2
540	325,76	0,2
550	325,76	0,2
560	325,76	0,2
570	325,76	0,2
580	325,76	0,2
590	325,76	0,2

Pengambilan Kedua		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
600	325,76	0,2
610	325,76	0,2
620	258,2	0,17
630	258,2	0,17
640	256,48	0,18
650	269,5	0,19
660	263,77	0,19
670	263,77	0,19
680	263,77	0,19
690	267,97	0,17
700	267,97	0,17
710	267,97	0,17
720	264,66	0,17
730	263,47	0,17
740	263,47	0,17
750	263,47	0,17
760	263,47	0,17
770	263,47	0,17
780	263,47	0,17
790	263,47	0,17
800	263,47	0,17
810	231,35	0,16
820	231,35	0,16
830	231,35	0,16
840	231,35	0,16
850	231,35	0,16
860	231,35	0,16

Pengambilan Kedua		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
870	231,35	0,16
880	231,35	0,16
890	231,35	0,16
900	268,28	0,16
910	268,28	0,16
920	268,28	0,16
930	268,28	0,16
940	230,85	0,15
950	246,99	0,16
960	246,99	0,16
970	246,99	0,16
980	255,34	0,15
990	258,2	0,16
1000	258,2	0,14

3. Pengambilan Ketiga pada Jarak 1,5 Km

Pengambilan Ketiga		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
10	258,2	0,16
20	258,2	0,16
30	258,2	0,16
40	258,2	0,16
50	258,2	0,16
60	258,2	0,16
70	257,63	0,15
80	257,63	0,15

Pengambilan Ketiga		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
90	247,53	0,14
100	258,2	0,15
110	250,28	0,14
120	254,77	0,14
130	254,77	0,14
140	254,77	0,14
150	254,77	0,14
160	254,77	0,14
170	231,35	0,16
180	234,37	0,14
190	234,37	0,14
200	234,37	0,14
210	234,37	0,14
220	234,37	0,14
230	234,37	0,14
240	234,37	0,14
250	234,37	0,14
260	234,37	0,14
270	234,37	0,14
280	234,37	0,14
290	230,85	0,14
300	230,85	0,14
310	231,35	0,14
320	230,85	0,14
330	231,35	0,14
340	231,35	0,14
350	231,35	0,14

Pengambilan Ketiga		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
360	231,35	0,14
370	231,35	0,14
380	231,35	0,14
390	231,35	0,14
400	231,35	0,14
410	231,35	0,14
420	231,35	0,14
430	231,35	0,14
440	231,35	0,14
450	221,65	0,14
460		
470		
480		
490		
500		
510		
520		
530		
540		
550		
560		
570		
580		
590	227,4	0,14
600	227,4	0,14
610	227,4	0,14
620	253,08	0,14

Pengambilan Ketiga		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
630	254,77	0,14
640	254,77	0,14
650	254,77	0,14
660	254,77	0,14
670	219,78	0,14
680	231,85	0,14
690	231,85	0,14
700	231,85	0,14
710	231,85	0,14
720	231,85	0,14
730	231,85	0,14
740	231,85	0,14
750	231,85	0,14
760	231,85	0,14
770	208,09	0,14
780	208,09	0,14
790	208,09	0,14
800	208,09	0,14
810	208,09	0,14
820	187,46	0,13
830	181,19	0,12
840	181,19	0,12
850	187,46	0,12
860	187,09	0,12
870	187,09	0,12

4. Pengambilan Keempat pada Jarak 500 m

Pengambilan Keempat		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
10	186,34	0,12
20	186,34	0,12
30	186,34	0,12
40	186,34	0,12
50	186,34	0,12
60	186,34	0,12
70	155,46	0,11
80	159,34	0,11
90	159,34	0,11
100	156,94	0,11
110	156,94	0,11
120	156,94	0,11
130	156,94	0,11
140	156,94	0,11
150		
160	156,34	0,11
170	169,47	0,11
180	154,58	0,11
190	154,58	0,11
200	151,41	0,11
210	154,58	0,1
220	154	0,1
230	154	0,1
240	154	0,1
250	154	0,1
260	154	0,1
270	151,41	0,1

Pengambilan Keempat		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
280	151,41	0,1
290	151,41	0,1
300	154,58	0,1
310	154,58	0,1
320	154,29	0,1
330	149,72	0,09
340	155,46	0,1
350	155,46	0,1
360	155,46	0,1
370	155,46	0,1
380	155,46	0,1
390	155,46	0,1
400	155,46	0,1
410	155,46	0,1
420	154,58	0,1
430	154,29	0,1
440	154,29	0,1
450	154,29	0,1
460	154	0,1
470	154	0,1
480	154	0,1
490	154	0,1
500	154	0,1
510	154	0,1
520	154	0,1
530	162,73	0,1
540	162,73	0,1

Pengambilan Keempat		
Waktu (Detik)	Sensor	
	MQ135 (PPM)	MQ136 (PPM)
550	162,73	0,1
560	162,73	0,1
570	162,73	0,1
580	162,73	0,1
590	169,8	0,1
600	170,13	0,1
610	170,13	0,1
620	186,71	0,11
630	174,17	0,1

.....*Halaman ini sengaja dikosongkan*.....

LAMPIRAN C

Dokumentasi Data Percobaan

1. Pengujian sensor gas per volume MQ135



(a) Konsentrasi 1 mL



(e) Konsentrasi 5 mL



(b) Konsentrasi 2 mL



(f) Konsentrasi 6 mL



(c) Konsentrasi 3 mL



(g) Konsentrasi 7 mL



(d) Konsentrasi 4 mL



(h) Konsentrasi 8 mL



(i) Konsentrasi 9 mL



(j) Konsentrasi 10 mL

2. Pengujian sensor gas per volume MQ136



(a) Konsentrasi 1 mL



(d) Konsentrasi 4 mL



(b) Konsentrasi 2 mL



(e) Konsentrasi 5 mL



(c) Konsentrasi 3 mL



(f) Konsentrasi 6 mL



(g) Konsentrasi 7 mL



(i) Konsentrasi 9 mL



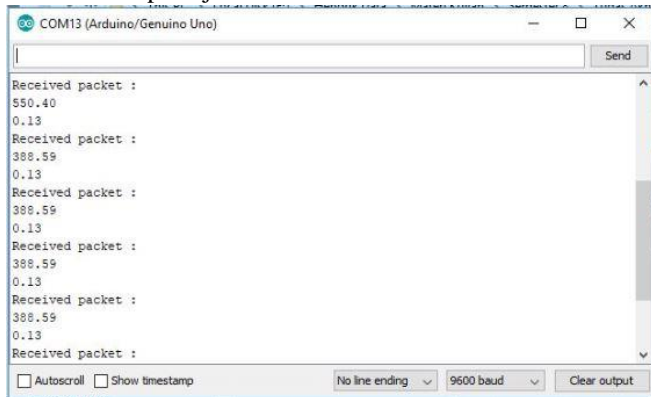
(h) Konsentrasi 8 mL



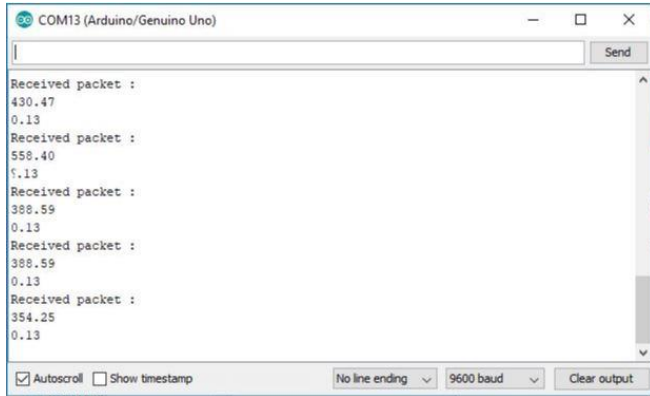
(j) Konsentrasi 10 mL

2. Pengujian LoRa dengan halangan gedung

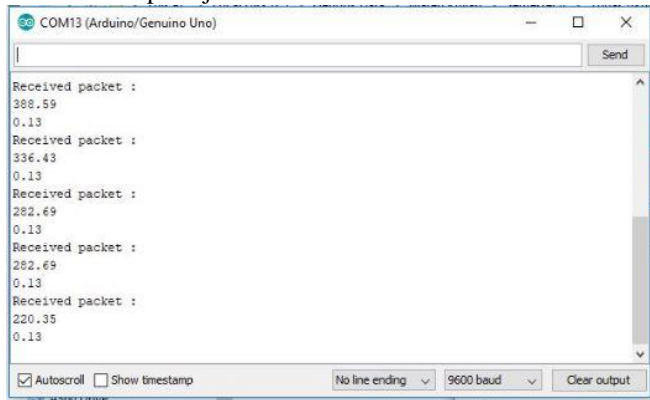
b. Data terbaca pada jarak 50 m



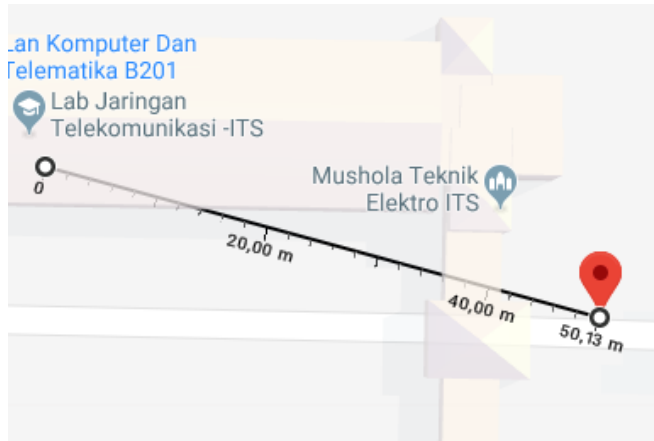
c. Data terbaca pada jarak 100 m



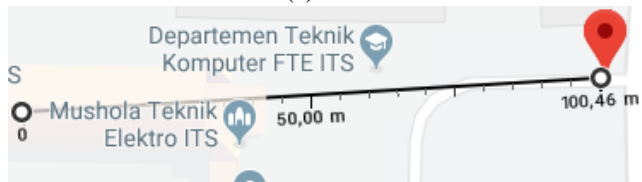
d. Data terbaca pada jarak 200 m



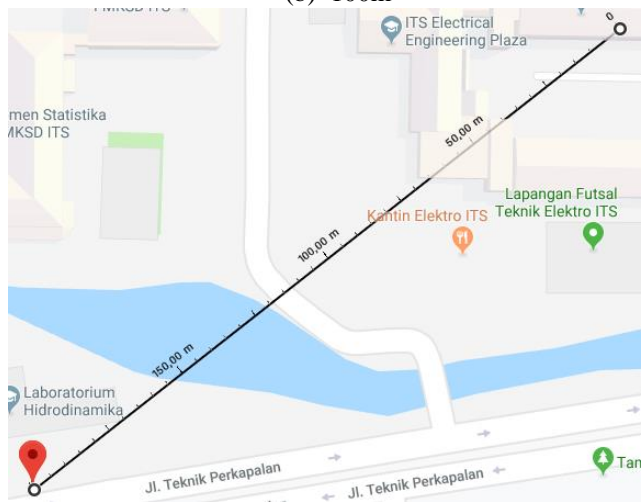
e. Pengukuran jarak pada aplikasi google map



(a) 50 m

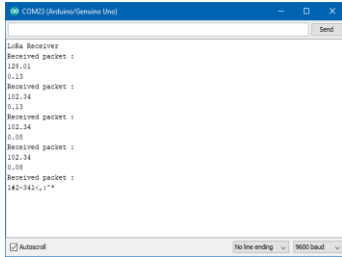


(b) 100m

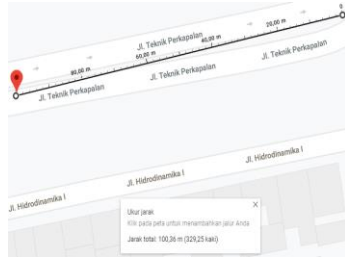


(c) 200m

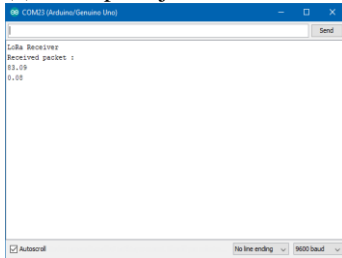
3. Pengujian LoRa pada jalan lurus



(a) Hasil pada jarak 100 m



(b) Jarak 100 m pada map



(c) Hasil pada jarak 200 m



(d) Jarak 200 m pada map

LAMPIRAN D

Dokumentasi Kegiatan

1. Pengujian keseluruhan



.....*Halaman ini sengaja dikosongkan*.....

LAMPIRAN E

Datasheet

1. Datasheet sensor MQ136

Statement

This manual copyright belongs to Zhengzhou Winsen Electronics Technology Co., LTD. Without the written permission, any part of this manual shall not be copied, translated, stored in database or retrieval system, also can't spread through electronic, copying, record ways.

Thanks for purchasing our product. In order to let customers use it better and reduce the faults caused by misuse, please read the manual carefully and operate it correctly in accordance with the instructions. If users disobey the terms or remove, disassemble, change the components inside of the sensor, we shall not be responsible for the loss.

The specific such as color, appearance, sizes &etc, please in kind prevail.

We are devoting ourselves to products development and technical innovation, so we reserve the right to improve the products without notice. Please confirm it is the valid version before using this manual. At the same time, users' comments on optimized using way are welcome.

Please keep the manual properly, in order to get help if you have questions during the usage in the future.

Zhengzhou Winsen Electronics Technology CO., LTD

MQ136 Gas Sensor for Hydrogen Sulfide

Profile

Sensitive material of MQ136 gas sensor is SnO₂ which with lower conductivity in clean air. When H₂S gas exists, the sensor's conductivity gets higher along with the gas concentration rising. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit. MQ136 gas sensor has high sensitivity to H₂S gas, also can monitor organic vapour including sulfur well. It a kind of low-cost sensor for kinds of applications.



Features

It has good sensitivity to H₂S gas in wide range, and has advantages such as long lifespan, low cost and simple drive circuit &etc.

Main Applications

It is widely used in domestic H₂S gas alarm, industrial H₂S gas leakage alarm and portable H₂S gas detector.

Technical Parameters **Stable.1**

Model		MQ136
Sensor Type		Semiconductor
Standard Encapsulation		Bakelite, Metal cap
Target Gas		Hydrogen Sulfide(H ₂ S gas)
Detection range		1~200ppm
Standard Circuit Conditions	Loop Voltage	V _L ≤24V DC
	Heater Voltage	V _H 5.0V±0.1V AC or DC
	Load Resistance	R _L Adjustable
Sensor character under standard test conditions	Heater Resistance	R _H 29Ω±3Ω (room tem.)
	Heater consumption	P _H ≤900mW
	Sensitivity	S R _S (in air)/R _S (50ppm H ₂ S)≥3
	Output Voltage	ΔVs ≥0.5V(in 50ppm H ₂ S)
	Concentration Slope	α ≤0.6(R _{200ppm} /R _{50ppm} H ₂ S)
Standard test conditions	Tem. Humidity	20°C±2°C : 55%±5%RH
	Standard test circuit	V _C : 5.0V±0.1V; V _H : 5.0V±0.1V
	Preheat time	Over 48 hours

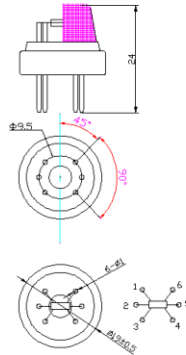


Fig1. Sensor Structure
 Unit: mm

NOTE: The change of Output voltage(ΔVs) is the difference value between V_{Rs} in test environment and

Basic Circuit

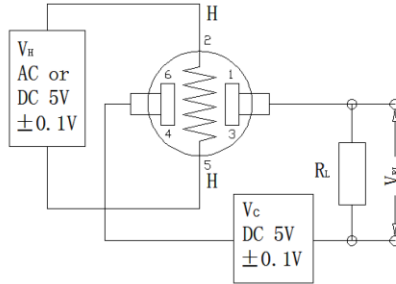


Fig2. MQ136 Test Circuit

Instructions: The above fig is the basic test circuit of MQ136. The sensor requires two voltage inputs: heater voltage (V_H) and circuit voltage (V_C). V_H is used to supply standard working temperature to the sensor and it can adopt DC or AC power, while V_{RL} is the voltage of load resistance R_L which is in series with sensor. V_C supplies the detect voltage to load resistance R_L and it should adopt DC power.

Description of Sensor Characters

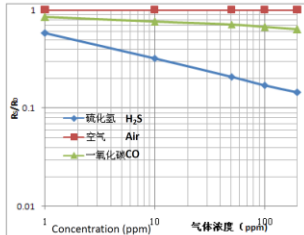


Fig3. Typical Sensitivity Curve

The ordinate is resistance ratio of the sensor (R_s/R_0), the abscissa is concentration of gases. R_s means resistance in target gas with different concentration, R_0 means resistance of sensor in clean air. All tests are finished under standard test conditions.

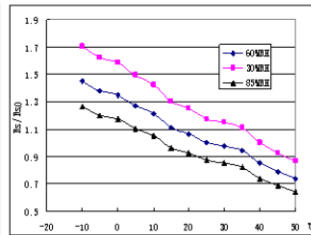


Fig4. Typical temperature/humidity characteristics

The ordinate is resistance ratio of the sensor (R_s/R_{50}). R_s means resistance of sensor in 50ppm H2S gas under different tem. and humidity. R_{50} means resistance of the sensor in 50ppm H2S gas under 20°C/55%RH.

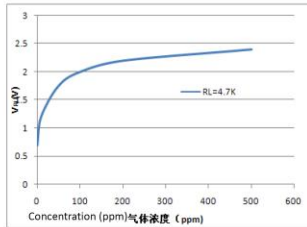


Fig5.Sensitivity Curve

Fig5 shows the V_{cc} in H_2S with different concentration. The resistance load R_L is 4.7 K Ω and the test is finished in standard test conditions.

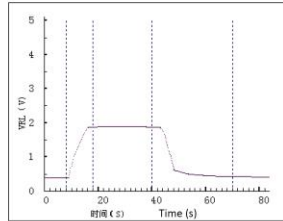


Fig6.Responce and Resume

Fig5 shows the changing of V_{cc} in the process of putting the sensor into target gas and removing it out.

Cautions

1 .Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Sensing material will lose sensitivity and never recover if the sensor absorbs organic silicon steam. Sensors must avoid exposing to silicon bond, fixature, silicon latex, putty or plastic contain silicon environment.

1.2 High Corrosive gas

If the sensors are exposed to high concentration corrosive gas (such as H_2S , SO_2 , Cl_2 , HCl etc.), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorine.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

Do avoid icing on sensor's surface, otherwise sensing material will be broken and lost sensitivity.

1.6 Applied higher voltage

Applied voltage on sensor should not be higher than stipulated value, even if the sensor is not physically damaged or broken, it causes down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, Pin 2&5 is heating electrodes, Pin (1,3)/(4,6) are testing electrodes (Pin 1 connects with Pin 3, while Pin 4 connects with Pin 6).if apply voltage on Pin 1&3 or 4&6, it will make lead broken; and no signal putout if apply on pins 2&4.

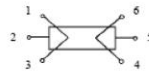


Fig7. Lead sketch

2. Following conditions must be avoided**2.1 Water Condensation**

Indoor conditions, slight water condensation will influence sensors' performance lightly. However, if water condensation on sensors surface and keep a certain period, sensors' sensitive will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if it is placed in high gas concentration for long time, sensors characteristic will be affected. If lighter gas sprays the sensor, it will cause extremely damage.

2.3 Long time storage

The sensors resistance will drift reversibly if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof bag without volatile silicon compound. For the sensors with long time storage but no electrify, they need long galvanical aging time for stability before using. The suggested aging time as follow:

Stable2.

Storage Time	Suggested aging time
Less than one month	No less than 48 hours
1 ~ 6 months	No less than 72 hours
More than six months	No less than 168 hours

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc., it will influence the sensors' performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then break. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage Conditions

2.7.1 For sensor, handmade welding is optimal way. The welding conditions as follow:

- Soldering flux: Rosin soldering flux contains least chlorine
- homothermal soldering iron
- Temperature: 250℃
- Time: less than 3 seconds

2.7.1 If users choose wave-soldering, the following conditions should be obey:

- Soldering flux: Rosin soldering flux contains least chlorine
- Speed: 1-2 Meter/ Minute
- Warm-up temperature: 100±20℃
- Welding temperature: 250±10℃
- One time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will reduce.

Zhengzhou Winsen Electronics Technology Co., Ltd
Add: No.299, Jinsuo Road, National Hi-Tech Zone,
 Zhengzhou 450001 China
Tel: +86-371-67169097/67169670
Fax: +86-371-60932988
E-mail: sales@winsensor.com
Website: www.winsentech.com

2. Datasheet sensor MQ135

TECHNICAL DATA**MQ-135 GAS SENSOR****FEATURES**

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃,NO_x, alcohol, Benzene, smoke,CO₂,etc.

SPECIFICATIONS**A. Standard work condition**

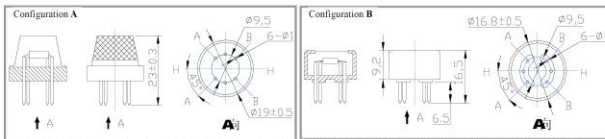
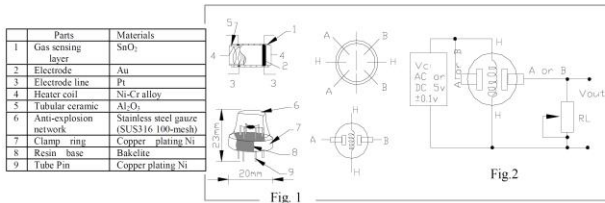
Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _{HT}	Heating voltage	5V±0.1	ACOR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃~45℃	
Tas	Storage Tem	-20℃~70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	30K Ω~200K Ω (100ppm NH ₃)	Detecting concentration scope: 10ppm~300ppm NH ₃ 10ppm~1000ppm Benzene 10ppm~300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	V _c :5V±0.1 V _H :5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

TEL: 86-371-67169070 67169080

FAX: 86-371-67169090

E-mail: sales@hwsensor.com

sensitive components. The enveloped MQ-135 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2
 E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

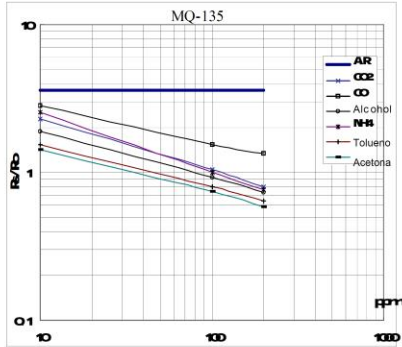


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. in their: Temp: 20°C, Humidity: 65%, O₂ concentration 21%, RL=20kΩ, R₀: sensor resistance at 100ppm of NH₃ in the clean air, R_s: sensor resistance at various concentrations of gases.

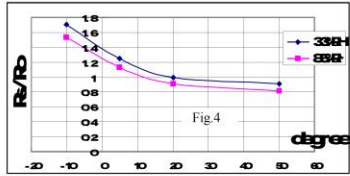
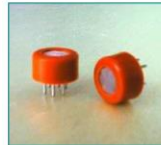


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity. R₀: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree. R_s: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use value of Load resistance (R_L) about 20 KΩ (10KΩ to 47 KΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.



Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2S , SO_x , Cl_2 , HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

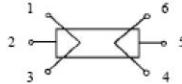
Do avoid icing on sensor's surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1, 3 pins or 4, 6 pins, it will make lead broken, and without signal when apply on 2, 4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, it will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stability before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature: $100 \pm 20^\circ C$

2.7.4 Welding temperature: $250 \pm 10^\circ C$

2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.

1.2. Product Versions

The features of the three product variants are detailed in the following table.

Table 48 RF96/97/98 Device Variants and Key Parameters

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
RF96	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
RF97	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
RF98	137 - 525 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

1.3. Pin Diagram

The following diagram shows the pin arrangement of the QFN package, top view.

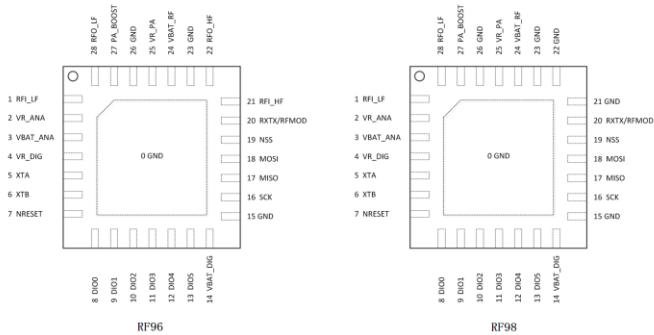


Figure 2. Pin Diagrams

1.4. Pin Description

Number	Name	Type	Description
0	GROUND	-	Exposed ground pad.
1	RFL_LF	I	RF input for lower bands
2	VR_ANA	-	Regulated supply voltage for analogue circuitry
3	VBAT_ANA	-	Supply voltage for analogue circuitry
4	VR_DIG	-	Regulated supply voltage for digital blocks
5	XTA	I/O	XTAL connection or TCXO input
6	XTB	I/O	XTAL connection.
7	NRESET	I/O	Reset trigger input.
8	DIO0	I/O	Digital I/O, software configured.
9	DIO1/DCLK	I/O	Digital I/O, software configured.
10	DIO2/DATA	I/O	Digital I/O, software configured.
11	DIO3	I/O	Digital I/O, software configured.
12	DIO4	I/O	Digital I/O, software configured.
13	DIO5	I/O	Digital I/O, software configured.
14	VBAT_DIG	-	Supply voltage for digital blocks
15	GND	-	Ground
16	SCK	I	SPI Clock input
17	MISO	O	SPI Data output
18	MOSI	I	SPI Data input
19	NSS	I	SPI Chip select input
20	RXTX/RF_MOD	O	Rx/Tx switch control: high in Tx
21	RFL_HF	I	RF input for upper bands
22	RFO_HF	O	RF output for upper bands
23	GND	-	Ground
24	VBAT_RF	-	Supply voltage for RF blocks
25	VR_PA	-	Regulated supply for the PA
26	GND	-	Ground
27	PA_BOOST	O	Optional high-power PA output, lower or upper bands
28	RFO_LF	O	RF output for lower bands

2. Electrical Characteristics

2.1. ESD Notice

The RF96/97/98 is a high performance radio frequency device. It satisfies:

- Class 2 of the JEDEC standard JESD22-A114-B (Human Body Model) on all pins.
- Class III of the JEDEC standard JESD22-C101C (Charged Device Model) on all pins



It should thus be handled with all the necessary ESD precautions to avoid any permanent damage.

2.2. Absolute Maximum Ratings

Stresses above the values listed below may cause permanent device failure. Exposure to absolute maximum ratings for extended periods may affect device reliability.

Table 49 Absolute Maximum Ratings

Symbol	Description	Min	Max	Unit
VDDmr	Supply Voltage	-0.5	3.9	V
Tmr	Temperature	-55	+115	°C
Tj	Junction temperature	-	+125	°C
Pmr	RF Input Level	-	+10	dBm

Note Specific ratings apply to +20 dBm operation (see Section 5.4.3).

2.3. Operating Range

Table 50 Operating Range

Symbol	Description	Min	Max	Unit
VDDop	Supply voltage	1.8	3.7	V
Top	Operational temperature range	-40	+85	°C
Clop	Load capacitance on digital ports	-	25	pF
ML	RF Input Level	-	+10	dBm

Note A specific supply voltage range applies to +20 dBm operation (see Section 5.4.3).

2.4. Chip Specification

The tables below give the electrical specifications of the transceiver under the following conditions: Supply voltage VDD=3.3 V, temperature = 25 °C, FXOSC = 32 MHz, F_{RF} = 169/434/868/915 MHz (see specific indication), Pout = +13dBm, 2-level FSK modulation without pre-filtering, FDA = 5 kHz, Bit Rate = 4.8 kb/s and terminated in a matched 50 Ohm impedance, shared Rx and Tx path matching., unless otherwise specified.

2.4.1. Power Consumption

Table 51 Power Consumption Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDSL	Supply current in Sleep mode		-	0.2	1	uA
IDDIDLE	Supply current in Idle mode	RC oscillator enabled	-	1.5	-	uA
IDDST	Supply current in Standby mode	Crystal oscillator enabled	-	1.6	1.8	mA
IDDFS	Supply current in Synthesizer mode	FSRx	-	5.8	-	mA
IDDR	Supply current in Receive mode	LnaBoost Off, higher bands	-	10.8	-	mA
		LnaBoost On, higher bands	-	11.5	-	mA
		Lower bands	-	12.1	-	mA
IDDT	Supply current in Transmit mode with impedance matching	RFOP = +20 dBm, on PA_BOOST	-	120	-	mA
		RFOP = +17 dBm, on PA_BOOST	-	87	-	mA
		RFOP = +13 dBm, on RFO_LF/HF pin	-	29	-	mA
		RFOP = +7 dBm, on RFO_LF/HF pin	-	20	-	mA

2.4.2. Frequency Synthesis

Table 52 Frequency Synthesizer Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
FR	Synthesizer frequency range	Programmable	137	-	175	MHz
			410	-	525	
			862	-	1020	
FXOSC	Crystal oscillator frequency		-	32	-	MHz
TS_osc	Crystal oscillator wake-up time		-	250	-	us
TS_FS	Frequency synthesizer wake-up time to PllLock signal	From Standby mode	-	60	-	us
TS_HOP	Frequency synthesizer hop time at most 10 kHz away from the target frequency	200 kHz step	-	20	-	us
		1 MHz step	-	20	-	us
		5 MHz step	-	50	-	us
		7 MHz step	-	50	-	us
		12 MHz step	-	50	-	us
		20 MHz step	-	50	-	us
FSTEP	Frequency synthesizer step	FSTEP = FXOSC/2 ¹⁹	-	61.0	-	Hz

FRC	RC Oscillator frequency	After calibration	-	62.5	-	kHz
BRF	Bit rate, FSK	Programmable values (1)	1.2	-	300	kbps
BRO	Bit rate, OOK	Programmable	1.2	-	32.768	kbps
BRA	Bit Rate Accuracy	ABS(wanted BR - available BR)	-	-	250	ppm
FDA	Frequency deviation, FSK (1)	Programmable FDA + BRF/2 =< 250 kHz	0.6	-	200	kHz

Note For Maximum Bit rate the maximum modulation index is 0.5.

2.4.3. FSK/OOK Mode Receiver

All receiver tests are performed with RxBw = 10 kHz (Single Side Bandwidth) as programmed in *RegRxBw*, receiving a PN15 sequence. Sensitivities are reported for a 0.1% BER (with Bit Synchronizer enabled), unless otherwise specified. Blocking tests are performed with an unmodulated interferer. The wanted signal power for the Blocking Immunity, ACR, IIP2, IIP3 and AMR tests is set 3 dB above the receiver sensitivity level.

Table 53 FSK/OOK Receiver Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
RFS_F_LF	Direct tie of RFI and RFO pins, shared Rx, Tx paths FSK sensitivity, highest LNA gain. Lower frequency bands	FDA = 5 kHz, BR = 1.2 kb/s FDA = 5 kHz, BR = 4.8 kb/s FDA = 40 kHz, BR = 38.4 kb/s* FDA = 20 kHz, BR = 38.4 kb/s** FDA = 62.5 kHz, BR = 250 kb/s***	-	-121 -117 -107 -108 -95	-	dBm dBm dBm dBm dBm
	Split RF paths, the RF switch insertion loss is not accounted for. Lower frequency bands	FDA = 5 kHz, BR = 1.2 kb/s FDA = 5 kHz, BR = 4.8 kb/s FDA = 40 kHz, BR = 38.4 kb/s* FDA = 20 kHz, BR = 38.4 kb/s** FDA = 62.5 kHz, BR = 250 kb/s***	-	-123 -119 -109 -110 -97	-	dBm dBm dBm dBm dBm
RFS_F_HF	Direct tie of RFI and RFO pins, shared Rx, Tx paths FSK sensitivity, highest LNA gain. Higher frequency bands	FDA = 5 kHz, BR = 1.2 kb/s FDA = 5 kHz, BR = 4.8 kb/s FDA = 40 kHz, BR = 38.4 kb/s* FDA = 20 kHz, BR = 38.4 kb/s** FDA = 62.5 kHz, BR = 250 kb/s***	-	-119 -115 -105 -105 -92	-	dBm dBm dBm dBm dBm
	Split RF paths, LnaBoost is turned on, the RF switch insertion loss is not accounted for. Higher frequency bands	FDA = 5 kHz, BR = 1.2 kb/s FDA = 5 kHz, BR = 4.8 kb/s FDA = 40 kHz, BR = 38.4 kb/s* FDA = 20 kHz, BR = 38.4 kb/s** FDA = 62.5 kHz, BR = 250 kb/s***	-	-123 -119 -109 -109 -96	-	dBm dBm dBm dBm dBm
RFS_O	OOK sensitivity, highest LNA gain shared Rx, Tx paths	BR = 4.8 kb/s BR = 32 kb/s	-	-117 -108	-	dBm dBm
CCR	Co-Channel Rejection		-	-9	-	dB

ACR	Adjacent Channel Rejection	FDA = 5 kHz, BR=4.8kb/s Offset = +/- 25 kHz or +/- 50kHz 169MHz Band 434 MHz Band 8-900 MHz Band	- - -	59 56 50	- - -	dB dB dB
BI_HF	Blocking Immunity, higher bands	Offset = +/- 1 MHz Offset = +/- 2 MHz Offset = +/- 10 MHz	- - -	71 76 84	- - -	dB dB dB
BI_LF	Blocking Immunity, lower bands	Offset = +/- 1 MHz Offset = +/- 2 MHz Offset = +/- 10 MHz	- - -	71 72 78	- - -	dB dB dB
IIP2	2nd order Input Intercept Point Unwanted tones are 20 MHz above the LO	Highest LNA gain	-	+55	-	dBm
IIP3_HF	3rd order Input Intercept point Unwanted tones are 1MHz and 1.995 MHz above the LO	Higher bands Highest LNA gain G1 LNA gain G2, 4dB sensitivity hit	- -	-12.5 -8.5	- -	dBm dBm
IIP3_LF	3rd order Input Intercept point Unwanted tones are 1MHz and 1.995 MHz above the LO	Lower bands Highest LNA gain G1 LNA gain G2, 2.5dB sensitivity hit	- -	-22 -16	- -	dBm dBm
BW_SSB	Single Side channel filter BW	Programmable	2.7	-	250	kHz
IMR	Image Rejection	Wanted signal 3dB over sensitivity BER=0,1%	-	48	-	dB
IMA	Image Attenuation		-	57	-	dB
DR_RSSI	RSSI Dynamic Range	AGC enabled	Min Max	-127 0	- -	dBm dBm

- * $RxBw = 83 \text{ kHz}$ (Single Side Bandwidth)
 ** $RxBw = 50 \text{ kHz}$ (Single Side Bandwidth)
 *** $RxBw = 250 \text{ kHz}$ (Single Side Bandwidth)

2.4.4. FSK/OOK Mode Transmitter

Table 54 Transmitter Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
RF_OP	RF output power in 50 ohms on RFO pin (High efficiency PA).	Programmable with steps Max Min	+11 -	+14 -1	- -	dBm dBm
Δ RF_OP_V	RF output power stability on RFO pin versus voltage supply.	VDD = 2.5 V to 3.3 V VDD = 1.8 V to 3.7 V	- -	3 8	- -	dB dB
RF_OPH	RF output power in 50 ohms, on PA_BOOST pin (Regulated PA).	Programmable with 1dB steps Max Min	- -	+17 +2	- -	dBm dBm

.....*Halaman ini sengaja dikosongkan*.....

BIODATA PENULIS



Hendrik Hermawan lahir di Lamongan pada tanggal 13 Juni 1997 merupakan anak kedua dari dua bersaudara. Penulis menyelesaikan pendidikan dasar di SD Negeri Nglebur, dilanjutkan pendidikan pendidikan tingkat menengah di SMP Negeri 1 Kedungpring dan sekolah tingkat atas di SMA Negeri 1 Kedungpring. Penulis memulai kehidupan perkuliahan pada tahun 2015 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Selama masa perkuliahan, penulis aktif dalam berbagai kegiatan laboratorium di bidang studi elektronika. Penulis juga menjadi asisten praktikum, koordinator praktikum, dan koordinator asisten laboratorium di bidang studi elektronika

Email : hermawanhendrik1@gmail.com
Hp/WA : 085608629578
Facebook : Hendric Alvarest
Line : @hendicalvarest