

**SKRIPSI**

**PURWARUPA SISTEM MONITORING INDEKS POLUSI  
UDARA**



**Disusun Oleh :**

**ADHE MUHAMMAD YUSUF F**

**DBC 114 124**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2020**

**SKRIPSI**

**PURWARUPA SISTEM MONITORING INDEKS POLUSI UDARA**

Sebagai salah satu syarat menyelesaikan Program Strata - 1  
pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

**OLEH :**

**ADHE MUHAMMAD YUSUF F**

**NIM. DBC 114 124**

Disetujui untuk diajukan dalam Seminar Akhir Skripsi,

Palangka Raya, 9 November 2020

Pembimbing I



**NAHUMI NUGRAHANINGSIH, S.T., M.T., Ph.D**  
**NIP. 19791009 200801 2 016**

Pembimbing II



**FELICIA SYLVIANA, S.T., M.M**  
**NIP. 19760118 200312 2 003**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKA RAYA**

**2020**

# PURWARUPA SISTEM MONITORING INDEKS POLUSI UDARA

## SKRIPSI

Sebagai salah satu syarat untuk menyelesaikan Program Strata-1 pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya




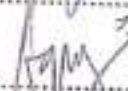
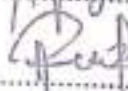
Oleh

**ADHE MUHAMMAD YUSUF F**  
DBC 114 124

Telah dipertahankan didepan tim penguji, pada :

Hari/Tanggal : Senin, 9 November 2020

Waktu : 13.00-14.30 WIB

- |  |   |  |           |
|--|---|--|-----------|
| 1. VIKTOR H. PRANATAWIJAYA, S.T., M.T<br>NIP. 19810606 200501 1 001      | : |   | (Ketua)   |
| 2. NAHUMI NUGRAHANINGSIH, S.T., M.T., Ph.D<br>NIP. 19791009 200801 2 016 | : |  | (Anggota) |
| 3. FELICIA SYLVIANA, S.T., M.M<br>NIP. 19760118 200312 2 003             | : |   | (Anggota) |
| 4. AGUS S. SARAGIH, S.T., M.Eng<br>NIP. 19850818 201212 1 003            | : |   | (Anggota) |
| 5. RESSA PRISKILA, ST., MT<br>NIP. 19940301 201903 2 016                 | : |   | (Anggota) |

Mengetahui :

Fakultas Teknik  
Universitas Palangka Raya  
Dekan



**I. WALUYO NUGWANTORO, M.T.**  
NIP. 19651119 199302 1 001

Jurusan / Program Studi Teknik Informatika  
Fakultas Teknik Universitas Palangka Raya  
Ketua Jurusan,



**ABERTUN SAGIT SAHAY, S.T., M.Eng**  
NIP. 19751212 200312 1 002

## PERNYATAAN

Dengan ini saya menyatakan dengan sebenar - benarnya bahwa dalam Skripsi ini tidak terdapat karya ilmiah yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi, serta tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam Skripsi ini dan disebutkan dalam Tinjauan Pustaka.

Palangka Raya, 14 November 2020



**Adhe Muhammad Yusuf F**

**DBC 114 124**

## RIWAYAT PENYUSUN

### Data Diri

Nama : ADHE MUHAMMAD YUSUF F  
NIM : DBC 114 124  
Fakultas : Teknik  
Jurusan/Program Studi : Teknik Informatika  
Jenjang : Strata 1 ( S-1 )  
Jenis Kelamin : Laki-laki  
Tempat, Tanggal Lahir : Palangkaraya, 06 Mei 1996  
Agama : Islam  
Status dalam Keluarga : Anak Kandung  
Anak ke - : Pertama (Satu)  
Alamat : Jl. TJILIK RIWUT KM. 2.5  
No. Telpon/HP : +62 8956 0550 4193



### Data Orang Tua

Nama Ayah : FRANKY FRANSNATA XANANA  
Pekerjaan Ayah : SWASTA  
Nama Ibu : I KETUT DERTI  
Pekerjaan Ibu : PNS  
Alamat Orang Tua : Jl. TJILIK RIWUT KM. 2.5  
No. Telpon/HP : +62 8525 2715 333

### Riwayat Pendidikan \*)

SD : SDN 9 Palangka (Tahun Lulus 2008)  
SMP : SMPN 3 Palangka Raya (Tahun Lulus 2011)  
SMA : SMKN 3 Palangka Raya (Tahun Lulus 2014)

Palangka Raya, 14 November 2020



ADHE MUHAMMAD YUSUF F  
DBC 114 124

Keterangan:

\*) Nama, Tempat, Tahun Lulus

## **PERSEMBAHAN**

Alhamdulillah Rabbil Alamin

Sujud syukur kepada Allah SWT, yang Maha Esa, yang Maha Agung dan Maha Pengasih lagi Maha Penyayang, Tuhan semesta raya.

Shalawat dan salam kepada baginda Rasulullah SAW, yang telah mengenalkan Tuhan Yang Maha Esa sebagai kebenaran sejati pada jiwa-jiwa pencinta-Nya, untuk jiwa-jiwa suci yang senantiasa menghembuskan nafas-Nya, menuliskan keberkahan nama-Nya.

Dengan segenap rasa cinta dan kasih, ku persembahkan karya kecil ini untuk orang-orang terkasih...

1. Orangtua tercinta, yaitu Ayah (Franky Fransnata Xanana) dan Ibu (I Ketut Derti). Terima kasih buat dukungan, semangat, dan perjuangannya dalam mendidik saya dari lahir hingga saat ini untuk menjadi orang yang lebih baik di masa depan, segala dukungan dan cinta kasih yang tak terhingga. Serta mendoakanku, dan menasehatiku untuk menjadi lebih baik.
2. Adikku tersayang Dhea Fitri Agustin Fransnita Xanana, yang selalu memberi semangat sehingga menambah warna dalam hidupku.
3. Tiada pantas kata selain terimakasih yang tak terhingga untuk Bapak/Ibu Dosen atas ilmu yang telah engkau berikan, jasa-jasamu takkan pernah ku lupa.
4. Untuk orang terdekat, tercinta, tersayang, dan sahabat-sahabat terbaikku yang tak bisa aku sebutkan satu persatu, yang senantiasa banyak memberiku dorongan motivasi dan doa.
5. Teman-teman Teknik Informatika UPR Angkatan 2014.
6. Teman-teman KKN Kalampangan yang selalu kompak dalam segala hal, pengalaman bersama kalian takkan ku lupakan.
7. Almamaterku tercinta, yaitu UPR.

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya, penulis dapat menyelesaikan laporan Skripsi dengan judul "PURWARUPA SISTEM MONITORING INDEKS POLUSI UDARA". Dengan diangkatnya laporan skripsi ini, saya selaku penulis mengucapkan terima kasih kepada :

1. Kedua orang tua yang telah memberikan semangat serta dukungan sepenuhnya.
2. Dosen pembimbing skripsi ibu Nahumi Nugrahaningsih, S.T., M.T., Ph.D dan juga ibu Felicia Sylviana, S.T., M.M.
3. Teman-teman Teknik Informatika UPR angkatan 2014 juga selalu memberi semangat serta dukungan selama pengerjaan skripsi ini.

Saya berharap laporan Skripsi ini dapat berguna dalam rangka menambah wawasan dan pengetahuan kita mengenai pengertian, prinsip kerja serta penerapan pada Arduino dan NodeMCU. Saya juga menyadari bahwa di dalam skripsi ini masih jauh dari sempurna. Oleh karena itu, saya mengharapkan pendapat, kritik, maupun saran yang bersifat membangun demi kesempurnaan laporan ini. Akhir kata, semoga laporan Skripsi ini dapat bermanfaat bagi kita semua terutama Mahasiswa/(i) jurusan Teknik Informatika Universitas Palangkaraya.

Palangka Raya, 18 Oktober 2020



Adhe Muhammad Yusuf E

NIM. DBC 114 124

## “PURWARUPA SISTEM MONITORING INDEKS POLUSI UDARA”

Adhe Muhammad Yusuf F (NIM DBC 114 124)  
Jurusan Teknik InformatikaFakultas Teknik Universitas Palangka Raya  
Kampus Tunjung Nyaho Jl. Yos Sudarso Palangka Raya 73112  
*E-mail* : adshnz@gmail.com

### ABSTRAK

Polusi udara merupakan hadirnya satu atau lebih zat fisika, kimia, maupun biologi di udara dalam jumlah yang dapat membahayakan kesehatan makhluk hidup, mengganggu estetika dan kenyamanan. Pencemaran udara di Provinsi Kalimantan Tengah pada umumnya bersumber dari kondisi lalu lintas kendaraan bermotor (sumber bergerak) dan kebakaran hutan. Sehingga untuk mengukur dampak polusi udara, dibutuhkan pengukuran beberapa parameter pada beberapa tempat yang berbeda pada suatu wilayah. Teknologi Internet of Things (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel dan Internet. IoT mengacu pada pengidentifikasian suatu objek yang direpresentasikan secara virtual di dunia maya atau internet.

Membangun, merancang dan mengembangkan sebuah desain *Internet of Things* untuk memonitoring kualitas udara di alam bebas dengan studi kasus polusi udara. Dengan memanfaatkan mikrokontroler ESP-8266 jenis wifi module sebagai pengiriman data ke web server dengan menambahkan mikrokontroler Arduino Uno sebagai proses mengolah nilai daa sensor lalu dikirimkan ke NodeMCU. Ada 4 sensor yang digunakan yaitu adalah CO (MQ-7), O3 (MQ 131), Dust Sensor (PM2.5), dan NO2 (MQ 135). Lokasi perekaman data dan monitoring data menggunakan protokol HTTP webservice.

Penelitian di lakukan pada 2 lokasi yang berbeda dengan waktu yang telah di tentukan, yaitu di lingkungan area Rumah dan juga di Taman Makan Pahlawan Sanaman Lampang.

*Kata-kata kunci* :*Internet of Things, Webservice, mikrokontroler ESP8266 NodeMCU*

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>iv</b>
<b>HALAMAN RIWAYAT PENYUSUN .....</b>	<b>v</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>ABSTRAK .....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan.....	3
1.5. Manfaat <b>Penelitian</b> .....	3
1.6. Alat dan Bahan .....	3
1.7. Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI</b>	
2.1. Tinjauan Pustaka.....	7
2.2. Polusi Udara .....	11
2.3. Internet Of Things (IOT) .....	16
2.4. Mikrokontroler.....	17
2.5. WIFI Module ESP8266 .....	18
2.6. Arduino IDE .....	21
2.7. Serial Komunikasi .....	24
2.8. Protokol Hypertext Transfer Protocol (HTTP).....	26
2.9. PHP.....	27

2.10. XAMPP .....	27
2.11. Localhost.....	28

### **BAB III METODOLOGI PENELITIAN**

3.1. Metodologi Penelitian.....	29
3.2. Desain Arsitektur Sistem .....	31
3.3. Alat dan Bahan .....	31
3.4. Rangkaian Keseluruhan Alat .....	35
3.5. Cara Kerja Alat.....	36
3.6. Pembuatan Alat.....	38
3.7. Perhitungan ISPU .....	44
3.8. Konversi nilai ke PPM.....	47

### **BAB IV HASIL DAN PEMBAHASAN**

4.1. Uji Coba Perangkat Keras .....	56
4.2. Pengambilan Pengambilan Data di sekitar area Rumah .....	58
4.3. Pengambilan Data di Taman Makam Pahlawan Sanaman Lampang .....	59
4.4. Analisis Data.....	60
4.5. Permasalahan dan Solusi .....	67
4.6. Kode Program.....	69
4.7. Hasil Pengujian Data dengan Metode Black Box Testing.....	82

### **BAB V PENUTUP**

5.1. Kesimpulan.....	85
5.2. Saran .....	86

### **DAFTAR PUSTAKA**

## DAFTAR TABEL

Tabel 1.1. Jadwal Pelaksanaan .....	6
Tabel 2.1 Tinjauan Pustaka .....	10
Tabel 2.2 Indeks Standar Pencemar Udara (ISPU).....	12
Tabel 2.3 Rentang Indeks Standar Pencemar Udara.....	15
Tabel 2.4 Batas Indeks Pencemar Udara Dalam Satuan SI .....	16
Tabel 4.1 Tabel Pengujian Sensor dalam ruangan .....	83
Tabel 4.2 Tabel Pengujian Sensor di luar ruangan .....	84

## DAFTAR GAMBAR

Gambar 2.1 jenis-jenis Mikrokontroler.....	17
Gambar 2.2 Jenis dan generasi WiFi module ESP8266.....	18
Gambar 2.3 Interface ESP8266 NodeMcu ESP-12E .....	19
Gambar 2.4 Tampilan Arduino IDE.....	21
Gambar 2.5 Tampilan dashboard Arduino IDE.....	22
Gambar 2.6 Keterangan pada Arduino IDE.....	26
Gambar 2.7 Istilah Pada Serial Komunikasi .....	25
Gambar 2.8 Ilustrasi Pengiriman Data Komunikasi Serial .....	27
Gambar 3.1 Desain arsitektur sistem .....	32
Gambar 3.2 Mikrokontroler ESP8266 jenis NodeMcu Ver.1.0.....	33
Gambar 3.3 Pinout Header NodeMCU .....	33
Gambar 3.4 Sensor MQ-7 .....	34
Gambar 3.5 GP2Y1010AU0F Optical Dust Sensor.....	34
Gambar 3.6 Sensor O3 (MQ-131).....	35
Gambar 3.7 Sensor O3 (MQ-135).....	35
Gambar 3.8 Board Arduino Uno.....	36
Gambar 3.9 Diagram blok sistem .....	36
Gambar 3.10 Rangkaian sensor .....	38
Gambar 3.11 Tampilan desain rangkaian skematik sensor.....	39
Gambar 3.12 Proses pengerjaan kotak pertama .....	40
Gambar 3.13 Proses perangkaian alat dalam kotak .....	41
Gambar 3.14 Proses perangkaian sensor.....	42
Gambar 3.15 Proses pemasangan alat pada tiang besi.....	42
Gambar 3.16 Tampilan aplikasi Arduino Ide.....	43
Gambar 3.17 Tampilan Preferences .....	44
Gambar 3.18 Tampilan Board Manager.....	44

Gambar 3.19 Tampilan Menunggu Hasil Instalasi .....	44
Gambar 3.20 Tampilan code untuk mencari nilai Rs.....	48
Gambar 3.21 Tampilan serial monitor nilai Rs.....	50
Gambar 3.22 Tampilan saat selesai melakukan penitikan grafik sensitivitas mq-7 .	51
Gambar 3.23 Tampilan grafik hubungan rs/ro dengan ppm di excel dari grafik sensitivitas mq-7 .....	51
Gambar 3.24 Tampilan code untuk mencari nilai PPM.....	52
Gambar 3.25 Tampilan serial monitor nilai PPM dari sensor CO MQ7.....	54
Gambar 4.1 Uji coba alat menggunakan asap pembakaran kertas.....	55
Gambar 4.2 Perbaikan program dan pengecekan rangkaian alat .....	56
Gambar 4.3 Uji coba alat dalam cuaca panas .....	57
Gambar 4.4 Proses pengambilan data di lingkungan kampus .....	58
Gambar 4.5 Pengambilan data di Taman Makam Pahlawan Sanaman Lampung.....	59
Gambar 4.6 Batas Nilai-Nilai Indeks.....	59
Gambar 4.7 Data CO di Area Rumah .....	61
Gambar 4.8 Data PM2.5 di Area Rumah .....	62
Gambar 4.9 Data O3 di Area Rumah.....	63
Gambar 4.10 Data NO2 di Area Rumah .....	63
Gambar 4.11 Data NO2 di Taman Makam Pahlawan Sanaman Lampung.....	64
Gambar 4.12 Data CO di Taman Makam Pahlawan Sanaman Lampung.....	65
Gambar 4.13 Data PM2.5 di Taman Makam Pahlawan Sanaman Lampung.....	66
Gambar 4.14 Data O3 di Taman Makam Pahlawan Sanaman Lampung.....	66

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pencemaran udara dapat mempengaruhi kesejahteraan manusia, baik secara langsung ataupun secara tidak langsung. Pengaruh pencemaran udara secara langsung dapat berupa penyakit dan kematian, sedangkan pengaruhnya secara tidak langsung adalah terganggunya berbagai sumberdaya alam yang penting untuk kehidupan dan kesejahteraan manusia.

Pencemaran udara adalah kondisi dimana turunnya kualitas udara dan udara terkontaminasi oleh zat-zat yang tidak baik bagi kesehatan manusia, baik dalam ruangan (*indoor*) maupun luar ruangan (*outdoor*) dengan agen kimia, fisik, atau biologi yang telah mengubah karakteristik alami dari atmosfer. Sehingga udara mengalami penurunan mutu dalam penggunaannya dan akhirnya tidak dapat dipergunakan lagi sebagaimana mestinya sesuai dengan fungsinya. Adapun bahan pencemar (polutan) utama yang dapat menimbulkan masalah kesehatan, yaitu partikulat, karbon monoksida (CO), ozon (O<sub>3</sub>), nitrogen dioksida (NO<sub>2</sub>), dan sulfur dioksida (SO<sub>2</sub>).

Internet of Things (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa perlu interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel dan Internet. IoT mengacu pada pengidentifikasian suatu objek yang direpresentasikan secara virtual di dunia maya atau internet. Biaya alat yang terjangkau namun dapat memberikan informasi mengenai kondisi tingkat kadar pencemaran udara. Dengan menggunakan teknologi IoT, kondisi dan kualitas udara atau tingkat polusi udara berupa kandungan gas karbon monoksida (CO), kadar gas nitrogen (NO<sub>2</sub>), kadar gas ozon (O<sub>3</sub>), dan partikulat matter (PM 2.5) dapat dibaca menggunakan sensor dan kemudian dikirimkan ke server untuk diolah menjadi informasi. Data yang didapatkan dari sensor secara real time kemudian dimasukkan ke dalam sebuah web server melalui jaringan dengan menggunakan WiFi Module. Data tersebut dapat diakses secara real time melalui web server.

Dari penjelasan diatas, bagaimana penerapan teknologi IoT untuk dapat membantu memecahkan permasalahan dengan menggunakan platform Arduino untuk dijadikan bahan penulisan Skripsi yang berjudul :"**Purwarupa Sistem Monitoring Indeks Polusi Udara**" guna untuk memantau dan mengetahui kadar kualitas udara serta dapat sigap mengambil keputusan saat terjadi tingkat pulusi udara yang sangat membahayakan.

### **1.2 Rumusan Masalah**

1. Bagaimana merancang dan membangun sistem monitoring indeks polusi udara?

### **1.3 Batasan Masalah**

1. Radius modul sensor terbatas, hanya berjarak sekitar kurang dari 10 meter.
2. Standar indeks baku kualitas udara yang digunakan adalah keputusan Menteri Lingkungan Hidup no 45 tahun 1997.
3. Sistem ini menekankan pada monitoring saja, tanpa controlling atau feedback.
4. Menggunakan Mikrokontroler Arduino Uno R3 untuk melakukan pengolahan data dan diakses menggunakan modul WiFi sebagai perantara dari sensor ke webservernya.
5. Data yang diambil merupakan data mentah yang diterima dari hasil pemantauan melalui website.
6. Sistem ini menggunakan 4 buah sensor, yaitu sensor CO (MQ 7) untuk mengukur kadar karbon monoksida, sensor O3 (MQ 131) untuk mengukur kadar gas Ozon, sensor NO2 (MQ 135) untuk mengukur kadar gas Nitrogen Dioksida dan sensor PM 2.5 (GP2Y1014AU0F) digunakan untuk mengukur kadar partikulat, yang di kemas dalam satu alat.

#### 1.4 Tujuan

1. Membuat sebuah perangkat prototype IoT untuk mengetahui tingkat kualitas udara.
2. Membuat sebuah sistem untuk mengumpulkan data, yang kemudian akan di analisis untuk menghasilkan sebuah informasi yang di kirim ke dalam sebuah website.
3. Menyelesaikan Skripsi sebagai syarat kelulusan di jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

#### 1.5 Manfaat Penelitian

1. Memberikan informasi kepada semua pihak mengenai keadaan kualitas udara yang terjadi untuk mengantisipasi keadaan yang tidak diinginkan.
2. Dapat diterapkan pada sistem yang nyata dan dapat dijadikan sebagai indikator atau sebuah peringatan pengambilan tindakan antisipasi secara dini dan agar dapat mengurangi dampak pencemaran udara lingkungan.
3. Mengaplikasikan teori yang di peroleh.
4. Menambah wawasan dan pengalaman Peneliti.

#### 1.6 Alat dan Bahan

##### 1. Alat :

- a. Solder
- b. Pistol Lem
- c. Gunting
- d. Lakban

##### 2. Bahan :

- a. Mikrokontroler ESP-8266 NodeMcu
- b. Arduino Uno R3
- c. Sensor CO ( MQ 7 )
- d. Sensor PM 2.5 ( Optik Sharp GP2Y1014AU0F )

- e. Sensor NO2 ( MQ 135 )
- f. Sensor O3 ( MQ 131 )
- g. Adaptor power supply 12v, 1.5a
- h. Modul WiFi

### 3. Hardware komputer :

- a. Prosesor : Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz, 2401 Mhz, 4 Core(s), 8 Logical Processor(s)
- b. Installed Physical Memory (RAM) : 8,00 GB
- c. Graphics Card : NVIDIA GeForce GT 745M
- d. Harddisk : 931,51 GB Standard SATA AHCI Controller

### 4. Software komputer :

- a. Arduino IDE
- b. Visio Project
- c. Website

## 1.7 Sistematika Penulisan

### BAB I PENDAHULUAN

Pada bab ini, di uraikan mengenai latar belakang masalah, perumusan masalah, ruang lingkup, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

### BAB II LANDASAN TEORI

Landasan teori yang digunakan untuk menguraikan mengenai suatu pedoman atau teori yang dikemukakan oleh pakar-pakar dalam suatu bidang tertentu untuk memecahkan masalah.

### BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan proses desain/perancangan program yang dilakukan dalam menyelesaikan proyek Tugas Akhir

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini menampilkan hasil pembuatan program dan menjelaskan bagaimana program ini dijalankan/digunakan.

**BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang menjawab dari rumusan masalah dan saran yang nantinya berguna apabila dilain waktu ingin mengembangkan perangkat lunak yang telah dibuat untuk Skripsi ini.



Tabel 1.1 Jadwal Pelaksanaan

No	Kegiatan	Bulan																															
		Bulan Ke 1				Bulan Ke 2				Bulan Ke 3				Bulan Ke 4				Bulan Ke 5				Bulan Ke 6				Bulan Ke 7				Bulan Ke 8			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Penyusunan dan pengumpulan proposal	■	■	■	■																												
2	Seminar Proposal																																
3	Analisa Kebutuhan																																
4	Desain Sistem																																
5	Pengkodean dan Pengujian																	■	■	■	■												
6	Pembuatan Laporan dan Evaluasi																	■	■	■	■												
7	Seminar Hasil																					■											
8	Seminar Akhir																													■			

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 TINJAUAN PUSTAKA**

##### **2.1.1 Sistem Pemantauan Kualitas Udara**

Azis Sugianto. Sistem Pemantauan Kualitas Udara. Tugas Akhir Jurusan Teknik Informatika, fakultas teknik, Universitas Widyatama, Bandung. Salah satu upaya penanggulangan pencemaran udara adalah dengan cara mengukur kualitas udara untuk mengategorikan kualitas udara. Pembangunan sistem ini bertujuan untuk mengukur dan memantau kualitas udara. Sistem dibangun menggunakan development board Arduino Uno berbasis ATmega328P, sensor MQ-135 untuk mendeteksi kontaminan udara, sensor DHT11 digunakan untuk mengukur suhu dan kelembaban udara, serta Arduino Ethernet Shield sebagai penghubung ke webserver. Sedangkan perangkat lunaknya dibangun menggunakan bahasa C Arduino untuk sistem benamnya, dan PHP-SQL untuk membangun aplikasi web di mana pengguna dapat melihat informasi kualitas udara melalui situs web. Pemanfaatan sensor MQ-135 untuk mengukur kadar kontaminan udara diharapkan mampu membantu pengguna meminimalisasi risiko menghirup udara berbahaya. Peran web server dalam sistem ini adalah untuk menginformasikan kualitas udara kepada masyarakat melalui internet seefektif mungkin.

##### **2.1.2. Pengembangan Sistem Deteksi Karbon Monoksida Berbasis IoT**

Salman Al Farizi, Eko Sakti Pramukantoro, Heru Nurwarsito. Pengembangan Sistem Deteksi Karbon Monoksida Berbasis IoT. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, e-ISSN: 2548-964X : Vol. 2, No. 10, Oktober 2018, him. 4164-4170. Karbon monoksida sebagai polutan utama berhak mendapatkan sorotan lebih dalam penanganannya. Penelitian ini fokus terhadap pendeteksian gas karbon monoksida di dalam ruangan. Pada penelitian sebelumnya telah

dikembangkan IoT middleware yang dapat memecahkan masalah interoperabilitas sintaksis. Sistem yang dikembangkan akan diintegrasikan dengan IoT middleware yang sudah ada Untuk memudahkan interaksi dengan pengguna, pengembangan dilakukan dengan pembuatan aplikasi untuk telepon pintar berbasis sistem operasi Android yang memberikan antarmuka yang lebih mudah dimengerti oleh pengguna dan dapat memberikan pemberitahuan di aplikasi apabila nilai pembacaan sensor melebihi batas aman. Terdapat beberapa parameter untuk menguji sistem yang dikembangkan, diantaranya integrasi, pengujian sistem berbasis konteks, dan Quality of Service (QoS). Berdasarkan hasil pengujian pada sistem ini, didapatkan bahwa sistem yang dikembangkan memiliki integritas data yang sempurna. Disisi lain, sistem yang dikembangkan memenuhi kaidah komputasi berbasis konteks.

### **2.1.3 Rancang Bangun Sistem Pemantauan Kualitas Udara Menggunakan Wireless Sensor Networks Dengan Topologi Cluster**

Dyah Lestari Saraswati, Jusak Jusak, Yosefine Triwidyastuti. Rancang Bangun Sistem Pemantauan Kualitas Udara Menggunakan Wireless Sensor Networks Dengan Topologi Cluster. *Journal of Control and Network Systems*. JCONES Vol. 5, No. 1 (2016) 126-135. Hal ini umumnya dipahami bahwa efek polusi udara merugikan bagi kesehatan manusia, dapat menyebabkan penyakit dan menyebabkan todeath. Oleh karena itu, sangat dibutuhkan untuk memantau tingkat udara secara terus menerus. Dalam penelitian ini, sistem pemantauan memanfaatkan jaringan sensor nirkabel dengan metode cluster. Kami menganggap bahwa ada dua cluster dimana setiap cluster head memiliki dua node anak yang dilengkapi dengan sensor CO dan sensor CO<sub>2</sub> untuk setiap node. Data dari node anak dikirim ke server melalui parent node dan cluster masing-masing cluster. Kami menggunakan radio xbee sebagai transmisi nirkabel dari node anak ke node induk, kami menggunakan modul Tonylabs WiFi CC3000 sebagai transmisi nirkabel dari

orang tua ke server. Hasilnya menunjukkan bahwa delay dan packet loss antara node anak dan node induk melalui xbee masing-masing sekitar 4,75 detik dan 5%. Ini berarti sistem memiliki delay yang pendek dan packet loss yang relatif rendah. Di sisi lain ada packet loss dari node induk ke server relatif lebih tinggi daripada simpul dari node anak ke node induk yaitu kira-kira 14,17% , Menariknya dapat dicapai mendekati nol detik selama transmisi.



**Tabel 2.1** Tinjauan Pustaka

	Judul Penelitian			
	Sistem Pemantauan Kualitas Udara (Azis Sugianto)	Pengembangan Sistem Deteksi Karbon Monoksida Berbasis IoT (Heru Nurwarsito)	Rancang Bangun Sistem Pemantauan Kualitas Udara Menggunakan Wireless Sensor Networks Dengan Topologi Cluster (Yosefine Triwidyastuti)	Purwarupa Sistem Monitoring Indeks Polusi Udara (Adhe Muhammad Yusuf F)
Sensor	Sensor NO2 ( MQ 135 ), Sensor Kelembaban Udara (DHT11)	Sensor CO ( MQ 7 )	Sensor CO ( MQ 7 ), Sensor CO2 (MG811)	Sensor CO ( MQ 7 ), Sensor PM 2.5 (GP2Y1014AU0F ), Sensor NO2 ( MQ 135 ), Sensor O3 ( MQ 131 )
Mikrokontroler	Arduino Uno ATmega328P	Arduino Uno	Arduino Uno ATmega328P	ESP-8266 NodeMcu
Modul Komunikasi	Arduino Ethernet Shield	ESP8266	Tonylabs WiFi CC3000	ESP8266
Monitoring	Aplikasi Web	Android	-	Aplikasi Web
Lokasi pengambilan data	Di luar ruangan	Di dalam ruangan	-	Di luar ruangan

## 2.2 Polusi Udara

Berdasarkan jenisnya polutan polusi udara ini dibedakan menjadi dua, yakni: Polutan Primer, adalah suatu substansi pencemar yang mana secara langsung ditimbulkan dari sumber polusinya. Polutan ini bentuk bahkan komposisinya sama dengan saat pencemarannya itu. Sebagai contoh polutan primer ini adalah CO<sub>2</sub>, CO, Hidrokarbon, SO, nitrogen Oksida, dan juga berbagai partikel yang lain. Polutan Sekunder, adalah substansi pencemar yang mana terbentuk dari reaksi polutan primer yang terjadi di atmosfer. Reaksi ini dapat terjadi secara otomatis maupun dengan bantuan dari katalisator, seperti misalnya saja sinar matahari. (Sunu, 2001)

Dampak langsung dari polusi udara ini adalah pada kesehatan makhluk hidup khususnya pada manusia itu sendiri., berdampak pada hujan asam, efek ruma kaca, kerusakan lapisan ozon, dan lain sebagainya. Dan dampak yang sering kita jumpai (umum) saat adanya polusi ini adalah Infeksi Saluran Pernafasan Akut atau ISPA termasuk juga didalamnya ada asma serta bronkitis. Beberapa zat polutan ini dapat bersifat toksik bahkan juga karsinogenik. Yang mana intinya berdampak buruk bagi kesehatan makhluk hidup dan juga lingkungan. Polusi udara adalah suatu keadaan dimana udara mengandung bahan kimia, partikel, atau bahan biologis lainnya yang menyebabkan kerugian atau ketidaknyamanan pada manusia atau organisme hidup lainnya, atau menyebabkan kerusakan pada lingkungan alam atau lingkungan binaan, ke atmosfer. (Wardhana, 2004)

Polusi udara merupakan masalah lingkungan paling serius yang dihadapi peradaban kita saat ini. Paling sering, hal ini disebabkan oleh aktivitas manusia seperti penambangan, konstruksi, transportasi, pekerjaan industri, pertanian, peleburan, dll. Namun, proses alami seperti letusan gunung berapi dan kebakaran hutan juga dapat mencemari udara, tetapi kejadiannya jarang terjadi dan biasanya memiliki efek lokal, tidak seperti aktivitas manusia yang merupakan penyebab polusi udara dan berkontribusi terhadap polusi global udara setiap hari. (Wardhana, 2004)

Tabel 2.2 Indeks Standar Pencemar Udara (ISPU)

Pengaruh Pencemaran Udara Terhadap Makhluk Hidup						
Kategori	Rentang	Karbon Monoksida (CO)	Nitrogen (NO <sub>2</sub> )	Ozon (O <sub>3</sub> )	Sulfur Dioksida (SO <sub>2</sub> )	Partikulat (PM 2.5)
Baik	0-50	Tidak ada efek	Sedikit berbau	Luka pada Beberapa spesies tumbuhan akibat kombinasi dengan SO <sub>2</sub> (Selama 4 Jam)	Luka pada Beberapa spesies tumbuhan akibat kombinasi dengan O <sub>3</sub> (Selama 4 Jam)	Tidak ada efek
Sedang	51 – 100	Perubahan kimia darah tapi tidak terdeteksi	Berbau	Luka pada Beberapa spesies tumbuhan	Luka pada Beberapa spesies tumbuhan	Terjadi penurunan pada jarak pandang
Tidak Sehat	101 – 199	Peningkatan pada kardiovaskular pada perokok yang sakit jantung	Bau dan kehilangan warna. Peningkatan reaktivitas pembuluh tenggorokan pada penderita asma	Penurunan kemampuan pada atlet yang berlatih keras	Bau, Meningkatnya kerusakan tanaman	Jarak pandang turun dan terjadi pengotoran debu di mana-mana
Sangat Tidak Sehat	200-299	Meningkatnya kardiovaskular pada orang bukan perokok yang berpenyakit jantung, akan tampak beberapa kelemahan terlihat secara nyata	Meningkatnya sensitivitas pasien yang berpenyakit asma dan <i>bronchitis</i>	Olah raga Ringan mengakibatkan pengaruh pernafasan pada pasien yang berpenyakit paru-paru kronis	Meningkatnya sensitivitas pada pasien berpenyakit asma dan <i>bronchitis</i>	Meningkatnya sensitivitas pada pasien berpenyakit asma dan <i>bronchitis</i>
Berbahaya	300 – lebih	Tingkat yang berbahaya bagi semua populasi yang terpapar				

(Sumber : Lampiran Keputusan Kepala Bapedal

Kep-107/KABAPEDAL/11/1997)

## **2.2.1 Jenis Polusi Udara**

Sejumlah besar kontaminan dapat mencemari udara dalam berbagai macam bentuk. Hampir semua bahan kimia beracun bisa masuk ke atmosfer untuk mencemari udara yang kita hirup. Partikel aerosol (awan partikel cair dan padat dalam gas) yang ditemukan di udara juga dapat mengandung polutan. Senyawa kimia yang menurunkan kualitas udara biasanya disebut sebagai polutan udara. (Sunu, 2001)

### **2.2.1.1 Karbon Monoksida (Carbon Monoxide; CO)**

Karbon monoksida adalah polutan yang dikenal hampir di mana-mana di kota-kota masa kini di seluruh dunia. Ini dihasilkan selama proses pembakaran bahan bakar (bensin, minyak, solar, kayu, arang dll.). Ini berarti ada di knalpot kendaraan dan asap rokok, serta ruang yang tidak digunakan di mana bahan bakar sedang dibakar. (Sunu, 2001)

### **2.2.1.2 Partikulat Debu (Particulate Matter; PM 2.5)**

Particulate matter (PM) adalah istilah untuk partikel padat atau cair yang ditemukan di udara. Partikel dengan ukuran besar atau cukup gelap dapat dilihat sebagai jelaga atau asap. Sedangkan partikel yang sangat kecil dapat dilihat dengan mikroskop elektron. Partikel berasal dari berbagai sumber baik mobile dan stasioner (diesel truk, woodstoves, pembangkit listrik, dll), sehingga sifat kimia dan fisika partikel sangat bervariasi. Partikel dapat langsung diemisikan atau terbentuk di atmosfer saat polutan gas seperti SO<sub>2</sub> dan NO<sub>x</sub> bereaksi membentuk partikel halus. (Sunu, 2001)

PM-2.5 Standar merupakan partikel kecil yang bertanggung jawab untuk efek kesehatan yang merugikan karena kemampuannya untuk mencapai daerah yang lebih dalam pada saluran pernapasan. PM-2.5 termasuk partikel dengan diameter 10 mikrometer atau kurang.

Standar kesehatan berdasarkan PP No. 41 Tahun 1999 untuk PM-2.5 adalah 150 ug/Nm<sup>3</sup> (24 jam). Efek utama bagi kesehatan manusia dari paparan PM-2.5 meliputi: efek pada pernapasan dan sistem pernapasan, kerusakan jaringan paru-paru, kanker, dan kematian dini. Orang tua, anak-anak, dan orang-orang dengan penyakit paru-paru kronis, influenza, atau asma, sangat sensitif terhadap efek partikel. PM-2.5 yang asam juga dapat merusak bahan buatan manusia dan merupakan penyebab utama berkurangnya jarak pandang. (Sunu, 2001)

#### **2.2.1.3 Nitrogen dioksida (NO<sub>2</sub>)**

Senyawa inilah yang dianggap paling berbahaya, bahkan mematikan. NO<sub>2</sub> bisa secara signifikan menyebabkan peradangan pernapasan jika dihirup dalam jangka waktu lama. Biasanya senyawa ini ditemukan pada asap pembakaran; seperti asap pemanas, pembangkit listrik, mesin kendaraan, dan asap kapal. (Sunu, 2001)

#### **2.2.1.4 Ozon (O<sub>3</sub>)**

Senyawa ini berasal dari polutan dan sinar matahari, sehingga O<sub>3</sub> mudah ditemukan dan dihirup ketika cerah siang hari. O<sub>3</sub> disebabkan oleh asap kendaraan dan asap pabrik yang bercampur dengan panas matahari. Jika dihirup dalam jangka waktu lama, O<sub>3</sub> bisa menyebabkan gangguan paru-paru, seperti asma dan penyakit paru-paru lainnya. (Sunu, 2001)

### **2.2.2 Indek Standar Pencemar Udara (ISPU)**

Saat ini Indeks standar kualitas udara yang dipergunakan secara resmi di Indonesia adalah Indek Standar Pencemar Udara (ISPU), hal ini sesuai dengan Keputusan Menteri Negara Lingkungan Hidup Nomor KEP 45/ MENLH/1997 Tentang Indeks Standar Pencemar Udara. Dalam keputusan tersebut yang

dipergunakan sebagai bahan pertimbangan diantaranya : bahwa untuk memberikan kemudahan dari keseragaman informasi kualitas udara ambien kepada masyarakat di lokasi dan waktu tertentu serta sebagai bahan pertimbangan dalam melakukan upaya-upaya pengendalian pencemaran udara perlu disusun Indeks Standar Pencemar Udara. (Sunu, 2001)

Indeks Standar Pencemar Udara adalah angka yang tidak mempunyai satuan yang menggambarkan kondisi kualitas udara ambien di lokasi dan waktu tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan makhluk hidup lainnya. Indeks Standar Pencemar Udara ditetapkan dengan cara mengubah kadar pencemar udara yang terukur menjadi suatu angka yang tidak berdimensi. Rentang Indeks Standar Pencemar Udara dapat dilihat pada tabel berikut ini. (Sunu, 2001)

**Tabel 2.3** Rentang Indeks Standar Pencemar Udara

ISPU	Tingkat Pencemaran Udara	Dampak Kesehatan
0-50	Baik	Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan dan tidak berpengaruh pada tumbuhan, bangunan ataupun nilai estetika
51-100	Berlang	Kualitas udara yang tidak berpengaruh pada kesehatan manusia ataupun hewan tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika
101-199	Tidak Baik	Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitive atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika
200-299	Sangat Tidak Baik	Tingkat kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar
300-500	Berbahaya	Tingkat kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius pada populasi

(Sumber : Lampiran Keputusan Kepala Bapedal  
Kep-107/KABAPEDAL/11/1997)

Data Indeks Standar Pencemar Udara diperoleh dari pengoperasian Stasiun Pemantauan Kualitas Udara Ambien Otomatis. Baku mutu udara ambien adalah ukuran batas atau kadar zat, energi, dan/atau komponen yang ada atau yang seharusnya ada

dan/atau unsur pencemar yang ditenggang keberadaannya dalam udara ambien.

Sedangkan Parameter Indeks Standar Pencemar Udara meliputi :

- Partikulat (PM 2.5).
- Karbondioksida (CO).
- Nitrogen dioksida (NO<sub>2</sub>).
- Ozon (O<sub>3</sub>).

**Tabel 2.4** Batas Indeks Pencemar Udara Dalam Satuan SI

ISPU	PM 2.5 ( $\mu\text{g}/\text{m}^3$ )	CO ( $\mu\text{g}/\text{m}^3$ )	O <sub>3</sub> ( $\mu\text{g}/\text{m}^3$ )	NO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )
50	50	5	120	-
100	150	10	235	-
200	350	17	400	1130
300	420	34	800	2260
400	500	46	1000	3000
500	600	57,5	1200	3750

(Sumber : Lampiran Keputusan Kepala Bapedal  
Kep-107/KABAPEDAL/11/1997)

Formula untuk menghitung indeks dari setiap parameter adalah sebagai berikut.

$$I = \frac{I_a - I_b}{X_a - X_b} (X_x - X_b) + I_b$$

Keterangan : I = ISPU terhitung  
I<sub>a</sub> = ISPU batas atas  
I<sub>b</sub> = ISPU batas bawah  
X<sub>a</sub> = *Ambient* batas atas  
X<sub>b</sub> = *Ambient* batas bawah  
X<sub>x</sub> = Kadar *ambient* nyata hasil pengukuran

### 2.3 Internet of Things (IoT)

Internet of things (IoT) adalah konsep komputasi yang menggambarkan gagasan objek fisik sehari-hari yang terhubung ke internet dan mampu mengidentifikasi diri ke perangkat lain. Istilah ini erat diidentifikasi dengan RFID sebagai metode komunikasi,

meskipun itu juga mungkin termasuk teknologi sensor lain, teknologi nirkabel atau kode QR. (Kevin Ashton, 1999)

IOT signifikan karena suatu objek yang dapat merepresentasikan dirinya secara digital menjadi sesuatu yang lebih besar dari objek itu sendiri. Tidak lagi objek berhubungan hanya dengan pengguna, tetapi sekarang terhubung ke objek dan data database sekitarnya. Ketika banyak objek bertindak bersama-sama, mereka dikenal sebagai memiliki "kecerdasan ambient". Internet hal adalah konsep yang sulit untuk didefinisikan secara tepat. Bahkan, ada banyak kelompok berbeda yang telah mendefinisikan istilah tersebut, meskipun penggunaan awalnya telah dikaitkan dengan Kevin Ashton, seorang pakar inovasi digital. Setiap definisi berbagi gagasan bahwa versi pertama internet adalah tentang data yang dibuat oleh orang-orang, sedangkan versi berikutnya adalah tentang data yang dibuat oleh berbagai hal. (Kevin Ashton, 1999)

#### 2.4 Mikrokontroler

Mikrokontroler adalah sebuah mikroprosesor yang memiliki bagian-bagian tambahan yang memungkinkan untuk mengatur atau mengontrol benda lain. Dapat diartikan bahwa mikrokontroler menjalankan program yang telah dibuat oleh user dan disimpan dalam memory. Mikrokontroler merupakan alat yang sangat berguna yang dapat membantu perancang untuk memanipulasi data yang didapat dari input ataupun output. (Chamim, 2012)



**Gambar 2.1** jenis-jenis Mikrokontroler

(Sumber: <https://en.wikipedia.org/wiki/Microcontroller>)

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas. (Chamim, 2012)

## 2.5 WIFI Module ESP8266

ESP8266 merupakan modul WiFi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan WiFi dan membuat koneksi TCP/IP. (Chamim, 2012)

Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode WiFi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler. (Chamim, 2012)

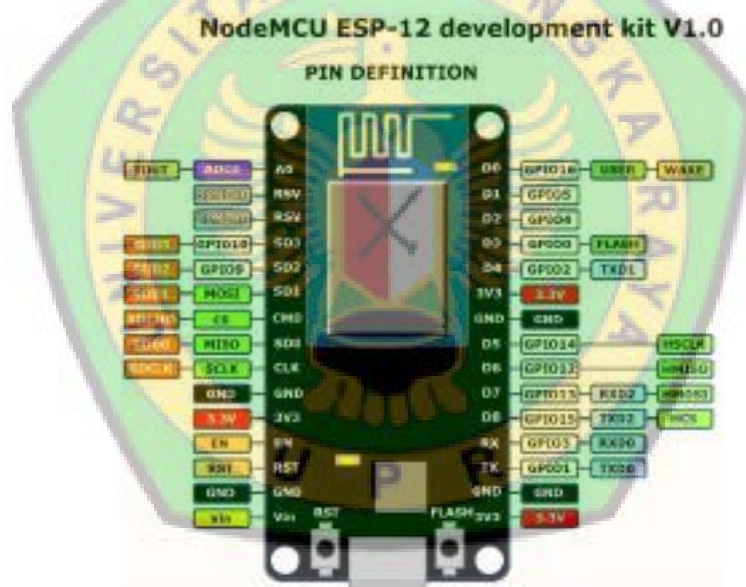


**Gambar 2.2** Jenis dan generasi WiFi module ESP8266

(Sumber : <http://arduino-esp8266.readthedocs.io/en/>)

### 2.5.1 NodeMCU

NodeMCU adalah platform IoT open source. Ini termasuk firmware yang berjalan pada ESP8266 Wi-Fi SoC dari Sistem Espressif, dan perangkat keras yang didasarkan pada modul ESP-12E. Istilah "NodeMCU" secara default mengacu pada firmware daripada perangkat pengembangan. Firmware menggunakan bahasa scripting Lua . Lua yang dalam bahasa indonesia berarti bulan adalah bahasa pemrograman dinamis yang berbasis skrip. Sebagai bahasa skrip, Lua memiliki API dalam bahasa C yang relatif lebih sederhana dibandingkan bahasa skrip lainnya. Ini didasarkan pada proyek eLua, dan dibangun di atas Espressif Non-OS SDK untuk ESP8266. Ini menggunakan banyak proyek sumber terbuka, seperti lua-cjson, dan spiffs. (Chamim, 2012)



**Gambar 2.3** Interface ESP8266 NodeMcu ESP-12E

(Sumber : [https://s3-ap-southeast-1.amazonaws.com/iotmaker-image-sync/9.nodemcu/images/nodemcudevkit\\_v1-0\\_io.jpg](https://s3-ap-southeast-1.amazonaws.com/iotmaker-image-sync/9.nodemcu/images/nodemcudevkit_v1-0_io.jpg))

Sebagai Arduino.cc mulai mengembangkan papan MCU baru berdasarkan pada prosesor non-AVR seperti ARM / SAM MCU dan digunakan dalam Arduino Due, mereka perlu memodifikasi Arduino IDE sehingga akan

relatif mudah untuk mengubah IDE untuk mendukung alat alternatif rantai untuk memungkinkan Arduino C/C++ untuk dikompilasi ke prosesor baru ini. Mereka melakukan ini dengan memperkenalkan Board Manager dan SAM Core. "Inti" adalah kumpulan komponen perangkat lunak yang diperlukan oleh Manajer Dewan dan Arduino IDE untuk mengkompilasi file sumber Arduino C/C++ ke bahasa mesin MCU sasaran. Beberapa penggemar kreatif ESP8266 telah mengembangkan inti Arduino untuk ESP8266 WiFi SoC yang tersedia di halaman web GitHub ESP8266 Core. Inilah yang populer disebut "ESP8266 Core untuk Arduino IDE". (Chamim, 2012)

### 2.5.2 MicroPython

Chip ESP8266 adalah platform hebat yang juga untuk menggunakan bahasa pemrograman MicroPython. Tutorial ini akan memandu Anda melalui pengaturan MicroPython, mendapatkan prompt, menggunakan WebREPL, menghubungkan ke jaringan dan berkomunikasi dengan Internet, menggunakan periferal perangkat keras, dan mengendalikan beberapa komponen eksternal. (Damien George, 2016)

Perangkat lunak MicroPython mendukung chip ESP8266 itu sendiri dan setiap papan harus berfungsi. Karakteristik utama dari sebuah papan adalah berapa banyak flash yang dimilikinya, bagaimana pin GPIO terhubung ke dunia luar, dan apakah itu termasuk konverter serial USB bawaan untuk membuat UART tersedia untuk PC yang di gunakan. Persyaratan minimum untuk ukuran flash adalah 1Mbyte. Ada juga build khusus untuk board dengan 512KB, tetapi sangat terbatas dibandingkan dengan build normal: tidak ada dukungan untuk filesystem, dan dengan demikian fitur yang bergantung padanya tidak akan berfungsi (WebREPL, upip, dll.). Dengan demikian, build 512KB akan lebih menarik bagi pengguna yang membangun dari source dan parameter fine-tune untuk aplikasi khusus. (Damien George, 2016)

### 2.5.3 AT Command

ESP8266 WiFi Module adalah chip terintegrasi yang dirancang untuk berkomunikasi ke dunia internet melalui sinyal radio WiFi. Ini adalah hubungan antara proyek dan internet. Ini memiliki pemrosesan on-board dan memori yang memungkinkan untuk diintegrasikan dengan elektronik melalui GPIO-nya. ESP8266 dapat digunakan untuk menghubungkan proyek Arduino Anda ke internet. Dalam hal ini, pemula akan memulai dengan dasar-dasar ESP8266 dan cara memprogramnya menggunakan perintah AT.

### 2.6 Arduino IDE

Untuk memprogram board Arduino, kita membutuhkan aplikasi IDE (Integrated Development Environment) bawaan dari Arduino. Aplikasi tersebut bisa di download di official website Arduino. Aplikasi ini berguna sebagai text editor untuk membuat, membuka, mengedit, dan juga mevalidasi kode serta untuk di upload ke board Arduino, Program yang digunakan pada Arduino disebut dengan istilah "sketch" yaitu file source code arduino dengan ekstensi .ino.



**Gambar 2.4** Tampilan Arduino IDE

(Sumber : <https://learn.adafruit.com/add-boards-arduino-v164/setup>)

Seperti teks editor pada umumnya yaitu memiliki fitur untuk cut / paste dan untuk find / replace teks. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengeksport dan juga sebagai tempat menampilkan kesalahan. Konsol

log menampilkan output teks dari Arduino Software (IDE), termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah jendela menampilkan papan dikonfigurasi dan port serial. Tombol toolbar memungkinkan Anda untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial.



**Gambar 2.5** Tampilan dashboard Arduino IDE

(Sumber : <https://www.microsoft.com/en-us/p/arduino-ide/9nblggh4rsd8?activetab=pivot:overviewtab>)

### 1) Verify

Pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi di-upload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk di-upload ke mikrokontroler.

**2) Upload**

Tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan dcompile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.

**3) New Sketch**

Membuka window dan membuat sketch baru.

**4) Open Sketch**

Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino.

**5) Save Sketch**

Menyimpan sketch, tapi tidak disertai dengan mengcompile.

**6) Serial Monitor**

Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.

**7) Keterangan Aplikasi**

Keterangan Aplikasi pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "Compiling" dan "Done Uploading" ketika kita mengcompile dan mengupload sketch ke board Arduino.

**8) Konsul Log**

Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika

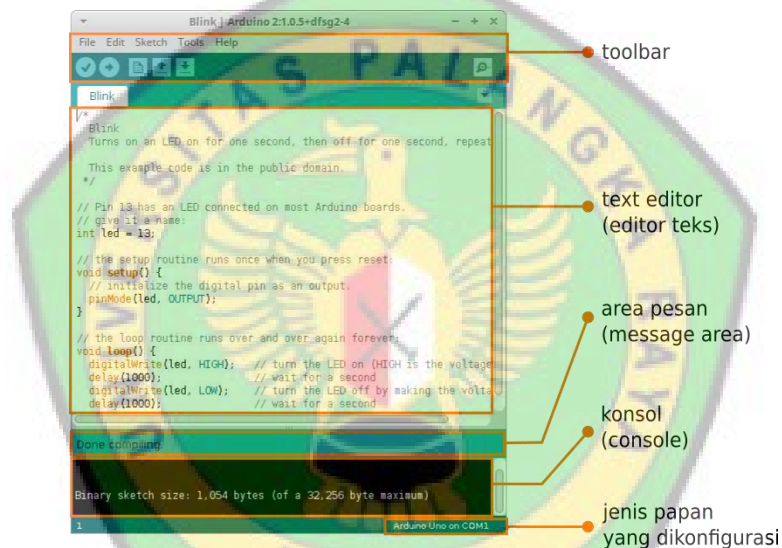
ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.

### 9) Baris Sketch

Bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.

### 10) Informasi Board dan Port

Bagian ini menginformasikan port yang dipakai oleh board Arduino.



**Gambar 2.6** Keterangan pada Arduino IDE

(Sumber : <https://dedyqoirudin.wordpress.com/2017/02/23/belajar-arduino-3-arduino-software-ide/>)

## 2.7 Serial Komunikasi

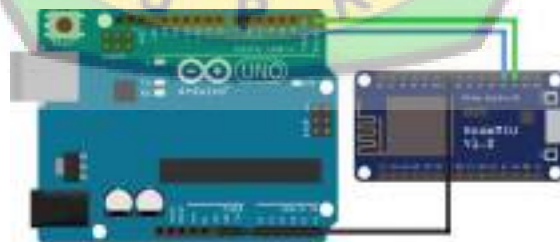
Komunikasi serial adalah salah satu metode komunikasi data dimana proses pengiriman data dilakukan per satu bit data secara berurutan dan bergantian yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Fungsinya digunakan untuk komunikasi data antara Arduino dengan komputer atau perangkat lain.

1.	Baudrate	Kecepatan transfer data dinyatakan dalam satuan bps (bit per second)
2.	Data Bit	Jumlah bit yang dikirimkan dalam 1 frame data (5bit – 9 bit)
3.	Parity Bit	Bit parity digunakan dalam pengecekan error untuk memastikan kesalahan yang mungkin terjadi selama transmisi data
4.	Start Bit	Bit yang digunakan sebagai indikator awal sebuah frame data (selalu Low)
5.	Stop Bit	Bit yang digunakan sebagai indikator akhir sebuah frame data (selalu high)

**Gambar 2.7** Istilah Pada Serial Komunikasi

(Sumber : [https://www.academia.edu/13193411/Pengiriman\\_Frame\\_Data\\_Pada\\_Komunikasi\\_Serial](https://www.academia.edu/13193411/Pengiriman_Frame_Data_Pada_Komunikasi_Serial))

Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai  $n = 1$ , atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya di mana  $n$ -bit data dikirimkan bersamaan.



**Gambar 2.8** Ilustrasi Pengiriman Data Komunikasi Serial

(Sumber : <https://iot-guider.com/esp8266-nodemcu/serial-communication-between-nodemcu-and-arduino/>)

## 2.8 Protokol Hypertext Transfer Protocol (HTTP)

*Hypertext Transfer Protocol (HTTP)* adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia.

Protokol HTTP didefinisikan oleh Tim *Berners-Lee* dalam RFC 1945 versi 1.0 dan digunakan sejak tahun 1990. Penyempurnaan protokol HTTP menjadi versi 1.1 yang dispesifikasikan oleh IETF dengan RFC 2616. HTTP bersifat *request – response*, yaitu HTTP *client* (*user* agen misalnya) mengirimkan permintaan (*request*) ke HTTP *server* dan *server* merespon sesuai *request* tersebut. *User* agen sebagai contoh adalah *Mozilla*, *Netscape*, *Google Chrome*, atau *browser berbasis* teks contohnya *Lynx* atau *links* dan sebagainya.

Pada protokol HTTP terdapat 3 jenis hubungan dengan perantara *proxy*, *gateway*, dan *tunnel*. *Proxy* bertindak sebagai agent penerus, menerima *request* dalam bentuk *Uniform Resource Identifier (URI)* absolut, mengubah format *request* dan mengirimkan *request* ke *server* yang ditunjukkan oleh URI. *Gateway* bertindak sebagai agen penerima dan menterjemahkan *request* ke protokol *server* yang dilayaninya. *Tunnel* bertindak sebagai titik *Relay* antara dua hubungan HTTP tanpa mengubah request dan response HTTP. *Tunnel* digunakan jika komunikasi perlu melalui sebuah perantara dan perantara tersebut tidak mengetahui isi pesan dalam hubungan tersebut.

Perbedaan mendasar antara HTTP/1.1 dengan HTTP/1.0 adalah penggunaan hubungan *persistent*. HTTP/1.0 membukasatu koneksi untuk tiap permintaan satu URI, sedangkan HTTP/1.1 dapat menggunakan sebuah koneksi TCP untuk beberapa permintaan URI (*persistent*) (*header Connection : keepAlive*), kecuali jika *client* menyatakan tidak hendak menggunakan hubungan *persistent* (*header Connection : close*). HTTP port TCP *default* adalah 80, namun itu bisa diganti dengan nomor TCP lain diantara 1023 – 65535.

## 2.9 PHP

Menurut Eko Prasetyo (2008, h. 19), php adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML. sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi sifat dinamis. sifat *server-side* berarti pengerjaan skrip dilakukan di server, baru kemudian hasilnya dikirimkan ke *browser*.

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di server.

## 2.10 XAMPP

Menurut Bunafit Nugroho (2008, h. 2), XAMPP merupakan paket php berbasis *open source* yang dikembangkan oleh sebuah komunitas *open source*. dengan menggunakan XAMPP, tidak usah lagi bingung untuk melakukan penginstalan program lain, karena semua kebutuhan telah disediakan oleh XAMPP.

XAMPP adalah perangkat lunak bebas mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa perangkat lunak. XAMPP dikembangkan oleh sebuah tim proyek bernama *Apache Friends*.

Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan

singkatan dari X berarti mendukung 4 sistem operasi, Apache, MySQL, PHP dan Perl.

### **2.11 Localhost**

Localhost merupakan istilah dalam komputer jaringan yang berarti komputer ini. localhost adalah nama standar yang diberikan sebagai alamat loopback network interface. Localhost selalu menerjemahkan loopback ip address 127.0.0.1 dalam IPv4, atau dalam IPv6. Localhost digunakan untuk mengantarkan web browser pada HTTP server yang terinstall di komputer lokal. Alamat `http://localhost` akan menampilkan website lokal pada komputer yang bersangkutan.

Jadi, ketika komputer 1 sebagai local server dan lainnya sebagai client, kemudian menghostingkan web didalamnya (localhost) untuk dijadikan tempat membangun website sementara dan kemudian dihostingkan secara online ke internet.

Dengan demikian menjadikan komputer kita sebagai local server, kita dapat bekerja secara offline tanpa harus takut menghadapi masalah biaya, waktu, dan kenyamanan.

### **2.12 Penentuan Kualitas Udara**

Penentuan kualitas udara ambien dengan parameter NO<sub>2</sub>, O<sub>3</sub>, CO, dan PM<sub>2.5</sub> menggunakan perhitungan kualitas udara yang merupakan bagian dari perhitungan Badan Pengendalian Dampak Lingkungan (BAPEDAL). BAPEDAL merupakan gambaran atau indikasi awal yang memberikan kesimpulan cepat dari suatu kondisi lingkungan hidup pada lingkup dan periode tertentu. Pada rumusan BAPEDAL yang dipublikasikan oleh BAPEDAL sejak tahun 1997, parameter yang digunakan untuk kualitas udara hanya SO<sub>2</sub> dan NO<sub>2</sub>. Penelitian ini bertujuan untuk melengkapi parameter kualitas udara yang digunakan untuk perhitungan BAPEDAL dengan

menambahkan parameter dan PM2.5. Idealnya ada 5 parameter yaitu SO<sub>2</sub>, NO<sub>2</sub>, CO, PM2.5, dan O<sub>3</sub> yang mewakili perhitungan kualitas udara untuk BAPEDAL. PM2.5 merupakan pencemar utama yang memberi dampak besar terhadap kesehatan manusia. Dengan data dari penelitian ini diharapkan dapat melengkapi parameter Indeks Kualitas Udara (IKU). Hal ini menunjukkan bahwa idealnya ada 5 parameter yang berperan dalam menentukan kriteria kualitas udara. (Diah Dwiana Lestiani, 2016)



## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi Penelitian**

##### **1. Metode Pengumpulan Data**

Merupakan tahap pengambilan data atau sampel yang berhubungan dengan permasalahan yang sedang dibahas. Dalam pengumpulan data tersebut menggunakan teknik-teknik pengumpulan data, yaitu :

a) **Observasi**

Merupakan peninjauan ke lokasi yang akan diteliti, sehingga mendapatkan data sementara guna untuk menguraikan permasalahan yang ada.

b) **Studi Literatur**

Mempelajari buku dan informasi dari internet yang berhubungan dengan proses pembuatan sistem. Melakukan pengamatan terhadap berbagai sistem teknologi IoT yang sejenis untuk melengkapi bahan referensi.

c) **Studi Konsultasi dan Diskusi**

Berupa tanya jawab dengan dosen pembimbing ataupun orang-orang yang berpengalaman dalam pembuatan sistem tersebut.

##### **2. Metode Analisis dan Perancangan**

Metode analisis melalui tahap alat dan bahan apa saja yang di butuhkan dari alat perangkat keras mikrokontroller dan sensorik kualitas udara maupun bahannya adalah perangkat lunak untuk pemrograman mikrokontroller serta pemrograman website sebagai monitoring indeks kualitas udara.

Kemudian metode perancangan meliputi perangkat lunak dan perangkat keras yang dibuat untuk purwarupa sistem monitoring indeks kualitas udara berdasarkan hasil analisis kebutuhan.

### **3. Metode Implementasi**

Semua perangkat keras yang telah dirancang dan direalisasikan kemudian disatukan menjadi sebuah sistem yang terintegrasi secara utuh. mikrokontroller yang mengendalikan perangkat sensorik disatukan menggunakan casing box.

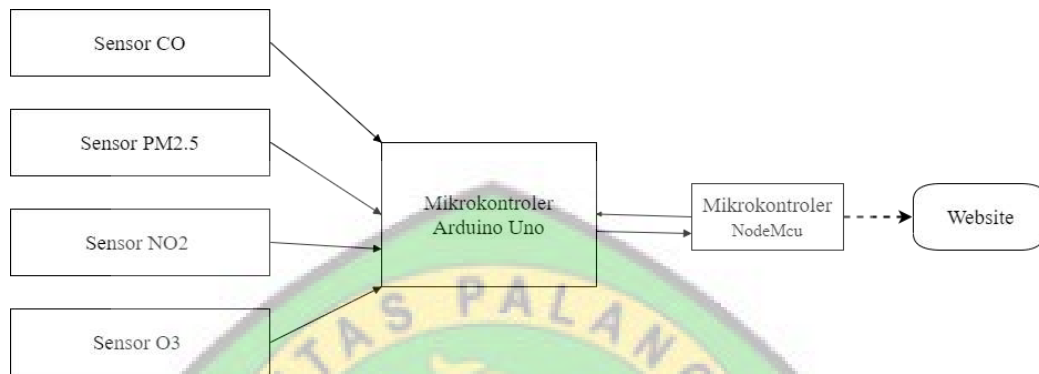
Implementasi Perangkat Lunak ini membahas mengenai instalasi firmware untuk development board mikrokontroller melalui Arduino IDE, implementasi koneksi web server dengan kontroler.

### **4. Metode Pengujian**

Metode uji coba yang memfokuskan pada keperluan software. Karena itu, uji coba black box memungkinkan pengembangan software untuk membuat himpunan kondisi input yang melatih seluruh syarat-syarat fungsional suatu program. Metode pengujian blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya fungsi-fungsi yang salah atau hilang. Untuk mengetahui kesalahan yang terjadi pada tahap perancangan dan juga dilakukan analisa untuk melakukan perbaikan dan pengembangan untuk bisa mendekati sesuai dengan kebutuhan yang di harapkan berdasarkan fungsinya.

### 3.2 Desain Arsitektur Sistem

Desain arsitektur sistem merupakan suatu rencana atau pemetaan kebutuhan-kebutuhan di dalam suatu organisasi dengan melibatkan seluruh komponen yang ada di dalamnya.



**Gambar 3.1** Desain arsitektur sistem

Terdapat dua unit mikrokontroler yang akan digunakan pada penelitian ini. Masing-masing unit tersebut akan mengirimkan nilai data 4 sensor ke web server dengan menggunakan media wireless.

Antara Arduino dengan NodeMCU dihubungkan dengan menggunakan port serial komunikasi masing-masing.

### 3.3 Alat dan Bahan

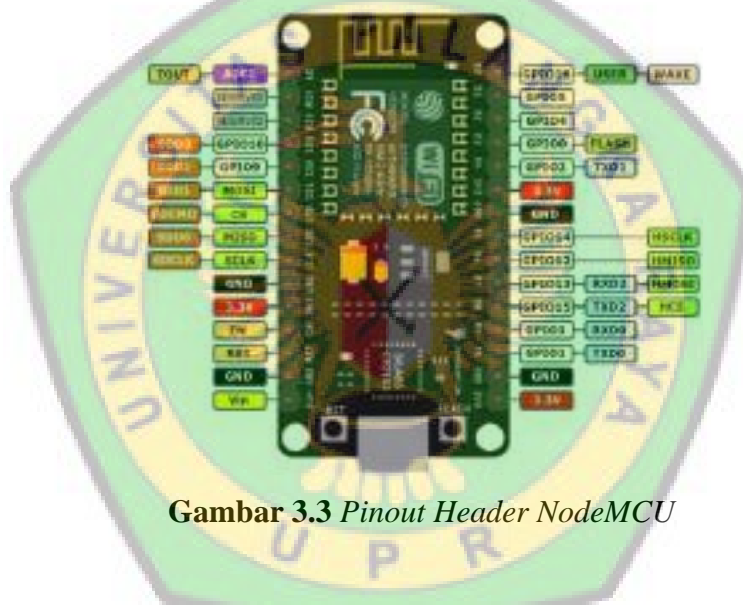
Alat dan bahan yang digunakan dalam penelitian ini meliputi peralatan yang digunakan untuk pengambilan data maupun perlengkapan pendukung lainnya yang digunakan untuk mendukung penelitian ini, yaitu sebagai berikut:

#### 3.3.1. Mikrokontroler ESP8266

Mikrokontroler ESP8266 adalah sebuah modul wifi serbaguna yang bersifat SoC (System on Chip), sehingga dapat melakukan programming langsung ke dalam ESP8266 tanpa memerlukan mikrokontroler tambahan. Kelebihan lainnya adalah ESP8266 ini dapat menjalankan peran sebagai akses poin maupun klien sekaligus.



**Gambar 3.2** Mikrokontroler ESP8266 jenis NodeMCU Ver.1.0



**Gambar 3.3** Pinout Header NodeMCU

### 3.3.2. Sensor Karbon Monoksida (MQ-7)

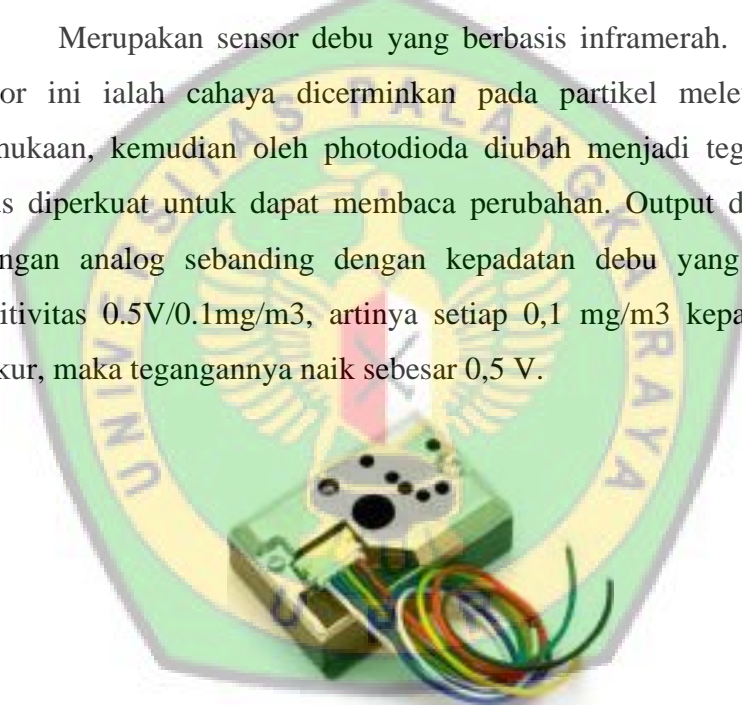
MQ7 adalah sensor Karbon Monoksida (CO) sederhana yang digunakan untuk mendeteksi konsentrasi CO di udara. Dapat mendeteksi konsentrasi CO-gas di mana saja dari 20 hingga 2000ppm. Sensitivitas dapat disesuaikan oleh potensiometer.



**Gambar 3.4** *Sensor MQ-7*

### 3.3.3. *Sensor PM2.5 (Dust Sensor)*

Merupakan sensor debu yang berbasis inframerah. Prinsip kerja dari sensor ini ialah cahaya dicerminkan pada partikel melewati keseluruhan permukaan, kemudian oleh photodiode diubah menjadi tegangan. Tegangan harus diperkuat untuk dapat membaca perubahan. Output dari sensor adalah tegangan analog sebanding dengan kepadatan debu yang terukur, dengan sensitivitas  $0.5V/0.1mg/m^3$ , artinya setiap  $0,1 mg/m^3$  kepadatan debu yang terukur, maka tegangannya naik sebesar  $0,5 V$ .



**Gambar 3.5** *GP2Y1010AU0F Optical Dust Sensor*

### 3.3.4. Sensor O<sub>3</sub> (MQ-131)

Sensor gas MQ131 berfungsi untuk mengukur besaran konsentrasi gas O<sub>3</sub>. Gas O<sub>3</sub> yang terdeteksi oleh sensor MQ131 dikonversi menjadi tegangan listrik yang berupa sinyal analog.



**Gambar 3.6** *Sensor O<sub>3</sub> (MQ-131)*

### 3.3.5. Sensor Nitrogen Dioksida (MQ-135)

MQ-135 Sensor adalah sensor yang memonitor kualitas udara untuk mendeteksi gas Nitrogen Dioksida.

Sensor ini melaporkan hasil deteksi kualitas udara berupa perubahan nilai resistensi analog di pin keluarannya. Pin keluaran ini bisa disambungkan dengan pin analog input Arduino.



**Gambar 3.7** *Sensor O<sub>3</sub> (MQ-135)*

### 3.3.6. Mikrokontroler Arduino Uno

Arduino Uno adalah sebuah board mikrokontroler yang menggunakan AVR Atmega328 sebagai processornya. Arduino Uno memiliki 14 pin digital input dan output, 6 pin input analog, 16 MHz ceramic resonator, sebuah koneksi Universal Serial Bus, sebuah Power Jack, sebuah ICSP header, dan tombol reset. Arduino uno dapat di hubungkan ke komputer melalui USB dan Arduino juga

dapat di jalankan dengan memberikan tegangan AC - to - DC menggunakan adapter atau battery. Arduino Uno menggunakan Atmega16U2 sebagai board processingnya.

ATmega328 menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX).



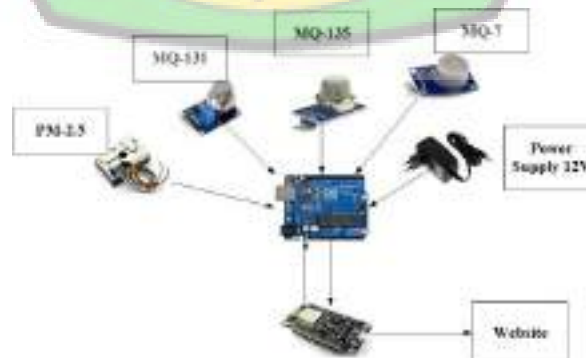
**Gambar 3.8** Board Arduino Uno

(Sumber : <https://www.tokopedia.com/muliakarunia/arduino-uno-r3-atmega328p-dip>)

### 3.4 Rangkaian Keseluruhan Alat

#### 3.4.1 Diagram Blok Sistem

Agar mudah dipahami, maka peneliti membuat diagram blok sistem dan berserta alur kerjanya untuk sistem monitoring kualitas udara pada gambar berikut ini.



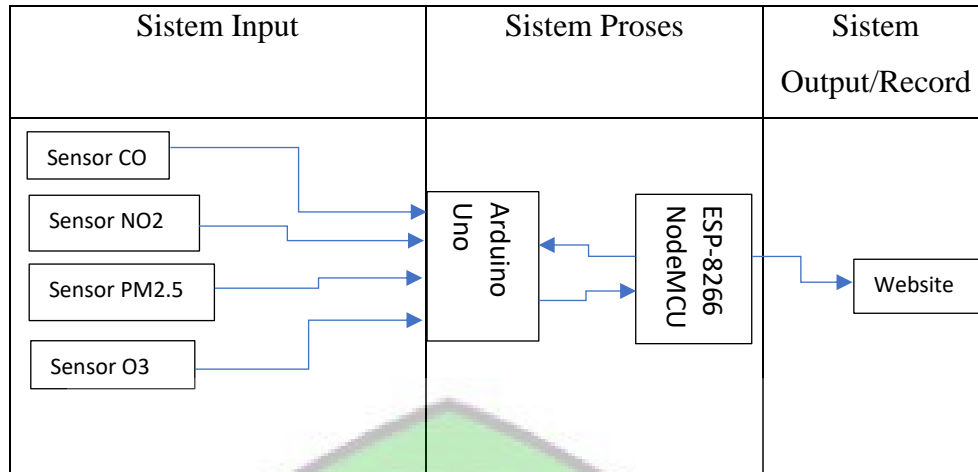
**Gambar 3.9** Diagram blok sistem

**Keterangan:**

1. Rangkaian Arduino uno sebagai mikrokontroler yaitu berfungsi untuk menerima, data dari sensor, lalu NodeMCU meminta data dari Arduino uno kemudian NodeMCU mengirim data ke website yang akan tersimpan di database melalui jaringan internet dan dapat dimonitoring secara *realtime*.
2. Sensor CO, O3, NO2 dan PM2.5 berfungsi sebagai input analog yaitu untuk mengirim data gas kepada Arduino Uno lalu data tersebut dikirim ke NodeMCU yang nantinya akan diolah.
3. Website berfungsi sebagai tempat monitoring dan menyimpan data yang diterima dari NodeMCU, data yang sudah di upload akan secara otomatis tersimpan di database melalui jaringan internet.
4. Power Supply digunakan untuk pemberi sumber tegangan arus listrik 12v ke NodeMCU.

### **3.5 Cara Kerja Alat**

Cara kerja dari alat monitoring kualitas udara berbasis IoT ini dapat dibagi atas 3 (tiga) bagian. Bagian pertama adalah sistem input, dimana sistem ini merupakan langkah awal dari kerja alat, kemudian sistem proses yang berkerja memproses sinyal yang telah diterima dari sistem input untuk di keluarkan pada bagian ketiga yaitu sistem output dan sistem record.



**Gambar 3.10** Rangkaian sensor

### 3.5.1 Sistem Input

Pada sistem input monitoring kualitas udara ini menggunakan sensor CO (MQ-7), NO<sub>2</sub> (MQ-135), PM-2.5 (GP2Y1014AU0F), O<sub>3</sub> (MQ-131) akan memberikan informasi yang diterima berupa sinyal analog, kemudian informasi akan dikirim ke NodeMCU untuk diproses.

### 3.5.2 Sistem Process

Pada sistem proses ini menggunakan Arduino Uno dan NodeMCU yang merupakan otak dari monitoring kualitas udara dan bertugas untuk mengeluarkan output atas input yang diterimanya dari Arduino Uno dan NodeMCU berdasarkan program yang telah disimpan. Setelah diolah menjadi data, NodeMCU akan mengambil keputusan untuk melakukan perintah.

### 3.5.3 Sistem Output/Record

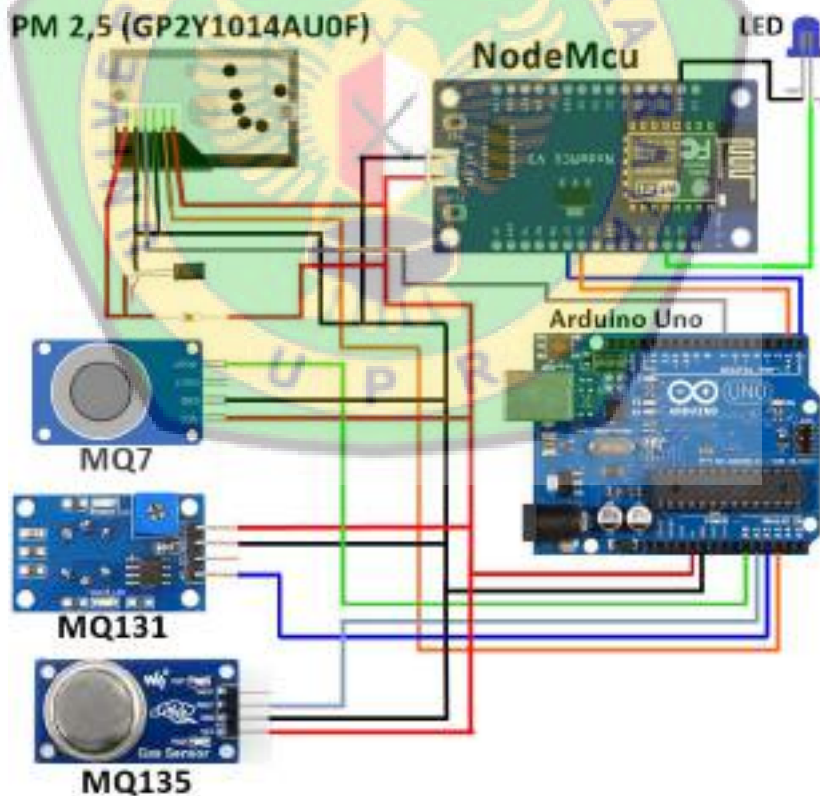
Sistem output pada sistem ini adalah dalam bentuk angka atau nilai yang di terima sensor dan sistem record, pada alat ini menggunakan Website yaitu sebagai tempat menyimpan data yang diterima dari Arduino Uno dan NodeMCU, data yang sudah di upload akan secara otomatis melalu jaringan internet.

### 3.6 Pembuatan Alat

Pada perancangan ini akan dibahas mengenai perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Dari kedua pembahasan perancangan ini dianggap penting untuk dibahas karena ingin menghasilkan sistem yang baik, serta menghasilkan sinkronisasi antara perangkat keras (*hardware*) dengan perangkat lunak (*software*).

#### 3.6.1 Rangkaian Skematik Seluruh Sensor

Rangkaian Skematik merupakan suatu rangkaian elektronika yang menggambarkan suatu rangkaian dengan menggunakan simbol-simbol listrik. Dalam skematik diagram simbol-simbol listrik tersebut di hubungkan dengan garis yang menggambarkan koneksi dan hubungan dari komponen listrik di dalam rangkaian.



**Gambar 3.11** Tampilan desain rangkaian skematik sensor

### 3.6.2 Proses Pengerjaan Perangkat Keras

Berikut ini akan dijelaskan tahapan-tahapan proses pembuatan dan perangkaian alat sesuai dengan prosedur dan desain rangkaian yang sudah di kerjakan.

#### 3.6.2.1 Proses Pengerjaan Kotak

Pada bagian ini, proses pengerjaan kotak dilapisi dengan papan PCB lobang dengan tujuan sebagai dasar untuk memberikan permukaan yang datar dan memudahkan untuk menempatkan seluruh alat di dalam nya dan di kunci dengan baut kecil.



**Gambar 3.12** *Proses pengerjaan kotak pertama*

### 3.6.2.2 Proses Perangkaian alat dalam Kotak

Proses ini menggabungkan komponen alat bersama dengan mikrokontroler sesuai dengan desain rangkaian. Setiap proses dilakukan pengecekan arus tegangan dan kualitas kabel jumper yang di gunakan.



**Gambar 3.13** *Proses perangkaian alat dalam kotak*

### 3.6.2.3 Proses Perangkaian Sensor

Pada proses ini, masing-masing sensor akan di tempatkan pada bagian bawah mangkok dan diikat dengan tali ties serta menggunakan 3 buah besi kecil sebagai penopang agar ada jarak antara sensor satu dengan yang lainnya.



**Gambar 3.14** *Proses perangkaian sensor*

### 3.6.2.4 Proses Pemasangan Alat pada Tiang Besi

Pada bagian ini, proses yang di kerjakan adalah menyatukan komponen pada kotak dan juga sensor dalam mangkok untuk diikatkan pada tiang besi.

**Gambar 3.15** *Proses pemasangan alat pada tiang besi*

### 3.6.3 Perancangan Perangkat Lunak (Software)

#### 3.6.3.1 Perancangan Perangkat Lunak (Software)

Perangkat lunak yang digunakan adalah software Arduino IDE versi 1.8.13 dimana versi ini merupakan versi terbaru. Software ini merupakan software yang kompatibel dengan device Arduino Uno dan NodeMCU. Karena NodeMCU ini merupakan device pengontrolan yang menggunakan platform berbasis Arduino, maka NodeMCU juga kompatibel dengan Software Arduino IDE ini. Software Arduino IDE digunakan untuk menuliskan source code program-program alat yang nantinya akan di upload kedalam Arduino Uno dan NodeMCU.



**Gambar 3.16** *Tampilan aplikasi Arduino Ide*

Pada mulanya board NodeMCU ini tidak terdaftar dalam software Arduino. Maka dari pada itu untuk menampilkan Board NodeMCU pada software Arduino kita harus menambah Library Board pada software Arduino ini. Sebelumnya kita harus terkoneksi internet terlebih dahulu. Untuk menambah library board kita harus memasukan link khusus pada Preferences untuk menampilkan hasil pencarian library. Caranya klik file lalu pilih dan klik Preferences dan akan muncul tampilan seperti berikut.



**Gambar 3.17** *Tampilan Preferences*

Lalu setelah itu kita tambahkan link [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) dengan paste link Additional Board Manager URLs dan klik Ok. Setelah memasukan link selanjutnya kita pilih dan klik menu Tools pilih Board dan klik Boards Manager dan akan tampil tampilan berikut ini



**Gambar 3.18** *Tampilan Board Manager*

Dan kemudian kita cari NodeMCU. dan klik install tunggu beberapa saat sampai proses instalasi selesai.



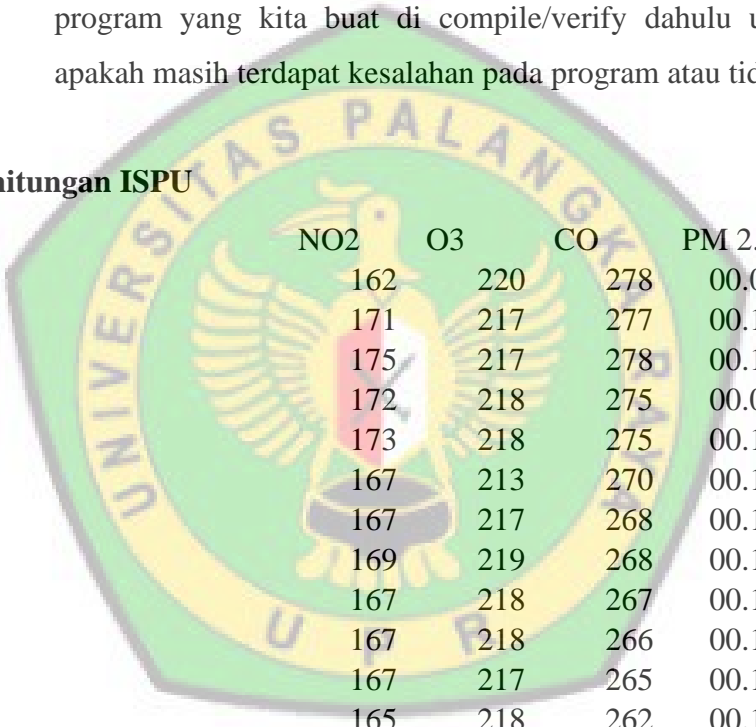
**Gambar 3.19** *Tampilan Menunggu Hasil Instalasi*

Jika sudah selesai kita dapat pilih Board NodeMCU dari menu Tools. Dan NodeMCU sudah dapat terhubung dengan Software Arduino IDE untuk memulai memprogram.

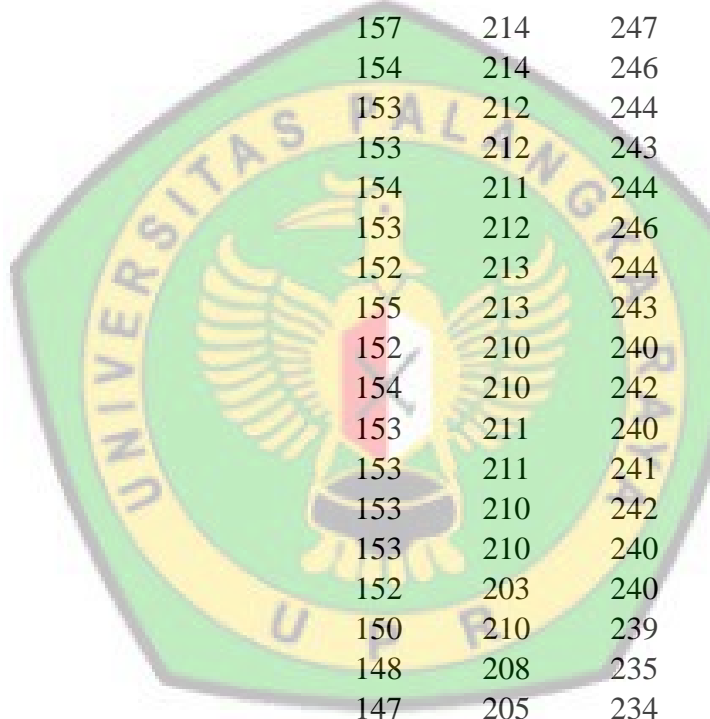
### 3.6.3.2 Upload Program

Setelah program selesai dibuat tahap selanjutnya adalah mengupload program ke dalam NodeMCU. Sebelum di upload biasanya program yang kita buat di compile/verify dahulu untuk mengecek apakah masih terdapat kesalahan pada program atau tidak.

## 3.7 Perhitungan ISPU



	NO2	O3	CO	PM 2.5
	162	220	278	00.01
	171	217	277	00.11
	175	217	278	00.11
	172	218	275	00.09
	173	218	275	00.14
	167	213	270	00.13
	167	217	268	00.11
	169	219	268	00.12
	167	218	267	00.12
	167	218	266	00.12
	167	217	265	00.13
	165	218	262	00.12
	165	218	262	00.13
	165	219	261	00.13
	166	220	261	00.14
	165	218	259	00.14
	162	219	258	00.14
	164	220	258	00.12
	163	219	256	00.13
	159	216	254	00.13
	159	217	252	00.15
	153	212	243	00.12
	156	213	244	00.12



160	214	245	00.01	
155	214	245	00.13	
155	213	245	00.11	
150	210	243	00.13	
150	208	242	00.13	
151	209	241	00.13	
150	209	242	00.01	
157	213	252	00.13	
155	213	248	00.13	
153	211	245	00.15	
157	214	247	00.13	
154	214	246	00.11	
153	212	244	00.13	
153	212	243	00.12	
154	211	244	00.11	
153	212	246	00.13	
152	213	244	00.12	
155	213	243	00.13	
152	210	240	00.13	
154	210	242	00.12	
153	211	240	00.12	
153	211	241	00.14	
153	210	242	00.14	
153	210	240	00.13	
152	203	240	00.12	
150	210	239	00.11	
148	208	235	00.13	
147	205	234	00.13	
148	205	234	00.13	
146	205	234	00.11	
146	206	233	00.12	
146	206	230	00.11	
145	204	231	00.12	
143	206	232	00.13	
144	203	228	00.12	
146	204	232	00.13	
<b>Nilai rata-rata</b>	<b>308</b>	<b>424</b>	<b>510</b>	<b>00.14</b>

ISPU	PM 2.5 ( $\mu\text{g}/\text{m}^3$ )	CO ( $\mu\text{g}/\text{m}^3$ )	O <sub>3</sub> ( $\mu\text{g}/\text{m}^3$ )	NO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )
50	50	5	120	
100	150	10	235	
200	350	17	400	1130
300	420	34	800	2260
400	500	46	1000	3000
500	600	57,5	1200	3750

$$I = \frac{I_a - I_b}{X_a - X_b} (X_x - X_b) + I_b$$

O<sub>3</sub>

Jenis Sensor	Konsentrasi Data Sensor	Konversi Data dengan ISPU	Hasil
O <sub>3</sub>	107.66 ppm	-	TIDAK SEHAT
NO <sub>2</sub>	198.78 ppm	-	TIDAK SEHAT
CO	38.96 ppm	341 ppm	BERBAHAYA

### 3.8 Konversi nilai ke PPM

```

1 #define pinSensor A0 // mendefinisikan bahwa pin yang digunakan
2 // untuk membaca sensor adalah pin A0
3
4 void setup()
5 {
6   Serial.begin(9600);
7 }
8
9 long RL = 10000; // 10000 Ohm
10
11 void loop()
12 {
13   int sensorValue = analogRead(pinSensor); // membaca nilai ADC dari sensor
14   float VRL = sensorValue*5.00/1024; // mengubah nilai ADC | 0 - 1023 | menjadi nilai voltase (0.0 - 5.00 Volt)
15   Serial.println("ADC : ");
16   Serial.println(VRL);
17   Serial.println(" VRL : ");
18
19   float Rs = ( 5.00 * RL / VRL ) - RL; // Rs = (5.00 / VRL) * RL - RL
20   Serial.println("Rs : ");
21   Serial.println(Rs);
22   Serial.println(" PPM : ");
23
24   Serial.println(" ");
25   delay(1000);
26 }

```

**Gambar 3.20** Tampilan code untuk mencari nilai  $R_s$

Code pada gambar 3.20 :

```
#define pinSensor A0 // mendefinisikan bahwa pin yang digunakan
// untuk membaca sensor adalah pin A0
```

```
void setup()
```

```
{
```

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

```
}
```

```
long RL = 10000; // 10000 Ohm
```

```
//long disini berfungsi untuk menampung bilangan bulat
```

```
void loop()
```

```
{
```

```

int sensorvalue = analogRead(pinSensor); // membaca nilai ADC dari sensor
float VRL= sensorvalue*5.00/1024; // mengubah nilai ADC ( 0 - 1023 ) menjadi nilai
voltase ( 0 - 5.00 volt )
Serial.print("VRL : ");
Serial.print(VRL);
Serial.println(" volt");

float Rs = ( 5.00 * RL / VRL ) - RL; //Rs = ( Vc * RL / V RL ) - RL
Serial.print("Rs : ");
Serial.print(Rs);
Serial.println(" Ohm");

Serial.println();
delay(1000);
}

```

Gambar 3.20 di atas menjelaskan bahwa pertama kali kita mencari nilai Rs nya dulu dengan rumus :

$$Rs = ( Vc * RL / V RL ) - RL$$

Dimana,

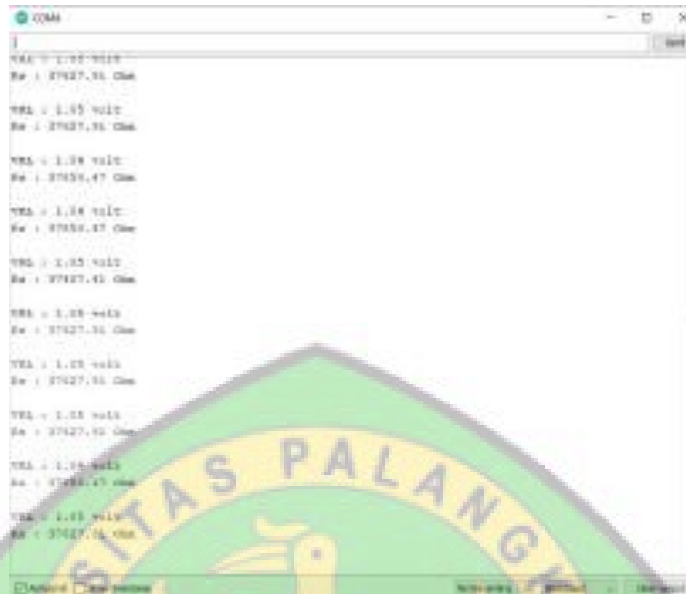
Rs = Tahanan pada sensor

Vc = Tegangan yang masuk ke Sensor

RL = Tahanan beban pada rangkaian

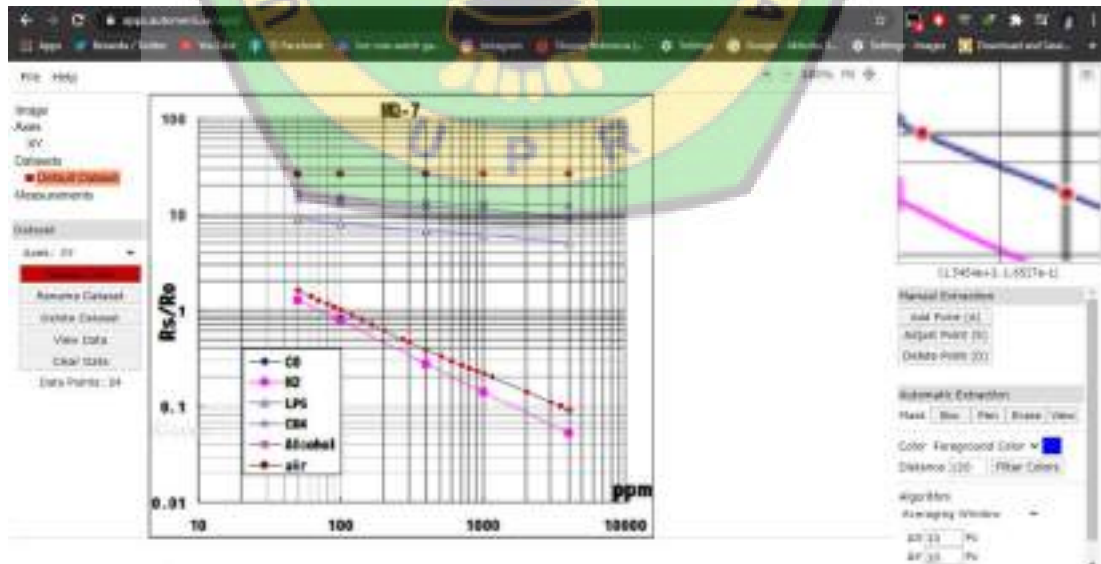
VRL = tegangan output rangkaian

Lalu didapatkan nilai Rs seperti gambar 3.21 di bawah.



Gambar 3.21 Tampilan serial monitor nilai Rs

Selanjutnya, saya perlu mencari nilai Ro menggunakan grafik dibawah ini.





**Gambar 3.22** Tampilan saat selesai melakukan penitikan grafik sensitivitas mq-7



**Gambar 3.23** Tampilan grafik hubungan rs/ro dengan ppm di excel dari grafik sensitivitas mq-7

Dimana :  $y$  (ppm) = 49.101

Berapa nilai  $x$  (Rs/Ro) ? Berikut perhitungannya :

$$y = 49.101 * (x^{-0.876})$$

$$10 = 49.101 * (x^{-0.876})$$

$$10 / 49.101 = x^{-0.876}$$

$$x^{-0.876} = 10/49.101$$

$$x^{-0.876} = 0.2$$

$$x = \sqrt[0.876]{0.2}$$

$$x = 6.279$$

$$R_s / R_o = 6.279$$

Untuk mencari  $R_o$ , masukan nilai  $R_s$  yang sudah kita dapatkan sebelumnya melalui coding pada gambar 3.21 tadi. Nilai  $R_s$ -nya 37627.91 ohm.

$$R_o = 37627.91 / 6.279$$

$$R_o = 5935 \text{ ohm}$$

Setelah kita tahu nilai  $R_o$  maka kita bisa mencari ppm dengan menggunakan grafik hubungan “ $y = 49.101 * (x - 0.876)$ ” tadi. Sehingga setelah dicoba masukan rumusnya ke code arduino menjadi seperti berikut :

```

NIRICO_PPM §
1 // Definisi pinSensor A0 // mendefinisikan pinnya pin yang digunakan
2 // untuk membaca sensor untuk nilai pin A0
3
4 void setup()
5 {
6   Serial.begin(9600);
7
8   long EL = 10000; // 10000 Ohm
9   long Rs = 6000; // 6000 ohm (SILANGAN BERSAMA)
10
11 void loop()
12 {
13   // Baca sensorvalue = analogRead(pinSensor); // membaca nilai ADC dari sensor
14   float VRL = sensorvalue * 5.0 / 1024; // mengubah nilai ADC (0-1023) menjadi nilai voltase (0 - 5.00 volt)
15   Serial.println("VRL = ");
16   Serial.println(VRL);
17   Serial.println(" Ra = ");
18
19   float Ra = ( 5.00 * EL / VRL ) - EL;
20   Serial.println("Ra = ");
21   Serial.println(Ra);
22   Serial.println(" Ra0 = ");
23
24   float ppm = 49.101 * ((Ra / Ra0) - 0.876); // ppm = 49.101 * ((Ra/Rs)-0.876)
25   Serial.println("PPM = ");
26   Serial.println(ppm);
27   Serial.println(" ppm = ");
28
29   Serial.println();
30   delay(1000);
31 }

```

**Gambar 3.24** Tampilan code untuk mencari nilai PPM

Code pada gambar 3.20 :

```
#define pinSensor A0 // mendefinisikan bahwa pin yang digunakan
                        // untuk membaca sensor adalah pin A0

void setup()
{
  Serial.begin(9600);
}

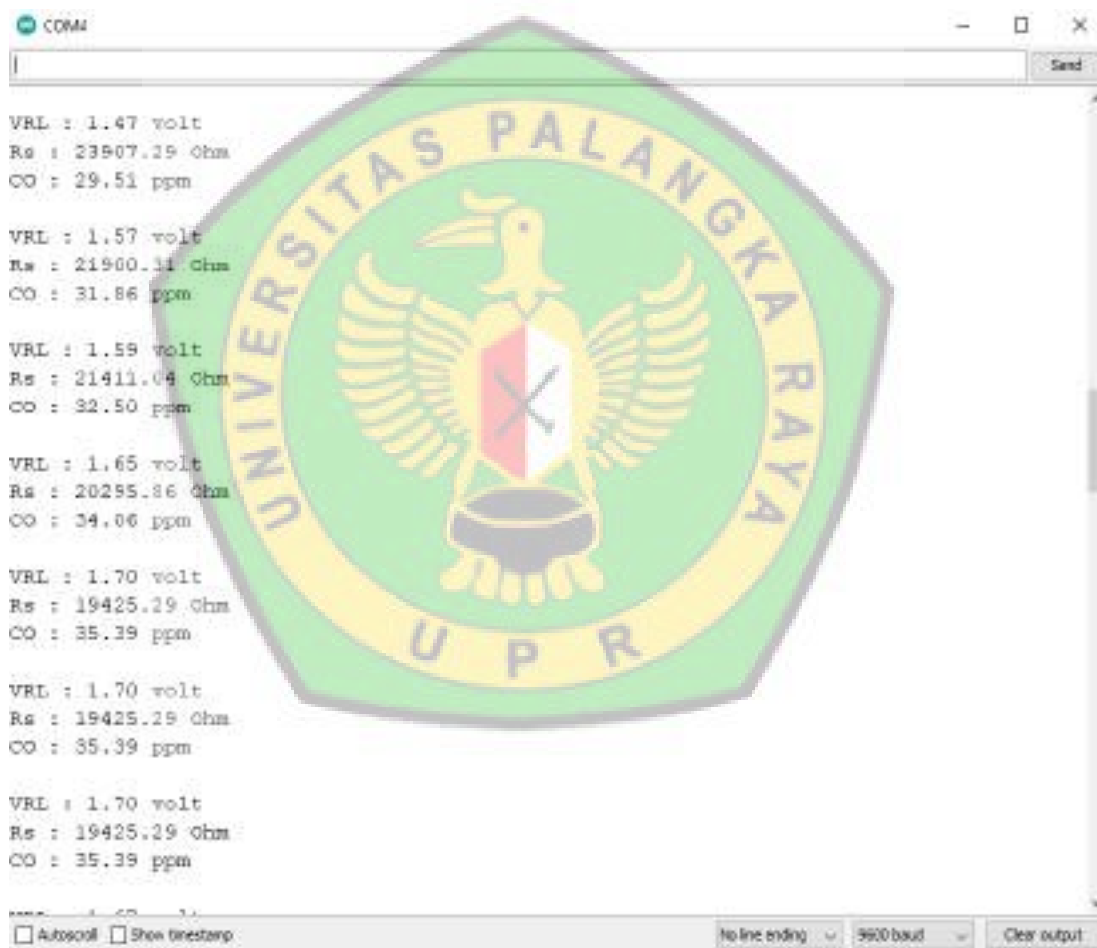
long RL = 10000; // 10000 Ohm
long Ro = 5935 ; // 5935 ohm ( SILAHKAN DISESUAIKAN)

void loop()
{
  int sensorvalue = analogRead(pinSensor); // membaca nilai ADC dari sensor
  float VRL= sensorvalue*5.00/1024; // mengubah nilai ADC ( 0 - 1023 ) menjadi
  nilai voltase ( 0 - 5.00 volt )
  Serial.print("VRL : ");
  Serial.print(VRL);
  Serial.println(" volt");

  float Rs = ( 5.00 * RL / VRL ) - RL;
  Serial.print("Rs : ");
  Serial.print(Rs);
  Serial.println(" Ohm");

  float ppm = 49.101 * pow(Rs / Ro,-0.876); // ppm = 49.101 * ((rs/ro)^-0.876)
  // pow disini berfungsi sebagai mencari bilangan berpangkat
  Serial.print("CO : ");
```

```
Serial.print(ppm);  
Serial.println(" ppm");  
  
Serial.println();  
delay(1000);  
}
```



**Gambar 3.25** Tampilan serial monitor nilai PPM dari sensor CO MQ7

## BAB IV

### HASIL DAN PEMBAHASAN

Setelah melakukan perancangan dan pemasangan komponen, selanjutnya adalah melakukan serangkaian uji coba pada masing-masing blok rangkaian yang bertujuan untuk mendapatkan kesesuaian spesifikasi dan hasil yang diinginkan. Untuk lebih jelas mengenai pembahasan hasil uji coba yang akan dilakukan dan dapat dilihat pada sub bagian berikut.

#### 4.1 Uji Coba Perangkat Keras

Setelah menyelesaikan tahap pembuatan dan perakitan seluruh komponen alat yang ada, maka tahap selanjutnya adalah masuk pada uji coba alat pada masing-masing sensor. Gunanya adalah untuk memastikan sensor tersebut sudah bekerja dengan benar atau masih ada terjadi kesalahan.



**Gambar 4.1** Uji coba alat menggunakan asap pembakaran kertas

Pada proses ini, uji coba yang dilakukan dengan menggunakan waktu pengiriman per 5 detik untuk melihat daya tangkap sensor terhadap lingkungan sekitar. Dan membandingkan dengan standar keadaan udara saat itu menurut BMKG sebagai acuan dasar. Namun, dalam hal ini, yang perlu di perhatikan adalah bahwa uji coba alat yang di lakukan disesuaikan dengan daya tangkap sensor terhadap keadaan sekitar.

Setelah melakukan uji coba alat, maka banyak di lakukan perbaikan baik dari segi program dan juga rangkaian solder alat.



**Gambar 4.2** Perbaikan program dan pengecekan rangkaian alat

Pada tahap perbaikan ini, kode program di sesuaikan dengan kemampuan daya yang di butuhkan sensor. Karena mengingat Arduino Uno hanya menyediakan power 5v dan NodeMcu hanya menyediakan power 3.3v.

#### **4.1.1. Uji Coba Alat dalam Cuaca Panas**

Pada proses tahapan yang ini, berguna untuk melihat daya tahan alat terhadap cuaca sekitar yang berubah-ubah. Karena suhu dalam kotak akan sedikit

lebih panas di banding keadaan di luar kotak. Begitu juga halnya keadaan sensor dalam mangkok.



**Gambar 4.3** Uji coba alat dalam cuaca panas

Dan hasil yang di dapatkan adalah bahwa sensor dan mikrokontroler tersebut mampu bertahan di dalam keadaan tidak terlalu panas.

#### **4.2 Pengambilan Data di sekitar area Rumah**

Setelah melakukan uji coba berkali-kali, maka tahap selanjutnya adalah pengambilan data secara realtime khusus di sekitar area Rumah.



**Gambar 4.4** Proses pengambilan data di sekitar area Rumah

Proses pengambilan data ini dilakukan pada akhir bulan September dan awal bulan Oktober sebagai tahap akhir pengambilan data. Untuk waktu pengiriman dilakukan setiap per 5 detik ke website.

#### **4.3 Pengambilan Data di Taman Makam Pahlawan Sanaman Lampung**

Penelitian dan pengambilan data selanjut nya dilakukan di Taman Makam Pahlawan yang berjarak sekitar 200m dari Rumah. Pelaksanaan pengambilan data ini dilakukan selama 3x dalam seminggu.



**Gambar 4.5** Pengambilan data di Taman Makam Pahlawan Sanaman Lampung

Keadaan saat pengambilan data di Taman Makam Pahlawan Sanaman Lampung mendung dan sedikit gerimis. Proyek alat yang di gunakan ini adalah alat yang sama dengan yang di lakukan di area Rumah.

#### 4.4 Analisis Data

Indeks Standar Pencemar Udara adalah angka yang tidak mempunyai satuan yang menggambarkan kondisi kualitas udara ambien di lokasi dan waktu tertentu yang didasarkan kepada dampak terhadap kesehatan manusia, nilai estetika dan makhluk hidup lainnya.

ISPU	Tingkat Pencemaran Udara	Dampak Kesehatan
0-50	Baik	Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan dan tidak berpengaruh pada tumbuhan, bangunan ataupun nilai estetika
51-100	Berdang	Kualitas udara yang tidak berpengaruh pada kesehatan manusia ataupun hewan tetapi berpengaruh pada tumbuhan yang sensitive dan nilai estetika
101-199	Tidak Sehat	Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitive atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika
200-299	Sangat Tidak Sehat	Tingkat kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar
300-500	Berbahaya	Tingkat kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius pada populasi

**Gambar 4.6** Batas Nilai-Nilai Indeks

Pengaruh konsentrasi gas karbonmonoksida (CO) terhadap kesehatan manusia dan makhluk hidup, sebagai berikut: Indeks ISPU berkategori baik (0-100), tidak menimbulkan efek apapun bagi manusia dan makhluk hidup. Nilai ISPU pada kisaran 101-299 berkategori tidak sehat, paparan gas CO mulai meningkatkan kardiovaskular pada perokok yang sakit jantung. Pada nilai ISPU di atas 300, atau masuk kategori berbahaya, paparan gas CO berbahaya bagi semua populasi. (Agusta Kurniawan, 2017)

Pengaruh konsentrasi gas Nitrogen dioksida (NO<sub>2</sub>) terhadap kesehatan manusia dan makhluk hidup, sebagai berikut: Indeks ISPU berkategori baik (0-100), paparan gas NO<sub>2</sub> menimbulkan sedikit bau tertentu. Pada kisaran lebih tinggi lagi pada indeks ISPU 101-299 berkategori tidak sehat paparan gas NO<sub>2</sub> mulai meningkatkan bau lebih tajam dan mulai kehilangan warna gas, memberikan efek peningkatan reaktivitas pembuluh tenggorokan pada penderita asma. Pada nilai ISPU di atas 300, atau masuk kategori berbahaya, paparan gas NO<sub>2</sub> berbahaya bagi semua populasi. (Agusta Kurniawan, 2017)

Pengaruh konsentrasi gas Ozon Permukaan (O<sub>3</sub>) terhadap kesehatan manusia dan makhluk hidup, sebagai berikut: Indeks ISPU berkategori baik (0-100), paparan gas O<sub>3</sub> mengakibatkan luka pada beberapa spesies tumbuhan. Pada kisaran indeks ISPU 101-299 berkategori tidak sehat, paparan gas O<sub>3</sub> mulai mengakibatkan penurunan kemampuan pada atlet yang berlatih keras. Pada nilai ISPU di atas 300, atau masuk kategori berbahaya, paparan gas O<sub>3</sub> berbahaya bagi semua populasi. (Agusta Kurniawan, 2017)

Pengaruh Partikulat (PM<sub>2.5</sub>) terhadap kesehatan manusia dan makhluk hidup, yaitu tidak ada efek apapun bila kategori ISPU berkategori baik (0-100). Pada kisaran indeks ISPU 101-299 berkategori tidak sehat, partikulat menyebabkan jarak pandang turun secara signifikan, dan terjadi pengotoran debu di mana-mana. Pada nilai ISPU di atas 300, atau masuk kategori berbahaya, paparan partikulat (PM<sub>2.5</sub>) berbahaya bagi semua populasi. (Agusta Kurniawan, 2017)

#### 4.4.1. Analisis Data pada Area Rumah

Analisis data yang di lakukan di area rumah ini akan di bahas satu per satu sesuai dengan data sensor yang ada. Proses pengambilan data yang di bahas ini adalah data yang di ambil pada tanggal : 21 September 2020.



Gambar 4.7 Antarmuka website monitoring seluruh sensor

##### 4.4.1.1. Data Sensor CO

Pengambilan data yang di analisis ini adalah pada tanggal 21 September 2020. Ada beberapa variasi konsentrasi CO di daerah Palangkaraya. Keadaan saat pengambilan data ini **sehat**.



**Gambar 4.8** Data CO di Area Rumah

Meskipun efek kesehatan dari polusi udara telah diketahui selama bertahun-tahun, sayangnya pertumbuhan penduduk, peningkatan lalu lintas kota, dan penggunaan udara luar biasa, tetapi kita dapat memperkirakan situasi polutan lain dari konsentrasi CO. Karbon Monoksida (CO) gas terbentuk terutama ketika bahan bakar karbon tidak dibakar sepenuhnya.

#### 4.4.1.2. Data Sensor PM2.5

Pengambilan data yang di analisis ini adalah pada tanggal 21 September 2020. Ada beberapa variasi konsentrasi PM2.5 di daerah Palangkaraya. Keadaan saat pengambilan data ini **sehat**.



**Gambar 4.9** Data PM2.5 di Area Rumah

#### 4.4.1.3. Data Sensor O3

Pengambilan data yang di analisis ini adalah pada tanggal 21 September 2020. Ada beberapa variasi konsentrasi O3 di daerah Palangkaraya. Keadaan saat pengambilan data ini **sehat**.



**Gambar 4.10** Data O3 di Area Rumah

#### 4.4.1.4. Data Sensor NO2

Pengambilan data yang di analisis ini adalah pada tanggal 21 September 2020. Ada beberapa variasi konsentrasi NO2 di daerah Palangkaraya. Keadaan saat pengambilan data ini **sehat**.



Gambar 4.11 Data NO2 di Area Rumah

#### 4.4.2. Analisis Data pada Taman Makam Pahlawan Sanaman Lampung

Analisis data yang di lakukan di area Taman Makam Pahlawan ini akan di bahas mulai dari CO, O3, NO2 dan PM2.5. Proses pengambilan data yang di bahas ini adalah data yang di ambil pada tanggal : 27 September 2020.

##### 4.4.2.1. Data Sensor NO2

Pengambilan data yang di analisis ini adalah pada tanggal 27 September 2020. Ada beberapa variasi konsentrasi NO2 di daerah Palangkaraya. Keadaan saat pengambilan data ini **tidak sehat**.



**Gambar 4.12** Data NO2 di Taman Makam Pahlawan Sanaman Lampung

Meskipun efek kesehatan dari polusi udara telah diketahui selama bertahun-tahun, sayangnya pertumbuhan penduduk, peningkatan lalu lintas kota, dan penggunaan udara luar biasa, tetapi kita dapat memperkirakan situasi polutan lain dari konsentrasi NO2. Karbon monoksida (NO) gas terbentuk terutama ketika bahan bakar karbon tidak dibakar sepenuhnya.

#### 4.4.2.2. Data Sensor CO

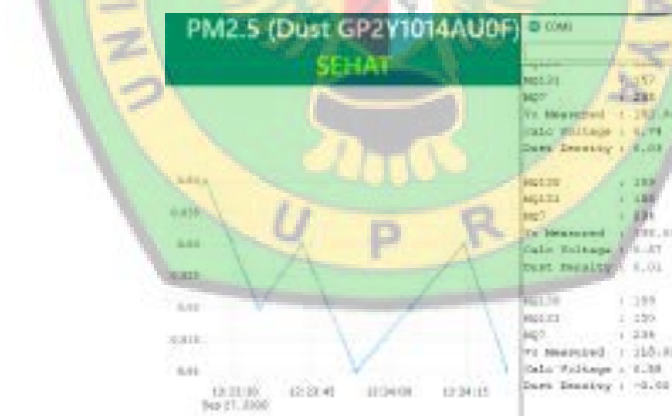
Pengambilan data yang di analisis ini adalah pada tanggal 27 September 2020. Ada beberapa variasi konsentrasi CO di daerah Palangkaraya. Keadaan saat pengambilan data ini **tidak sehat**.



Gambar 4.13 Data CO di Taman Makam Pahlawan Sanaman Lampung

#### 4.4.2.3. Data Sensor PM2.5

Pengambilan data yang di analisis ini adalah pada tanggal 27 September 2020. Ada beberapa variasi konsentrasi PM2.5 di daerah Palangkaraya. Keadaan saat pengambilan data ini **sehat**.



Gambar 4.14 Data PM2.5 di Taman Makam Pahlawan Sanaman Lampung

#### 4.4.2.4. Data Sensor O3

Pengambilan data yang di analisis ini adalah pada tanggal 27 September 2020. Ada beberapa variasi konsentrasi O3 di daerah Palangkaraya. Keadaan saat pengambilan data ini **tidak sehat**.



**Gambar 4.15** Data O3 di Taman Makam Pahlawan Sanaman Lampung

## 4.5 Permasalahan dan Solusi

### 4.5.1. Permasalahan Internal

Permasalahan internal yang di hadapi adalah :

1. Jika sensor di aktifkan terus selama 24 jam nonstop, maka akan memperpendek umur sensor. Karena masing-masing sensor memiliki data sheet dimana sensor bekerja normal selama 48 jam.
2. Prototype alat yang di buat hanya 1 buah, jadi tidak bisa dibuat perbandingan data dengan lokasi yang berbeda secara bersamaan. Karena terkendala pada jumlah alat yang di sediakan.
3. Jika jaringan wireless di lokasi dalam keadaan normal, maka pengiriman data tidak mengalami masalah. Namun sebaliknya, jika saat pengambilan data dan jaringan wireless pun terganggu, maka proses pengiriman datanya pun akan terhalang.

### 4.5.2. Solusi Permasalahan Internal

Solusi yang ditawarkan dan dikerjakan adalah sebagai berikut :

1. Proses pengambilan di atur sesuai jadwal yang diinginkan. Contohnya : Data Siang di ambil dari pukul 12.00 – 17.00 WIB. Kemudian untuk data Malam diambil dari pukul 18.00 – 21.00 WIB. Namun untuk saat ini,

proses mematikan dan menghidupkan alat masih secara manual, belum menggunakan Remote Control berbasis IoT.

2. Jika jaringan sedang dalam keadaan tidak stabil untuk pengambilan data, maka ada solusi terbaik yang di tawarkan oleh para devoloper hardware yaitu penggunaan kartu SD Card untuk record data secara offline. Namun, mengingat projek ini merupakan langkah awal atau proses penelitian awal, maka alternatif yang di gunakan adalah merekam data menggunakan Serial Monitor yang disediakan oleh Arduino Ide dan kemudian di copy paste kedalam MS.Exel.

#### **4.5.3. Permasalahan Eksternal**

Berikut ini beberapa permasalahan eksternal yang di hadapi :

1. Alat harus diawasi oleh orang-orang disekitar, karena rawan pencurian.
2. Sensor sangat sensitif dengan asap rokok dan pembakaran sampah.
3. Posisi alat mudah jatuh jika tidak ditahan dengan penyangga atau pengikat kaki-kaki tiang.

#### **4.5.4. Permasalahan Eksternal**

Solusi yang ditawarkan untuk permasalahan yang ada adalah :

1. Alat yang di pasang saat ini masih di sekitar rumah dan diawasi oleh saya sendiri. Sebagai contoh, ketika pengambilan data di area rumah, maka yang bertanggung jawab untuk mengawasi alat adalah saya sendiri. Dan ketika pengambilan data di Taman Makam Pahlawan Sanaman Lampang, maka yang bertanggung jawab untuk mengawasi alat tersebut adalah saya dan 1 orang teman.
2. Di larang untuk merokok dan membakar sampah di sekitar alat sensor saat pengambilan data berlangsung. Karena akan mengakibatkan Noise pada data yang di tangkap. Sehingga grafik data yang masuk akan terganggu.

3. Untuk tata letak alat harus pada permukaan tanah yang keras dan datar. Karena tiang besi sudah didesain dengan pengikat pada kaki-kaki tiang, maka dengan mudah untuk mengikat kaki tiang pada sebuah kayu ataupun di tancapkan pada tanah.

#### 4.6 Kode Program

Dalam membangun desain IoT ini, kode yang digunakan adalah menggunakan bahasa C yaitu pada NodeMCU yang manual mengetikkan alamat IP agar terhubung ke website Monitoring.

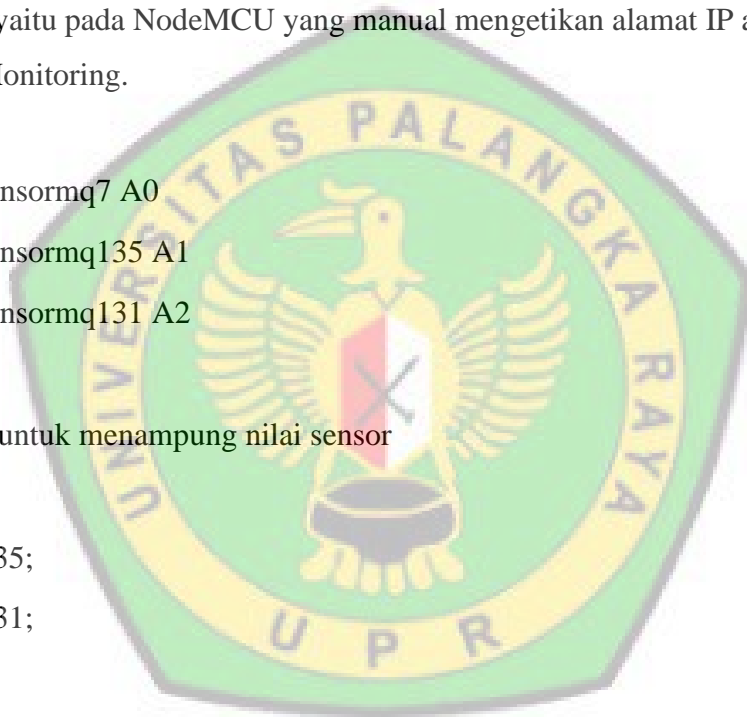
```
#define sensormq7 A0
#define sensormq135 A1
#define sensormq131 A2

//variabel untuk menampung nilai sensor
float mq7;
float mq135;
float mq131;

int measurePin = A3; //Connect dust sensor to Arduino A0 pin
int ledPower = 7; //Connect 3 led driver pins of dust sensor to Arduino D2

int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;

float voMeasured = 0;
```



```
float calcVoltage = 0;
float dustDensity = 0;

long RL_mq135 = 20000; // 10000 Ohm
long Ro_mq135 = 91655 ; // 5935 ohm ( SILAHKAN DISESUAIKAN)

long RL_mq131 = 20000; // 10000 Ohm
long Ro_mq131 = 67895 ; // 5935 ohm ( SILAHKAN DISESUAIKAN)

long RL_mq7 = 10000; // 10000 Ohm
long Ro_mq7 = 5935 ; // 5935 ohm ( SILAHKAN DISESUAIKAN)

void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(ledPower,OUTPUT);
}

void loop() {

  //baca permintaan dari NodeMCU
  String minta = "";
  //baca permintaan NodeMCU

  // kirimdata();
  // Serial.println();
```

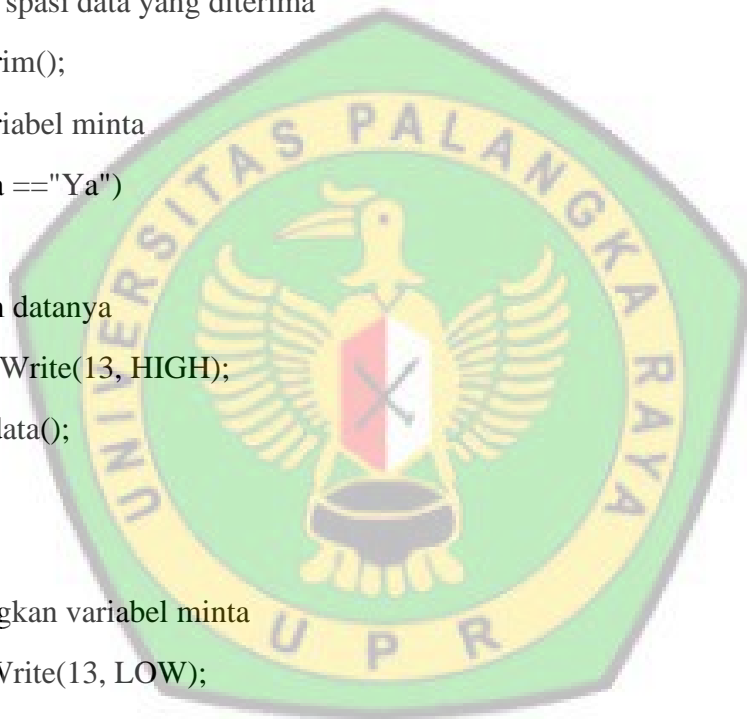
```
while(Serial.available()>0)
{
  minta += char(Serial.read());
}

//buang spasi data yang diterima
minta.trim();

//uji variabel minta
if(minta == "Ya")
{
  //kirim datanya
  digitalWrite(13, HIGH);
  kirimdata();
}

//kosongkan variabel minta
digitalWrite(13, LOW);
minta = "";
delay(1000);
}

void kirimdata()
{
  //baca nilai sensor LDR
```



```

digitalWrite(ledPower,LOW); // power on the LED
delayMicroseconds(samplingTime);

voMeasured = analogRead(measurePin); // read the dust value

delayMicroseconds(deltaTime);
digitalWrite(ledPower,HIGH); // turn the LED off
delayMicroseconds(sleepTime);

// 0 - 5V mapped to 0 - 1023 integer values
// recover voltage
calcVoltage = voMeasured * (5.0 / 1024.0);

// linear equation taken from http://www.howmuchsnow.com/arduino/airquality/
// Chris Nafis (c) 2012
dustDensity = 0.17 * calcVoltage - 0.1;

int pinmq135 = analogRead(sensormq135);

float VRL_mq135 = pinmq135 * 5.00 / 1024; // mengubah nilai ADC (
0 - 1023 ) menjadi nilai voltase ( 0 - 5.00 volt )

float Rs_mq135 = (5.00 * RL_mq135 / VRL_mq135) - RL_mq135;

float ppm_mq135 = 100 * pow(Rs_mq135 / Ro_mq135, -0.007); // ppm = 49.101
* ((rs/ro)^-0.876)

// pow disini berfungsi sebagai mencari

int pinmq131 = analogRead(sensormq131);

float VRL_mq131 = pinmq131 * 5.00 / 1024; // mengubah nilai ADC (
0 - 1023 ) menjadi nilai voltase ( 0 - 5.00 volt )

```

```

float Rs_mq131 = (5.00 * RL_mq131 / VRL_mq131) - RL_mq131;

float ppm_mq131 = 100 * pow(Rs_mq131 / Ro_mq131, -0.912); // ppm = 49.101
* ((rs/ro)^-0.876)

// pow disini berfungsi sebagai mencari

int pinmq7 = analogRead(sensormq7);

float VRL_mq7 = pinmq7 * 5.00 / 1024; // mengubah nilai ADC ( 0
- 1023 ) menjadi nilai voltase ( 0 - 5.00 volt )

float Rs_mq7 = (5.00 * RL_mq7 / VRL_mq7) - RL_mq7;

float ppm_mq7 = 100 * pow(Rs_mq7 / Ro_mq7, -0.876); // ppm = 49.101 *
((rs/ro)^-0.876)

// pow disini berfungsi sebagai mencari

mq135 = ppm_mq135;
mq131 = ppm_mq131;
mq7 = ppm_mq7;

// Idr = analogRead(0);
// //baca temperature dan humidity
// temperature = dht.readTemperature();
// humidity = dht.readHumidity();

//sediakan variabel penampung data yang akan dikirim
// datakirim = "10#29.05#84.50"

String datakirim = String(mq135) + "#" + String (mq131) + "#" + String (mq7) +
"# + String (voMeasured) + "#" + String (calcVoltage) + "#" + String (dustDensity);

//kirim data ke nodemcu

```

```
Serial.println(data kirim);  
}
```



```
#include <SoftwareSerial.h>
//#include <ThingerESP8266.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>

//buat variabel untuk software serial (Rx, Tx)
SoftwareSerial DataSerial (12, 13);

//millis sebagai pengganti delay
unsigned long previousMillis = 0;
const long interval = 5000;

//variabel array untuk data parsing
String arrData[6];

//konfigurasi ThingerIO
#define USERNAME "adshnz"
#define DEVICE_ID "Serial_ThingerIO"
#define DEVICE_CREDENTIAL "kIv+e?KbqXU0"

//variabel PIN LED
#define LED_PIN 4 //pin D2 NodeMCU

//variabel untuk thingerIO
//ThingerESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
```



```
//konfigurasi WiFi
const char* ssid = "Frans Xanana";
const char* password = "6271030605940006";
const char* host = "192.168.1.5";

//sediakan variabel untuk menampung nilai LDR, Temp, Hum yang siap dikirim ke
ThingierIO

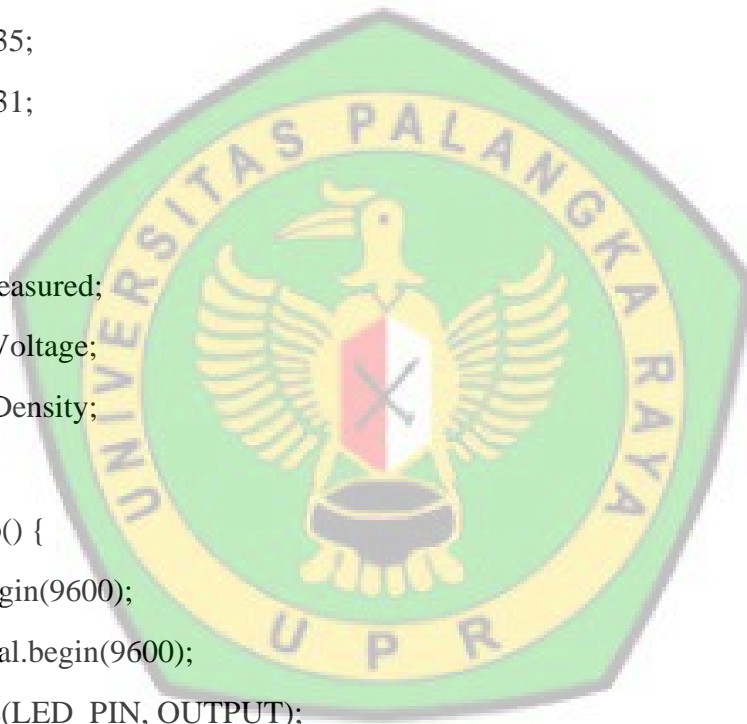
float mq135;
float mq131;
float mq7;

float voMeasured;
float calcVoltage;
float dustDensity;

void setup() {
  Serial.begin(9600);
  DataSerial.begin(9600);
  pinMode(LED_PIN, OUTPUT);

//koneksi ke WiFi
WiFi.hostname("NodeMCU");
WiFi.begin(ssid, password);

//cek koneksi
while(WiFi.status() != WL_CONNECTED)
{
```



```

    delay(500);
    digitalWrite(LED_PIN, LOW);
}
//apabila terkoneksi
digitalWrite(LED_PIN, HIGH);
//hubungkan NodeMCU ke ThingerIO
// thing.add_wifi(ssid, password);

    //data yang akan dikirim
// thing["Dataku"] >> [(pson & out)
// {
//   out["Mq135"] = mq135;
//   out["Mq131"] = mq131;
//   out["Mq7"] = mq7;
//   out["Vo Measured"] = voMeasured;
//   out["Calc Voltage"] = calcVoltage;
//   out["Dust Density"] = dustDensity;
// };

}

void loop() {
    //konfigurasi millis
    unsigned long currentMillis = millis(); //baca waktu millis saat ini
    if(currentMillis - previousMillis >= interval)
    {

```

```

//update previousMillis
previousMillis = currentMillis;

//prioritaskan pembacaan data dari arduino uno(hasil kiriman data)
//baca data serial
String data = "";
while(DataSerial.available(>0)
{
  data += char(DataSerial.read());
}
//buang spasi datanya
data.trim();

//uji data
if(data != "")
{
  //format data "10#29.45#89.50" = array (setelah parsing)
  //parsing data (pecah data)
  int index = 0;
  for(int i=0; i<= data.length(); i++)
  {
    char delimiter = '#' ;
    if(data[i] != delimiter)
      arrData[index] += data[i] ;
    else
      index++; //variabel index bertambah 1
  }
}

```

```

}

//pastikan bahwa data yang dikirim lengkap (MQ135, MQ131, Vo Measured,
Calc Voltage)

//urutannya 0=MQ135, 1=MQ131, 2=MQ7, 3=Vo Measured, 4=Calc Voltage,
5=Dust Density

if(index == 5)
{
//tampilkan nilai sensor ke serial monitor
Serial.println("MQ135      : " + arrData[0]); //MQ135
Serial.println("MQ131      : " + arrData[1]); //MQ131
Serial.println("MQ7        : " + arrData[2]); //MQ7
Serial.println("Vo Measured : " + arrData[3]); //Vo Measured
Serial.println("Calc Voltage : " + arrData[4]); //Calc Voltage
Serial.println("Dust Density : " + arrData[5]); //Dust Density
Serial.println();
}

//isi variabel yang akan dikirim
mq135    = arrData[0].toFloat();
mq131    = arrData[1].toFloat();
mq7      = arrData[2].toFloat();
voMeasured = arrData[3].toFloat();
calcVoltage = arrData[4].toFloat();
dustDensity = arrData[5].toFloat();

//picu pengiriman data ke ThingierIO
//  thing.handle();

```

```

//proses cek koneksi ke server
WiFiClient client ;
if(!client.connect(host, 80))
{
Serial.println("Connection Failed");
return ;
}

//proses pengiriman data ke server
// String Link ;
HTTPClient http;
// Link = "http://192.168.1.14/chart/kirimdata.php?DataSerial=" + String(data);
// //eksekusi link
// http.begin(Link);
// //mode GET
// http.GET();
// http.end();
// Serial.println(data);

```

String MQ135Post, MQ131Post, MQ7Post, DustDensityPost, voMeasuredPost,  
postData;

MQ135Post = String(mq135); //String to interger conversion

MQ131Post = String(mq131); //String to interger conversion

MQ7Post = String(mq7); //String to interger conversion

DustDensityPost = String(dustDensity); //String to interger conversion

```

voMeasuredPost = String(voMeasured);

postData = "mq135=" + MQ135Post
+ "&mq131=" + MQ131Post
+ "&mq7=" + MQ7Post
+ "&dustDensity=" + DustDensityPost
+ "&voMeasured=" + voMeasuredPost;
// Serial.println (postData);

http.begin("http://192.168.1.5/chart2/kirimdata.php?"); //Specify request
destination

http.addHeader("Content-Type", "application/x-www-form-urlencoded");
//Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

http.end();

arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
arrData[3] = "";
arrData[4] = "";
arrData[5] = "";

```

```
}  
//minta data ke arduino uno  
DataSerial.println("Ya");  
}  
}
```

#### **4.7 Hasil Pengujian Data dengan Metode Black Box Testing**

Hasil pengujian merupakan bagian yang penting dalam siklus pengembangan sistem. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari sistem. Tujuan dari pengujian adalah untuk menjamin sistem yang di bangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam. Kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya. Tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui input dan output).

#### 4.7.1. Pengujian Sensor dalam Ruangan

Tabel 4.1 Tabel Pengujian Sensor dalam ruangan

No.	Jenis Sensor	Hasil Tes	
1.	CO	Berdasarkan hasil uji coba dan penelitian untuk sensor CO memiliki daya tangkap rata-ratanya ialah 16 ppm.	✓
2.	O3	Berdasarkan hasil uji coba dan penelitian untuk sensor O3 memiliki daya tangkap rata-ratanya ialah 49 ppm.	✓
3.	PM2.5	Berdasarkan hasil uji coba dan penelitian untuk sensor PM2.5 memiliki daya tangkap rata-ratanya ialah 0.10 $\mu\text{g}/\text{cm}^3$ .	✓
4.	NO2	Berdasarkan hasil uji coba dan penelitian untuk sensor NO2 memiliki daya tangkap rata-ratanya ialah 99 ppm.	✓

#### 4.7.2. Pengujian Sensor di luar Ruangan

Tabel 4.2 Tabel Pengujian Sensor di luar ruangan

No.	Jenis Sensor	Hasil Tes	
1.	CO	Berdasarkan hasil uji coba dan penelitian untuk sensor CO memiliki daya tangkap rata-ratanya ialah 233 ppm.	✓
2.	O3	Berdasarkan hasil uji coba dan penelitian untuk sensor O3 memiliki daya tangkap rata-ratanya ialah 150 ppm.	✓
3.	PM2.5	Berdasarkan hasil uji coba dan penelitian untuk sensor PM2.5 memiliki daya tangkap rata-ratanya ialah 0.01 $\mu\text{g}/\text{cm}^3$ .	✓
4.	NO2	Berdasarkan hasil uji coba dan penelitian untuk sensor NO2 memiliki daya tangkap rata-ratanya ialah 163 ppm.	✓

## BAB V

### KESIMPULAN DAN SARAN

Kesimpulan adalah pernyataan singkat, jelas, dan sistematis dari keseluruhan hasil analisis, pembahasan, dan pengujian hipotesis dalam sebuah penelitian. Saran adalah usul atau pendapat dari seorang peneliti yang berkaitan dengan pemecahan masalah yang menjadi objek penelitian ataupun kemungkinan penelitian lanjutan. Berikut ini adalah merupakan kesimpulan dan saran dari hasil yang di dapatkan oleh peneliti :

#### 5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah di lakukan oleh peneliti, dapat di simpulkan sebagai berikut :

1. Teknologi yang digunakan adalah *Internet of Things* dengan menggunakan website sebagai..
2. Membangun desain *Internet of Things* dengan menggunakan ESP-8266 NodeMCU yang di gabungkan dengan Arduino Uno sebagai pembantu mikrokontroler mendapatkan data nilai sensor gas.
3. Tujuan penelitian dilakukan adalah untuk mempelajari sistem cara kerja berbasis IoT pada studi kasus Pemantauan Kualitas Udara. Dalam hal ini, sampel data yang di ambil dari area rumah dan juga di Taman Makam Pahlawan Sanaman Lampung.
4. Prototype pemantauan kualitas udara ini dapat memberikan kemudahan untuk memonitoring tingkat kualitas udara secara sehingga mendapatkan sebuah informasi yang nyata dan cepat.
5. Prototype pemantauan kualitas udara ini dirancang dengan menggunakan bahasa pemrograman IDE (*Integrated Development Environment*) ada pada Arduino dan NodeMCU.

6. Hasil data penelitian yang dilakukan pada 2 lokasi berbeda adalah sebagai berikut :

a. Lokasi di Area Rumah

Jenis Sensor	Konsentrasi Data Sensor	Konversi Data dengan ISPU	Hasil
O3	107.66 ppm	-	TIDAK SEHAT
NO2	198.78 ppm	-	TIDAK SEHAT
CO	38.96 ppm	341 ppm	BERBAHAYA
PM2.5	0.10 ppm	-	SEHAT

b. Lokasi di Taman Makam Pahlawan Sanaman Lampung

```

MQ135      : 159
MQ131      : 150
MQ7        : 234
Vo Measured : 118.00
Calc Voltage : 0.58
Dust Density : -0.00
  
```

## 5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan oleh peneliti, banyak ditemukan kekurangan yang menjadi bahan evaluasi untuk peneliti selanjutnya. Berikut ini saran yang di paparkan oleh peneliti :

1. Pengecekan dan pemeliharaan alat secara rutin.
2. Sistem keamanan alat dan data yang baik.
3. Alat ini dapat di tambahkan power daya yang besar seperti aki dengan menggunakan sistem *auto charging* yang di bantu dengan solar panel.
4. Penambahan Alarm, kamera pengawas dan Lampu peringatan

5. Alat ini juga bisa ditambahkan dengan pada bagian program nya dengan mode *sleep time* untuk menghemat pemakaian daya yang cukup besar serta untuk memperpanjang umur sensor agar tidak 24 jam aktif terus, tapi di berikan jeda dan mode mati/hidup secara otomatis.
6. Di alat ini, bisa juga di tambahkan nilai perhitungan data setiap sensornya.

