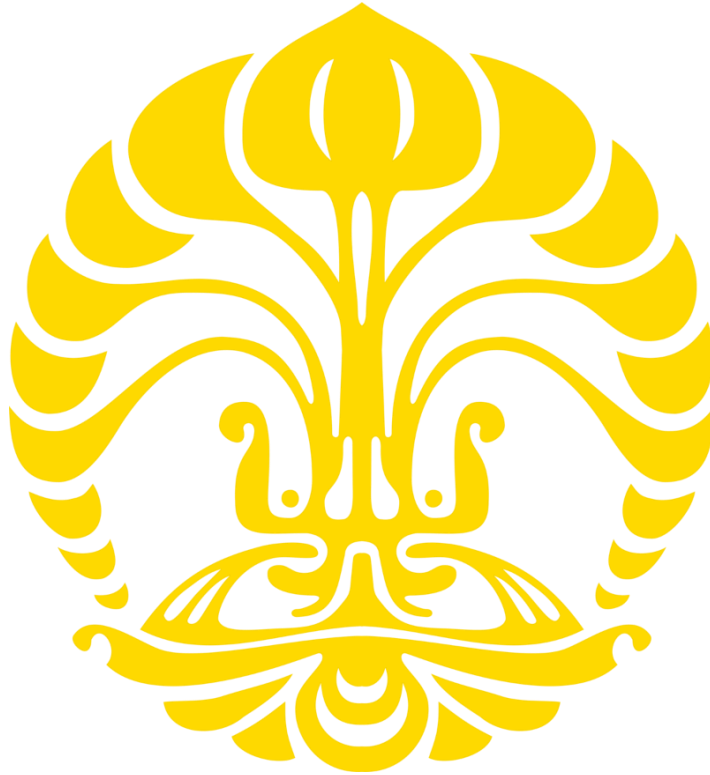


LAPORAN SISTEM EMBEDDED

MONIRA (Monitoring Udara)



BY GROUP:

Aufa Dhiya Aydan	1706043260
Muhamad Fadil	1706042812
Muhammad Farhan	1706042876
Saint Vitra Dhiyaulhaq	1706986063
Zafran Hibatullah Tsany	1706986082

TEKNIK KOMPUTER
FAKULTAS TEKNIK
UNIVERSITAS INDONE

KATA PENGANTAR

Kami mengucapkan terima kasih kepada Tuhan Yang Mahakuasa, Allah SWT, karena atas berkah, taufiq, dan inayahNya kami diberikan kesempatan untuk membuat laporan Sistem Embedded yang berjudul "Air Quality" ini sehingga dapat diselesaikan tepat waktu.

Dalam pekerjaan ini pula, tidak lupa kami juga berterima kasih kepada dosen Sistem Embedded kami Fransiskus Astha Ekadiyanto S.T., M.Sc dan juga teman-teman yang membantu kami untuk ikut berkontribusi dengan memberikan masukan berupa ide-ide cemerlang sehingga laporan Air Quality ini dapat disusun dengan baik dan rapi.

Kami berharap semoga laporan Air Quality yang kami buat ini dapat menambah wawasan dan pengetahuan bagi mereka yang membacanya. Kami mohon maaf yang sebesar-besarnya karena laporan ini masih jauh dari kata sempurna, jadi kami menantikan kritik dan saran dari kalian yang sifatnya membangun agar laporan selanjutnya bisa lebih baik lagi.

Depok, 18 November 2019

Kelompok Monira (Monitoring Udara)

DAFTAR ISI

JUDUL.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Manajemen.....	4
BAB II PENJELASAN PROJECT	5
2.1 Dasar Teori	5
2.1.1 Node MCU ESP36	5
2.1.2 MQ135	6
2.1.3 DHT11	10
2.1.4 Thingsboard	11
2.1.5 Parameter Skala Udara	13
2.2 Perancangan Project	13
2.2.1 UML	13
2.2.2 Bahasa Program	16
2.2.3 Schematic Alat	17
2.2.4 Fitur-fitur Platform	18
2.2.5 Hasil Data	21

2.3	Pencapaian22
2.4	Kendala23
2.5	Product Physics24
2.5.1	Kerangka fisik24
2.5.2	Kerangka elektronik25
2.6	Implementasi26
2.7	Lokasi (Share Locate).....	28
2.8	Penempatan IoT29
BAB III PENUTUP		30
3.1	Kesimpulan	30
3.2	Saran	30
REFERENSI		31

BAB I

PENDAHULUAN

1.1. Latar Belakang

Udara adalah campuran gas yang memiliki berbagai elemen di dalamnya dan memiliki dampak besar pada kelangsungan hidup yang ada di seluruh permukaan bumi. Oleh karena itu, kita harus memperhatikan kelestarian udara di sekitar kita dengan mencari tahu apakah kualitas udara masih cocok untuk kita atau sudah di zona kritis.

Mengapa kita mesti tahu kualitas udara suatu daerah? Sebab area dengan kualitas udara buruk dapat menyebabkan penyakit berbahaya seperti kanker kulit, asma, penyakit paru obstruktif kronis (PPOK), infeksi saluran pernapasan atas (ISPA), dan kematian karena penyakit jantung.

Penyebab kualitas udara yang buruk adalah emisi kendaraan dengan bahan bakar minyak bumi. Bahan bakar minyak bumi mengandung senyawa hidrokarbon yang kemudian dibakar menjadi karbon dioksida (CO_2) dan air (H_2O). Hasil pembakaran dari kendaraan, menghasilkan karbon monoksida (CO) yang merupakan gas beracun, dan kemudian nitrogen oksida dan senyawa organik yang mudah menguap akan menjadi polutan, dan kemudian gas akan mempengaruhi kualitas udara.

Dari masalah-masalah tersebutlah, kami ingin membuat suatu software yang dapat digunakan untuk menentukan udara mana yang baik dan udara mana yang buruk bagi tubuh kita (semua makhluk hidup di dunia). Kami ingin menggabungkan software dengan suatu hardware untuk mendeteksi kualitas udara. Output dari perangkat keras menjadi input untuk perangkat lunak. Kami membuat softwrenya dengan HTML, CSS, bahasa pemrograman PHP dan hardwarenya adalah sensor ESP32 dan MQ135. Untuk itulah kami membuat MONIRA (Monitoring Udara).

1.2. Rumusan Masalah

Berdasarkan latar belakang dan permasalahan yang ada, maka diperoleh rumusan masalah sebagai berikut :

- Apa itu Air Quality?
- Apa penyebab kualitas udara buruk?
- Bagaimana cara mengetahui kualitas udara?
- Apa dampak dari kualitas udara yang buruk?
- Cara mempertahankan kualitas udara yang baik?

1.3. Tujuan

Kualitas udara diukur dengan Indeks Kualitas Udara, atau AQI. AQI bekerja seperti termometer yang beroperasi dari 0 hingga 500 derajat. Namun, alih-alih menunjukkan perubahan suhu, AQI adalah cara untuk menunjukkan perubahan dalam jumlah polusi di udara.

Karena itu, kita harus memperhatikan kualitas udara. Di kota, polusi udara disebabkan oleh asap dari kendaraan. Ini disebut ozon permukaan tanah (kabut asap perkotaan). Ozon permukaan tanah meningkat di kota-kota ketika udaranya tenang, suhunya hangat, dan matahari keluar. Kombinasi ini menjebak polusi di udara. Pesawat terbang juga menyebabkan polusi udara. Hal-hal lain, seperti kendaraan konstruksi dan asap tembakau juga menyebabkan polusi udara. Di daerah pedesaan, polusi udara luar seringkali disebabkan oleh debu dari traktor yang membajak ladang, truk dan mobil yang mengemudi di jalan tanah atau kerikil, tambang batu, dan asap dari kayu dan cemara tanaman.

Efek buruk dari kualitas udara buruk pada manusia adalah dapat dipecah menjadi efek jangka pendek dan efek jangka panjang. Efek jangka pendek, yang bersifat sementara, termasuk penyakit seperti pneumonia atau bronkitis. Mereka juga termasuk ketidaknyamanan seperti iritasi pada hidung, tenggorokan, mata, atau kulit. Polusi udara juga dapat menyebabkan sakit kepala, pusing, dan mual. Bau busuk yang dibuat oleh pabrik, sampah, atau sistem saluran pembuangan juga dianggap sebagai polusi udara. Bau ini kurang serius tetapi masih tidak menyenangkan. Efek jangka panjang dari polusi udara dapat berlangsung selama bertahun-tahun atau seumur hidup. Mereka bahkan dapat menyebabkan kematian seseorang.

Efek kesehatan jangka panjang dari polusi udara termasuk penyakit jantung, kanker paru-paru, dan penyakit pernapasan seperti emfisema. Polusi udara juga dapat menyebabkan kerusakan jangka panjang pada saraf, otak, ginjal, hati, dan organ orang lain. Beberapa ilmuwan menduga polusi udara menyebabkan cacat lahir. Hampir 2,5 juta orang meninggal di seluruh dunia setiap tahun karena efek polusi udara luar atau dalam ruangan. Untuk efek pada lingkungan, ini dapat membunuh tanaman atau mengurangi hasil panen mereka. Itu dapat membunuh pohon muda dan tanaman lainnya. Itu 10 langkah mudah untuk kualitas udara yang lebih bersih, seperti:

1. Berjalan kaki, bersepeda, carpool, atau naik angkutan umum
2. Kurangi kebutuhan pemanas Anda dengan membuat rumah Anda lebih hemat energi
3. Katakan tidak untuk pembakaran halaman belakang
4. Gunakan alat berkebun bertenaga tangan
5. Katakan tidak untuk bensin atau diesel
6. Periksa tekanan ban Anda untuk meningkatkan efisiensi bahan bakar
7. Kurangi penggunaan kembali daur ulang
8. Bebas menganggur
9. Kebun tanpa pestisida
10. Terlibat untuk mendukung upaya nasional dan lokal untuk membersihkan udara

1.4 Manfaat

1. Data yang dikumpulkan dari pemantauan kualitas udara membantu kami menilai dampak yang disebabkan oleh kualitas udara yang buruk pada udara kesehatan masyarakat
2. Data kualitas udara membantu kita menentukan apakah suatu daerah memenuhi standar kualitas udara yang dirancang oleh WHO
3. Data yang dikumpulkan dari pemantauan kualitas udara terutama akan membantu kami mengidentifikasi area yang tercemar, tingkat polusi, dan tingkat kualitas udara.
4. Pemantauan kualitas udara akan membantu dalam menentukan apakah programmer pengontrol polusi udara yang dirancang di suatu daerah bekerja secara efisien atau tidak

5. Data kualitas udara membantu kita memahami tingkat kematian di setiap lokasi karena polusi udara. Kita juga dapat menilai dan membandingkan penyakit/gangguan jangka pendek dan jangka panjang yang merupakan akibat dari polusi udara
6. Berdasarkan data yang dikumpulkan, langkah-langkah pengendalian dapat dirancang untuk melindungi lingkungan dan kesehatan semua organisme hidup.
7. Mengurangi polusi udara membantu mengatasi perubahan iklim
8. Mengambil tindakan yang masuk akal dari lokal ke global
9. Kunci informasi untuk meminimalkan paparan

1.5 Manajemen

Process:

Dengan menggunakan metode Waterfall.

- Requirement: spesifikasi software yang dibutuhkan (device, apps mobile, web)
- Design:
- Implementation : digunakan untuk mendeteksi kadar CO₂, temperature, serta kelembapan disekitar FTUI
- Testing: diimplementasikan di sekitar FTUI apakah sesuai dengan kebutuhan (*customer* masih *developer* belum warga FTUI)
- Maintenance: pengupdatean *software* sesuai kebutuhan yang ada atau perubahan spesifikasi agar software dapat lebih optimal

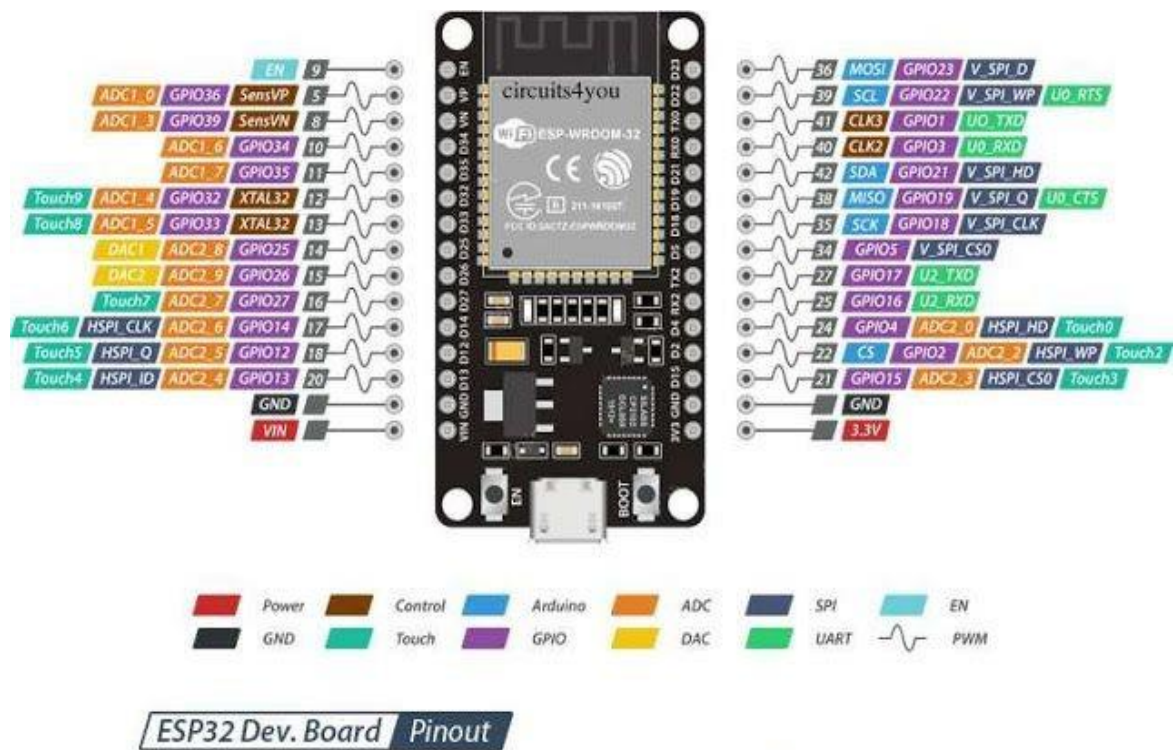
BAB II

PENJELASAN PROJECT

2.1 Dasar Teori

2.1.1 Node MCU ESP32

NodeMCU terkenal dengan modul ESP8266E dengan bahasa pemrograman LUA. ESP32 adalah big brother dari ESP8266. Muncul dengan prosesor dual-core 32-bit, built-in WiFi dan Bluetooth, lebih banyak RAM dan memori Flash, lebih banyak GPIO, lebih banyak ADC, dan banyak perangkat lain. NodeMCU ESP32 adalah modul ESP-WROOM-32 di breadboard yang friendly, kita dapat mengembangkan proyek kita dalam menggunakan mikrokontroler ini di papan breadboard. Dikarenakan pada ESP32 memiliki lebih banyak pin GPIO dibanding dengan ESP8266, maka dapat diputuskan pin mana yang akan difungsikan sebagai UART, I2C, atau SPI sesuai dengan kode program yang akan dibuat.



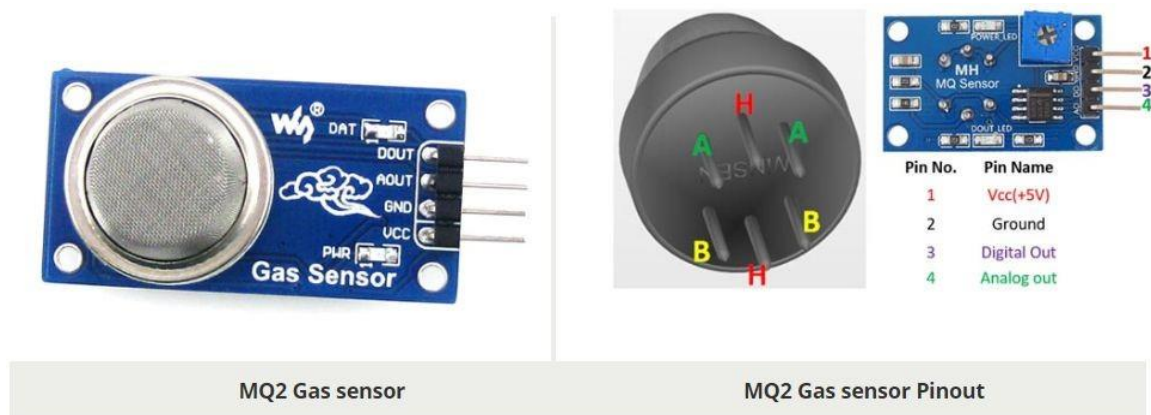
Gambar 1. Datasheet ESP32

Fitur:

- NodeMCU berdasarkan modul ESP-WROOM-32

- ❑ Berdasarkan ESP32 DEVKIT DOIT
- ❑ 30 Versi GPIO
- ❑ ESP32 adalah prosesor 32-bit dual core dengan built-in 2,4 GHz Wi-Fi dan Bluetooth
- ❑ Memori flash 4MByte
- ❑ RAM 520KByte
- ❑ 2.2 tp 3.6V Operating voltage range
- ❑ Di breadboard friendly breakout
- ❑ USB microB untuk komunikasi daya dan Serial, digunakan juga untuk memuat program dan debugging serial

2.1.2 MQ135



Gambar 2. Pin-pin pada MQ135

Modul Pin :

Pin no.	Pin Name	Keterangan
1	Vcc(+5V)	Tegangan yang dibutuhkan sebesar +5 Voltage
2	Ground	Untuk menghubungkan modul ke ground
3	Digital Out	Untuk mendapatkan nilai digital dengan menggunakan nilai thershold

4	Analog Out	Tegangan yang dihasilkan oleh pin analog 0-5 Voltage
---	------------	--

Tabel 1. Kterangan pin pada sensor

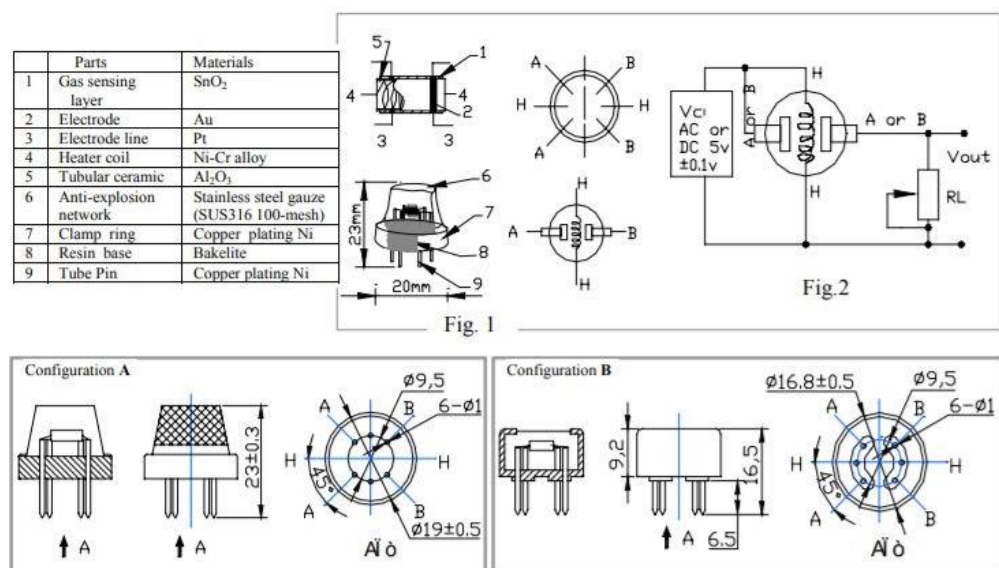
Modul Sensor:

Pin no.	Pin Name	Keterangan
1	A-pins	Pin A dan B dapat saling ditukarkan sesuai dengan kebutuhan. Nantinya akan dihubungkan ke suply
2	B-pins	Pin A dan B dapat saling ditukarkan. Nantinya satu pin dijadikan sebagai output dan satunya dihubungkan ke ground
3	H-pins	Terdapat dua pin H, satu pin dihubungkan ke supply dan yang kedua duhubungkan ke ground

Tabel 2. Kterangan pin pada modul

Spesifikasi MQ135

- Structure basic measuring circuit



Gambar 3. Structure basic circuit

- Standar work condition

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

Tabel 3. Standard work condition

- Environment condition

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

Tabel 4. Environment Condition

- Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Ramark 2
Rs	Sensing Resistance	30KΩ-200KΩ (100ppm NH ₃)	Detecting concentration scope 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20 ±2 Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

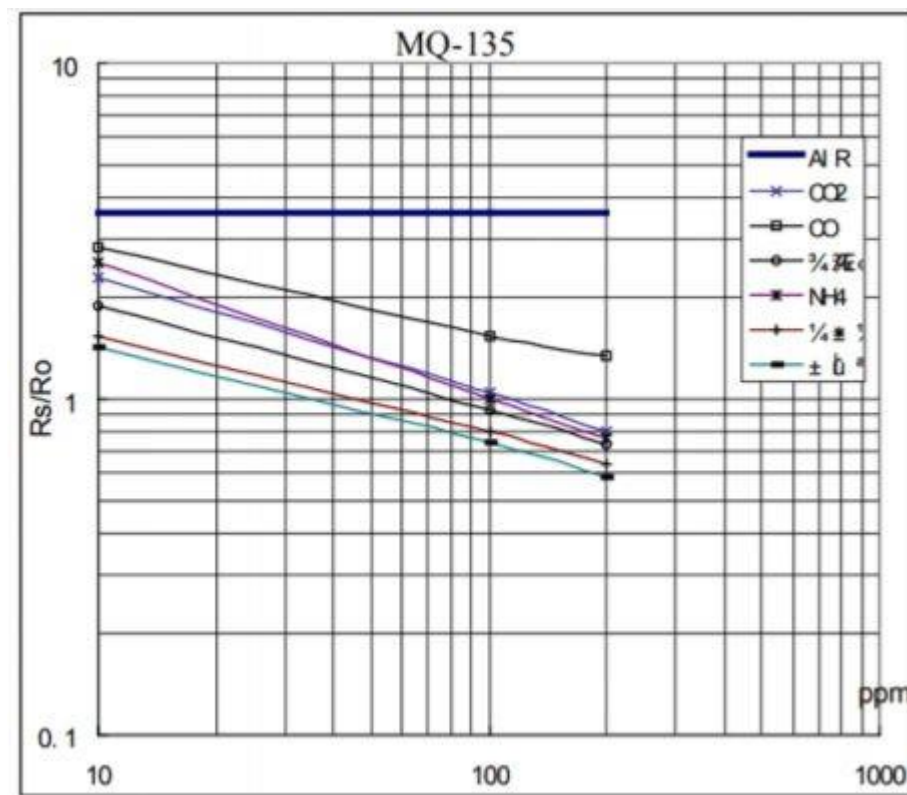
Tabel 5. Sensitivity Characteristic

Kegunaan MQ135:

- Mendeteksi pada ruang lingkup yang luas
- Mempunyai respons yang cepat dan sensitivitas yang tinggi
- Output tegangan analog 0 - 5 V
- Output tegangan digital 0 - 5 V
- Dapat digunakan sebagai sensor digital atau analog
- Mempunyai stabilitas yang tinggi dan tahan lama
- Beroperasi pada tegangan +5 V
- Digunakan untuk mengukur/mendeteksi NH₃, Nox, alcohol, Benzene, asap, dan Co₂

MQ135 untuk mengukur PPM

Pada MQ135, pada proyek embedded kami digunakan untuk mendeteksi kadar Co2. Berikut merupakan cara pengukuran Co2. Untuk mengukur PPM dengan menggunakan MQ135 mula-mula perlu mengikuti grafik PPM yang terdapat pada datasheet sensor ini. Berikut merupakan grafiknya:



Gambar 4. Grafik PPM dan gas lin pada MQ135

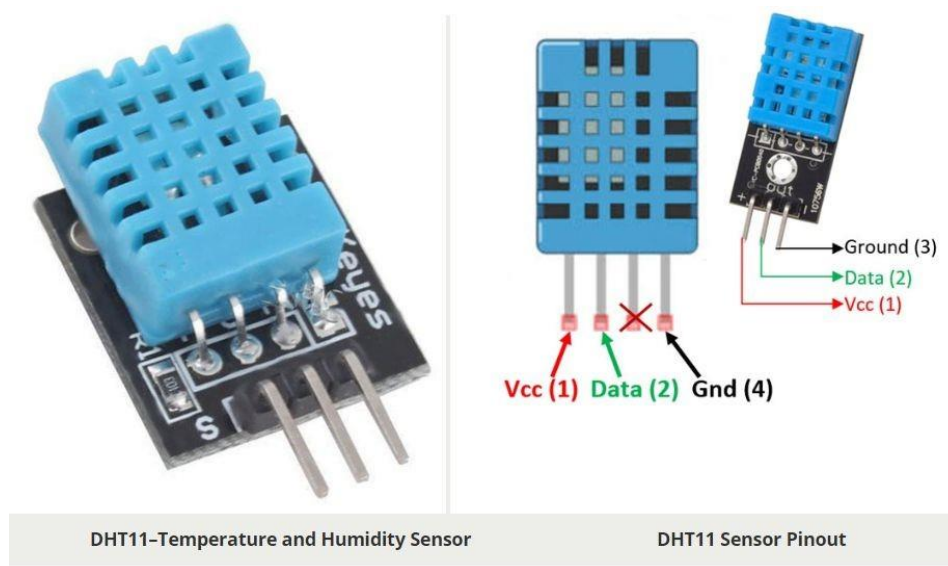
Pada gambar tersebut menunjukkan karakteristik tingkat sensitivitas dari MQ135 pada pendeteksian gas. Grafik yang menunjukkan R_s/R_o , menunjukkan bahwa,

R_o : merupakan resistansi sensor pada 100ppm NH_3 udara bersih

R_s : merupakan nilai resistansi sensor pada konsentrasi udara/gas

formula sebagai berikut, resistanse sensor (**R_s**): **$R_s = (V_c/V_{RL} - 1) \times R_L$**

2.1.3 DHT11



Gambar 5. Data pin DHT11

Pin Sensor:

Pin no.	Pin name	Keterangan
1	Vcc	Power supply 3.5 - 5.5 V
2	Data	Data keluaran adalah temperature dan kelembapan melalui serial data
3	NC	No connection
4	Ground	Dihubungkan ke ground

Tabel 6. Pin sensor DHT11

Pin modul:

Pin no.	Pin name	Keterangan
1	Vcc	Power supply 3.5 - 5.5 V
2	Data	Data keluaran adalah temperature dan kelembapan melalui serial data
3	Ground	Dihubungkan ke ground

Tabel 7. Pin modul DHT11

Fitur pada DHT11

- Beroperasi pada tegangan 3.5 - 5.5 V
- Beroperasi pada arus 0.3 mA (measuring) 60uA (standby)
- Data keluaran adalah serial data
- Temperature range: 0°C to 50°C
- Humidity range: 20% to 90%
- Resolution: temperature and humidity both are 16 bit
- Tingkat akurasi: $\pm 1^{\circ}\text{C}$ and $\pm 1\%$

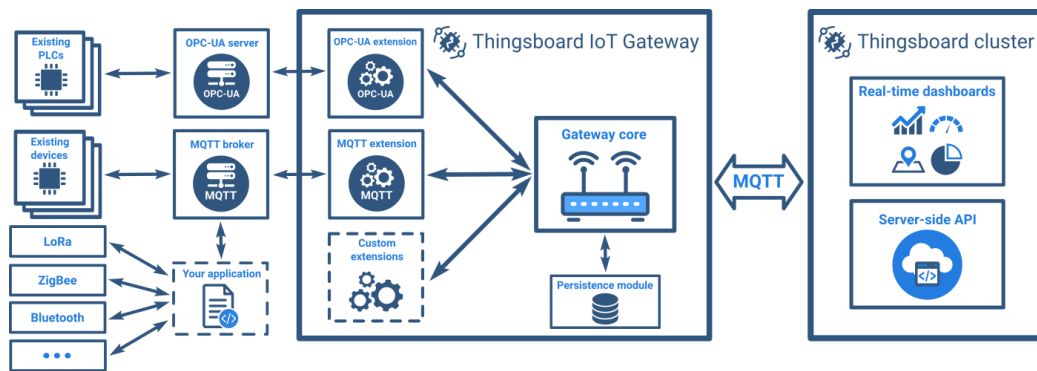
2.1.4 Thingsboard



Gambar 6. Tampilan menu login Thingsboard

Thingsboard merupakan platform IoT yang open source. Thingsboard digunakan untuk menghubungkan device ke suatu webserver yang akan diolah pada database. Data tersebut dapat dilihat dan diamati serta dapat dilakukan export data dalam bentuk CSV atau file-file lainnya seperti JSON dan lain-lain. Thingsboard dapat diakses dimanapun.

Fitur-fitur pada thingsboard:

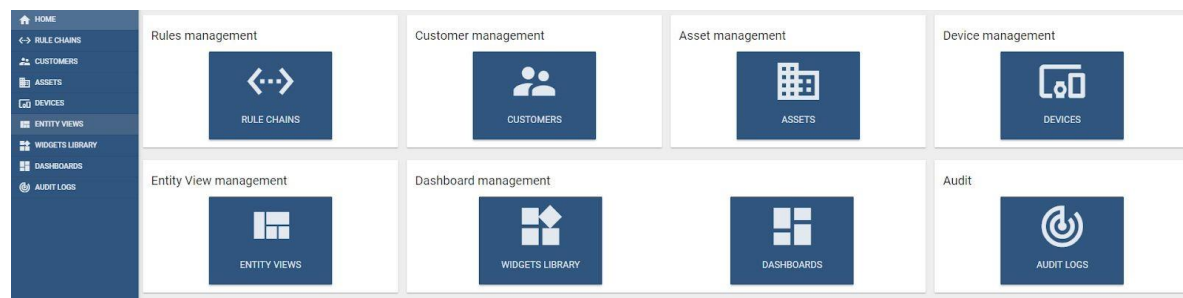


Gambar 7. Fitur yang dapat digunakan pada Thingsboard

Gambar diatas merupakan fitur-fitur yang dapat dijalankan dan dioperasikan pada thingsboard. Pengiriman data dilakukan dengan menggunakan MQTT.

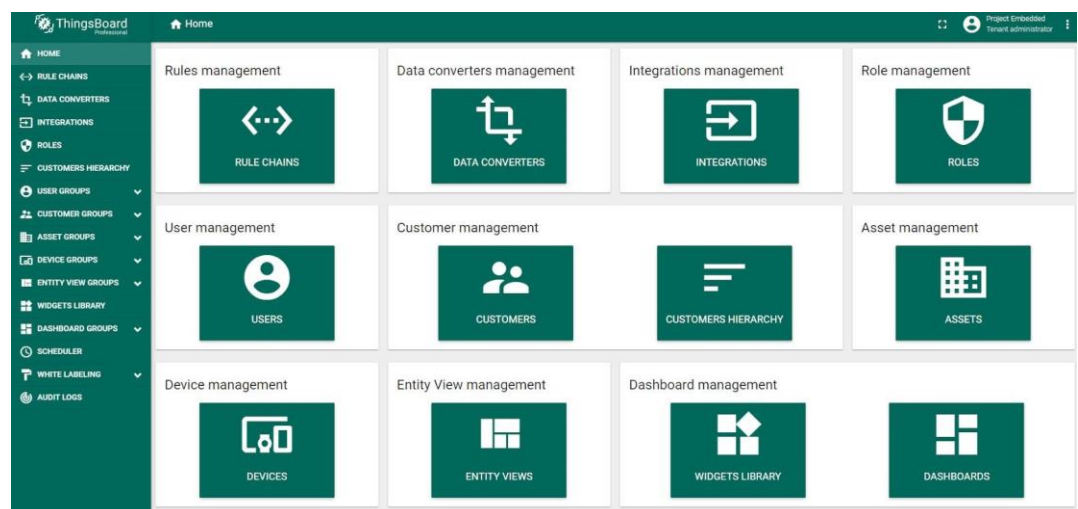
Tampilan thingsboard

- Live Demo



Gambar 8. Menu Live Demo pada thingsboard

- Profesional



Gambar 9. Menu Professional mode pada thingsboard

Perbedaan keduanya lebih terdapat pada fitur-fitur yang diberikan, untuk mode professional mempunyai fitur-fitur yang banyak dan lengkap.

2.1.5 Parameter Skala Udara

ISPU	Pencemaran Udara Level	Dampak kesehatan
0 - 50	Baik	tidak memberikan dampak bagi kesehatan manusia atau hewan.
51 - 100	Sedang	tidak berpengaruh pada kesehatan manusia ataupun hewan tetapi berpengaruh pada tumbuhan yang peka.
101 - 199	Tidak Sehat	bersifat merugikan pada manusia ataupun kelompok hewan yang peka atau dapat menimbulkan kerusakan pada tumbuhan ataupun nilai estetika.
200 - 299	Sangat Tidak Sehat	kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar.
300 - 500	Berbahaya	kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius pada populasi (misalnya iritasi mata, batuk, dahak dan sakit tenggorokan).

sumber: <https://www.climate4life.info/2019/08/kualitas-udara-memburuk-pahami-indeks-standar-pencemar-udara-dan-dampaknya-pada-kesehatan.html>

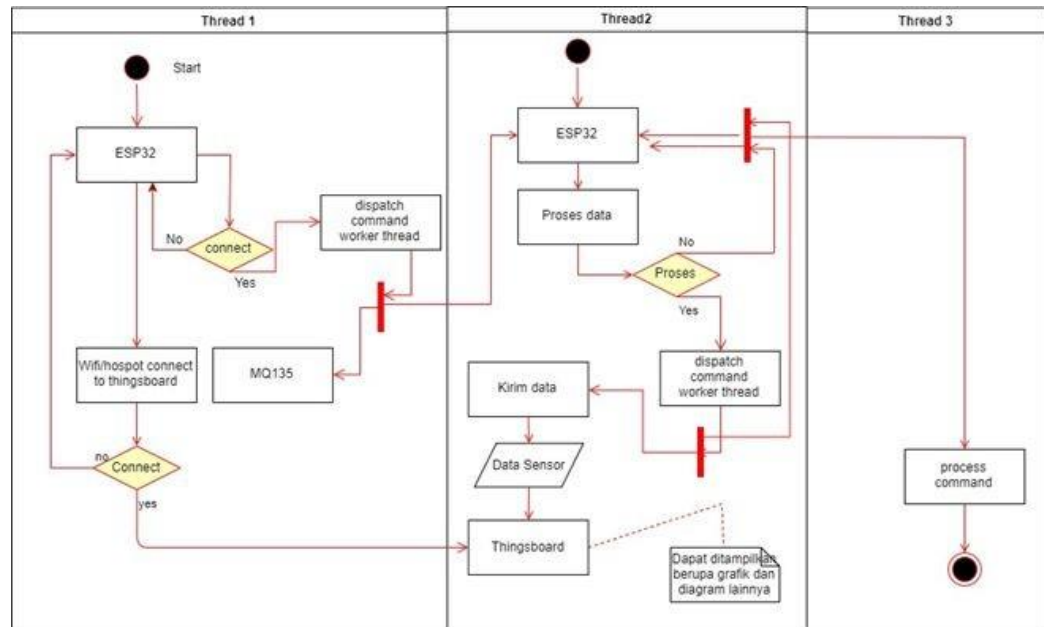
Gambar 10. Skala index udara

2.2 Perancangan Project

2.2.1 UML

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang terdiri dari serangkaian diagram terintegrasi, yang dikembangkan untuk membantu pengembang sistem dan software untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem software, serta untuk pemodelan bisnis dan non-software system pemodelan lainnya. UML mewakili sekumpulan praktik teknik terbaik yang telah terbukti berhasil dalam pemodelan sistem besar dan kompleks. UML adalah bagian penting dari pengembangan software berorientasi objek dan proses pengembangan perangkat lunak. Berikut ini adalah contoh tentang UML project kita, seperti:

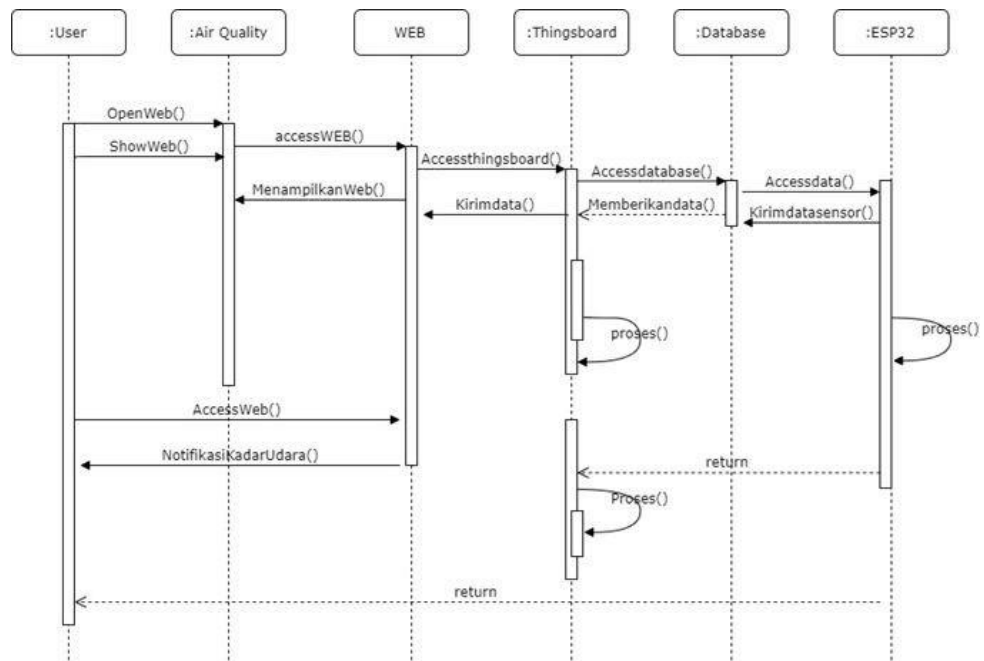
a. active-UML



Gambar 11. active-UML

Menjelaskan serangkaian alur aktivitasnya, digunakan untuk menggambarkan aktivirasnya yang dibentuk dalam suatu operasi sehingga juga dapat digunakan untuk aktivitas lain seperti use cases atau berbagai interaksi.

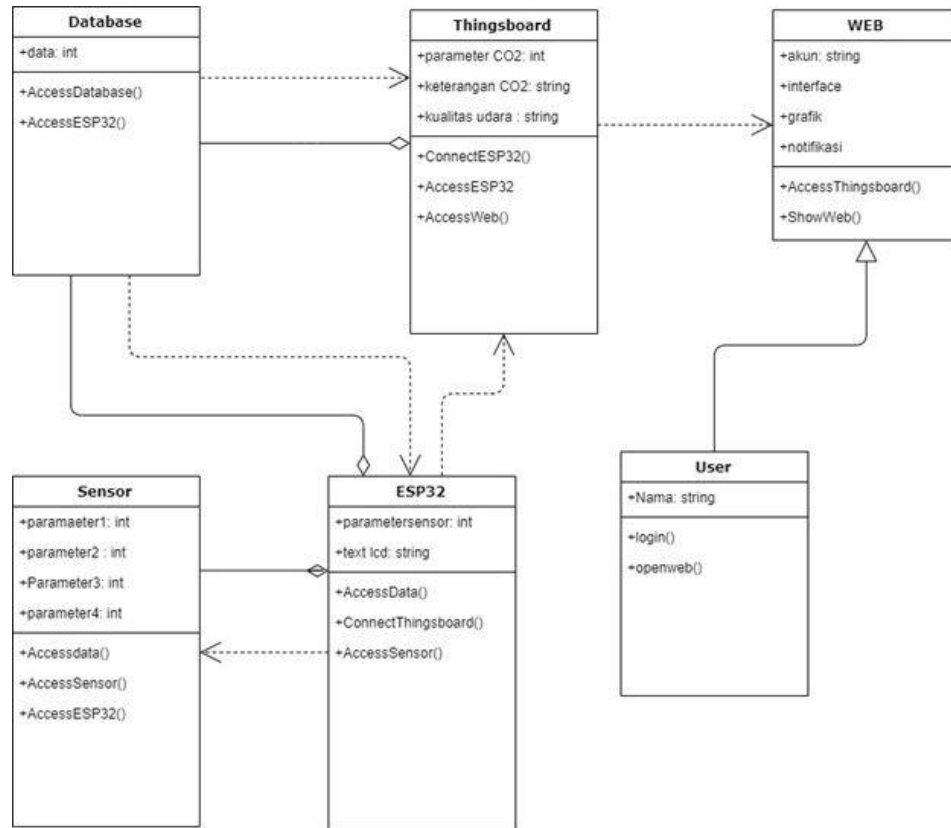
b. sequence-UML



Gambar 12. sequence-UML

Menjelaskan kolaborasi dinamis antara beberapa objek. Tujuannya adalah untuk menunjukkan urutan pesan yang dikirim antara objek serta interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

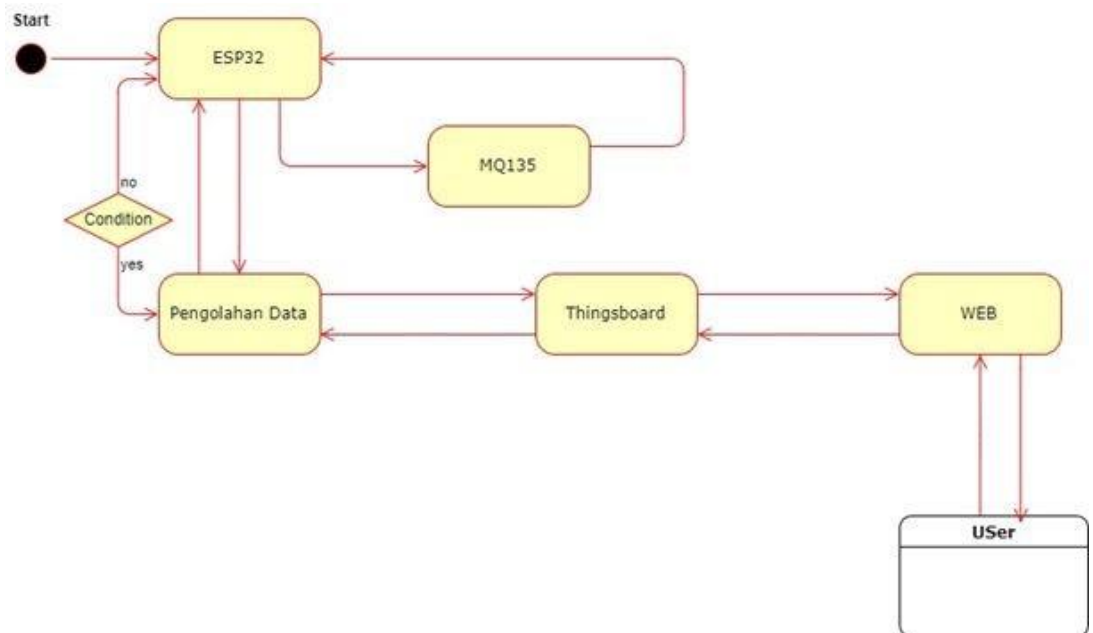
c. case-UML



Gambar 13. case-UML

Menjelaskan struktur statis dalam sistem. Kelas-kelas mewakili sesuatu yang dihandle oleh sistem. Kelas-kelas dapat dihubungkan satu sama lain dalam berbagai cara: associated (connected), dependent (satu kelas tergantung pada atau menggunakan kelas lain), specialied (satu kelas adalah spesialisasi kelas lain), atau package (kelompok bersama sebagai satu unit). Suatu sistem biasanya memiliki beberapa kelas diagram.

d. state-UML



Gambar 14. state-UML

Menjelaskan beberapa aktor eksternal dan hubungannya dengan use case yang disediakan oleh sistem. Use case adalah deskripsi dari fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi simbol use case tetapi juga dapat dilakukan dalam diagram aktivitas. Use cases hanya dijelaskan yang dilihat dari luar oleh aktor (keadaan lingkungan sistem yang dilihat pengguna) dan bukan bagaimana fungsinya dalam sistem.

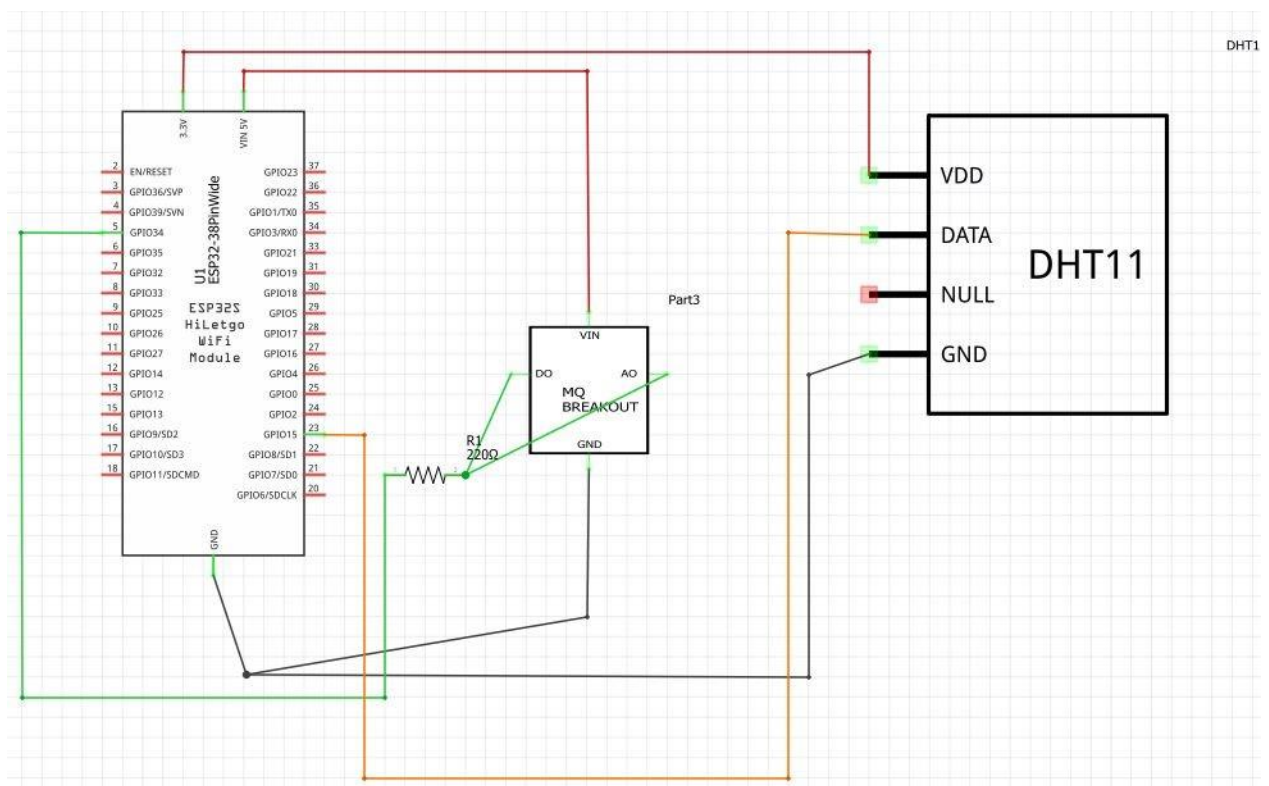
2.2.2 Bahasa Program

Bahasa program yang kami gunakan untuk program ESP32 adalah bahasa pemograman Arduino dengan menggunakan Arduino IDE (Integrated Development Environment). Software ini merupakan software text editor untuk melakukan program editing code, pembuatan code, dan mengcompile code ke board ESP32. Program yang dicompile akan otomatis diberi folder sendiri dan nama sketch yang sama dengan nama program yang disimpan. Sketch pada Arduino seperti pada bahasa C/C++ terdapat 3 bagian utama structure, function, dan values. Structure seperti setup() dan loop(), function seperti fungsi segmentasi code yang berisi tugas-tugas yang dapat

diulang atau yang lainnya. Dan values merupakan variable dan konstanta. Berikut merupakan sedikit dari program yang kami telah buat.

Pengiriman data yang dilakukan menggunakan MQTT. MQTT merupakan protokol komunikasi untuk IoT. MQTT berjalan di atas Transmission Control Potokol/Internet Protocol (TCP/IP). Digunakannya MQTT karena lebih ringan dibandingkan dengan protokol HTTP 1.1, sehingga cocok untuk digunakan pada perangkat berdaya rendah yang diharuskan mengirimkan dan menerima data dengan ukuran sekecil mungkin.

2.2.3 Schematic Alat



Gambar 15. Schematic hubungan ESP32 dengan sensor

Pada wire yang berwarna merah merupakan wire yang terhubung ke tegangan. Untuk MQ135 wire dihubungkan ke tegangan 5 V. Sedangkan untuk DHT11 wire dihubungkan ke 3 V.

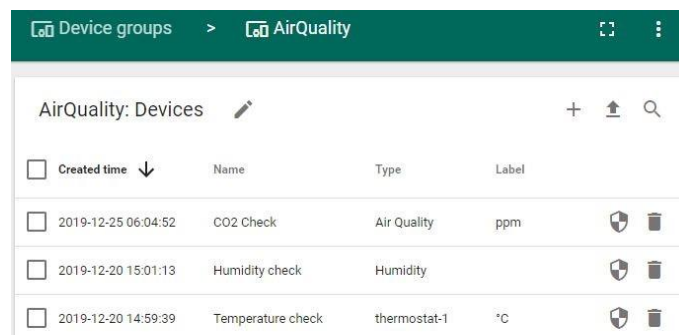
Untuk wire yang berwarna hijau, merupakan pin dari MQ135 yang terhubung ke pin 35. Dan juga wire yang berwarna orange dihubungkan ke pin 15 untuk sensor DHT11.

Kemudian, wire berwarna hitam dihubungkan ke ground untuk setiap sensor.

2.2.4 Fitur-fitur Platform yang digunakan

Data yang diambil oleh sensor dan dikirim ke thingsboard merupakan data yang realtime. Pengambilan data dilakukan beberapa detik, sehingga mendapatkan hasil yang realtime. Pada thingsboard dapat dilakukan pengambilan data realtime yang dirata-ratakan atau dijumlah sesuai kebutuhan user. Untuk proyek yang kami buat, kami merata-ratakan hasil dari beberapa detik itu, dan hasil yang ditampilkan pada dashboard adalah data rata-rata pada detik tertentu. Berikut tampilan dari fitur thingsboard yang kami digunakan sebagai berikut, yaitu:

- Device

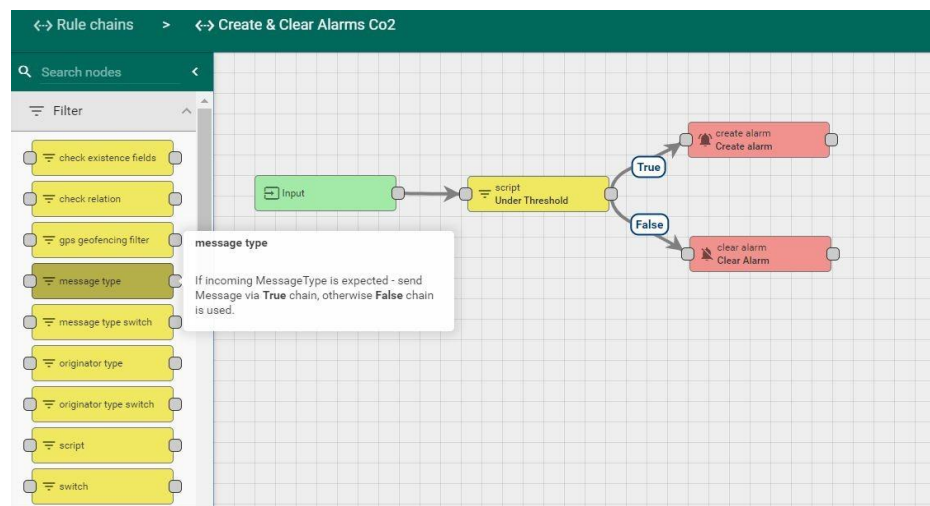


Created time	Name	Type	Label
2019-12-25 06:04:52	CO2 Check	Air Quality	ppm
2019-12-20 15:01:13	Humidity check	Humidity	
2019-12-20 14:59:39	Temperature check	thermostat-1	°C

Gambar 16. device yang dipakai

Gambar device tersebut merupakan device nama-nama device yang digunakan untuk mendeteksi, device yang kami gunakan adalah ESP32.

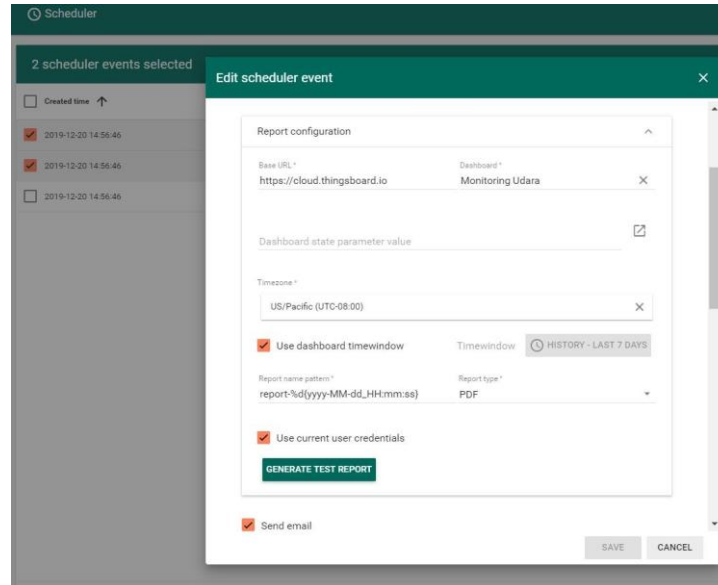
- Rule Chains



Gambar 17. Rule chains

Rule chains yang kami gunakan untuk memberitahu notifikasi jikalau suhu, kelembapan, dan Co2 melebihi batas yang sudah ditentukan.

- Scheduler



Gambar 18. scheduler

Scheduler digunakan untuk menampilkan dan update data pada waktu periode yang sudah ditentukan.

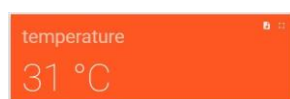
- Dashboard

1. Co2 widget



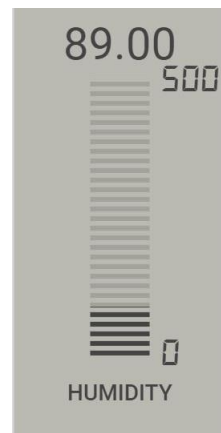
Gambar 19. Co2 widget

2. Temperature widget



Gambar 20. temperature widget

3. Humidity widget



Gambar 21. humidity widget

4. Table realtime data widget

New Timeseries table

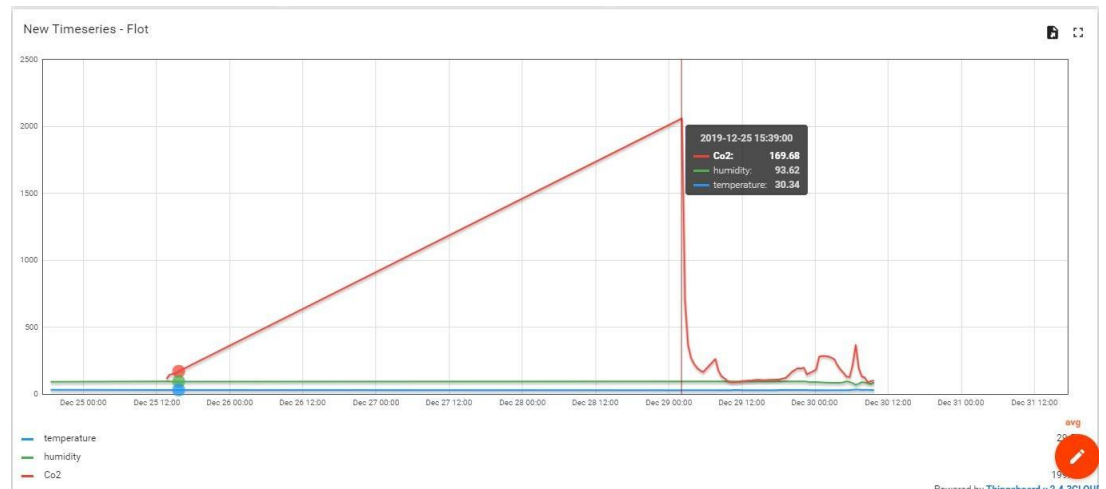
🕒 Realtime - last 5 minutes

Timestamp ↓	temperature	humidity	Co2
2019-12-29 06:27:44		95	190
2019-12-29 06:27:44	29.3		
2019-12-29 06:27:42		95	193
2019-12-29 06:27:41	29.3		
2019-12-29 06:27:40	29.3	95	193
2019-12-29 06:27:37	29.3		
2019-12-29 06:27:36		95	189
2019-12-29 06:27:35	29.3		

Page: 1 1 - 10 of 285 < >

Gambar 22. data realtime widget

5. Grafik widget

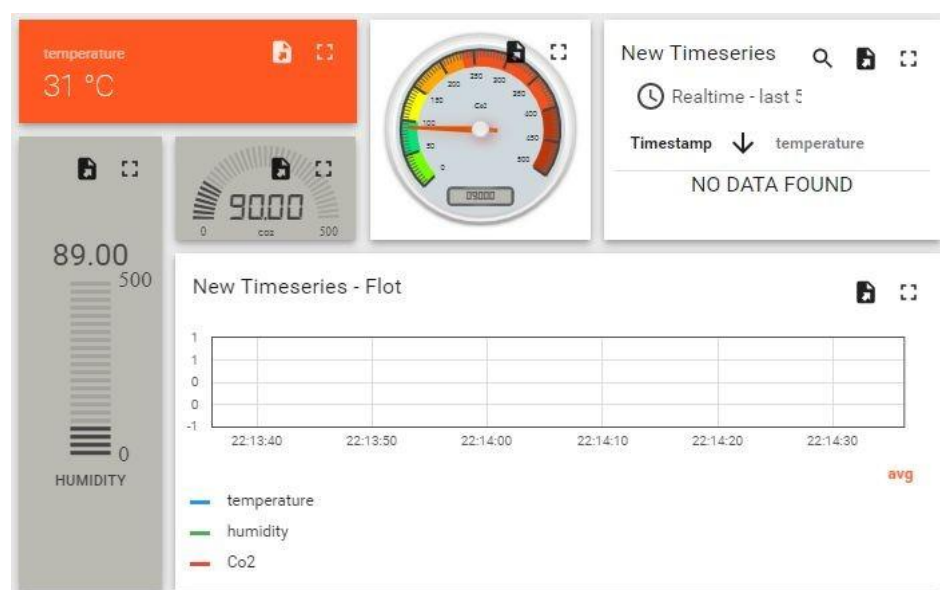


Gambar 23. Grafik widget

Grafik pada Gambar 23 merupakan grafik history dari data yang diambil pada awal pengambilan tanggal 24 Desember 2019 sampai dengan 31 Desember 2019. Dapat dilihat grafik mengalami kenaikan yang drastis pada sensor MQ135 dikarenakan sensor terkena pertikan air. Sehingga data yang diambil tidak stabil.

2.2.5 Hasil Data

- Lokasi 1



Gambar 24. tampilan dashboard lokasi 1

- Format pengambilan data secara realtime

Timestamp	Co2	humidity	temperature
12/29/2019 22:59	155	92	
12/29/2019 22:59	155	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	157	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	154	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	157	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	157	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	160	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	153	92	
12/29/2019 22:59	155	92	28.5
12/29/2019 22:59			28.5
12/29/2019 22:59	154	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	157	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	158	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	156	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	158	92	
12/29/2019 22:59			28.5
12/29/2019 22:59	156	92	
12/29/2019 22:59			28.5

Tabel 8. Format data tabel pengampilan data pada Thingsboard

2.3 Pencapaian

Projek kami sudah dapat mengambil data dan mengkoneksikannya ke platform IoT seperti yang kami gunakan adalah thingsboard serta data yang didapatkan dapat ditampilkan melalui telegram bot.

Untuk pengambilan data, kami sudah mendapatkan data secara realtime. Sehingga dapat dipantau data secara kontinu pada dashboard thingsboard untuk lokasi 1 dan lokasi 2.

Sumber daya yang kami gunakan, sudah dapat bertahan sekitar 3 hari dengan pengambilan data yang tidak berhenti.

Namun, untuk pengambilan data yang dilakukan dengan kondisi mode *deepsleep* pada ESP32 menggunakan waktu 5 menit pengambilan data dan 10 menit *deepsleep*. Hal itu dapat membuat perangkat dapat mengambil data sekitar kurang dari 7 hari.

Dan juga data yang sudah didapat tersimpan pada thingsboard yang akan ditampilkan pada grafik history. Sehingga dapat dilihat pola naik-turun data pada periode yang diinginkan.

2.4 Kendala/Permasalahan

Projek yang kami buat mempunyai beberapa kendala yang mungkin akan menjadi kurang efisien jika dilakukan pengamatan yang lama, yaitu sebagai berikut:

1. Sumber daya supply baterai yang bertahan tidak lebih dari seminggu (menggunakan deepsleep)
2. Ukuran produk yang masih terlalu besar
3. Masih membutuhkan data paket dari hotspot atau wifi pribadi, belum wifi umum (Hotspot UI)
4. Belum menggunakan mesh antar ESP32 (satu ESP32 menjadi router)

Kendala yang ada adalah ketika musim penghujan sensor yang ada riskan sekali terkena percikan air walaupun sudah ditempatkan pada lokasi yang aman. Ini akan mengakibatkan sensor mengalami ketidakstabilan dalam mengambil data. Hal ini terjadi ketika kami mengambil data dan kemudian sensor, terutama sensor MQ135 yang datanya menjadi melonjak sangat tinggi dari data awal 185 menjadi 3024 dan seterusnya. Keadaan ini kembali stabil sekitar 1 sampai 2 menit ke depan.

2.5 Product Physics



Gambar 25.. Project secara fisik

Pada produk yang kami kerjakan dalam proyek Sistem Embedded 1 ini, kami menggunakan desain berbentuk Bis Kuning yang menggambarkan salah satu ikon dari UI yang diharapkan cukup menarik perhatian dan *aesthetic*.

Bahan yang digunakan sebagai rangka luar produk adalah bahan akrilik yang tebal sehingga diharapkan melindungi bagian dalam produk dari debu dan air. Akrilik tersebut direkatkan menggunakan lem agar dapat merekat dengan baik, lalu bahan akrilik tersebut dilapisi dengan bahan kertas berwarna kuning yang menyesuaikan warna dari Bis Kuning serta dihias dengan spidol untuk menambahkan *line art*. Bahan ban dari produk menggunakan plastik tutup dari botol bekas sebanyak 4 buah agar bisa menopang produk.

Kekurangan dari desain ini adalah belum dibuatnya lubang akses untuk *switch on/off* dari produk, sehingga ketika ingin mematikan atau menyalakan produk harus membuka bagian atas terlebih dahulu lalu menekan tombol *switch* yang kecil. Selain itu, pada proses pengambilan data sensor MQ-135 dan DHT-11 sempat *down* dikarenakan terkena cipratan air saat hujan. Hal ini bisa terjadi dikarenakan pada desain produk dibuat sebuah lubang agar kedua sensor tersebut pada posisi ke arah luar sehingga sensor terkena air yang menyebabkan *down* selama beberapa saat.

Untuk perkembangan kedepannya, kami akan melakukan beberapa perubahan dalam segi desain produk yaitu seperti penambahan kanopi pada bagian sensor sehingga sensor tidak terkena cipratan air untuk kedepannya.

2.5.1 Kerangka fisik

- > Kerangka menggunakan akrilik

- > Ukuran dari akrilik:

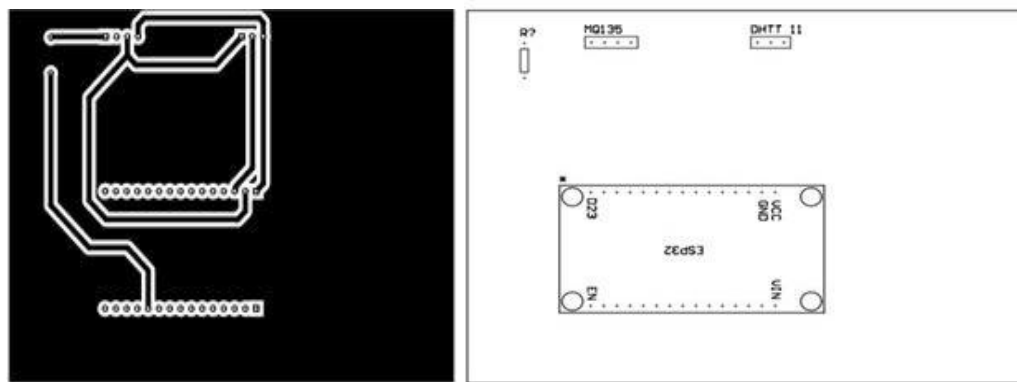
- Lebar : 10 cm
- Panjang : 15 cm
- Tinggi : 8 cm

Akrilik tersebut dibentuk menjadi seperti bentuk balok sebagai body dari Monira yang berbentuk seperti bus kuning

> Mengapa kita menggunakan akrilik karena akrilik mudah dibentuk dan bahkan disuhu dingin pun dia masih bisa mempertahankan bentuknya. Bahan akrilik juga tahan terhadap cuaca, sehingga alat-alat elektronik yang berada didalam akrilik tersebut tidak terhantar dampak dari cuaca tersebut. Selain itu juga, akrilik transparan, sehingga alat elektroniknya bisa dari bagian atas.

2.5.2 Kerangka elektronik

> schematic pcb



Gambar 26. Schematic PCB

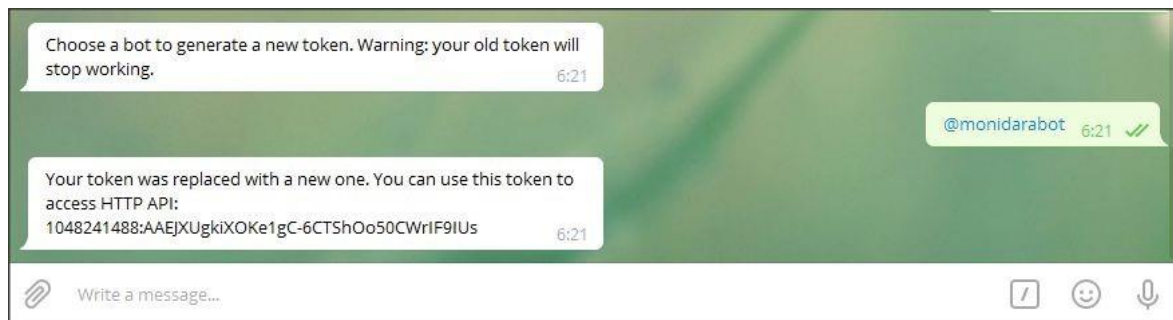
> ukuran

- Tebal : 0.2 cm
- Lebar : ± 7 cm
- Panjang : ± 9 cm

2.6 Implementasi

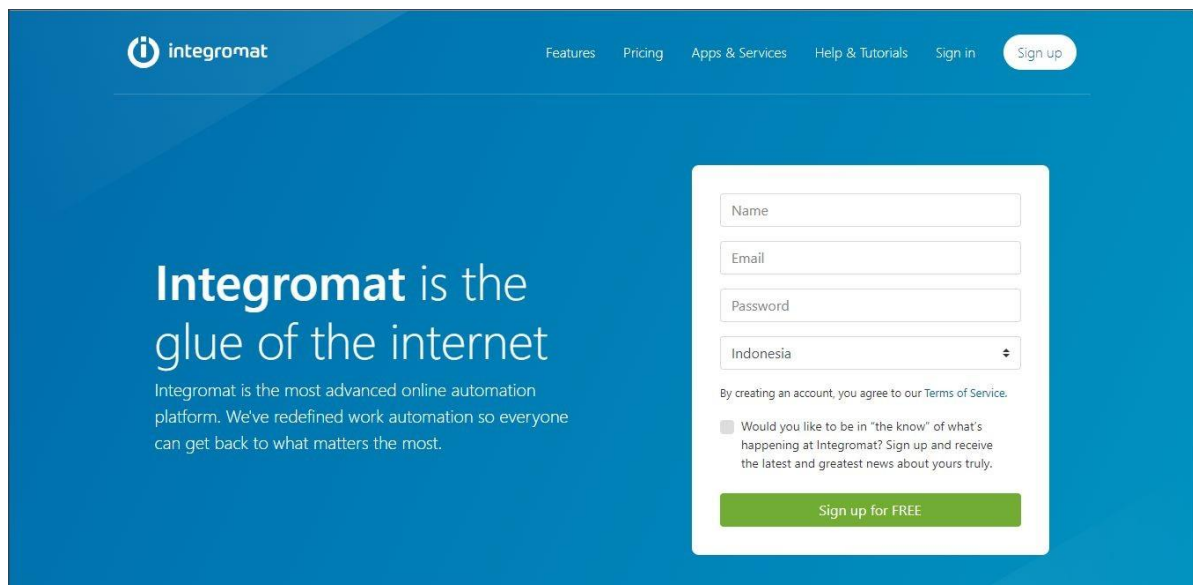
Kami mencoba menerapkan project ini dengan menggunakan integrasi aplikasi yang mana dapat memudahkan pengguna memantau secara langsung data yang diambil oleh sensor kapanpun dan dimanapun pengguna berada. Dalam hal ini kami memanfaatkan fitur Bot pada aplikasi android yaitu Telegram.

Langkah pertama yang dilakukan adalah membuat terlebih dahulu bot Telegram melalui “Bot Father”, yaitu fitur Telegram yang dapat memudahkan pengguna membuat Bot dan mendapatkan API Token dari bot tersebut.

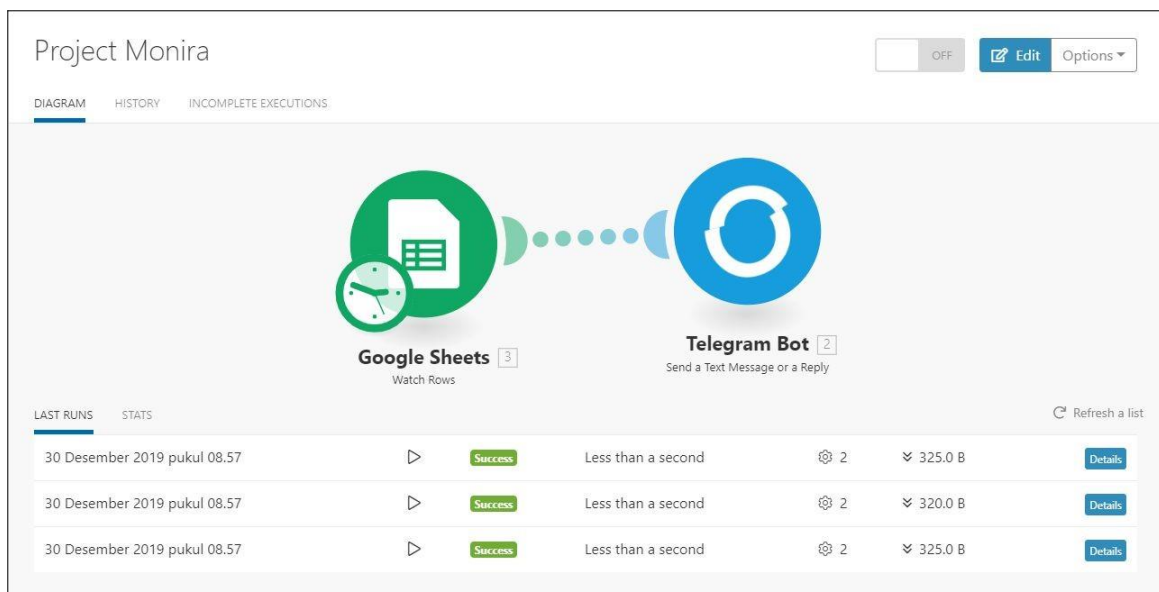


Gambar 27. API token bot

Dengan menggunakan API Token tersebut kami bisa mengintegrasikannya dengan data sensor yang telah dihasilkan (Format file Google Spreadsheet) dengan menggunakan platform Integromat.

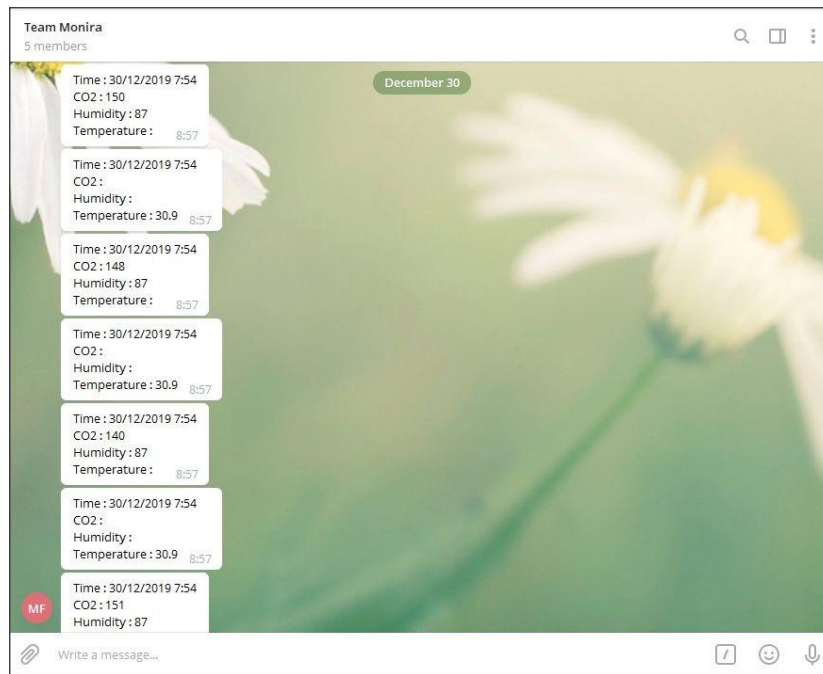


Gambar 28.(a)Koneksi datasheet dari thingsboard ke telegram



Gambar 28.(b)Koneksi datasheet dari thingsboard ke telegram

Dengan menerapkan beberapa konfigurasi pada platform ini kami dapat dengan mudah mengintegrasikan data yang dihasilkan dengan aplikasi telegram. Hasilnya yaitu sebagai berikut :

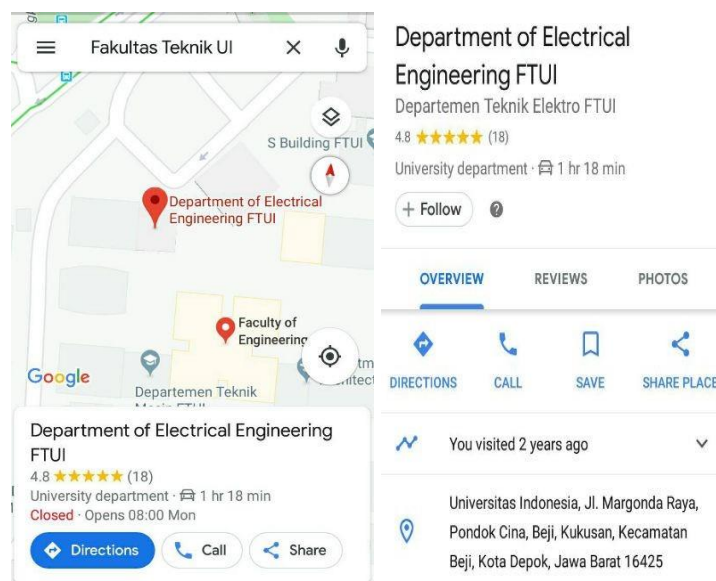


Gambar 29. Tampilan data pada telegram

Data akan secara otomatis terkirim ke Grup ini dengan menggunakan fitur bot telegram yang telah diintegrasikan dengan menggunakan API.

2.7 Lokasi (Share Locate)

Alat yang kita gunakan diletakkan di kolam depan Departemen Teknik Elektro dimana kita menggunakan wifi sendiri untuk mengambil datanya.



Gambar 30. Lokasi Embedded

2.8 Penempatan IoT



Gambar 31. Lokasi penempatan alat di kolam Departemen Elektro UI

Alat yang kami buat, kami letakan di kolam Departemen Teknik Elektro Universitas Indonesia. Kami letakan ditempat tersebut untuk mengetahui kelembapan yang terjadi di lokasi tersebut. Tidak hanya itu, kami juga ingin mengetahui kadar Co2 dan temperature di lokasi tersebut.

BAB III

PENUTUP

3.1 Kesimpulan

Monira merupakan alat monitoring udara yang desain fisiknya seperti bikun UI dimana dia dapat mendeksi kadar CO₂, kelembapan, maupun suhu di sekitar sensor pada Monira tersebut. Data yang kita ambil bertempat di FT UI tepatnya di kolam depan Departemen Teknik Elektro. Dengan adanya Monira ini, kita dapat mengetahui bagaimana keadaan lingkungan di sekitar alat tersebut dan juga harapannya Monira dapat menolong lingkungan sekitar agar menjadi lebih sehat dan jauh dari dampak-dampak yang buruk dengan memonitornya dari situs web. Kami juga berharap orang-orang menjadi lebih peduli dengan kualitas udara di sekitar mereka.

3.2 Saran

Monira tentunya akan menampilkan kadar CO₂, kelembapan, maupun suhu dimana pada platform yang kita buat akan menampilkan batas kadar udaranya, jadi kita bisa tau kadar udara di tempat yang kita jadikan tempat monitoring tersebut (FT UI) masih dalam status aman atau darurat. Oleh karena itu, kami mengharapkan dengan adanya Monira ini, civitas FT UI bisa mengurangi pemakaian kendaraan bermotor, pembakaran sampah, dan hal-hal lain yang dapat menyebabkan CO₂ meningkat

REFERENSI

- <https://www.cytron.io/p-nodemcu-esp32>
- <https://www.nyebarilmu.com/perbedaan-module-wifi-esp8266-vs-esp32/>
- <https://www.arduinoindonesia.id/2019/07/memanfaatkan-nilai-adc-pada-esp32-untuk.html>
- <https://www.instructables.com/id/Air-Quality-Meter/>
- <https://www.climate4life.info/2019/08/kualitas-udara-memburuk-pahami-indeks-standar-pencemar-udara-dan-dampaknya-pada-kesehatan.html>