

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebagai bahan pertimbangan dalam menyelesaikan tugas akhir ini penulis melakukan studi literatur terkait sensor – sensor yang berkaitan dengan kualitas udara. Berikut beberapa penelitian terkait.

1. Berdasarkan penelitian yang dilakukan oleh Rodhotul Muttaqin, Wasi Sakti Wiwit Prayitno, Natalia Erna Setyaningsih, Upik Nurbaiti dengan judul “Rancang Bangun Sistem Pemantauan Kualitas Udara Berbasis Iot (Internet Of Things) dengan Sensor DHT11 dan Sensor MQ135”, Penelitian ini bertujuan untuk merancang dan membangun sistem pemantauan kualitas udara berbasis Internet of Things (IoT) di Laboratorium Fisika Material Universitas Negeri Semarang. Sistem ini menggunakan dua sensor utama, yaitu sensor MQ-135 untuk mendeteksi gas berbahaya seperti CO₂, alkohol, dan aseton, serta sensor DHT11 untuk mengukur suhu dan kelembapan udara. Kedua sensor dihubungkan dengan mikrokontroler NodeMCU ESP32 yang memiliki konektivitas Wi-Fi, sehingga data hasil pengukuran dapat dikirim secara real-time ke cloud dan dipantau melalui platform Blynk IoT. Hasil pengujian menunjukkan bahwa sistem mampu memantau perubahan kadar gas dan parameter lingkungan dengan baik. Grafik pengukuran menunjukkan pola regresi linier dengan koefisien determinasi tinggi untuk suhu (0,98), kelembapan (0,94), alkohol (0,85), aseton (0,97), dan CO₂ (0,93), yang menandakan keandalan sensor dalam merespon perubahan di lingkungan laboratorium. Sistem ini juga dilengkapi dengan buzzer sebagai alarm jika kualitas udara berada pada level tidak aman.[4]
2. Berdasarkan penelitian yang dilakukan oleh Rifa Aghorru dan Muhammad Kopravi dengan judul “Rancang Bangun Sistem Pemantau Kualitas dan Polusi Udara PM_{2.5} yang Terintegrasi dengan Platform IoT”, Penelitian ini membahas perancangan dan implementasi alat pemantau kualitas udara, khususnya polutan partikulat PM_{2.5}, yang terintegrasi dengan platform Internet of Things (IoT).

Alat ini menggunakan sensor debu PM2.5 GP2Y1010AU0F dan sensor suhu serta kelembapan DHT22 yang dikendalikan oleh mikrokontroler NodeMCU dengan modul WiFi ESP8266. Data hasil pengukuran dapat dipantau secara real-time melalui platform Blynk yang dapat diakses menggunakan smartphone atau desktop dengan koneksi internet. Sistem ini juga dilengkapi dengan fitur notifikasi otomatis yang akan mengirimkan peringatan kepada pengguna jika konsentrasi PM2.5 melebihi ambang batas yang ditetapkan oleh BMKG. Hasil pengujian menunjukkan bahwa alat ini memiliki presisi yang baik, dengan standar deviasi pengukuran PM2.5 sebesar 22,1 dan rata-rata 47,0 $\mu\text{g}/\text{m}^3$. Pengujian dilakukan di area dekat jalan raya pada berbagai waktu (pagi, siang, sore, malam) untuk memastikan keandalan alat dalam berbagai kondisi lingkungan.[5]

Pada penelitian yang pernah dilakukan sebelumnya, terkhusus mengenai pemantauan kualitas udara, jika dilihat dari parameter – parameter yang diukur seperti gas CO₂, NH₃, suhu kelembapan dan PM 2.5 belum ada penelitian yang mengukur parameter – parameter tersebut secara bersamaan. Jika dilihat dari komponen sensor yang digunakan seperti penggunaan sensor MQ135, DHT22, dan DSM501A untuk memantau kualitas udara dengan dukungan ESP32 sebagai mikrokontroler belum ada ditemukan. Media visualisasi dari sistem ini menggunakan DMD P10, serta memanfaatkan data logger berbasis cloud dengan menggunakan platform Arduino IoT Cloud. Dikarenakan sistem ini menggunakan DMD P10 dan memanfaatkan platform Arduino IoT Cloud memungkinkan sekitar untuk memantau kualitas udara secara real time dan jarak jauh. Sistem ini juga menyediakan visualisasi tampilan yang menyorot pada satu blok parameter yang diukur apabila parameter yang diukur melebihi ambang batas.

2.2 Dasar Teori

2.2.1 Kualitas Udara

Udara adalah campuran gas yang sangat penting bagi kehidupan manusia dan makhluk hidup lainnya, memiliki sifat fisik yang dimana udara tidak memiliki rasa, maupun warna. Tanpa udara yang baik dan sehat manusia tidak dapat bertahan

hidup, demikian juga makhluk hidup lainnya. Oleh karena itu udara wajib dijaga kebersihan dan kualitasnya agar memenuhi syarat bagi kesehatan makhluk hidup.[1]

Dewasa ini, seiring berkembangnya zaman dan semakin meningkatnya kebutuhan manusia, mengakibatkan kualitas udara menjadi semakin menurun atau dengan kata lain mengakibatkan pencemaran udara. Sumber pencemaran yang kerap terjadi berasal dari kegiatan – kegiatan yang dilakukan manusia/makhluk hidup sehingga menimbulkan gangguan bagi lingkungan sekitarnya antara lain emisi dari pembakaran sampah, transportasi, dan aktivitas industri. Udara tercemar adalah udara yang tercampur dengan zat, energi, dan atau komponen lain yang tidak diinginkan. Gas yang berada di udara bervariasi, beberapa jenis gas yang terkandung di udara sebagai berikut.[2]

Dampak kesehatan yang sering terjadi oleh pencemaran udara adalah infeksi pada saluran pernapasan. Jika kualitas udara yang berada di sekitar buruk, maka akan mengganggu kesehatan dan kenyamanan lingkungan. Kualitas udara perlu memenuhi standar kesehatan. Dengan upaya penanggulangan kualitas udara yang buruk untuk menaikkan dan menstabilkan kualitas udara tersebut.

2.2.1.1 Gas Karbon Dioksida (CO₂)

Karbon dioksida (CO₂) adalah gas tak berwarna dan tak berbau, sejenis senyawa kimia yang terdiri dari dua atom oksigen yang terikat secara kovalen dengan sebuah atom karbon. Gas ini merupakan komponen penting dalam siklus kehidupan di bumi. Namun, aktivitas manusia meningkatkan konsentrasi gas CO₂, salah satu sumber penyumbang karbon dioksida adalah pembakaran bahan bakar fosil dan pengundulan hutan. Meningkatnya konsentrasi CO₂ berimplikasi luas bagi lingkungan luar ruangan dan kualitas udara dalam ruangan. Umumnya di lingkungan ruangan dengan ventilasi buruk, kadar CO₂ dapat meningkat dengan cepat. Ketika seseorang terpapar konsentrasi CO₂ yang tinggi, mereka dapat mengalami sejumlah gejala kesehatan.[3]

Gejala awal paparan CO₂ tinggi meliputi sakit kepala ringan, pusing, sesak napas, serta gangguan kognitif yang terlihat, seperti kesulitan berkonsentrasi dan

mengambil keputusan. Seiring meningkatnya kadar CO₂, dampak yang dirasakan dapat menjadi lebih serius, termasuk sakit kepala hebat, disorientasi, gangguan penglihatan dan pendengaran, bahkan kehilangan kesadaran. Dalam situasi ekstrem, hal ini bisa berujung pada kondisi fatal. Karena itu, pemantauan dan pengendalian kadar CO₂ dalam ruangan bukan hanya untuk menjaga kenyamanan penghuni, tetapi juga merupakan aspek krusial dalam kesehatan dan keselamatan publik.[3]

Menurut +iQAir sebuah platform untuk pemantauan kualitas udara, berikut penjelasan terkait besar konsentrasi gas CO₂ pada udara dapat dilihat pada Tabel 2.1 berikut.

Tabel 2. 1 Keterangan Kategori Konsentrasi CO₂[3]

Kategori (PPM)	Keterangan
400 – 1.000	Kondisi Normal
1.000 – 2.000	Menyebabkan gejala seperti kantuk, perasaan tersumbat, kebingungan ringan, disorientasi
2.000 – 5.000	Menyebabkan sakit kepala, mengantuk, kurang konsentrasi, mual
5.000 – 40.000	Kehilangan kesadaran, bahkan kematian

2.2.1.2 Gas Amonia (NH₃)

Senyawa utama yang dihasilkan oleh kendaraan bermotor adalah Karbon Monoksida (CO), namun terdapat pula senyawa lain yang terbentuk akibat gas buang, seperti Amonia (NH₃). Amonia merupakan senyawa anorganik yang tersusun atas satu atom nitrogen dan terikat dengan jenis ikatan kovalen dengan tiga atom hidrogen dan ditandai sebagai gas tidak berwarna dengan bau menyengat serta pemaparannya yang terjadi melalui saluran pernapasan atau dengan terkontak langsung. Biasanya, sumber utama Amonia (NH₃) berasal dari proses biologis di tanah serta pengolahan limbah industri. Namun demikian, NH₃ juga dapat menjadi polutan atmosfer yang berperan dalam beberapa masalah pencemaran udara. Sumber emisi gas NH₃ sebagian besar berasal dari kegiatan pertanian namun

di daerah perkotaan sumber emisi gas tersebut berasal dari transportasi atau kendaraan bermotor. Penelitian terbaru juga menunjukkan bahwa emisi NH_3 dari kendaraan bermotor berperan penting dalam pembentukan partikulat di atmosfer seperti $\text{PM}_{2.5}$. [6]

Dampak langsung dari NH_3 terhadap kesehatan manusia ditemukan pada beberapa kasus yang menyerang sistem pernapasan manusia. Selain itu, gas NH_3 adalah prekursor utama pembentukan aerosol sekunder yang erat kaitannya dengan $\text{PM}_{2.5}$, sehingga membuat pentingnya pemantauan NH_3 di udara terutama di daerah perkotaan [6]. Keputusan Menteri Negara Lingkungan Hidup Nomor 50 Tahun 1996 tentang Baku Tingkat Kebauan menyatakan konsentrasi NH_3 yang masih berada pada batas aman berada pada konsentrasi NH_3 yang tidak melebihi 2 PPM. Apabila konsentrasi NH_3 yang terukur melebihi dari 2 PPM, kualitas udara dikategorikan tercemar dan dapat membahayakan manusia. [7]

2.2.1.3 Particulate Matter 2.5 ($\text{PM}_{2.5}$)

Particulate matter (PM) adalah campuran kompleks partikel padat dan cair yang tersuspensi di udara. Sumber utama debu di atmosfer berasal dari tanah, kebakaran semak belukar, pembakaran sampah rumah tangga, kendaraan bermotor. PM memiliki ukuran besar yang beragam, faktor ukuran dari partikulat ini dianggap sangat penting dalam aspek kesehatan, sebab partikulat yang memiliki ukuran kecil yang kurang dari $10\mu\text{m}$ cenderung dapat merugikan manusia apabila terhirup. Partikulat berukuran kecil tidak dapat terlihat oleh mata dan apabila terhirup rambut – rambut di dalam hidung tidak dapat menyaring debu tersebut dan akan terjebak di sistem pernafasan manusia. [8]

$\text{PM}_{2.5}$ adalah partikel udara yang berukuran lebih kecil atau sama dengan $2,5\mu\text{m}$ (mikrometer). Sumber utama $\text{PM}_{2.5}$ berasal dari lalu lintas dan industri termasuk pembakaran bahan bakar dari pembangkit listrik, dan lain – lain. $\text{PM}_{2.5}$ termasuk kedalam PM yang kurang dari $10\mu\text{m}$, sehingga dapat merugikan manusia khususnya bagi masalah kesehatan manusia [8]. Tabel 2.2 menjelaskan keterangan dari besar konsentrasi $\text{PM}_{2.5}$ di udara, standar ini didapatkan berdasarkan penjelasan tingkat konsentrasi $\text{PM}_{2.5}$ di udara oleh BMKG (Badan Meteorologi,

Klimatologi, dan Geofisika).

Tabel 2. 2 Keterangan Konsentrasi PM 2.5 di udara[9]

Konsentrasi PM 2.5	Keterangan
0 – 15.5 $\mu\text{g}/\text{m}^3$	Baik
15.6 – 55.4 $\mu\text{g}/\text{m}^3$	Sedang
55.5 – 150.4 $\mu\text{g}/\text{m}^3$	Tidak Sehat
150.5 – 250.4 $\mu\text{g}/\text{m}^3$	Sangat Tidak Sehat
>250.4 $\mu\text{g}/\text{m}^3$	Berbahaya

2.2.1.4 Part Per Million (PPM)

Part per million (PPM) atau bagian per juta adalah satuan konsentrasi suatu zat yang digunakan untuk menunjukkan kandungan suatu senyawa dalam larutan, polutan dalam air atau udara, atau konsentrasi bahan kimia dalam suatu formulasi. Gas CO₂ dan NH₃ merupakan gas yang terkandung dalam udara menggunakan satuan PPM untuk menyatakan besar pengukuran konsentrasi gas tersebut.[1]

2.2.1.5 Suhu dan Kelembapan

Kualitas Fisik udara khususnya dalam ruangan rumah adalah parameter yang mengidikasi kondisi fisik seperti kelembapan, suhu, dan partikulat. Temperatur suhu di udara adalah ukuran energi kinetik rata – rata dari pergerakan molekul – molekul. Suhu suatu benda ialah keadaan yang menentukan kemampuan benda tersebut, untuk memindahkan (transfer) panas ke benda – benda lain atau menerima panas dari benda – benda lain tersebut. Dalam sistem dua benda, benda yang kehilangan panas dikatakan benda yang bersuhu lebih tinggi.[1]

Temperatur suhu yang terlalu rendah dapat merugikan manusia, sehingga manusia mengalami gangguan kesehatan seperti *hypotermia*, dan sebaliknya apa bila cuaca panas dapat mengakibatkan dehidrasi sampai dengan *heat stroke*. Oleh karena itu batas aman temperatur suhu yang dianjurkan oleh Peraturan Menteri Kesehatan RI No 1077 tahun 2011 bahwasnya suhu yang baik berada pada rentang 18 -30°C.[10]

Kelembapan udara adalah banyaknya kandungan uap air di udara

(atmosfer). Udara atmosfer adalah campuran dari udara kering dan uap air. Kelembapan udara ditentukan oleh banyaknya uap air dalam udara. Kalau tekanan uap air dalam udara mencapai maksimum, maka mulailah terjadi pengembunan. Temperature dimana terjadi pengembunan disebut titik embun. Tingkat kelembapan bervariasi menurut suhu. Semakin hangat suhu udara, semakin banyak uap air yang dapat ditampung. Semakin rendah suhu udara, semakin sedikit jumlah uap air yang dapat ditampung. Jadi pada siang hari yang panas dapat menjadi lebih lembab dibandingkan dengan hari yang dingin. Kemampuan udara untuk menampung uap air dipengaruhi oleh suhu. Jika udara jenuh uap air dinaikkan suhunya, maka udara tersebut menjadi tidak jenuh uap air. Sebaliknya, jika udara tidak jenuh uap air suhunya diturunkan dan kerapatan airnya dijaga konstan, maka udara tersebut akan mendekati kondisi jenuh uap air. Jadi ketika udara hangat naik dan mulai mendingin, lama kelamaan akan kehilangan kemampuan untuk menahan /menampung uap air.[1]

Dampak dari kelembapan persentase perhitungan yang terukur apabila tinggi maupun rendah dapat menyebabkan suburnya pertumbuhan mikroorganisme. Baiknya persentase kelembapan yang dianjurkan oleh standar Peraturan Menteri Kesehatan RI No 1077 tahun 2011 berada pada rentang nilai $>40\%$, sebaliknya apabila konsentrasi kelembapan melebihi $<60\%$ maka kelembapan tersebut tidak baik.[10]

2.2.2 Sistem Pemantauan

Sistem pemantauan adalah proses pemantauan dan pengawasan terhadap suatu sistem, proses, atau aktivitas untuk memperoleh informasi tentang kondisi, performa, atau perubahan yang terjadi. Sistem monitoring digunakan untuk memantau berbagai aspek dan parameter yang relevan dengan tujuan tertentu, seperti kinerja sistem komputer, lingkungan, jaringan, atau infrastruktur.[11]

Tujuan utama dari sistem pemantauan adalah untuk memastikan bahwa sistem atau proses yang diamati berfungsi dengan baik dan sesuai dengan harapan. Dengan memantau sistem, kita dapat mendeteksi masalah atau ketidaknormalan yang terjadi sehingga tindakan perbaikan atau perbaikan dapat diambil segera.

Sistem monitoring juga membantu dalam mengumpulkan data dan informasi yang diperlukan untuk analisis, pelaporan, dan pengambilan keputusan.[11]

Sistem pemantauan biasanya melibatkan penggunaan alat atau perangkat lunak khusus yang memungkinkan pengumpulan data secara terus-menerus atau periodik, analisis data secara real-time, dan pengiriman notifikasi atau alarm jika terjadi perubahan atau situasi yang tidak diinginkan. Data yang dikumpulkan melalui sistem monitoring dapat berupa metrik kinerja, log kejadian, data sensor, atau informasi lainnya yang relevan dengan sistem yang sedang dipantau.

2.2.3 Sensor MQ-135

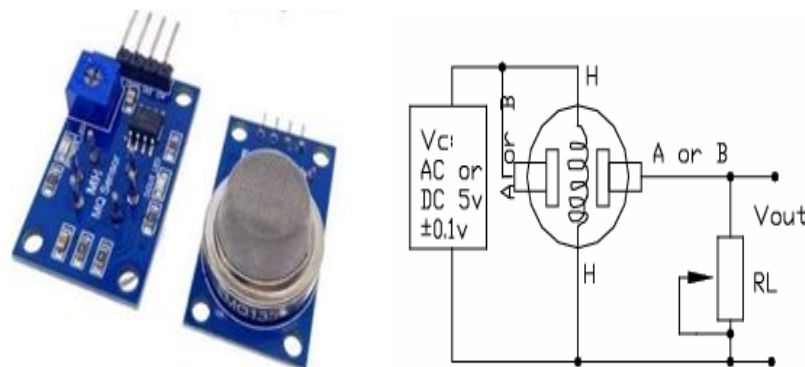
MQ-135 (Gambar 2.1) adalah sensor udara yang dirancang untuk mendeteksi gas amonia (NH_3), natrium-(di)oksida (NO_x), alkohol/ethanol ($\text{C}_2\text{H}_5\text{OH}$), benzena (C_6H_6), karbon dioksida (CO_2). Gas – gas tersebut apabila konsentrasi melebihi ambang batas dapat mengganggu kualitas udara dan kesehatan manusia. Sensor ini melaporkan hasil deteksi kualitas udara berupa perubahan nilai resistansi analog di pin keluarannya, serta keluaran digital yang menunjukkan keberadaan gas juga melewati ambang batas tertentu. Sensor MQ-135 adalah sensor gas yang sensitive terhadap berbagai gas, tetapi memerlukan kalibrasi dan penyesuaian sensitivitas yang tepat untuk mendapatkan pengukuran yang akurat. Suhu dan kelembapan juga menjadi faktor yang dapat memengaruhi kinerja sensor. Sensor ini bekerja pada tegangan 5 Volt dan dilengkapi dengan elemen pemanas internal yang menjaga suhu kerja sensor agar stabil, sensor ini membutuhkan sesi preheat sebelum pembacaan yang stabil[12]. Karakteristik lengkap sensor gas MQ-135 tertera pada Tabel 2.3 berikut.

Tabel 2. 3 Spesifikasi sensor gas MQ-135[13]

Keterangan MQ-135	Spesifikasi
Tegangan Operasi	5 Volt AC/DC
Heating voltage	5 Volt AC/DC
Target Gas	Karbon Dioksida (CO_2), Amonia (NH_3), Nitrogen Oksida (NO_x), Alkohol, Benzene, Smoke

Keterangan MQ-135	Spesifikasi
Sinyal keluaran	Analog
Heater resistance	$33\Omega \pm 5\%$
Standard Detecting Condition	Temp: $20^{\circ}\text{C} \pm 2^{\circ}\text{C}$ Vc: $5\text{V} \pm 0.1$ Humidity: $65\% \pm 5\%$ Vh: $5\text{V} \pm 0.1$

Ukuran fisik MQ 135 tidak terlalu besar, namun MQ-135 memiliki performa yang paling baik di kelasnya. Bentuk dan model dari modul sensor Gas MQ-135 ditunjukkan oleh Gambar 2.1.



Gambar 2. 1 Sensor Gas MQ-135[12], [13]

Tabel 2. 4 Keterangan Pin Sensor MQ-135

Pin	Keterangan	Fungsi Pin
1	VCC	Pin ini digunakan untuk memasok tegangan listrik ke sensor (5 VDC), sehingga sensor dapat terhubung ke sumber daya dan beroperasi
2	Ground	Pin ini berperan penting untuk menjaga kestabilan tegangan dan memastikan sensor bekerja dengan benar serta aman
3	DO (Digital Output)	Pin output digital ini memberikan sinyal High atau Low. Jika konsentrasi gas terdeteksi melebihi nilai ambang, pin ini akan berubah status, sehingga

Pin	Keterangan	Fungsi Pin
		dapat digunakan untuk deteksi sederhana dengan output digital
4	AO (Analog Output)	Pin output analog ini memberikan sinyal analog yang merepresentasikan kadar gas yang terdeteksi secara kontinu

Dalam proses kalibrasi sensor MQ135 untuk mendeteksi gas seperti karbon dioksida (CO_2) dan amonia (NH_3), kurva karakteristik sensor yang tercantum dalam datasheet memainkan peran yang sangat krusial. Kurva ini menunjukkan hubungan antara rasio resistansi sensor (R_s/R_o) dan konsentrasi gas (PPM – parts per million), yang disajikan dalam skala logaritmik. Rasio ini sendiri diperoleh dari perbandingan resistansi sensor saat terpapar gas (R_s) dengan resistansi referensi pada udara bersih (R_o).

Karena MQ135 merupakan sensor gas multi-komponen yang merespons berbagai jenis gas khususnya karbon dioksida dan amonia, maka pendekatan pengolahan data harus selektif. Di sinilah pentingnya pemahaman dan penerapan kurva karakteristik—karena hanya dengan regresi yang tepat, sinyal tegangan analog dari sensor dapat diinterpretasikan secara spesifik sebagai estimasi kadar CO_2 atau NH_3 .

Sebagaimana telah dibahas sebelumnya, pendekatan regresi yang digunakan dapat bervariasi, tergantung pada bentuk data yang diperoleh dari kurva karakteristik. Untuk CO_2 , regresi power digunakan secara langsung, menghasilkan model matematis. Sementara untuk amonia digunakan regresi log-log. Agar regresi ini akurat dan representatif, diperlukan data numerik yang valid dari kurva karakteristik tersebut. Oleh karena itu, langkah yang dilakukan adalah mencari semua titik koordinat dari grafik menggunakan aplikasi WebPlotDigitizer. Titik-titik ini mewakili pasangan nilai R_s/R_o dan PPM yang kemudian digunakan untuk membentuk model regresi, baik dalam bentuk power maupun log-log yang mencerminkan karakteristik respons sensor terhadap masing-masing gas target.

1. Gas Karbon Dioksida

Tabel 2. 5 Titik Potong Karbon Dioksida Berdasarkan Datasheet MQ135

x	y
10	2.2801
20.1356	1.7926
30.2117	1.5664
39.7300	1.4177
49.9157	1.3136
59.9146	1.2460
69.7609	1.1889
79.9988	1.1211
90.3539	1.0760
101.5331	1.0511
199.3239	0.8025

Setelah mendapatkan seluruh titik potong langkah selanjutnya adalah mencari nilai a dan b yang diperoleh dari seluruh titik potong diatas yang merujuk pada koefisien dalam persamaan regresi power ($PPM=a \times (R_s/R_o)^b$). Nilai a adalah konstanta regresi (faktor skala), tanpa konstanta regresi, maka tidak bisa mengubah output sensor ke satuan karbon dioksida (PPM). Konstanta ini membuat model matematis relevan dengan kenyataan fisik. Sedangkan nilai b adalah eksponen yang menunjukkan bagaimana sensitifnya perubahan konsentrasi gas terhadap perubahan rasio resistansi sensor. Menggunakan aplikasi sebuah power regression calculator maka didapatkanlah nilai a sebesar 116.6020682 dan nilai b adalah - 2.769034857. Nilai ini dapat digunakan kedalam program kalibrasi sensor MQ135 dalam mendeteksi karbon dioksida.

2. Gas Amonia

Tabel 2. 6 Titik Potong Amonia Berdasarkan Datasheet MQ135

x	y
10	2.5594

x	y
20	1.9115
30	1.6325
40	1.4532
50	1.3089
60	1.2265
70	1.1583
80	1.0960
90	1.0516
100	0.9990

Setelah seluruh titik koordinat diperoleh dari ekstraksi kurva karakteristik sensor MQ135 menggunakan aplikasi *WebPlotDigitizer*, langkah berikutnya adalah melakukan analisis regresi untuk membentuk model matematis hubungan antara rasio resistansi sensor (R_s/R_o) dan konsentrasi gas amonia (NH_3) dalam satuan PPM. Karena kurva karakteristik disajikan dalam skala logaritmik, maka pendekatan yang digunakan adalah regresi log-log, yang mengasumsikan bentuk hubungan sebagai berikut $\log_{10}(R_s/R_o) = m \times \log_{10}(\text{PPM}) + b$.

Dalam persamaan tersebut m adalah gradien (kemiringan garis regresi) yang menggambarkan sensitivitas perubahan R_s/R_o terhadap variasi konsentrasi gas, dan b adalah intersep atau titik potong dengan sumbu y dalam grafik log-log. Melalui proses regresi menggunakan data logaritmik dari seluruh titik koordinat hasil ekstraksi kurva NH_3 , diperoleh nilai gradien $m = -0.417$ dan intersep $b = 0.858$. Dengan demikian, hasil regresi ini dapat digunakan sebagai dasar dalam pemrograman mikrokontroler untuk mendeteksi dan mengukur kadar amonia. Persamaan tersebut kemudian dibalik untuk menghitung PPM berdasarkan nilai pembacaan R_s/R_o , menjadi $\log_{10}(\text{PPM}) = (\log_{10}(R_s/R_o) - b)/m \Rightarrow \text{PPM} = 10^{((\log_{10}(R_s/R_o) - b)/m)}$. Persamaan inilah yang kemudian diimplementasikan ke dalam kode program mikrokontroler, sehingga tegangan analog yang dihasilkan sensor MQ135 dapat diubah menjadi nilai estimasi konsentrasi amonia secara real-time. Dengan pendekatan ini, sistem pengukuran menjadi jauh lebih akurat dan

reliabel dalam mengonversi sinyal analog menjadi informasi digital yang bermakna.

```
void MQ135_begin() {
  Serial.println("Initializing MQ-135 Sensor...");
  pinMode(PIN_MQ_A0, INPUT);
  analogReadResolution(12); // ADC 12-bit (0-4095)
}

float adc = analogRead(PIN_MQ_A0);
```

Gambar 2. 2 Contoh program MQ135

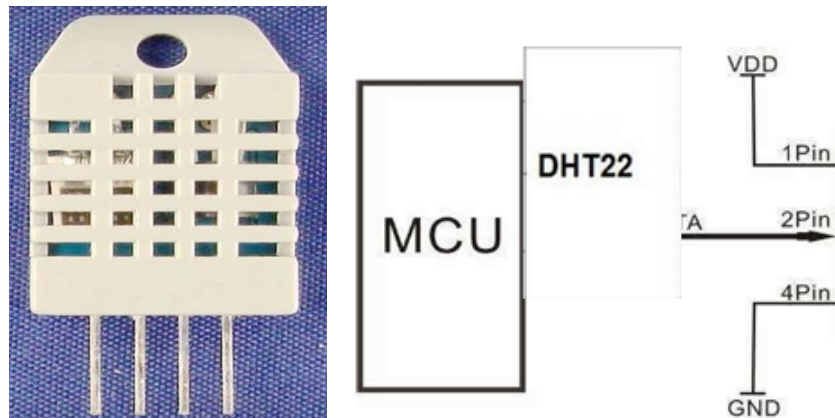
Prinsip kerja dari sensor MQ135 adalah mendeteksi konsentrasi gas tertentu dengan mengubah kadar gas menjadi tegangan analog yang dibaca oleh ESP32 melalui pin analog. Baris diatas merupakan program untuk menunjukkan cara membaca nilai analog dari sensor MQ135, yang kemudian dapat dikonversi ke satuan PPM.

2.2.4 Sensor DHT-22

DHT-22/AM2302 adalah sensor digital yang digunakan untuk mengukur kelembapan dan suhu, dengan output berupa digital-output. DHT-22 mempunyai fitur kalibrasi yang sangat akurat dan tingkat stabilitas yang sangat baik. Sensor ini menggunakan elemen kapasitif untuk mendeteksi kelembapan dan termistor untuk mengukur suhu, kemudian mengubah hasil pengukuran tersebut menjadi sinyal digital yang dapat dibaca oleh mikrokontroler atau komputer. OTP program memory menyimpan koefisien kalibrasi, sehingga pada saat terdeteksi sesuatu oleh internal sensor, maka koefisien tersebut disertakan oleh DHT-22 pada perhitungannya. DHT-22 beroperasi pada rentang tegangan 3,3 V hingga 6 V. Bentuk dan model dari DHT-22 disajikan oleh Gambar 2.3.[14]

Sensor DHT22 bekerja dengan prinsip kapasitif untuk kelembapan dan termistor untuk suhu. Saat kelembapan udara berubah, kapasitor polimer di dalam sensor mengubah kapasitansinya, sementara termistor mengubah resistansinya sesuai suhu lingkungan. Sensor kemudian mengolah perubahan ini menjadi sinyal digital melalui komunikasi satu kawat (*single wire digital communication*). Data suhu dan kelembapan dikirim secara serial ke mikrokontroler dengan interval

pengukuran sekitar 2 detik. Sensor sudah terkalibrasi secara internal dan menyediakan data digital yang siap pakai tanpa perlu konversi analog ke digital tambahan.



Gambar 2. 3 Sensor DHT-22[14]

Pada Tabel 2.5 memberikan penjelasan terkait keterangan spesifikasi oleh sensor suhu dan kelembapan DHT-22, dan di Tabel 2.6 menjelaskan terkait fungsi – fungsi dari pin yang terdapat pada Sensor DHT-22.

Tabel 2. 7 Spesifikasi Sensor DHT-22[14]

Keterangan MQ-135	Spesifikasi
Sumber Tegangan	3,3 – 6 Volt DC
Konsumsi Arus	1 – 1,5 mA
Rentang operasi kelembapan dan suhu	0 – 100% RH dan -40°C ~ +80°C
Sinyal keluaran	Digital
Sensing period	2 detik
Akurasi	H= +-2% RH(Max +-5%RH); T = <+-0,5°C

Tabel 2. 8 Keterangan Pin Sensor DHT-22[14]

Pin	Keterangan	Fungsi Pin
1	VDD-Power Supply	Pin memberikan tegangan operasional ke sensor agar komponen internal seperti kapasitor polimer dan termistor dapat berfungsi.
2	Data-Signal	Pin ini mengirimkan data hasil pengukuran secara digital ke mikrokontroler
3	NULL	Pada sensor 4 pin, pin ini tidak digunakan.
4	GND	Pin ini berperan penting untuk menjaga kestabilan tegangan dan memastikan sensor bekerja dengan benar serta aman

```

void DHT22_begin() {
  Serial.println("Initializing DHT22 Sensor...");
  dhtSensor.setup(PIN_DHT22, DHTesp::DHT22);
}

TempAndHumidity data = dhtSensor.getTempAndHumidity();
float humi_val = data.humidity;

TempAndHumidity data = dhtSensor.getTempAndHumidity();
float temp_val = data.temperature;

```

Gambar 2. 4 Contoh Program DHT22

Prinsip kerja DHT22 mengukur suhu dan kelembapan udara, lalu mengirimkan data digital ke ESP32 melalui pin digital. Baris program diatas menunjukkan bahwa pemanggil fungsi library DHT untuk membaca suhu dan kelembapan secara digital dari sensor DHT22.

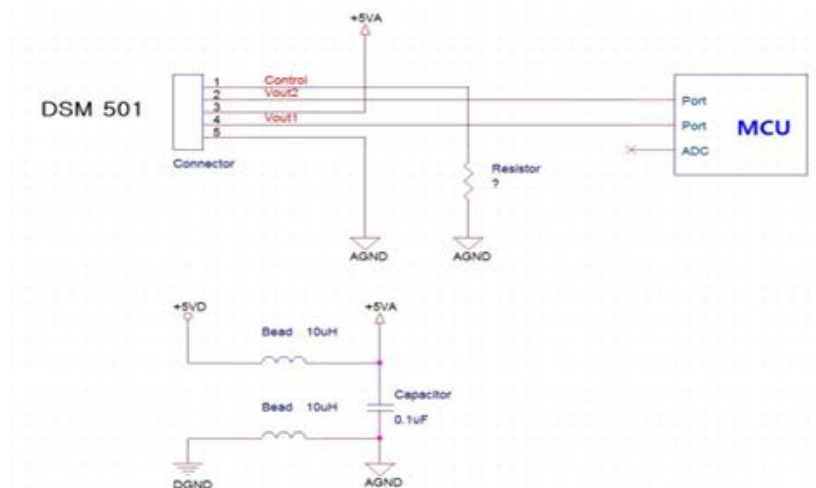
2.2.5 Sensor Debu DSM 501A

Sensor debu DSM 501A merupakan sebuah modul sensor yang dapat mendeteksi konsentrasi partikel debu di udara, sensor ini dapat bekerja dengan baik dalam mendeteksi debu dengan ukuran minimal debu sebesar 1 μ m dengan. Sensor debu DSM 501 dapat diintegrasikan kedalam sistem pemantauan ataupun sistem pembersih udara. Output dari pengukuran konsentrasi debu berupa sinyal PWM, sinyal digital yang memiliki pulsa – pulsa dengan lebar yang bervariasi. Lebar pulsa dipengaruhi oleh tingkat konsentrasi debu tersebut. [15]

Sensor DSM501 menghisap udara sekitar melalui modulnya dengan bantuan pemanas internal yang menciptakan aliran udara paksa. Di dalam sensor terdapat LED inframerah yang memancarkan cahaya ke udara yang masuk. Ketika partikel debu melewati area pengukuran, cahaya dari LED tersebut tersebar oleh partikel-partikel debu dan diterima oleh phototransistor (detektor cahaya). Sinyal yang diterima oleh phototransistor kemudian diperkuat oleh op-amp dan diolah menjadi sinyal PWM. Lebar pulsa PWM tersebut berbanding lurus dengan jumlah partikel debu yang terdeteksi. Sensor dapat membedakan partikel debu berdasarkan ukuran, yaitu partikel berukuran lebih dari 1 μm dan lebih dari 2.5 μm , yang masing-masing memiliki output PWM berbeda. Gambar 2.5 menjelaskan bentuk dari sensor debu DSM 501A.[16]



Gambar 2. 5 Sensor DSM 501A[16]



Gambar 2. 6 Skematik Sensor Debu DSM 501A[16]

Tabel 2. 9 Spesifikasi Sensor Debu DSM 501A[16]

Parameter DSM 501A	Spesifikasi
VCC	4,5 – 5,5 Volt
ICC	90 mA
Rentang Suhu operasi dan Rentang Suhu Memori	-10°C ~ +65°C dan -20°C ~ +80°C
Sinyal keluaran	PWM Output
Rentang Kelembapan Operasi	95 %RH
Target deteksi	Debu, PM 2.5 dan PM 1.0

Pada Tabel 2.7 menjelaskan terkait spesifikasi dari sensor debu DSM 501A, sedangkan pada tabel 2.8 memberikan penjelasan terkait Pin dan fungsi – fungsi pin yang ada pada sensor debu.

Tabel 2. 10 Keterangan Pin Sensor Debu DSM 501A[16]

No Pin	Nama Pin	Keterangan	Fungsi Pin
1	Kontrol	Vout 1 Kontrol	Pin berfungsi untuk mengatur rentang ukuran partikel debu yang dideteksi oleh output Vout 1 dengan menghubungkan ke ground atau membiarkannya tidak terhubung.
2	Vout2	V out 2 (PWM)	Pin ini menghasilkan sinyal PWM yang mewakili konsentrasi partikel debu > 1 μm
3	VCC	DC 5 V Input	Pin dihubungkan dengan supplay daya sehinga dapat menyediakan tegangan operasi 5V untuk sensor
4	Vout1	V out 1 (PWM)	Pin ini menghasilkan sinyal PWM untuk partikel debu > 2.5 μm

No Pin	Nama Pin	Keterangan	Fungsi Pin
5	GND	Ground	Pin ini berperan penting untuk menjaga kestabilan tegangan dan memastikan sensor bekerja dengan benar serta aman

```
void DSM501A_begin() {
  Serial.println("Initializing DSM501A (PM2.5 Sensor)..."); // Placeholder dihilangkan
  pinMode(PIN_DSM501, INPUT);
}

duration_local = pulseInLong(PIN_DSM501, LOW, 100000);
```

Gambar 2. 7 Contoh Program DSM501A

Prinsip kerja dari sensor ini mengukur konsentrasi partikulat debu PM2,5 di udara dengan membaca durasi pulsa sinyal digital pada pin digital tertentu. Prinsip kerja dari sensor ini mengukur konsentrasi partikulat debu PM2,5 di udara.

2.2.6 Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil yang dikemas dalam satu chip IC (Integrated Circuit) yang dirancang khusus untuk mengendalikan sistem atau perangkat tertentu dengan menjalankan program yang tersimpan di dalamnya. Di dalam mikrokontroler terdapat komponen-komponen penting seperti CPU (Central Processing Unit) sebagai otak pengolah data, memori (RAM dan ROM) untuk menyimpan program dan data, serta port input/output untuk berinteraksi dengan sensor dan aktuator. Berbeda dengan komputer umum, mikrokontroler memiliki fungsi khusus dan digunakan dalam sistem tertanam (embedded system) yang mengontrol perangkat secara otomatis, seperti pengukuran sensor, kendali motor, dan komunikasi data.[17]

2.2.7 ESP32 DEV KIT V1

ESP32 adalah salah satu contoh mikrokontroler modern yang sangat populer, karena menggabungkan kemampuan prosesor ganda (dual-core), memori yang cukup besar, serta fitur konektivitas WiFi dan Bluetooth bawaan. Dengan kemampuan tersebut, ESP32 sangat cocok untuk aplikasi Internet of Things (IoT)

yang membutuhkan pengolahan data sensor secara real-time, komunikasi nirkabel, dan pengendalian perangkat secara efisien dalam satu modul kecil dan hemat energi. ESP32 adalah mikrokontroler yang dikembangkan oleh Espressif Systems sebagai penerus dari mikrokontroler ESP8266. Mikrokontroler ini sudah dilengkapi dengan modul WiFi yang terintegrasi dalam satu chip, sehingga sangat mendukung pembuatan aplikasi Internet of Things (IoT). Jadi, ESP32 merupakan mikrokontroler canggih yang mengintegrasikan berbagai fungsi penting dalam satu chip, memudahkan pengembangan sistem embedded yang kompleks dan terhubung secara wireless.[18]

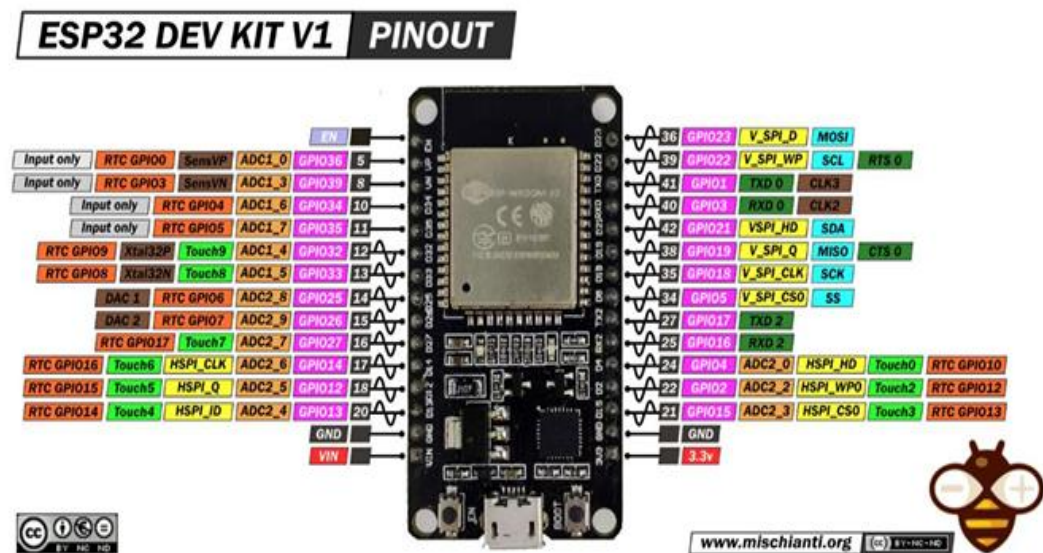
ESP32 DEVKIT V1 bekerja dengan menjalankan program yang diunggah ke dalam flash memori mikrokontroler. Prosesor dual-core memungkinkan eksekusi multitasking secara efisien, misalnya menjalankan koneksi WiFi dan Bluetooth secara bersamaan dengan pengolahan sensor atau kontrol perangkat. Board ini menyediakan berbagai pin input/output (GPIO) yang dapat dikonfigurasi sebagai digital input/output, ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), PWM (Pulse Width Modulation), serta antarmuka komunikasi seperti UART, SPI, dan I2C. Dengan kemampuan ini, ESP32 DEVKIT V1 dapat menghubungkan sensor, aktuator, dan perangkat lain untuk membangun sistem IoT yang kompleks dan responsive.[18]

Keluarga ESP32 termasuk dalam chip ESP32-D0WDQ6 (ESP32- D0WD), ESP32- D2WD, ESP32-S0WD, dan sistem dalam paket (SiP) ESP32-PICO-D4. Mikroprosesor yang digunakan oleh ESP32 adalah Tensilica Xtensa LX6 dual-core atau single-core dengan clock rate hingga 240 MHz. Dengan rilisnya ESP32, espressif system.

Espressif System mengeluarkan beberapa model board ESP32 untuk menunjang para pengguna yang ingin membuat proyek elektronika berbasis Internet of Things, salah satu model yang dikeluarkan oleh Espressif System yaitu ESP32 Development Kit (Board untuk membuat aplikasi IoT dengan ESP32) atau ESP32 DEVKIT V1. [19]

- 1 : 30 meliputi pin tegangan dan GPIO
- 2 : 15 pin ADC

- 3 : 3 UART Interface
- 4 : 3 SPI Interface
- 5 : 2 I2C Interface
- 6 : 16 pin PWM (Pulse Width Modulation)
- 7 : 2 pin DAC (Digital to Analog Converter)
- 8 : Flash memory 4MB
- 9 : Clock Speed 240MHz



Gambar 2. 8 Board ESP32 DEVKIT[18]

Tabel 2. 11 Konfigurasi Pin ESP32 DEVKIT V1[20][18]

Pin	Fungsi utama	Cara Kerja
3.3V (VCC)	Pin suplai tegangan 3.3 Volt	Pin ini memberikan daya stabil ke modul ESP32 dan komponen terkait.
GND	Ground (referensi tegangan nol)	Menjadi jalur kembali arus listrik dan referensi tegangan untuk kestabilan rangkaian.
GPIO (General	Pin input/output digital umum	Bisa dikonfigurasi sebagai input atau output digital, digunakan untuk membaca sensor atau

Pin	Fungsi utama	Cara Kerja
Purpose Input Output)		mengendalikan aktuator. Beberapa GPIO juga mendukung fungsi khusus seperti ADC, DAC, PWM, dan komunikasi serial.
ADC Pins	Analog to Digital Converter	Pin ini mengubah sinyal analog dari sensor menjadi data digital yang dapat diproses mikrokontroler.
DAC Pins	Digital to analog converter	Pin ini menghasilkan sinyal analog dari data digital untuk mengendalikan perangkat analog.
UART Pins	Serial communication	Pin ini digunakan untuk komunikasi serial dengan perangkat lain seperti computer atau modul lain.
SPI Pins	Serial Peripheral Interface	Pin ini merupakan antarmuka komunikasi cepat untuk sensor dan modul eksternal
I2C Pins	Inter-Integrated Circuit	Antarmuka komunikasi dua kabel untuk sensor dan perangkat lain.

Berikut contoh pin GPIO dan fungsinya:

1. GPIO 0, 2 4, 5, 12–19, 21–23, 25–27, 32, 33: Pin ini dapat digunakan sebagai input maupun output digital, serta mendukung fungsi ADC dan PWM pada beberapa pin tertentu.
2. GPIO 34, 35, 36, 39: Pin ini hanya berfungsi sebagai input dan tidak memiliki internal pull-up, biasanya digunakan untuk input sensor analog.
3. Pin UART (GPIO 1, 3, 16, 17): Digunakan untuk komunikasi serial data, misalnya menghubungkan ESP32 dengan komputer atau modul lain.
4. Pin I2C (GPIO 21 sebagai SDA, GPIO 22, sebagai SCL): Digunakan untuk komunikasi dengan sensor atau modul I2C

Setiap pin GPIO dapat diprogram untuk berfungsi sebagai input atau output. Sebagai input, pin dapat membaca status logika HIGH atau LOW dari sensor atau tombol. Sebagai output, pin dapat mengendalikan perangkat seperti LED, relay, atau motor dengan mengirimkan sinyal HIGH atau LOW. Pin yang mendukung ADC dapat membaca nilai tegangan analog dan mengubahnya menjadi data digital untuk pemrosesan lebih lanjut. Pin PWM dapat menghasilkan sinyal pulsa dengan lebar variabel untuk mengatur kecepatan motor atau kecerahan LED.

ESP32 adalah mikrokontroler dengan dua inti prosesor (dual-core), yang terdiri dari dua unit pemrosesan independen. Kedua inti ini memungkinkan ESP32 menjalankan tugas secara bersamaan atau menjalankan dua program secara multitasking tanpa saling mengganggu. Keunggulan dual-core pada ESP32 menjadikannya ideal untuk aplikasi IoT yang memerlukan pemrosesan paralel, seperti pengolahan data sensor sekaligus komunikasi nirkabel secara simultan. Dengan fitur ini, ESP32 dapat menggunakan FreeRTOS, yaitu sistem operasi waktu nyata (RTOS) yang dirancang untuk mengelola tugas pada mikrokontroler.[21]

FreeRTOS berperan dalam penjadwalan tugas, penanganan interrupt, dan manajemen sumber daya, sehingga memastikan aplikasi berjalan dengan stabil dan responsif. Integrasi FreeRTOS dalam ESP-IDF dan Arduino IDE membantu pengembang dalam merancang sistem IoT dan embedded yang kompleks, serta menawarkan fitur-fitur seperti task ke core tertentu, timer, dan manajemen daya.[21]

```
Serial.println("Creating FreeRTOS Tasks...");
xTaskCreatePinnedToCore(TaskUpdateIoT, "TaskUpdateIoT", 8192, NULL, 1, NULL, 0);
xTaskCreatePinnedToCore(TaskUpdateRTC, "TaskUpdateRTC", 2048, NULL, 2, NULL, 0);
xTaskCreatePinnedToCore(TaskSensor, "TaskSensor", 4096, NULL, 1, NULL, 1);
xTaskCreatePinnedToCore(TaskDisplay, "TaskDisplay", 8192, NULL, 1, NULL, 1);
xTaskCreatePinnedToCore(TaskRefresh, "TaskRefresh", 2048, NULL, 3, NULL, 1);
Serial.println("Setup complete. Tasks running.");
```

Gambar 2. 9 Contoh Program RTOS FREE32

Gambar 2.9 merupakan contoh penggunaan kode untuk membuat beberapa task pada sistem operasi FreeRTOS di ESP32. Setiap task memiliki fungsi, prioritas, dan core yang berbeda untuk mengoptimalkan kinerja sistem. Fungsi *xTaskCreatePinnedToCore* digunakan untuk membuat task dan menempatkannya

pada core tertentu. Dengan membagi tugas ke core yang berbeda, proses multitasking menjadi lebih efisien.

2.2.8 Software Arduino Ide

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak yang memungkinkan pengguna untuk menulis, mengompilasi, dan mengunggah kode program (dikenal sebagai sketch) ke board Arduino. Software ini dibangun menggunakan bahasa pemrograman Java dan dilengkapi dengan pustaka C/C++ yang dikenal sebagai Wiring, yang menyederhanakan operasi input dan output, menggunakan Arduino IDE, pengembang dapat dengan mudah membuat prototipe sistem monitoring daya listrik yang efisien dan efektif. Kemudahan dalam menulis kode serta dukungan pustaka yang luas menjadikan Arduino IDE sebagai alat yang ideal untuk proyek-proyek berbasis IoT.[17]

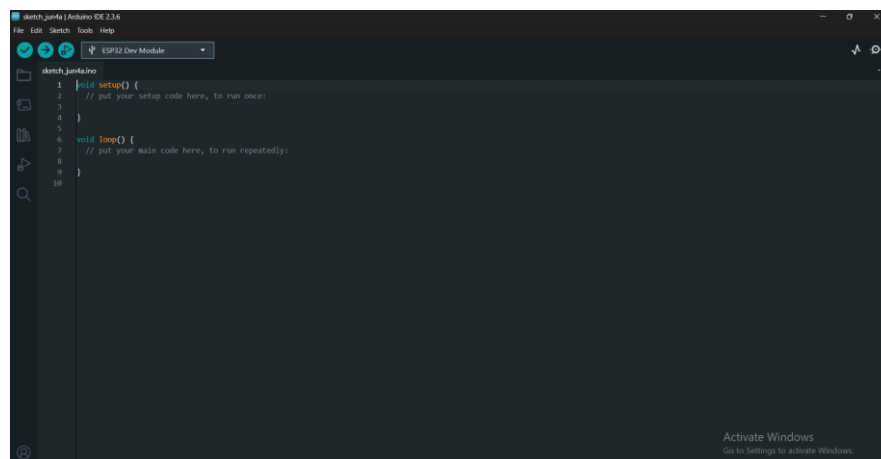
Langkah – Langkah Menggunakan Arduino IDE:

1. Menginstal dan menghubungkan aplikasi dengan komputer

Jika belum memiliki aplikasi Arduino IDE, Anda dapat mengunduhnya terlebih dahulu. Setelah Arduino IDE terinstal, langkah berikutnya adalah menghubungkan papan mikrokontroler ESP32 ke komputer menggunakan kabel USB. Untuk papan ESP32, biasanya digunakan kabel USB tipe C. Selain berfungsi sebagai media transfer data antara komputer dan ESP32, kabel ini juga menyediakan daya listrik sementara selama proses pemrograman. Setelah kabel USB tersambung, sistem operasi komputer umumnya akan mendeteksi perangkat ESP32 secara otomatis dan menginstal driver yang diperlukan. Keberhasilan koneksi dapat dipastikan melalui beberapa indikator, seperti LED yang menyala pada papan ESP32 serta munculnya port baru di pengaturan Device Manager komputer. Koneksi yang stabil sangat penting, karena menjadi jalur utama untuk mengunggah program dari Arduino IDE ke papan mikrokontroler.

Gambar 2.10 menampilkan tampilan sketch atau workspace pada Arduino IDE. Workspace biasanya terdiri dari dua fungsi utama, yaitu `setup()` dan `loop()`, yang membentuk struktur dasar dalam pemrograman Arduino. Fungsi `setup()` dieksekusi satu kali saat sistem dinyalakan, sedangkan `loop()` akan terus berjalan

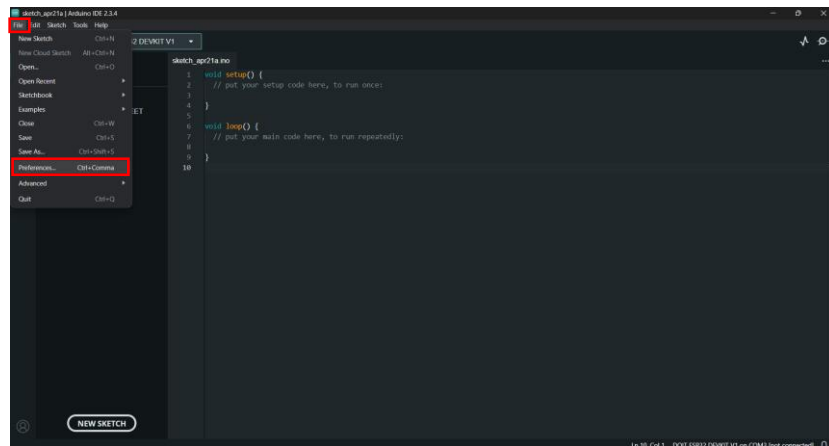
selama mikrokontroler aktif. Workspace memungkinkan pengguna untuk menulis berbagai fungsi dan logika sistem, seperti pembacaan sensor, pemrosesan data, pengendalian aktuator, serta komunikasi data. Selain itu, workspace mendukung fitur highlighting sintaks, yang memudahkan pengguna dalam menulis dan membaca kode program dengan lebih jelas dan efisien



Gambar 2. 10 Tampilan Sketch Arduino IDE

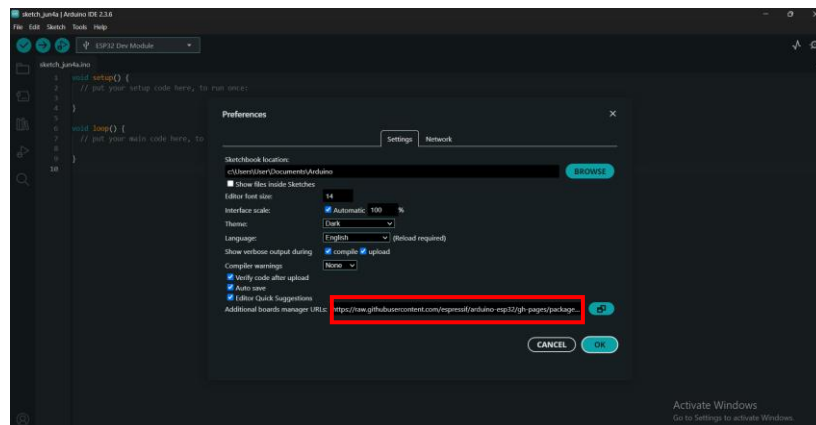
2. Memilih preferences dan menambahkan tautan board tambahan pada preferences

Setelah membuka Arduino IDE, pengguna dapat masuk ke menu *File > Preferences* untuk menyesuaikan pengaturan sesuai kebutuhan. Menu ini memungkinkan pengguna mengatur berbagai preferensi, termasuk menambahkan tautan (URL) guna mengakses board tambahan yang tidak tersedia secara default di Arduino IDE.



Gambar 2. 11 Tampilan memilih preferences

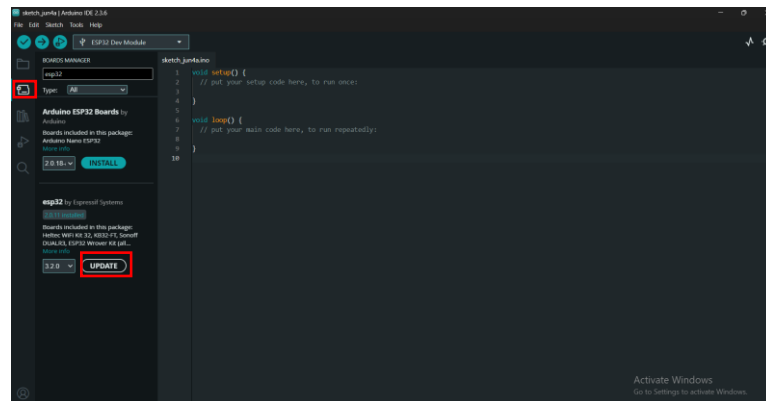
Langkah ini diperlukan agar Arduino IDE dapat mengenali dan mengunduh paket board yang dimaksud melalui Board Manager.



Gambar 2. 12 Menambahkan Tautan board pada Preferences

3. Menginstal Board melalui board manager

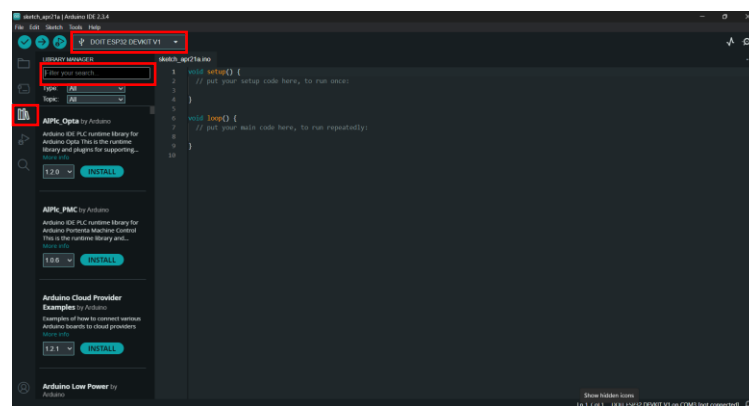
Setelah menambahkan URL, langkah selanjutnya adalah membuka *Tools > Board > Boards Manager*. Di dalam Board Manager, pengguna mencari board yang sesuai, seperti ESP32 by Espressif Systems, kemudian menekan tombol Install untuk mengunduh dan menginstal board tersebut ke dalam Arduino IDE.



Gambar 2. 13 Menginstal Board

4. Menginstall Library yang dibutuhkan

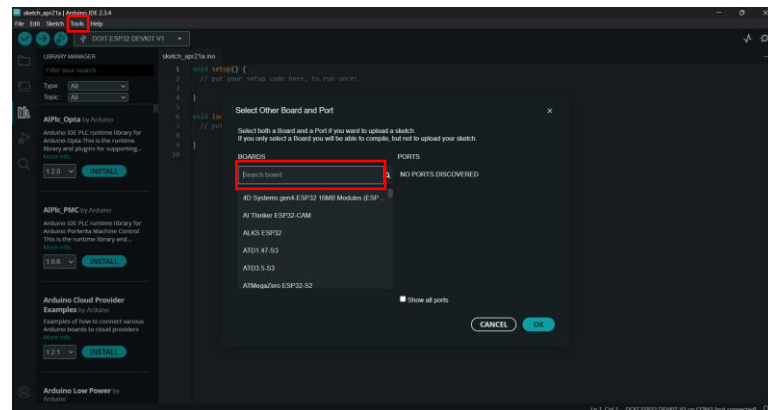
Untuk mendukung fungsi-fungsi tambahan pada proyek, pengguna perlu menginstal pustaka tertentu. Instalasi pustaka dilakukan melalui **Sketch > Include Library > Manage Libraries**. Pengguna dapat mencari pustaka yang dibutuhkan, kemudian melakukan proses instalasi



Gambar 2. 14 Menginstall Library

5. Memilih board dan port yang akan digunakan

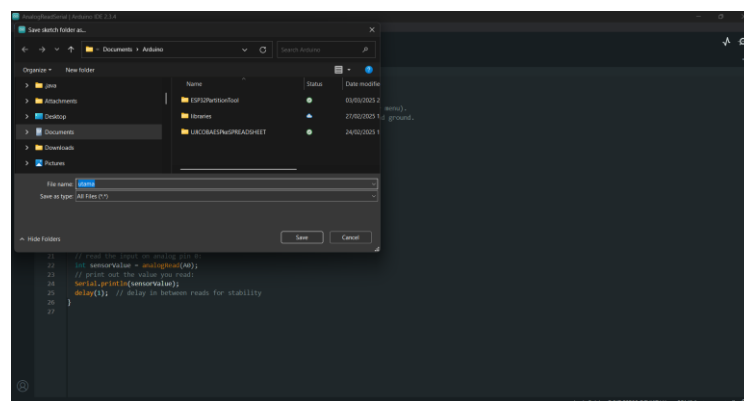
Sebelum mengunggah program ke perangkat keras (mikrokontroler), pengguna perlu menentukan jenis board dan port komunikasi yang digunakan. Pengaturan ini dilakukan pada menu **Tools > Board** untuk memilih jenis board (misalnya: ESP32 Dev Module) dan **Tools > Port** untuk memilih port komunikasi serial (COM) yang sesuai.



Gambar 2. 15 Memilih board dan port

6. Menulis program dan menyimpan program

Setelah seluruh konfigurasi selesai, pengguna dapat mulai menulis kode program (sketch) pada area editor Arduino IDE. Dan Untuk menyimpan program, pengguna memilih menu File > Save As. Arduino IDE akan meminta pengguna untuk memberi nama file, dan secara otomatis membuat folder proyek dengan nama yang sama. Setelah itu, pengguna memilih direktori (lokasi) pada komputer untuk menyimpan folder proyek. Arduino IDE akan menyimpan file program dengan ekstensi .ino di dalam folder yang telah dibuat.



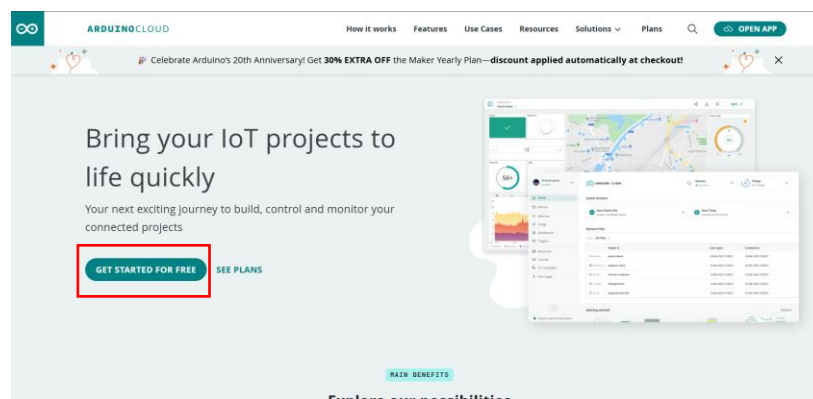
Gambar 2. 16 Menyimpan program

2.2.9 Arduino IOT Cloud

Arduino IoT Cloud adalah sebuah platform berbasis cloud yang dikembangkan oleh Arduino untuk memudahkan pembuatan dan pengelolaan

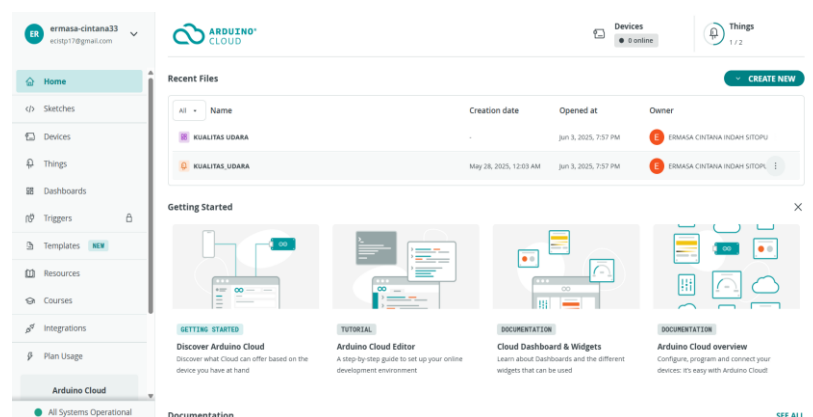
proyek Internet of Things (IoT). Platform ini bersifat open source dan menyediakan antarmuka yang intuitif untuk menghubungkan perangkat mikrokontroler Arduino atau ESP32 ke internet sehingga data sensor dapat dikirim, dipantau, dan dikendalikan secara real-time dari jarak jauh.

1) Pembuatan Akun dan Tampilan Home



Gambar 2. 17 Tampilan Platform Arduino IOT Cloud

Untuk dapat menggunakan platform ini, pengguna memerlukan pembuatan akun terlebih dahulu, dengan klik *GET STARTED FOR FREE*. Setelah membuat akun tampilan berubah seperti tampilan pada gambar 2.17 sebagai berikut. Tampilan berikut merupakan tampilan home sebelum memulai pembuatan dashboard.

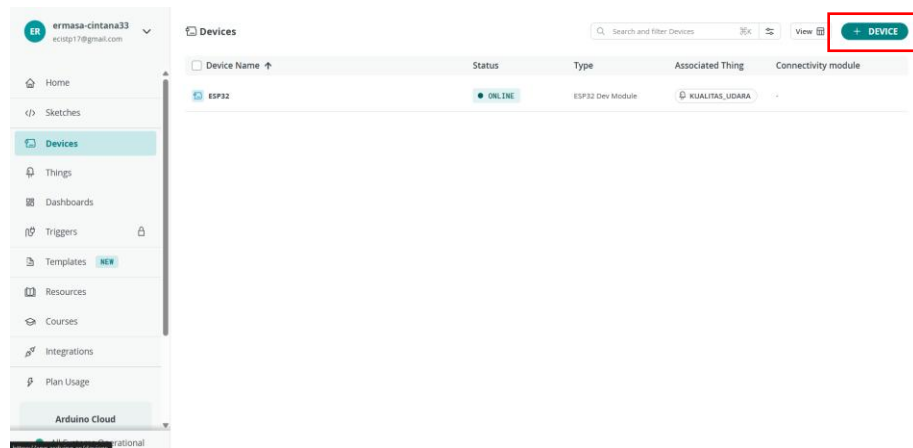


Gambar 2. 18 Tampilan Home Arduino IOT Cloud

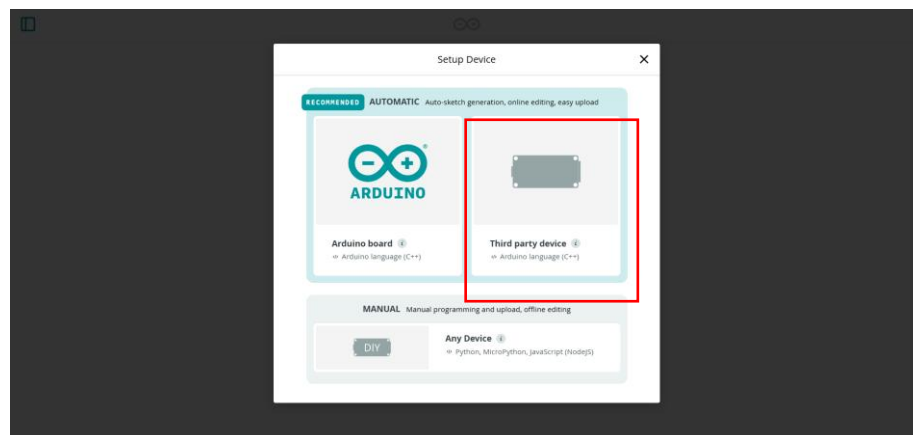
1. **Things:** Di Arduino IoT Cloud, setiap proyek IoT disebut sebagai "Thing". Thing adalah representasi virtual dari perangkat fisik yang berisi konfigurasi perangkat, variabel yang digunakan, dan kredensial jaringan. Pengguna membuat Thing untuk menghubungkan perangkat Arduino ke cloud.
2. **Devices:** Devices adalah perangkat fisik Arduino yang terhubung ke Thing. Arduino IoT Cloud mengelola koneksi perangkat ini secara otomatis, termasuk pengaturan jaringan WiFi dan komunikasi data.
3. **Sketches:** Sketch adalah kode program Arduino yang sudah disediakan secara otomatis oleh platform sesuai dengan Thing dan device yang dibuat. Sketch ini berisi fungsi untuk mengirim dan menerima data dari cloud, serta mengelola variabel yang telah didefinisikan.
4. **Dashboard:** Dashboard adalah antarmuka pengguna berbasis web yang memungkinkan pemantauan dan pengendalian perangkat secara real-time melalui widget seperti grafik, tombol, slider, dan indikator.
5. **Triggers:** Fitur ini memungkinkan pengguna membuat aturan otomatisasi, misalnya mengirim notifikasi atau mengaktifkan aktuator saat kondisi tertentu terpenuhi berdasarkan data sensor.
6. **Arduino API:** Arduino IoT Cloud menyediakan API yang memungkinkan integrasi dengan aplikasi pihak ketiga, sehingga data dan kontrol perangkat dapat diakses atau dikendalikan melalui aplikasi lain.

2) Pembuatan Device

Langkah ini merupakan tahap awal di mana pengguna mendaftarkan perangkat keras (board) yang akan digunakan untuk terhubung ke platform cloud. Dengan klik “+DEVICE” pada pojok kanan atas.

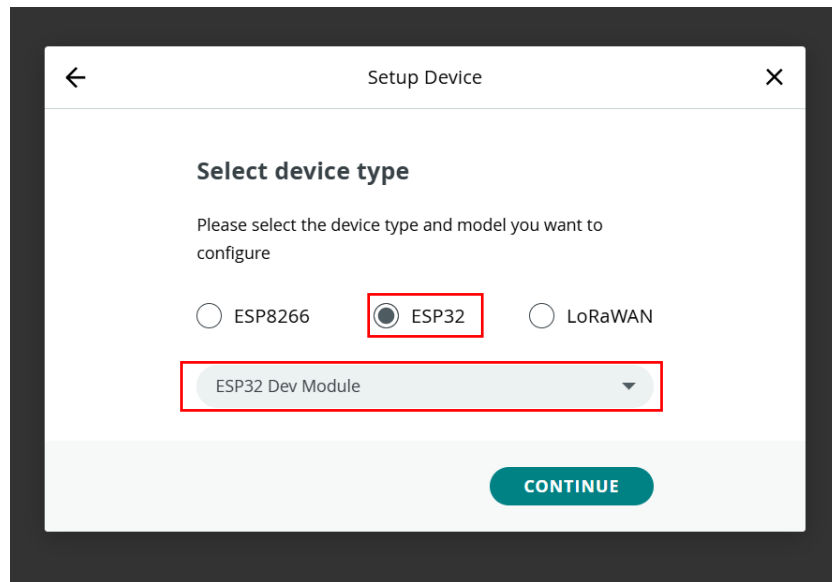


Gambar 2. 19 Tampilan Sebelum Buat Devices



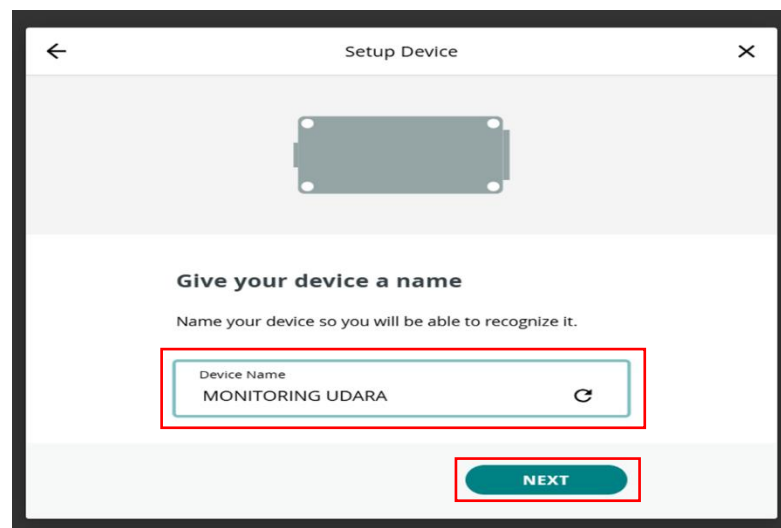
Gambar 2. 20 Tampilan untuk memilih Setup Device

Gambar 2.20 memperlihatkan sebuah proses dimana pengguna memulai dengan mengklik opsi “*Third party device*” pada antarmuka Arduino IOT Cloud, setelah nya tampilan akan berubah secara otomatis untuk menampilkan form pengisian jenis board dan mikrokontroler yang akan digunakan dalam proyek, seperti tipe dan model board yang kompatibel dengan platform.



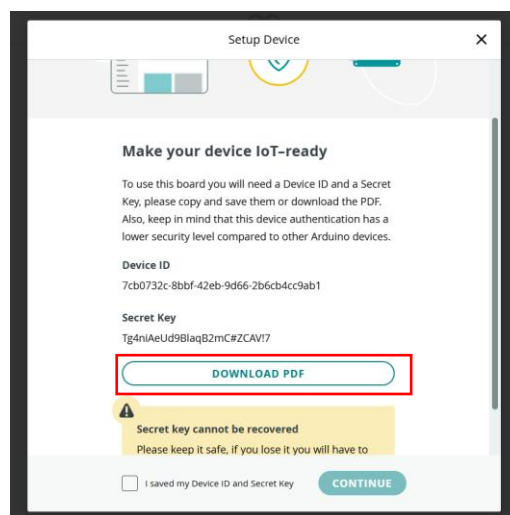
Gambar 2. 21 Tampilan Pemilihan Jenis Device dan Modul

Gambar 2.21 menjelaskan bahwa dengan mengatur informasi tersebut memungkinkan Arduino IOT Cloud mengenali perangkat yang akan dihubungkan, sehingga dapat mengatur konfigurasi koneksi dan pengelolaan data secara tepat. Dengan demikian, langkah ini merupakan bagian penting dalam pembuatan device agar perangkat dapat terintegrasi dengan layanan cloud dan siap untuk diprogram serta dikendalikan secara remote sesuai kebutuhan proyek IoT yang sedang dikembangkan.



Gambar 2. 22 Pemberian Nama Device

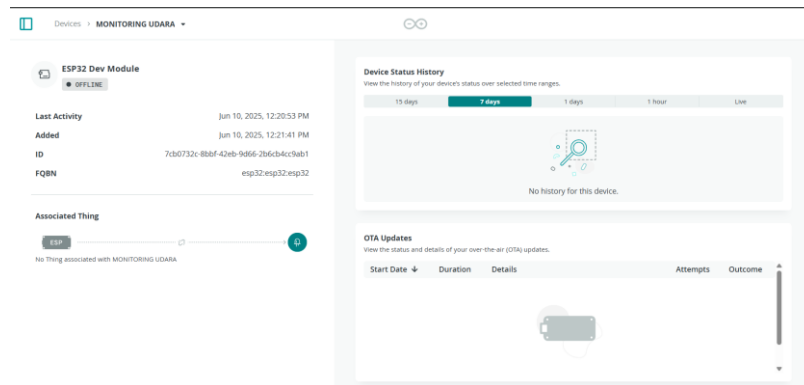
Setelah mengisi informasi mengenai jenis board yang digunakan, langkah selanjutnya adalah memberikan nama unik untuk perangkat tersebut agar mudah dikenali dalam daftar device Anda di Arduino IoT Cloud. Isilah kolom nama perangkat dengan nama yang deskriptif dan sesuai dengan fungsi atau proyek yang sedang di kerjakan. Setelah memastikan nama sudah diisi dengan benar, klik tombol “Next” untuk melanjutkan ke tahap berikutnya dalam proses pembuatan device. Dengan mengklik “Next”, akan diarahkan ke pengaturan koneksi jaringan atau konfigurasi tambahan yang diperlukan agar perangkat dapat terhubung dan berkomunikasi dengan platform Arduino IoT Cloud secara optimal.



Gambar 2. 23 Download Data Device ID dan Secret Key

Saat membuat perangkat (device) di Arduino IoT Cloud, pengguna akan menerima kredensial penting berupa Device ID dan Secret Key yang digunakan untuk menghubungkan dan mengautentikasi perangkat pengguna dengan layanan cloud. Setelah proses pembuatan device selesai, pengguna diberikan opsi untuk mengunduh file PDF yang berisi Device ID dan Secret Key tersebut. Mengunduh dan menyimpan PDF ini sangat penting karena Secret Key hanya ditampilkan sekali saat pembuatan device dan tidak dapat diakses kembali setelahnya. Jika kehilangan Secret Key, pengguna harus menghapus device dan membuatnya ulang dari awal. Oleh karena itu, menyimpan PDF berisi kredensial ini di tempat yang aman sangat dianjurkan agar dapat dengan mudah mengonfigurasi perangkat di Arduino IDE

atau aplikasi dan menjaga keamanan koneksi perangkat dengan Arduino IoT Cloud.

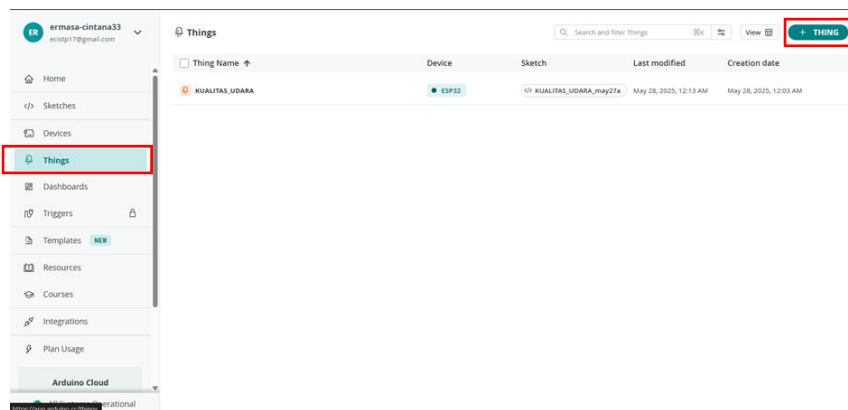


Gambar 2. 24 Tampilan Devices yang sudah di tautkan

Setelah berhasil membuat device di Arduino IoT Cloud, tampilan yang muncul biasanya berupa halaman dashboard perangkat yang baru saja di daftarkan. Pada halaman ini, terdapat ringkasan informasi penting seperti nama device, Device ID, dan status koneksi perangkat. Selain itu, terdapat opsi untuk mengunduh file konfigurasi yang berisi Device ID dan Secret Key, yang diperlukan untuk menghubungkan perangkat fisik Anda ke cloud.

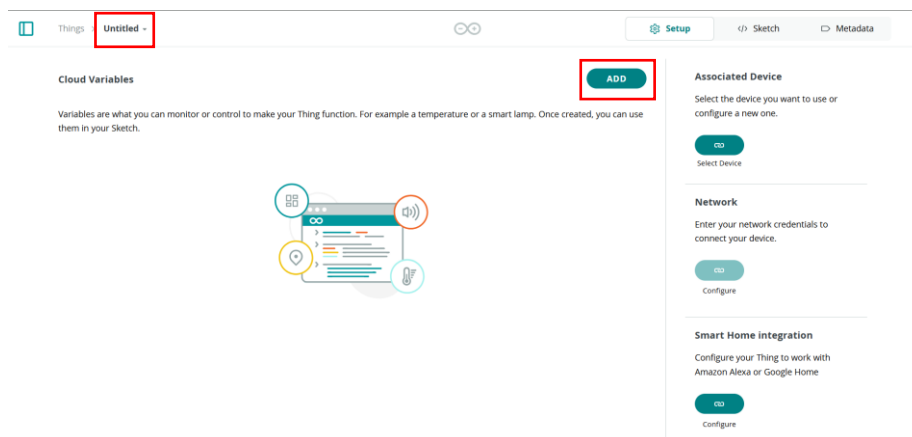
3) Pembuatan Things

Langkah ini merupakan tahap selanjutnya setelah pembuatan device, langkah pembuatan "Things" di Arduino IoT Cloud berfungsi untuk mengonfigurasi proyek IoT secara menyeluruh. Pada tahap ini, pengguna dapat mengatur variabel yang akan digunakan dalam program, seperti sensor suhu, kelembapan, atau gas, serta menghubungkan variabel tersebut dengan device yang sudah didaftarkan.



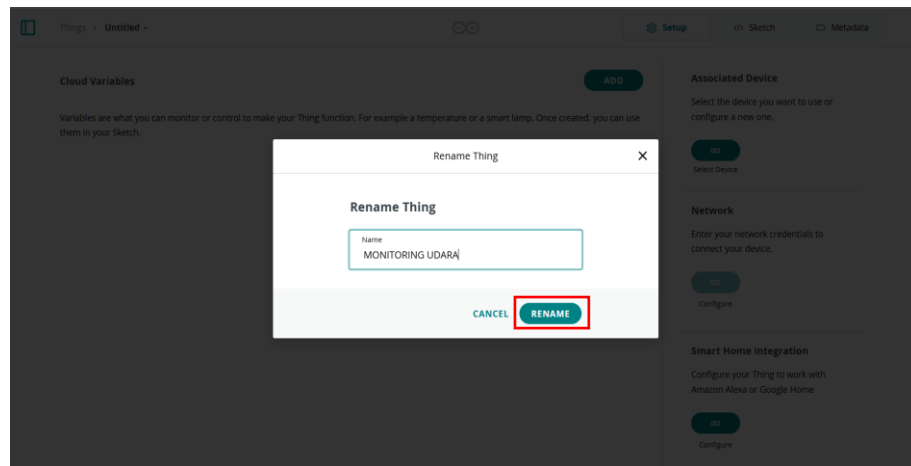
Gambar 2. 25 Tampilan Things Arduino IOT Cloud

Tahap selanjutnya adalah dengan klik “+ THING” yang kemudian tampilan akan berubah otomatis menjadi tampilan seperti gambar 2.26. Sebelum melanjutkan proses biasanya, form ini berisi beberapa opsi pengaturan dasar seperti pemberian nama perangkat



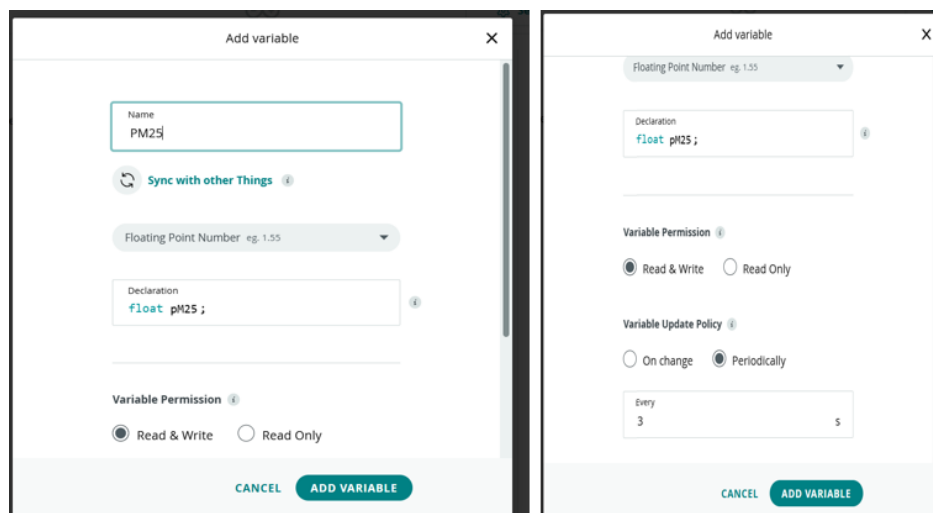
Gambar 2. 26 Pemberian Nama Things

Berikut merupakan tampilan antarmuka Things pada Arduino IOT Cloud, sebelum memulai seluruh konfigurasi dan pengaturan untuk sebuah proyek IOT. Biasanya, berisi beberapa opsi pengaturan dasar seperti pemberian nama perangkat dengan klik “Untitled” untuk mengubah nama things.



Gambar 2. 27 Rename Things

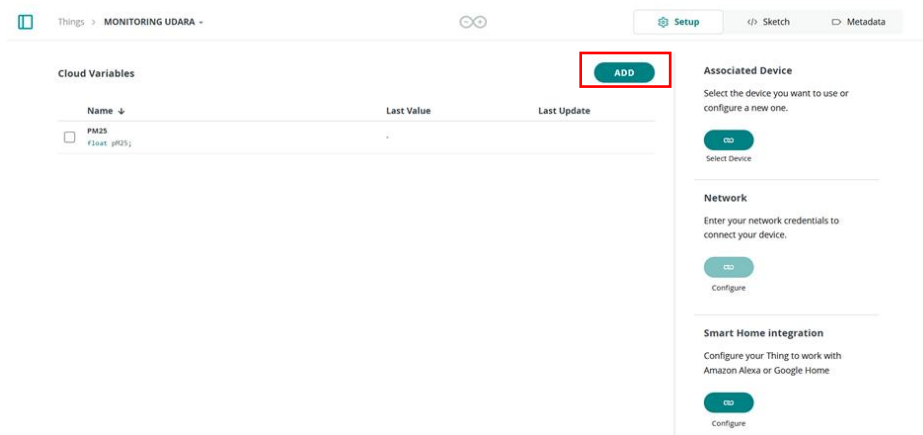
Dengan mengganti nama Things, pengguna dapat lebih mudah mengorganisir dan mengenali berbagai proyek IoT yang di kelola, terutama jika memiliki banyak device atau aplikasi yang berbeda pada platform Arduino IoT Cloud.



Gambar 2. 28 Tampilan ADD Variabel pada Things

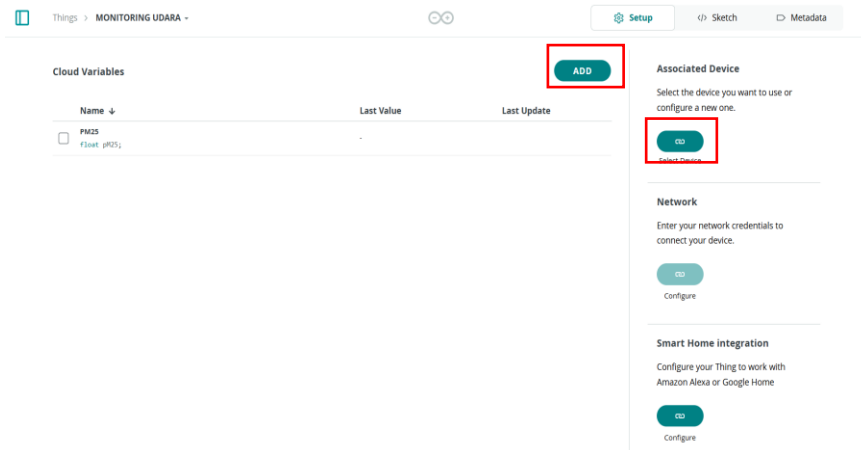
Pada tahap **Add Variabel** oleh Things di Arduino IoT Cloud, pengguna diminta untuk menambahkan variabel-variabel yang sesuai dengan kebutuhan proyek IoT yang sedang dikembangkan. Variabel ini merepresentasikan data yang akan dipantau atau dikendalikan, seperti suhu, kelembapan, konsentrasi gas, atau

nilai sensor lainnya. Penting untuk menyesuaikan jenis dan tipe variabel dengan sensor atau aktuator yang digunakan agar data yang dikirim dan diterima di cloud dapat diproses dengan benar. Misalnya, jika proyek menggunakan sensor suhu DHT22, maka variabel yang ditambahkan harus bertipe float atau integer untuk merepresentasikan nilai suhu secara akurat. Dengan menambahkan variabel yang tepat, pengguna dapat mengatur bagaimana data tersebut ditampilkan pada dashboard, serta menentukan apakah variabel tersebut bersifat baca saja (read-only), tulis saja (write-only), atau baca-tulis (read-write). Proses ini sangat krusial dalam pengembangan perangkat keras dan desain sistem IoT karena memastikan integrasi data sensor dengan platform cloud berjalan lancar dan sesuai dengan kebutuhan aplikasi yang diinginkan.



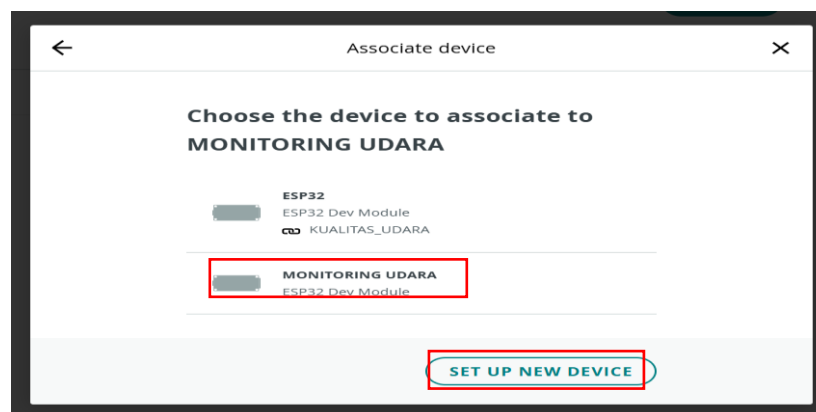
Gambar 2. 29 Tampilan Setup Things Arduino IOT Cloud

Setelah berhasil menambahkan variabel pada Arduino IoT Cloud, tampilan yang muncul menunjukkan daftar variabel yang telah dibuat dalam proyek "Things". Setiap variabel akan ditampilkan dengan nama, tipe data (seperti integer, float, boolean, atau tipe khusus seperti CloudTemperatureSensor), serta status aksesnya (READ, WRITE, atau READWRITE). Pada halaman ini juga dapat melihat opsi untuk mengedit properti variabel, seperti mengubah nama, tipe data, atau mode akses. Selain itu, tampilan ini menyediakan fitur untuk menambah variabel baru "ADD", menghapus variabel yang tidak diperlukan, dan mengatur interaksi variabel dengan device yang terhubung dengan mengklik kata ADD.



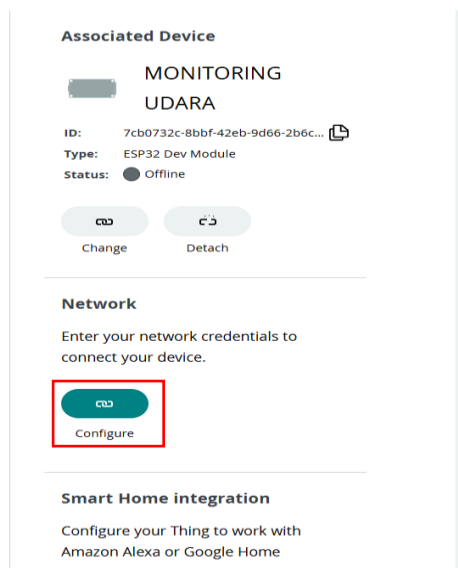
Gambar 2. 30 Tampilan Things dan Select Device

Tahap selanjutnya pada adalah klik “Select Device” pada halaman Things di Arduino IoT Cloud berarti pengguna memilih perangkat keras (board atau device) yang sudah didaftarkan sebelumnya untuk dihubungkan dengan proyek Things yang sedang dibuat atau dikonfigurasi. Dengan memilih device ini, variabel-variabel yang di tambahkan pada Things akan dikaitkan langsung dengan perangkat tersebut, sehingga data sensor atau kontrol yang dikirim dan diterima dapat berjalan secara sinkron antara device fisik dan platform cloud. Proses ini penting untuk memastikan bahwa proyek IoT dapat berkomunikasi dengan benar dan data yang dipantau atau dikendalikan sesuai dengan perangkat yang digunakan.



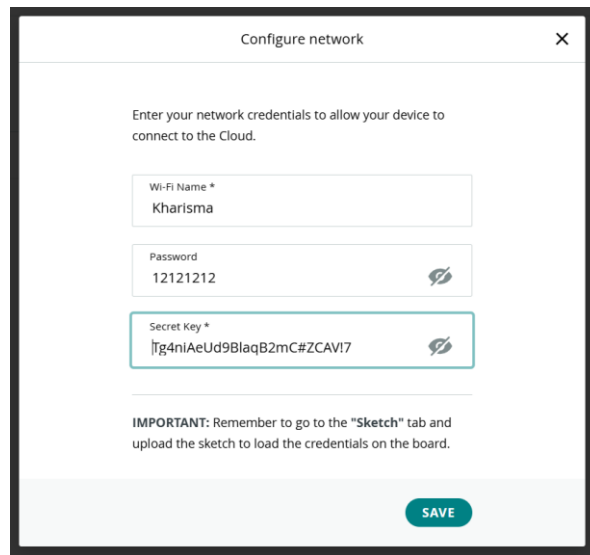
Gambar 2. 31 Set Up New Device

Setelah melakukan select device dan kemudian mengklik nama device yang sudah dibuat, misalnya “Monitoring Udara”, tampilan yang muncul akan menampilkan detail lengkap dari device tersebut dalam konteks proyek Things Anda. Pada halaman ini, dapat melihat status koneksi device, daftar variabel yang telah dikaitkan dengan device tersebut, serta pengaturan jaringan WiFi dan kredensial yang digunakan.



Gambar 2. 32 Tampilan Configure Network Things

Setelah berhasil select device pada halaman Things di Arduino IoT Cloud, tampilan akan berubah menampilkan detail device yang dipilih, seperti nama device dan status koneksi seperti gambar 2.32. Selanjutnya, ketika mengklik tombol Configure pada bagian Network, akan diarahkan ke halaman pengaturan jaringan. Di sini, perlu memasukkan kredensial Wi-Fi seperti nama jaringan (SSID) dan kata sandi (password), serta memasukkan Secret Key yang sebelumnya diberikan saat pembuatan device.



Configure network

Enter your network credentials to allow your device to connect to the Cloud.

Wi-Fi Name *
Kharisma

Password
12121212

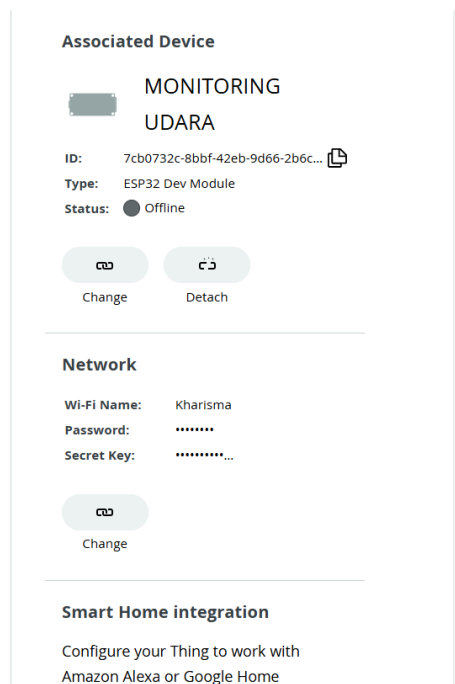
Secret Key *
lg4niAeUd9BlaqB2mC#ZCAVI7

IMPORTANT: Remember to go to the "Sketch" tab and upload the sketch to load the credentials on the board.

SAVE

Gambar 2. 33 Form Pengisian Configure Network

Langkah selanjutnya adalah mengisi bagian-bagian penting yang diperlukan agar perangkat dapat terhubung dengan jaringan dan sistem. Pada tahap ini, penulis mengisi informasi WiFi, Password, dan Secret Key yang sebelumnya sudah didapatkan dari file PDF yang telah diunduh.



Associated Device

MONITORING
UDARA

ID: 7cb0732c-8bbf-42eb-9d66-2b6c...

Type: ESP32 Dev Module

Status: Offline

Change Detach

Network

Wi-Fi Name: Kharisma

Password:

Secret Key:

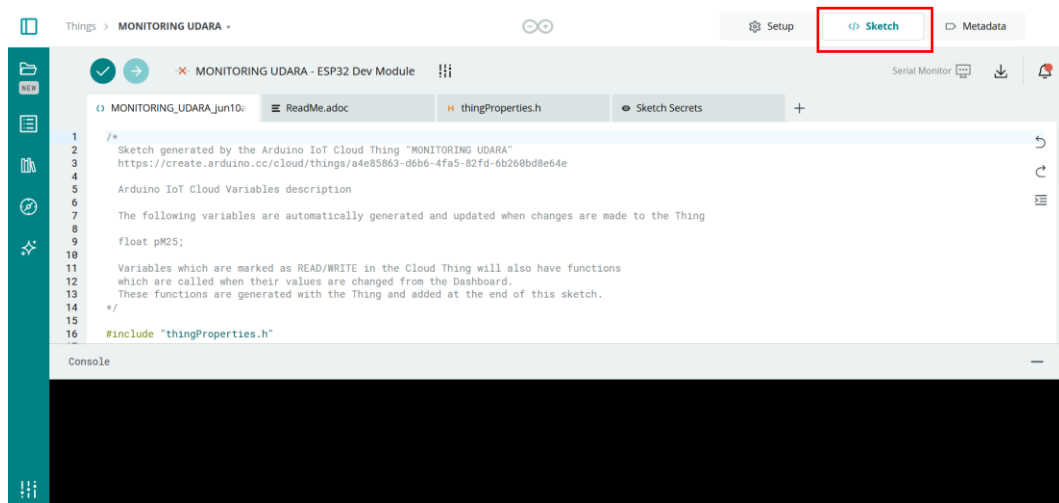
Change

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home

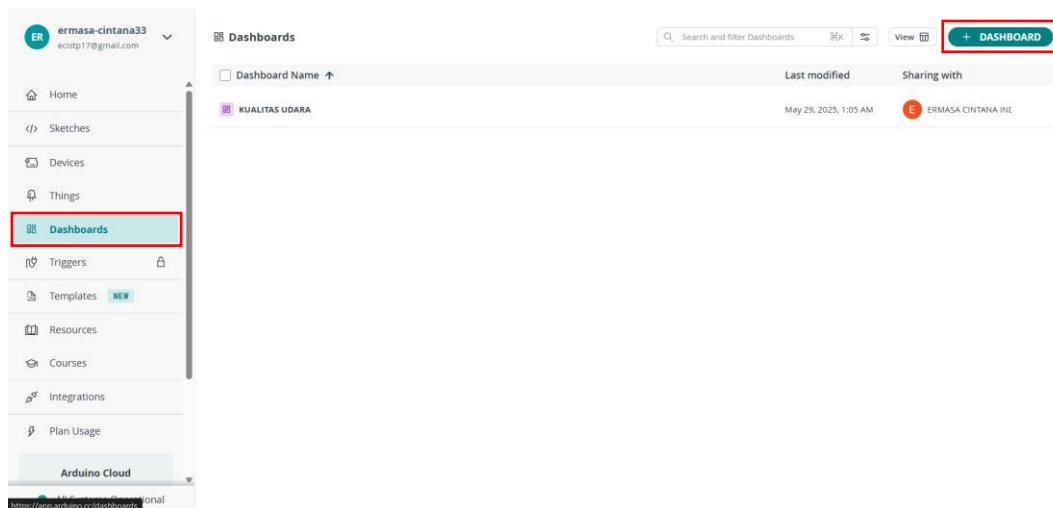
Gambar 2. 34 Tampilan Setelah Mengisi Setup dan configure Device

Setelah berhasil melakukan konfigurasi jaringan (configure network), tampilan antarmuka akan berubah secara otomatis untuk menandakan bahwa proses konfigurasi telah berhasil. Pada tampilan baru ini berisi informasi status koneksi perangkat yang menunjukkan bahwa perangkat sudah terhubung dengan jaringan WiFi dan siap berkomunikasi dengan platform IoT.



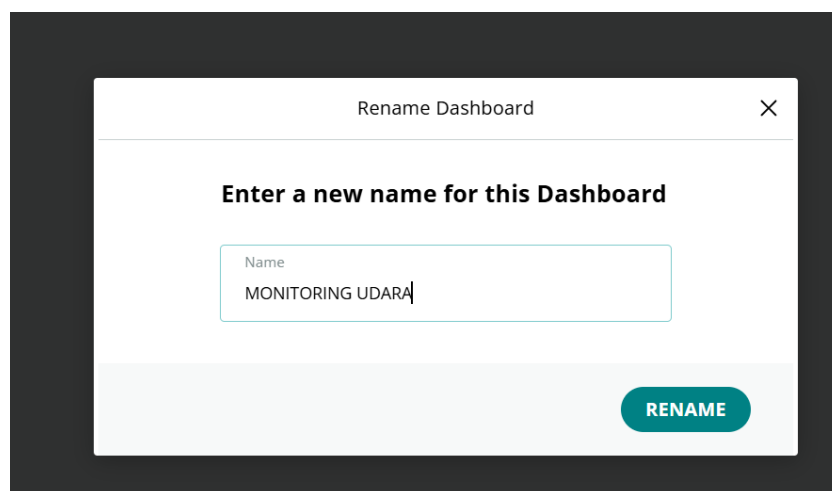
Gambar 2. 35 Tampilan Sketch Program

Setelah berhasil mengonfigurasi jaringan dan mendapatkan tampilan yang menandakan koneksi berhasil, langkah berikutnya adalah mengklik tombol Sketch yang terletak di pojok kanan atas antarmuka platform IoT. Dengan mengklik Sketch, kode program (program sketch) yang sudah dibuat secara otomatis sesuai dengan konfigurasi perangkat dan sensor yang digunakan akan muncul. Selanjutnya, salin kode program tersebut ke dalam Arduino IDE. Setelah kode berhasil disalin dan diunggah ke perangkat, langkah selanjutnya adalah kembali ke halaman Home pada platform IoT.



Gambar 2. 36 Tampilan Dashboard

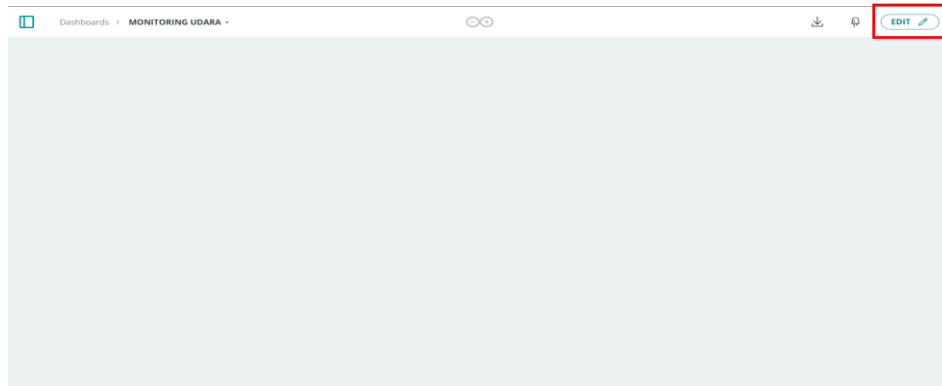
Dari halaman Home, klik menu Dashboard untuk mengatur tampilan antarmuka pengguna. Dashboard ini memungkinkan penulis mengelola bagaimana data dari perangkat akan ditampilkan, seperti grafik, indikator, atau kontrol yang dapat dioperasikan secara real-time. Pengaturan dashboard ini penting agar dapat memantau dan mengendalikan perangkat IoT dengan lebih mudah dan efektif sesuai kebutuhan proyek.



Gambar 2. 37 Rename Dashboard

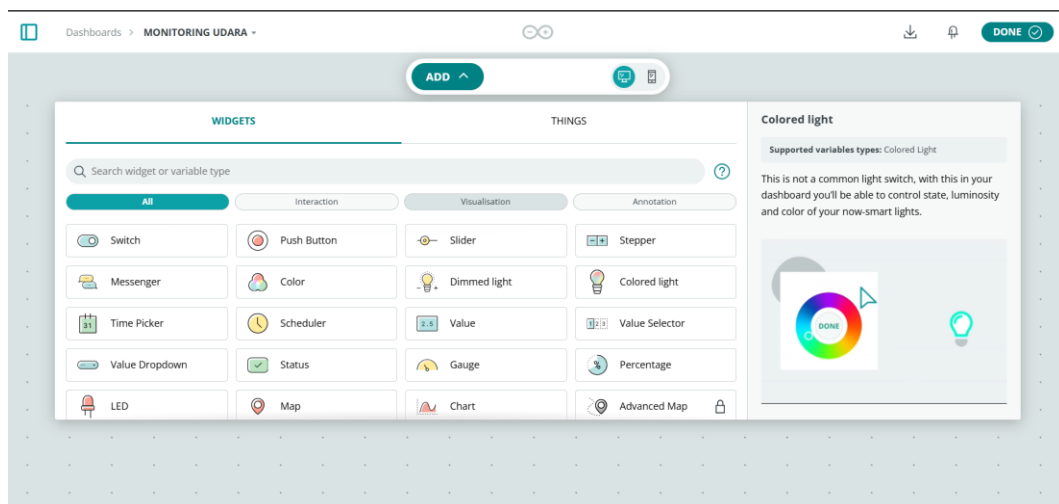
Setelah masuk ke halaman Dashboard, langkah selanjutnya adalah membuat nama baru untuk penamaan tampilan dashboard yang akan digunakan. Memberi

nama pada dashboard ini sangat penting agar dapat dengan mudah mengenali dan mengelola berbagai dashboard yang mungkin untuk proyek yang berbeda atau untuk berbagai perangkat.



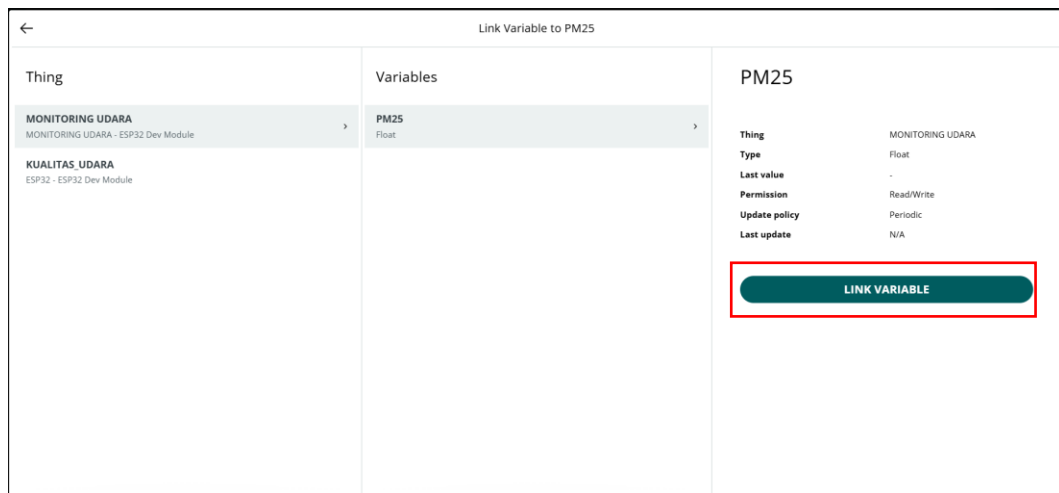
Gambar 2. 38 Tampilan sebelum ada variable

Langkah selanjutnya dengan mengklik “EDIT” pada pojok kanan atas tampilan antarmuka dashboard, kemudian aturlah sesuai dengan rancangan proyek IOT yang diinginkan.



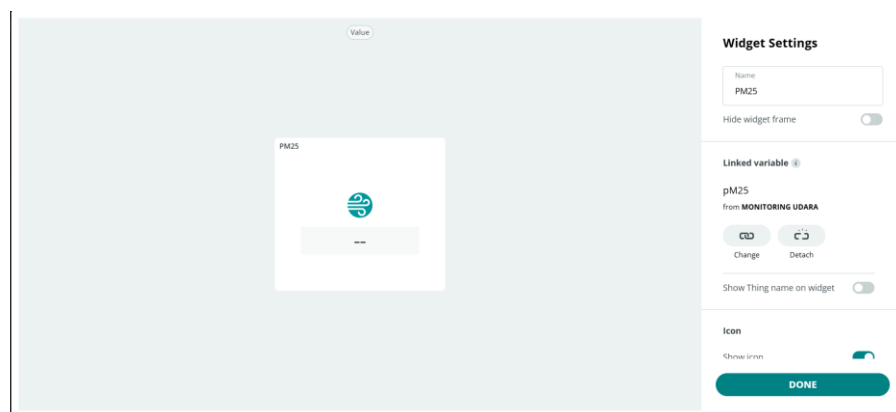
Gambar 2. 39 Tampilan ADD Dashboard

Pada langkah ini, pengguna dapat memilih jenis – jenis widget yang diperlukan untuk tampilan antarmuka proyek IOT kualitas udara, pada penelitian ini penulis menggunakan jenis Widgets seperti Gauge, Chart, Value untuk mendukung tampilan yang lebih baik lagi.



Gambar 2. 40 Tampilan LINK Variable

Langkah selanjutnya adalah tautkan link variabel dengan thing yang dibutuhkan. Proses ini berarti menghubungkan variabel-variabel data yang telah didefinisikan dalam perangkat (thing) ke elemen-elemen tampilan di dashboard, seperti grafik, indikator, atau kontrol. Dengan menautkan variabel ke thing, data yang dikirim oleh perangkat melalui jaringan akan secara otomatis diperbarui dan ditampilkan di dashboard secara real-time

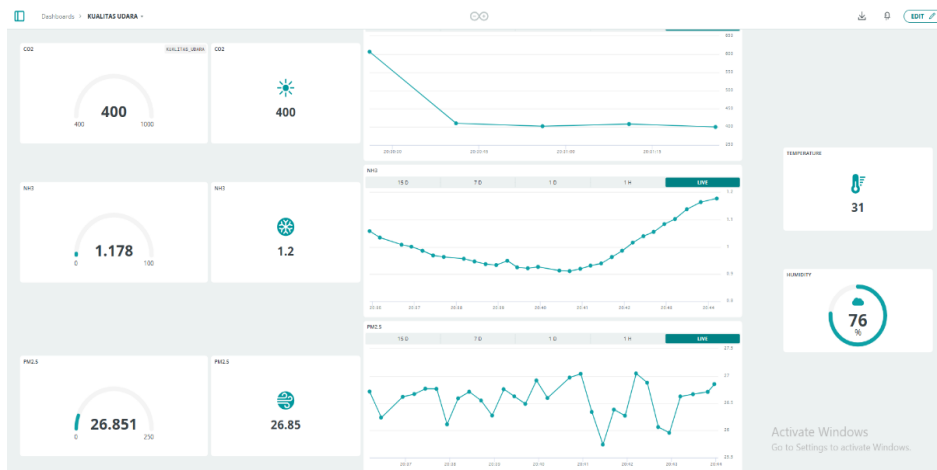


Gambar 2. 41 Tampilan Widgets Setting

Setelah selesai melakukan pengaturan pada dashboard dan mengklik tombol Done, tampilan akan berubah menampilkan dashboard yang sudah disesuaikan sesuai dengan konfigurasi yang di buat. Pada tampilan ini, pengguna dapat melihat

berbagai widget seperti grafik, indikator, tombol kontrol, atau elemen visual lain yang menampilkan data dari variabel yang telah ditautkan dengan thing.

4) Tampilan ketika Dashboard sudah dibuat dan sudah klik DONE



Gambar 2. 42 Tampilan Dashboard setelah di create

Gambar 2.42 memperlihatkan isi dashboard setelah di create yang berisi tampilan konsentrasi CO2, NH3, PM 2.5, suhu, kelembapan. Data tersebut dapat di download, namun untuk mengunduh ini pengguna lain juga harus memiliki akun. Gambar 2.15 memperlihatkan contoh program Arduino IOT Cloud yang digunakan.

```
const char DEVICE_LOGIN_NAME[] = "c384e2bd-c8dc-4dff-bacd-4c25163a7f0b";
```

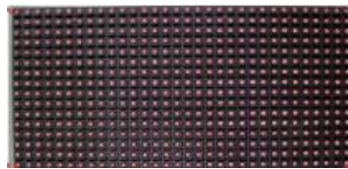
Gambar 2. 43 Contoh program Arduino IOT Cloud

Kelebihan dari platform Arduino IOT Cloud memberi kemudahan untuk integrasi dengan ke board ESP32, mendukung visualisasi data, dan menyediakan sketch otomatis. Namun, Arduino IOT Cloud maksimal pembuatan dua Things, dan dashboard hanya dapat create 5 parameter dalam satu things. Pemanfaatan data logger dimana ketika mengunduh nilai paramter CO2 file nya berbentuk “.csv”.

2.2.10 Dot Matriks Display P10

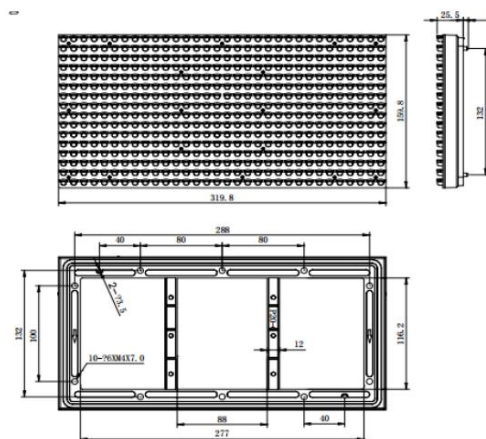
Panel P10 merupakan modul display yang dirancang untuk penggunaan semi-outdoor dengan fitur utama berupa tingkat kecerahan tinggi, konsumsi daya rendah, dan umur pakai yang panjang. Modul ini menggunakan konfigurasi 1/4 duty

scan drive dengan tingkat kecerahan antara 3500 hingga 4500 nits, konsumsi daya maksimum 20W, dan tegangan input DC 5V. Setiap panel P10 memiliki resolusi 32x16 pixel dengan pixel pitch 10mm, serta dilengkapi dengan konfigurasi 1W Pixel dan constant voltage drive untuk memastikan kecerahan yang stabil. Selain itu, panel P10 memiliki sudut pandang yang lebar, rasio kontras tinggi berkat louver/cover yang optimal, serta rating IP65 yang menunjukkan ketahanan terhadap air.[22]



Gambar 2. 44 DMD P10 Merah[22]

Terdapat beberapa varian warna yang tersedia, meliputi merah, kuning, biru, hijau, dan putih, dengan karakteristik elektrik yang serupa yaitu arus maksimum 4A, tegangan maksimum 5V, dan konsumsi daya rata-rata antara 8W hingga 10W. Perlu diperhatikan bahwa LED dan komponen panel P10 rentan terhadap benturan eksternal dan paparan terhadap zat pelarut organik. Selain itu, panel P10 juga sensitif terhadap listrik statis, sehingga disarankan untuk menggunakan earth band saat menangani modul. Paparan terhadap sinar UV dalam jangka waktu lama dapat menyebabkan perubahan warna pada LED dan PCB.



Gambar 2. 45 Skematik DMD P10[22]

```

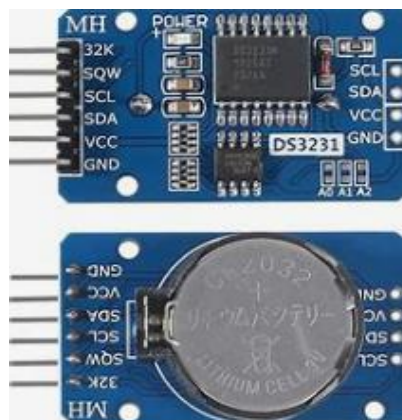
void P10_begin() {
  display.begin(4); // Sesuaikan dengan jumlah baris per scan panel Anda
  P10_clear();
  display.setBrightness(255); // Kecerahan (0-255)
  display.setTextColor(0xFF); // Atur warna teks default global setelah clear
}

```

Gambar 2. 46 Contoh Kode Program DMD P10

Kode program pada fungsi *P10_begin()* digunakan untuk menginisialisasi panel LED P10 sebelum digunakan. Pada bagian ini, *display.begin(4)* menjalankan inisialisasi panel dengan parameter yang menyesuaikan jumlah baris per scan pada panel yang digunakan. Fungsi *P10_clear()* akan membersihkan tampilan panel agar tidak ada data sisa sebelumnya. Selanjutnya, *display.setBrightness(255)* mengatur tingkat kecerahan panel ke nilai maksimum (skala 0-255), dan *display.setTextColor(0xFF)* mengatur warna teks default menjadi putih setelah panel dibersihkan

2.2.11 RTC DS3231

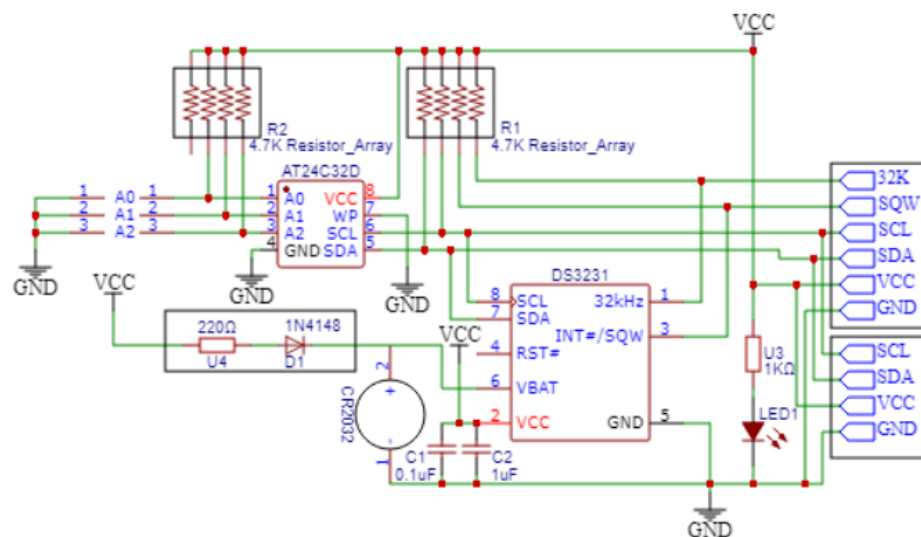


Gambar 2. 47 Modul RTC DS3231[23]

Gambar 2.47 merupakan bentuk dari modul RTC DS3231, modul Real-Time Clock (RTC) yang sangat presisi dan efisien dalam konsumsi daya. Menggunakan osilator kristal yang dikompensasi suhu (Temperature-Compensated Crystal Oscillator/TCXO), modul ini mampu mempertahankan akurasi waktu dengan sangat baik. Kemampuannya mencatat waktu mencakup detik, menit, jam, hari, tanggal, bulan, hingga tahun, dengan fitur otomatis yang menyesuaikan

tanggal pada bulan dengan kurang dari 31 hari serta memperhitungkan tahun kabisat.[23]

Dengan antarmuka komunikasi I2C, DS3231 memungkinkan pengaturan serta pembacaan waktu secara digital menggunakan berbagai mikrokontroler seperti Arduino, ESP32, dan lainnya. Selain itu, modul ini dilengkapi dengan dua alarm independen serta output gelombang persegi yang dapat diprogram sesuai kebutuhan. Keunggulan utama DS3231 terletak pada akurasinya yang tinggi—deviasi waktu hanya sekitar ± 2 ppm (setara dengan ± 1 menit per tahun) dalam rentang suhu 0°C hingga 40°C , serta tetap stabil dalam suhu ekstrem industri -40°C hingga $+85^{\circ}\text{C}$. Selain itu, adanya baterai cadangan memastikan waktu tetap berjalan meskipun sumber daya utama terputus, menjadikannya solusi ideal untuk sistem yang membutuhkan ketepatan waktu yang andal.[23]

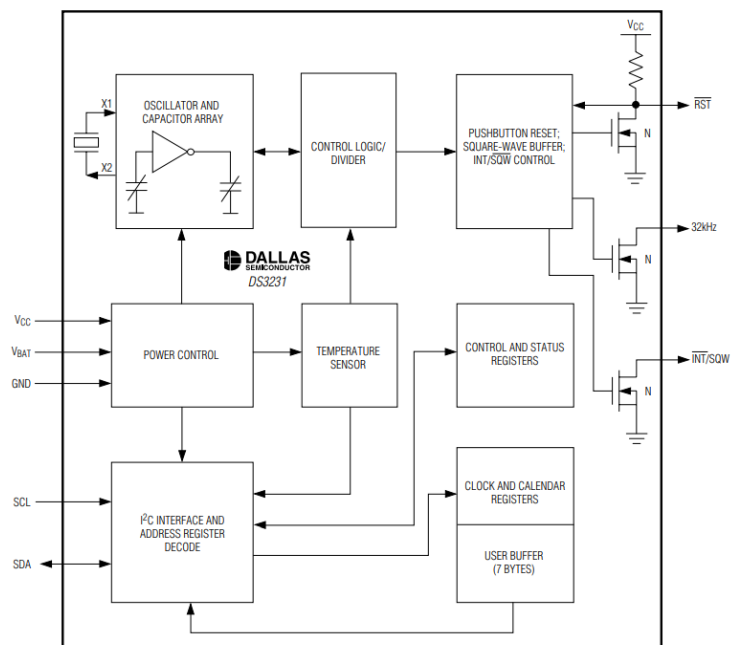


Gambar 2. 48 Skematik Modul RTC DS3231

Gambar 2.48 merupakan tampilan skematik dari modul RTC DS3231. Modul ini menggunakan IC DS3231 sebagai pencatat waktu utama dan AT24C32 sebagai memori EEPROM eksternal. Modul ini terhubung melalui protokol I2C menggunakan pin SCL dan SDA, yang juga dihubungkan ke resistor pull-up $4,7\text{ k}\Omega$ untuk memastikan komunikasi data yang stabil.

Tabel 2. 12 Spesifikasi Modul RTC DS3231[23]

Parameter	Keterangan
Tegangan Operasi (VCC dan VBAT)	2.3 V hingga 5.5 V
Rentang Suhu Operasi	Komersial: 0°C hingga +70°C, Industri: -40°C s/d +85°C
Akurasi Waktu	± 2 ppm (0°C hingga +40°C), ± 3.5 ppm (-40°C s/d +85°C)
Komunikasi	I2C (hingga 400kHz)
Output Frekuensi	32.768 kHz
Alarm	2 Alarm waktu Independen
Konsumsi Arus	Sangat rendah, cocok untuk aplikasi berbasis baterai



Gambar 2. 49 Diagram Blok modul RTC DS3231[23]

Modul DS3231 bekerja dengan menggunakan osilator kristal 32.768 kHz yang dikompensasi suhu untuk menjaga frekuensi osilasi tetap stabil meskipun terjadi perubahan suhu lingkungan. Sinyal osilator ini digunakan untuk menghitung waktu secara presisi.

1. Penghitungan Waktu: RTC mencatat waktu dalam format detik, menit, jam, hari, tanggal, bulan, dan tahun.
2. Komunikasi I2C: Mikrokontroler berkomunikasi dengan DS3231 melalui protokol I2C menggunakan dua jalur yaitu SDA (data) dan SCL (clock). Data waktu dapat dibaca dan diatur melalui register internal DS3231.
3. Backup Baterai: Ketika sumber daya utama terputus, modul otomatis beralih ke sumber baterai cadangan untuk menjaga waktu tetap berjalan tanpa kehilangan data.
4. Alarm dan Interrupt: DS3231 menyediakan dua alarm yang dapat diprogram untuk menghasilkan interrupt pada mikrokontroler, serta output gelombang persegi yang dapat diatur frekuensinya.
5. Output 32kHz: Pin output 32kHz dapat digunakan untuk aplikasi lain yang memerlukan sinyal referensi frekuensi stabil.

Tabel 2. 13 Konfigurasi Pin RTC DS3231[23]

Nama Pin	Keterangan Fungsi
32 kHz	Output sinyal frekuensi 32.768 kHz, open-drain, membutuhkan resistor pull-up eksternal jika digunakan.
VCC	Pin suplai tegangan utama modul, biasanya 3.3V atau 5V, perlu kapasitor decoupling 0.1 μ F–1 μ F.
INT/SQW	Output interrupt aktif rendah atau output gelombang persegi yang dapat diprogram frekuensinya.
RST	Pin reset aktif rendah, open-drain, dapat dihubungkan ke pushbutton untuk reset eksternal.
N.C	No Connection, biasanya dihubungkan ke ground atau tidak digunakan.
GND	Ground modul
VBAT	Input sumber daya cadangan (baterai), menjaga waktu tetap berjalan saat VCC mati.
SDA	Serial Data Line untuk komunikasi I2C, open-drain, memerlukan resistor pull-up eksternal.

Nama Pin	Keterangan Fungsi
SCL	Serial Clock Line untuk komunikasi I2C, open-drain, memerlukan resistor pull-up eksternal.

```
float RTC_update_time_and_get_temp() {
    DateTime now = rtc.now();
    currentTime.tahun = now.year();
    currentTime.bulan = now.month();
    currentTime.hari = now.day();
    currentTime.jam = now.hour();
    currentTime.menit = now.minute();
    currentTime.detik = now.second();
    return rtc.getTemperature();
}
```

Gambar 2. 50 Contoh Program RTC DS3231

Gambar 2.50 memperlihatkan contoh program yang dapat digunakan pada penelitian ini, namun pada penelitian ini penulis hanya menampilkan jam, menit, dan detik pada tampilan DMD P10. Program diatas dapat digunakan untuk engupdate (memperbarui) data waktu saat ini dari modul RTC (Real Time Clock)

2.2.12 Catu Daya 5 Volt

Catu daya merupakan sebuah peralatan yang digunakan untuk menghasilkan dan menyediakan sumber daya listrik yang cocok dan stabil pada suatu peralatan tertentu. Pada penelitian ini Catu daya 5 volt adalah sumber daya listrik yang digunakan untuk memberikan tegangan sebesar 5 volt kepada perangkat elektronik. Dalam konteks sistem pemantauan kualitas udara, catu daya 5 volt sering digunakan untuk mengaktifkan sensor-sensor yang memerlukan tegangan tersebut untuk beroperasi. Catu daya 5V umumnya disediakan melalui adaptor DC atau baterai yang diatur untuk menghasilkan tegangan yang stabil. Penggunaan catu daya yang tepat sangat penting untuk memastikan kinerja yang akurat dan stabil dari sensor dan perangkat lainnya dalam sistem pemantauan kualitas udara. Catu daya 5 Volt ditunjukkan gambar 2.51 berikut.[18]



Gambar 2. 51 Catu Daya 5V Switching

Klasifikasi Power Supply berdasarkan karakteristik dan metode konversinya terbagi menjadi beberapa jenis, yaitu:

1. Catu daya Switcing

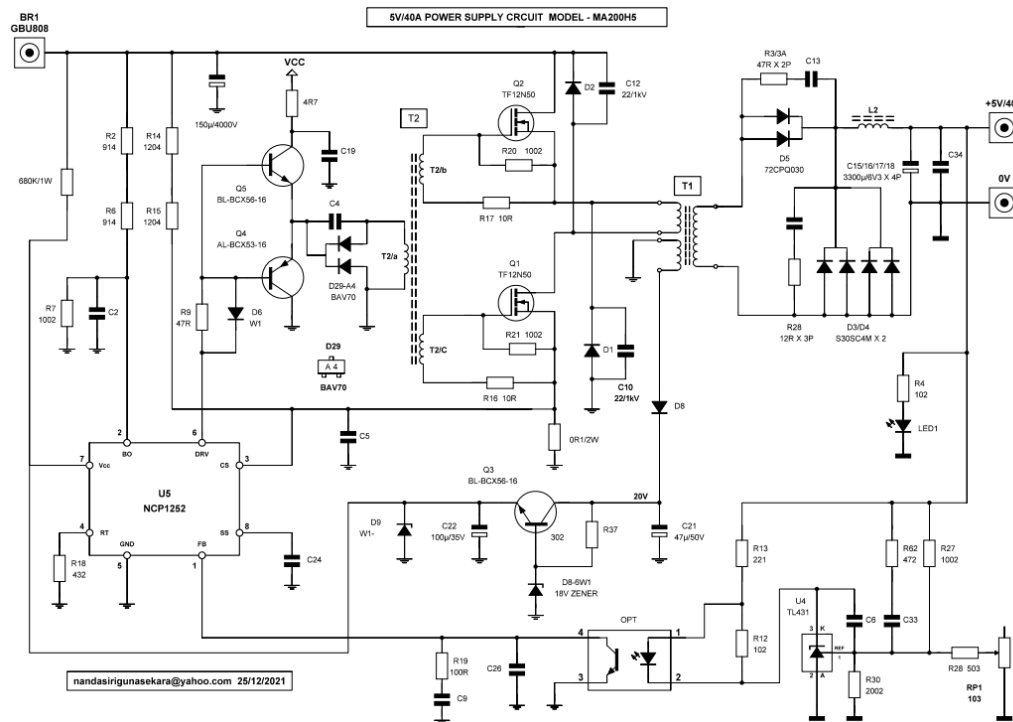
Catu daya mode saklar atau Switch mode power supply (SMPS) menggunakan teknik pensaklaran frekuensi tinggi untuk mengatur dan mengubah tegangan. Umum sekali digunakan pada benda elektronik

2. Catu daya Linear

Catu daya ini menggunakan trafo step – down untuk menurunkan tegangan AC, kemudian disearahkan dan distabilkan menggunakan regulator linear agar menghasilkan tegangan DC yang stabil.

3. Catu daya teregulasi dan tidak teregulas

power supply teregulasi menjaga tegangan output tetap konstan meskipun ada perubahan beban atau tegangan input, sedangkan power supply tidak teregulasi tidak memiliki mekanisme pengaturan tersebut.



Gambar 2. 52 Skematik Power Supplay 5V 40A

Tabel 2. 14 Spesifikasi Catu daya

Parameter	Spesifikasi
Application	Apply for Led Display
AC Input	200~240V
Input Frequency	50/60Hz
Tegangan Output	5V DC
Arus Maksimum	40 A
Tipe	Switch mode power supply