

**RANCANG BANGUN MONITORING KUALITAS UDARA DAN
PENANGANAN UDARA KOTOR BERBASIS IoT (*Internet of Things*)**

SKRIPSI



Oleh :

Nuril Fajril Huda

NPM : 2021501027

**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO
2025**

**RANCANG BANGUN MONITORING KUALITAS UDARA DAN
PENANGANAN UDARA KOTOR BERBASIS IoT (*Internet of Things*)**

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi Ibrahimi
Sukorejo Situbondo Jawa Timur



Oleh :

Nuril Fajril Huda

NPM : 2021501027

**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMI
SITUBONDO**

2025

PERNYATAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Nuril Fajril Huda
NPM : 2021501027
Program Studi : S-1 Ilmu Komputer
Fakultas : Fakultas Sains dan Teknologi

Menyatakan dengan sebenarnya, bahwa laporan Hasil Penelitian (SKRIPSI) ini secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali pada bagian-bagian yang dirujuk sebagai sumber referensi dan disebutkan dalam daftar Pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa laporan Hasil Penelitian (SKRIPSI) ini hasil plagiasi, maka saya siap menerima sanksi atas perbuatan tersebut.

Situbondo, 14 Juli 2025
Saya yang menyatakan,

Nuril Fajril Huda

PERSETUJUAN PEMBIMBING

Nama : Nuril Fajril Huda
NPM : 2021501027
Judul : Rancang Bangun Monitoring Kualitas Udara dan Penanganan
Udara Kotor Berbasis IoT (*Internet of Things*)

Telah disetujui oleh :

Pembimbing I

Ahmad Lutfi, M. Kom
NIDN. 0714108803

Pembimbing II

Farihin Lazim M.Tr.T.
NIDN. 0711099201

HALAMAN PENGESAHAN**SKRIPSI****RANCANG BANGUN MONITORING KUALITAS UDARA dan
PENANGANAN UDARA KOTOR BERBASIS INTERNET OF THINGS
(IoT)****NURIL FAJRIL HUDA****2021501027**

Telah dipertahankan didepan Dewan Penguji Sidang/munaqasah skripsi pada hari
..... tanggal.... sebagai salah satu syarat memperoleh gelar sarjana (S.Kom) pada
Fakultas Sains dan Teknologi Universitas Ibrahimy.

Tim Penguji,

Ketua Sidang,

Sekretaris Sidang,

Dr. Ach Khumaidi, M.P
NIDN. 0722049001**Miftahul Arifin, Amd. Pi**
NIDN. -

Penguji I,

Penguji II,

Ahmad Baijuri, M.Kom
NIDN. 0715078902**A. Hamdani, M.Kom**
NIDN. 0730118806Mengetahui
Dekan,**Abd. Ghofur, M. Kom**
NIDN. 0711088303

MOTTO

من احس بالنهاية فقد افسد البداية

Barang yang merasa sukses dalam menempuh tujuan, maka sebenarnya dia telah merusak jalan perjuangannya di masa lalu

(Abuya Sayyid Muhammad bin Alawai Al-Maliki)



KATA PENGANTAR

Segala puji syukur peneliti sampaikan kepada Allah SWT, karena atas rahmat dan hidayah-Nya, perencanaan, pelaksanaan dan penyelesaian tugas akhir/skripsi dengan judul ” Rancang Bangun Sistem Pembersih Saluran Air Berbasis IoT Pada Perumahan Dawuhan *Cluster* Kabupaten Situbondo ” sebagai salah satu syarat penyelesaian program diplomat/sarjana dapat terselesaikan dengan baik dan lancar.

Kesuksesan ini dapat peneliti peroleh karena dukungan beberapa pihak. Peneliti menyampaikan terima kasih kepada :

1. **KHR. Ach. Azaim Ibrahimy, S. Sy, M. HI** Selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo.
2. **KH. Ach. Fadlail, SH, M.H** Selaku Rektor Universitas Ibrahimy.
3. **Abd. Ghofur, M. Kom** Selaku Dekan Fakultas Sains dan Teknologi Universitas Ibrahimy.
4. **Farihin Lazim, M.Kom** Selaku Ketua Program Studi Ilmu komputer
5. **Ahmad Lutfi, M. Kom dan Farihin Lazim, M. Tr.T** selaku pembimbing I dan II
6. Teruntuk Abi dan Ummi (**Zainul Abidin, S.Pd dan Zahra Haviva Rasyida S.Pd**) yang slalu mencurahkan kasih sayang yang tiada batas, yang tanpa lelah mendo'akan dan mendukung peneliti untuk tetap kuat sehingga dapat menyelesaikan tugas akhir ini.
7. Keluarga kecil seperjuangan teman – teman angkatan 2021 yang selalu menghibur
8. Turuntuk teman hidup Penulis (*F*) yang selalu rela menjadi tempat pulang ketika duka, suka dan penguat dalam keadaan apapun, terima kasih atas doa dan dukungannya. *jazakumullahuKhoiron.*

Situbondo, 14 Juli 2025

Peneliti

DAFTAR ISI

PERSETUJUAN PEMBIMBING	iii
HALAMAN PENGESAHAN	iv
MOTTO	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Batasan Masalah	3
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
1.7 Metode Penelitian	4
1.7.1 Jenis Penelitian	4
1.7.2 Teknik Pengumpulan Data	4
1.7.3 Metode Pengembangan Sistem	5
1.8 Sistematika Pembahasan	7
BAB II TINJAUAN PUSTAKA	9
2.1 Penelitian Terdahulu	9
2.2 Landasan Teori	11
2.2.1 Rancang bangun	11
2.2.2 Monitoring	12
2.2.3 Kualitas Udara	12
2.2.4 Penanganan	12
2.2.5 Udara	13
2.3 Pemodelan	14

2.3.1	Flowchart	14
2.3.2	Diagram Blok	17
2.3.3	Bahasa C++	18
2.4	Perangkat Lunak yang Digunakan	19
2.4.1	Arduino Uno	19
2.4.2	Fritzing	19
2.4.3	Draw.io	20
2.4.4	Blynk	20
BAB III ANALISIS DAN PERANCANGAN SISTEM		22
3.1	Gambaran Umum Objek Penelitian	22
3.1.1	Objek Penelitian	22
3.1.2	Keadaan sistem yang bejalan	23
3.1.3	Kelebihan sistem	24
3.1.4	Kelemahan sistem	25
3.2	Alur Proses	26
3.2.1	Identifikasi Dan Analisis Proses Bisnis	27
3.2.2	Identifikasi dan Analisis Kebutuhan	29
3.2.3	Identifikasi dan Analisis Alternatif Solusi	32
3.3	Desain Sistem	33
3.3.1	Diagram Blok Sistem	33
3.3.2	Diagram Alir Sistem	35
3.4	Ilustrasi Desain	38
BAB IV PERANCANGAN SISTEM		50
4.1	Kontruksi Sistem	50
4.1.1	Kebutuhan sistem	50
4.1.2	Instalasi sistem	52
4.1.3	Segmen Program	57
4.2	Skenario Pengujian	74
4.3	Pengujian	77
4.4	Maintenance	90
BAB V PENUTUP		93
5.1	Kesimpulan	93

5.2	Saran.....	93
DAFTAR PUSTAKA.....		95
LAMPIRAN.....		99



DAFTAR TABEL

Tabel 2. 1 Flowchart Proses.....	16
Tabel 2. 2 Diagram Blok.....	17
Tabel 3. 1 Identifikasi Alternatif Solusi.....	32
Tabel 3. 2 Analisis Kelayakan Alternatif Solusi.....	33
Tabel 3. 3 Rencana Alat dan Anggaran.....	47
Tabel 4. 1 Spesifikasi Komputer.....	50
Tabel 4. 2 Pengujian Kualitas Udara dan Penanganan Udara Kotor.....	75
Tabel 4. 3 Pengujian Solar Tracking.....	76
Tabel 4. 4 Rentang Indeks Standart pencemaran Udara.....	80
Tabel 4. 5 Indeks Sumber Daya dan Tegangan.....	82
Tabel 4. 6 Persyaratan Kimia.....	83
Tabel 4. 7 Hasil Pengujian Monitoring dan Penanganan Udara Kotor.....	84
Tabel 4. 8 Pengujian Tegangan dan Arus Panel Surya.....	85
Tabel 4. 9 Pengujian Pengisian Baterai.....	85
Tabel 4. 10 Pengujian Beban.....	86
Tabel 4. 11 Perbandingan Alat IoT dengan Alat Konvensional.....	86
Tabel 4. 12 Perbandingan Pembacaan Sensor IoT dengan Konvensional.....	87
Tabel 4. 13 Pemeliharaan Perangkat Keras (Hardware).....	91
Tabel 4. 14 Pemeliharaan Perangkat Lunak (Software).....	92

DAFTAR GAMBAR

Gambar 1.1 Metode Engineering design Proses.....	6
Gambar 2.1 Logo arduino uno.....	19
Gambar 2.2 logo fritzing.....	19
Gambar 2.3 Logo draw.io.....	20
Gambar 2.4 logo aplikasi blynk.....	20
Gambar 3. 1 Diagram Blok Sistem Pemantauan Kualitas Udara dan Penanganan Udara Kotor.....	33
Gambar 3. 2 Diagram Blok Sistem Pada Solar Tracking.....	34
Gambar 3. 3 Flowchart Diagram Alir Pemantauan Kulaitas udara dan penanganan udara kotor.....	35
Gambar 3. 4 Flowchart Diagram Alir Solar Tracking.....	37
Gambar 3. 5 Sistem Monitoring Kualitas Udara / Air Quality.....	39
Gambar 3. 6 Monitoring Suhu Dan Kelembapan.....	40
Gambar 3. 7 Sistem Air Purifier Dan Air Humidity.....	41
Gambar 3. 8 Sistem Solar Tracking (Pelacak Matahari).....	42
Gambar 3. 9 Sistem Solar Panel Terintegrasi.....	43
Gambar 3. 10 Skema Keseluruhan Pemantauan Kualitas Udara Beserta Penanganannya.....	44
Gambar 3. 11 Skema Keseluruhan Solar Tracking Dengan Solar System.....	44
Gambar 3. 12 Tampilan Default Ketika Membuka Aplikasi Arduino UNO.....	45
Gambar 3. 13 Desain Case Seluruh Komponen.....	45
Gambar 3. 14 Case Alat Pemantau Dan Penjernih Udara Berbasis Iot.....	46
Gambar 4. 1 Instalasi Driver USB to serial pada Device Manager.....	52
Gambar 4. 2 Pembuatan New Sketch Aplikasi Arduino IDE.....	53
Gambar 4. 3 Tampilan Awal Sebelum Malakukan Input Program.....	53
Gambar 4. 4 Tampilan Menu Preferences.....	54
Gambar 4. 5 Tampilan Proses Konfigurasi pada Board.....	55
Gambar 4. 6 Proses Penginstalan Library Board ESP32 WROM.....	55
Gambar 4. 7 Tampilan Instlasi Library Sesuai Dengan Sketch Yang Kita Butuhkan	56

Gambar 4.8 Instalasi Blynk pada Arduino IDE.....	56
Gambar 4. 9 Tampilan Upload Program ke mikrokontroler ESP32 Wrom.....	57
Gambar 4. 10 Tampilan Upload Program ke ESP32.....	57
Gambar 4. 11 Diagram Warna Indeks Standart Pencemaran Udara.....	81
Gambar 4. 12 Grafik Indeks Standart Pencemaran Udara.....	81
Gambar 4. 13 Grafik Perbandingan.....	83
Gambar 4. 14 Tampilan Implementasi perangkat lunak.....	89
Gambar 4. 15 Implementasi Perangkat Keras.....	90



DAFTAR LAMPIRAN

A. Lampiran Code Pemrograman.....	100
B. Lampiran Penelitian.....	102
C. Skala dan Ukuran Tempat yang diteliti.....	103
D. Lampiran data Perbandingan alat konvensional.....	104
E. Dokumentasi.....	105



ABSTRAK

Nuril Fajril Huda 2021, **Rancang Bangun Monitoring Kualitas Udara Serta penanganan udara kotor berbasis *Internet of Things (IoT)***, Skripsi, Program Study Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Ibrahimy. Pembimbing: (I) Ahmad Lutfi, M.Kom., (II) Farihin Lazim, M.Tr.T

Kualitas udara dalam ruangan yang buruk dapat menimbulkan dampak negatif terhadap kesehatan manusia, seperti gangguan pernafasan dan penurunan produktifitas. Penelitian ini bertujuan untuk merancang dan membangun sistem monitoring kualitas udara dan penanganan udara kotor berbasis *Internet of Things (IoT)* yang dilengkapi dengan sumber daya tenaga surya sebagai solusi berkelanjutan. Sistem ini menggunakan *mikrokontroler* ESP32 dan beberapa sensor seperti MQ-135, MQ-7, dan GP2Y1010AU0F untuk mendeteksi gas berbahaya dan partikel debu, serta sensor DHT11 untuk suhu dan kelembapan. Data yang diperoleh ditampilkan secara *real-time* melalui LCD dan dikirim ke *platform Blynk* sebagai antarmuka pengguna. Sistem ini juga dilengkapi dengan mekanisme otomatis berupa kipas dan *air purifier* yang akan aktif jika kualitas udara berada di bawah ambang batas aman. Untuk menjamin ketersediaan energi, sistem terintegrasi dengan panel surya dan *solar charge controller* yang memungkinkan alat tetap beroperasi meskipun terjadi pemadaman listrik. Hasil pengujian menunjukkan bahwa sistem mampu bekerja secara stabil, responsif terhadap perubahan kualitas udara, dan efisien dalam konsumsi energi. Sistem ini diharapkan dapat menjadi solusi cerdas dan ramah lingkungan dalam menjaga kualitas udara dalam ruangan, khususnya di lingkungan perumahan.

Kata Kunci : *Kualitas Udara, Internet of Things (IoT), ESP32, Sensor Gas, Blynk, Panel Surya*

ABSTRACT

*Nuril Fajril Huda 2021, **Design and Development of Air Quality Monitoring and Polluted Air Treatment System Based on the Internet of Things (IoT)**, Undergraduate Thesis, Computer Science Study Program, Faculty of Science and Technology, Ibrahimi University. Advisors: (I) Ahmad Lutfi, M.Kom., (II) Farihin Lazim, M.Tr.T*

Poor indoor air quality can have negative impacts on human health, such as respiratory problems and decreased productivity. This study aims to design and develop an air quality monitoring and polluted air treatment system based on the Internet of Things (IoT), equipped with solar power as a sustainable solution. The system utilizes an ESP32 microcontroller and several sensors such as MQ-135, MQ-7, and GP2Y1010AU0F to detect harmful gases and dust particles, as well as a DHT11 sensor for temperature and humidity. The collected data is displayed in real-time through an LCD and sent to the Blynk platform as a user interface. The system is also equipped with an automatic mechanism consisting of a fan and an air purifier that activate when air quality falls below a safe threshold. To ensure energy availability, the system integrates solar panels and a solar charge controller, allowing it to operate even during power outages. Test results show that the system can operate stably, respond to changes in air quality, and be energy efficient. This system is expected to be a smart and environmentally friendly solution for maintaining indoor air quality, particularly in residential environments.

Keywords: Air Quality, Internet of Things (IoT), ESP32, Gas Sensor, Blynk, Solar Panel

BAB I PENDAHULUAN

1.1 Latar Belakang

Lingkungan yang sehat sangat berpengaruh terhadap kesehatan fisik makhluk hidup. Faktor penting penunjang lingkungan yang sehat adalah kualitas udara yang memenuhi standart kesehatan. Udara mengandung oksigen yang dibutuhkan untuk hidup. Faktor penting penunjang lingkungan yang sehat adalah kualitas udara yang memenuhi standart kesehatan. Udara mengandung oksigen yang dibuthkan untuk hidup. Namun selain oksigen, terdapat zat lainnya dalam udara seperti karbon monoksida, karbon dioksida, formaldehid, jamur, virus, bakteri, debu dan sebagainya. Oksigen didalam maupun di luar ruangan dapat terkontaminasi dengan zat – zat yang berbahaya bagi kesehatan. Dalam batasan tertentu kadar zat -zat tersebut masih dapat dinetralisir namun jika melampaui batas normal maka dapat mengganggu kesehatan.[1] *World Health Organization* (WHO) menyatakan bahwa terdapat zat berbahaya yang berasal dari bangunan, material konstruksi, peralatan, proses pembakaran atau pemanasan dapat memicu masalah kesehatan.[2] Menurut *Environtmental Protection Agency* (EPA), 40% dari waktu kita sehari, kita berada di dalam ruangan dalam rumah, kantor, kendaraan, supermarket, sekolah dan tempat umum lainnya.[3]

Pencemaran udara merupakan kombinasi dari berbagai zat kimia, partikel padat, dan gas yang dapat menyebabkan dampak negatif pada kesehatan manusia dan lingkungan.[4] Kualitas udara yang buruk dapat menyebabkan berbagai masalah kesehatan, termasuk gangguan pernapasan, iritasi mata, alergi, hingga penyakit paru-paru kronis dan bahkan kematian dini.[5] Selain itu, pencemaran udara juga berkontribusi pada perubahan iklim global melalui efek rumah kaca dan pemanasan global.[6] Besarnya dampak buruk yang diakibatkan polusi udara ini mengharuskan adanya tingkat perhatian khusus dalam pengendaliannya dari berbagai pihak mulai dari pemerintah maupun masyarakat.[7] Saat ini indeks standar kualitas udara yang dipergunakan secara resmi di Indonesia adalah Indek Standar Pencemar Udara (ISPU), hal ini sesuai dengan Keputusan Menteri Negara Lingkungan Hidup Nomor : KEP 45 /

MENLH / 1997 Tentang Indeks Standar Pencemar Udara. Dalam keputusan tersebut yang dipergunakan sebagai bahan pertimbangan diantaranya, bahwa untuk memberikan kemudahan dari keseragaman informasi kualitas udara ambien kepada masyarakat di lokasi dan waktu tertentu serta sebagai bahan pertimbangan dalam melakukan upaya-upaya pengendalian pencemaran udara perlu disusun Indeks Standar Pencemar Udara.[8] Kesadaran akan pentingnya menjaga kualitas udara yang bersih semakin meningkat, namun upaya untuk memonitoring dan mengelola kualitas udara masih menghadapi banyak tantangan. Salah satu tantangan utama adalah kurangnya data yang akurat dan real-time mengenai kualitas udara di berbagai lokasi.[9] Sebagian besar sistem monitoring kualitas udara yang ada saat ini masih menggunakan atau mengandalkan petugas kesehatan yang dimana dengan cara konvensional ini dengan menggunakan petugas kesehatan yang tidak *real-time* terpantau.[10]

Penggunaan suatu teknologi IoT dalam sistem monitoring dan deteksi dini pencemaran udara menawarkan sejumlah keuntungan yang luar biasa.[11] *Internet of Things* adalah teknologi baru yang memungkinkan interaksi perangkat komputasi yang dapat diidentifikasi secara unik yang dapat disematkan dengan antarmuka lain seperti mesin dan manusia, dihubungkan melalui jaringan kabel dan nirkabel, untuk menangkap data kontekstual dari lingkungan yang telah ditandai dan membuat informasi. IoT juga merupakan jaringan yang menyediakan fungsionalitas baru dan model bisnis digital.[12] Namun, tantangan lain muncul ketika sistem monitoring ini bergantung pada sumber daya listrik konvensional. Gangguan listrik dapat menghambat operasional perangkat IoT, yang pada akhirnya mengurangi efektivitas monitoring kualitas udara. Oleh karena itu, penggunaan energi surya sebagai sumber daya alternatif untuk sistem IoT ini menjadi solusi yang layak dipertimbangkan. Energi surya tidak hanya ramah lingkungan, tetapi juga dapat memastikan bahwa perangkat tetap beroperasi bahkan saat terjadi pemadaman listrik. Dengan integrasi energi surya, sistem monitoring kualitas udara berbasis IoT dapat memberikan manfaat ganda pengumpulan data kualitas udara yang berkelanjutan dan penggunaan energi yang lebih bersih. Hal ini akan membantu mengurangi dampak polusi udara serta mendukung inisiatif energi hijau yang

berkelanjutan. Oleh karena itu, pengembangan sistem monitoring kualitas udara dan penanganan udara kotor berbasis IoT dengan bantuan energi surya agar menjadi solusi yang relevan untuk mengurangi dampak negatif polusi udara sekaligus mendukung penggunaan energi terbarukan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat di tentukan identifikasi masalah yang didapat dalam Rancang Bangun Monitoring Kualitas Udara dan Penanganan Udara Kotor Berbasis IoT (*Internet of Things*) yaitu :

- a. Sulitnya memantau kualitas udara sehingga setiap melakukan pengecekan harus mendatangkan petugas kesehatan
- b. Kurangnya sistem monitoring kualitas udara dan penanganan udara kotor berbasis IoT yang terintegrasi dan mudah di akses oleh masyarakat.
- c. Ketergantungan pada sumber daya listrik konvensional yang rentan terhadap gangguan seperti pemadaman listrik.

1.3 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah dijelaskan di atas, rumusan masalah dalam penelitian ini adalah “Bagaimana merancang dan membangun sistem monitoring kualitas udara serta penanganan udara kotor berbasis *Internet of Things* (IoT) di Perumahan Dawuhan *Cluster* Kabupaten Situbondo”.

1.4 Batasan Masalah

Agar tidak terjadi penyimpangan masalah yang akan dibahas, maka permasalahan dalam menyusun skripsi ini terdapat beberapa batasan masalah sebagai berikut:

- a. Sistem hanya mencakup monitoring kualitas udara dan penanganan udara kotor di dalam ruangan
- b. Sistem ini akan dirancang untuk bekerja secara optimal dengan bantuan energi surya ketika pemadaman arus listrik,
- c. Area uji coba hanya dibatasi didalam satu ruangan, dan berada di lingkup perumahan Dawuhan Cluster Situbondo

1.5 Tujuan Penelitian

Tujuan penelitian ini adalah merancang dan membangun monitoring kualitas udara dan penanganan udara kotor berbasis IoT (*Internet of Things*) adalah sebagai berikut :

- a. Mengembangkan sistem monitoring kualitas udara dan penanganan udara kotor berbasis IoT yang mampu memberikan data *real – time* dan mudah di akses.
- b. Mengintegrasikan energi surya sebagai alternatif sumber daya listrik untuk memastikan keberlanjutan operasional sistem.
- c. Mengidentifikasikan parameter kualitas udara yang perlu dimonitoring secara berkala untuk mendapatkan informasi yang akurat.

1.6 Manfaat Penelitian

- a. Menyediakan solusi praktis dan ramah lingkungan untuk memantau serta menangani udara kotor secara *real-time*.
- b. Meningkatkan kesadaran masyarakat tentang pentingnya menjaga kualitas udara yang sehat.
- c. Mengurangi ketergantungan pada listrik konvensional, yang dapat menekan biaya operasional dan mendukung penggunaan energi terbarukan.

1.7 Metode Penelitian

1.7.1 Jenis Penelitian

Pada jenis penelitian ini yang digunakan adalah menggunakan penelitian dan pengembangan *Research and Development* (R&D), penelitian ini bertujuan untuk merancang dan mengembangkan sistem monitoring kualitas udara dan penanganan udara kotor berbasis *Internet of Things* (IoT) yang di implementasikan di perumahan Dawuhan *Cluster*.

1.7.2 Teknik Pengumpulan Data

Metode pengumpulan data adalah tahapan untuk memperoleh informasi-informasi yang diperlukan peneliti untuk menyelesaikan penelitiannya, pada penelitian ini menggunakan 3 metode pengumpulan data, yaitu :

a. *Studi Literatur*

Study Literatur dilakukan dengan meneliti berbagai sumber ilmiah seperti jurnal, buku dan artikel yang relevan dengan teknologi IoT. kualitas udara dan penggunaan energi surya. Kajian ini akan membantu memahami perkembangan terkini dalam bidang monitoring kualitas udara dan penanganan udara kotor dengan teknologi pendukungnya.

b. Observasi

Observasi lapangan akan dilakukan di lokasi yang dimana akan dilakukan uji coba untuk memahami kondisi lingkungan yang akan di pantau. Kegiatan ini meliputi pengukuran awal sebagai parameter kualitas udara menggunakan alat konvensional, serta identifikasi potensi gangguan yang dapat mempengaruhi kinerja sistem.

c. Wawancara

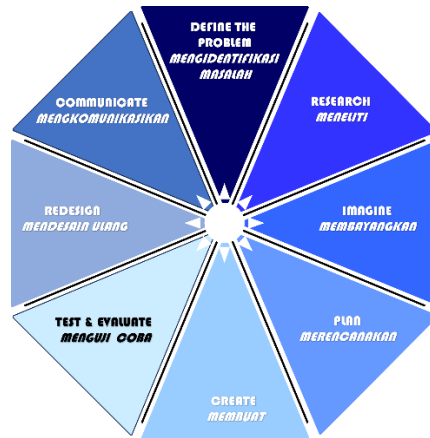
Wawancara adalah metode pengumpulan data yang melibatkan interaksi langsung antara peneliti dan responden. Tujuannya untuk menggali pandangan, pengalaman, dan pendapat responden mengenai kondisi saluran air, permasalahan yang dihadapi, serta kebutuhan dan harapan mereka terhadap pemantauan kualitas udara beserta penanganan udara kotor. Informasi ini membantu memastikan solusi yang dikembangkan sesuai dengan kebutuhan pengguna.

Dengan pendekatan ini, penelitian dapat lebih terfokus pada pengembangan yang sesuai dengan kebutuhan spesifik dan pengujian yang menyeluruh untuk memastikan efektivitas sistem monitoring kualitas udara dan penanganan udara kotor berbasis IoT (*Internet of Things*).

1.7.3 Metode Pengembangan Sistem

Untuk membangun sistem ini diperlukan tahapan-tahapan yang harus diselesaikan. Dengan menggunakan metode *Engineering Design Process* (EDP). *Engineering Design Process* (EDP) adalah salah satu strategi yang tersedia untuk mengimplementasikan pendidikan STEM.[13] Dalam pendapat ABET (*Accreditation Board for Engineering and Technology*) menjelaskan EDP adalah proses merancang suatu sistem,

komponen, atau proses untuk memenuhi kebutuhan yang diinginkan.[14]
Metode ini menggunakan alur siklus seperti **Gambar 1.1** dibawah ini :



Gambar 1. 1 Metode *Engineering Design Proses*

- a. Mengidentifikasi Masalah
Pada tahap ini, peneliti mengamati dan mengenali masalah yang terjadi. Fokus utama adalah memahami kebutuhan, tantangan, atau kesulitan yang dihadapi oleh pengguna atau situasi tertentu. Tujuannya adalah menentukan masalah yang benar – benar perlu diselesaikan.
- b. Meneliti
Setelah masalah diidentifikasi, langkah selanjutnya adalah mengumpulkan informasi. Penelitian bisa dilakukan melalui wawancara, *study literatur*, survei, observasi, atau eksperimen. Tahap ini penting untuk memahami konteks dan menemukan solusi yang relevan.
- c. Membayangkan
Tahap ini adalah *brainstorming* atau curah pendapat. Peneliti mencoba membayangkan berbagai kemungkinan solusi yang inovatif dan kreatif untuk menyelesaikan masalah berdasarkan hasil penelitian sebelumnya.
- d. Merencanakan
Setelah memiliki ide, pada tahap ini menyusun rencana aksi yang sistematis. Termasuk dalam tahap ini adalah membuat sketsa, diagram, daftar bahan, dan urutan pengerjaan.

- e. Membuat
Pada tahap ini, ide direalisasikan menjadi bentuk nyata atau prototipe. Ini bisa berupa model fisik, simulasi digital, atau produk awal yang bisa diuji coba.
- f. Menguji coba
Sistem dan alat yang dibuat untuk mengetahui apakah dapat bekerja sesuai tujuan. Pengujian ini bisa dilakukan dalam kondisi nyata atau simulasi, untuk mengetahui kelebihan dan kekurangannya..
- g. Mendesain ulang
Berdasarkan hasil pengujian, perbaikan dilakukan jika ada kekurangan. Sistem dan alat diperbaiki, ditingkatkan atau bahkan diubah total agar hasilnya lebih baik.
- h. Mengkomunikasikan
Tahap akhir adalah menyampaikan hasil dari seluruh proses, baik melalui laporan, presentasi, poster atau publikasi. Tujuannya adalah membagikan pengetahuan, solusi atau produk kepada orang lain.

1.8 Sistematika Pembahasan

Adapun sistematika pembahasan pada penelitian skripsi terdiri dari beberapa bab sebagai berikut, yang di susun secara sistematis untuk memberikan alur pembahasan yang jelas dan terstruktur :

BAB I PENDAHULUAN

Pada bab ini membahas tentang hal-hal yang bersangkutan tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian yang meliputi jenis penelitian, teknik pengumpulan data dan metode pengembangan sistem serta sistematika pembahasan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dipaparkan tentang tujuan atau kajian pustaka dan landasan teori tentang hal-hal yang berkaitan dengan proses sistem saluran air yang berbasis IoT.

BAB III OBJEK PENELITIAN DAN ANALISIS PROSES BISNIS

Pada bab ini menjelaskan tentang analisa, rancangan dan gambaran umum fitur dan alat yang dirancang.

BAB IV PERANCANGAN SISTEM

Bab ini berisi tentang gambaran umum obyek penelitian, alur proses dan desain atau perancangan sistem yang akan di buat.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dari awal penelitian hingga akhir penelitian yang dilakukan oleh peneliti. Serta kritik dan saran yang membangun terhadap media yang telah di buat.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam penelitian akan digunakan berbagai tinjauan studi yang nantinya akan mendukung penelitian yang akan dilakukan, dimana tinjauan studi yang di ambil antara lain :

a. Perancangan Sistem Monitoring Kualitas Udara dalam Ruangan Berbasis *Internet of Things*.

Jurnal ini di buat oleh Hendi Budianto dan Budi Sumanto tentang bagaimana merancang Sistem Monitoring Kualitas Udara dalam Ruangan Berbasis *Internet of Things*.

Saat ini, berbagai penelitian telah dilakukan untuk mengembangkan sistem pemantauan kualitas udara berbasis *Internet of Things* (IoT). Sistem ini memungkinkan pengumpulan data kualitas udara secara real-time yang dapat diakses melalui web atau perangkat lain. Dalam beberapa penelitian, digunakan berbagai sensor seperti MQ135 untuk deteksi CO₂, MQ7 untuk karbon monoksida, dan GP2Y1010AU0F untuk partikel debu. Sistem pemantauan berbasis IoT telah menunjukkan keefektifannya. Data dari sensor dikirimkan ke database melalui perangkat seperti ESP32 atau NodeMCU ESP8266, yang kemudian divisualisasikan dalam bentuk grafik pada sebuah dashboard web. Sistem ini memberikan kemudahan bagi pengguna untuk memantau kualitas udara di ruangan tanpa harus melakukan pengukuran secara manual. Beberapa penelitian sebelumnya menunjukkan bahwa sistem monitoring IoT dapat mengukur berbagai parameter kualitas udara, seperti konsentrasi gas, suhu, dan kelembaban. Misalnya, penelitian menggunakan ESP32 yang memanfaatkan protokol MQTT untuk mengirim data ke server, atau NodeMCU ESP8266 untuk pengolahan data secara real-time. Dalam penelitian yang dilakukan oleh Grace C. Rumampuk dkk., sistem ini memanfaatkan berbagai sensor untuk mendeteksi parameter polusi udara. Data hasil pembacaan sensor ditransmisikan melalui *Wi-Fi* dan ditampilkan dalam bentuk antarmuka web yang *user-friendly*. Sistem

pemantauan manual yang sering digunakan oleh petugas kesehatan lingkungan memerlukan kehadiran fisik untuk pengukuran di lokasi. Pendekatan ini tidak hanya memakan waktu, tetapi juga kurang efisien dibandingkan dengan sistem berbasis IoT yang dapat memberikan informasi secara otomatis dan *real-time*. Hal ini juga mendukung integrasi fitur tambahan, seperti alarm peringatan kualitas udara yang buruk.[15]

b. Perancangan Sistem Pembersih Udara Menggunakan Metode *Fuzzy Mamdani* untuk kontrol kipas berbasis IoT (Internet of Things).

Skripsi ini dibuat oleh Zaki Muhammad tentang bagaimana merancang sistem pembersih udara menggunakan metode *Fuzzy Mamdani* untuk kontrol kipas berbasis IoT (Internet of Things) .

Kesimpulan yang dapat diambil dari penelitian ini adalah sistem pembersih udara menggunakan metode *fuzzy mamdani* berbasis IoT telah berhasil dibangun. Sistem ini dibangun dengan input asap, PM 2,5 dan suhu untuk menghasilkan kecepatan kipas. Pembangunan sistem ini menggunakan Arduino, NodeMCU, sensor – sensor untuk inputan dan beberapa *hardware* lainnya. Perangkat lunak yang digunakan adalah web based dengan native javascript dan menggunakan *web server* serta database dari firebase. Alat ini telah berjalan dengan baik, pada pengujian sistem *fuzzy* dengan cara membandingkan hasil *output* mikrokontroler dengan perhitungan manual. Maka dengan menggunakan standart tingkat akurasi oleh Goronescu, akurasi sistem ini dapat digolongkan ke klasifikasi Excellent Classification. Pengujian selanjutnya dengan cara membandingkan output dari mikrokontroler dengan perhitungan matlab untuk mencari *error* dari output sistem *fuzzy*. Pengujian ini menghasilkan rata – rata *error* sebesar 5,8 % dapat dikatakan *error* yang didapatkan cukup rendah. Pemilihan metode *fuzzy mamdani* ini merupakan metode yang intuitif, sangat cocok untuk inputan manusia, *rule base* yang lebih mudah diinterpretasikan, dan memiliki penerimaan yang luas.[16]

c. Rancang Bangun Sistem Pemantauan dan Kontrol Kualitas Udara dalam ruangan berbasis IoT.

Jurnal ini di buat oleh Fredy Agung Dwi Cahyono dan Denny Irawan tentang bagaimana merancang dan membangun sistem pemantauan dan kontrol kualitas udara dalam ruangan berbasis IoT

Sistem pemantauan dan pengendalian kualitas udara dalam ruangan berbasis IoT yang telah dirancang berhasil mengukur parameter kualitas udara seperti CO, gas dan debu dengan tingkat akurasi yang memadai. Sistem ini dapat mengontrol secara otomatis, seperti mengaktifkan kipas DC berdasarkan data dari sensor. Dengan demikian, kualitas udara dalam ruangan dapat terjaga dengan baik. Perbedaan utama dalam penelitian ini terletak pada penerapan sensor GP2Y1010AU0F, yang belum banyak diintegrasikan dalam penelitian sebelumnya. Meskipun sistem telah berfungsi dengan baik, ketergantungan pada koneksi internet dan kebutuhan untuk kalibrasi sensor secara berkala menjadi tantangan. Ke-depan, sistem ini dapat dikembangkan dengan menambahkan sensor tambahan dan meningkatkan fitur pencatatan data untuk meningkatkan keandalan sistem.[17]

2.2 Landasan Teori

2.2.1 Rancang bangun

Widyawati menjelaskan rancang bangun adalah gambaran dari sistem untuk menciptakan sistem baru atau memperbaiki sistem sebelumnya. Adapun menurut Gunawan menjelaskan bahwa: “Rancang bangun adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi”.

Berdasarkan pendapat beberapa ahli tersebut, dapat disimpulkan bahwa rancang bangun merupakan proses mengembangkan dan memperbaiki sistem atau aplikasi yang sudah ada ataupun belum ada dengan beberapa komponen yang digunakan dari hasil proses analisa sistem dimana mengharapkan dapat menghasilkan aplikasi yang berkualitas sesuai dengan kebutuhan yang diinginkan.[18]

2.2.2 Monitoring

Monitoring adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang kegiatan / tindakan koreksi untuk penyempurnaan program / kegiatan itu selanjutnya. Monitoring adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*Awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. Monitoring akan memberikan informasi tentang status dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.[19]

2.2.3 Kualitas Udara

Kualitas udara merupakan alat untuk mengukur kondisi udara yang layak di sebuah area. Disebabkan oleh aktifitas manusia, udara menjadi lebih tercemar. Penurunan kualitas udara disebabkan oleh beberapa jenis gas, asap kendaraan, asap industri dan limbah rumah tangga. Sistem pernapasan, kesehatan umum, dan produktivitas dapat terpengaruh oleh pencemaran udara. Hal ini disebabkan oleh zat-zat yang tercemar dalam udara yang masuk ke paru-paru dan menyebabkan peradangan. Karena kemampuan sistem kekebalan tubuh yang lebih lemah, anak-anak lebih rentan terhadap peradangan ini daripada orang dewasa.

Alat pemantau kualitas udara berbasis IoT ini menggunakan sensor MQ untuk mendeteksi kadar gas serta sensor DHT 22 untuk mengukur kelembaban dan suhu. Sangat penting untuk memantau kadar gas-gas ini di udara untuk memastikan kesehatan dan keselamatan manusia. Adapun kadar aman untuk beberapa faktor yang akan diukur dalam alat ini, seperti : LPG (*Liquefied Petroleum Gas*), *Methane* (CH₄), *Hidrogen* (H₂), *Karbon Monoksida* (CO), *Karbon Dioksida* (CO₂), Kelembaban dan Suhu.[20]

2.2.4 Penanganan

Penanganan menurut Kamus Besar Bahasa Indonesia penanganan memiliki satu arti yaitu penanganan dan berasal dari kata dasar tangan. Penanganan memiliki

arti yang menyatakan sebuah tindakan yang dilakukan dalam melakukan sesuatu. Penanganan juga dapat berarti proses, cara, perbuatan menanganai sesuatu yang sedang dialami.[21]

2.2.5 Udara

Menurut Kamus Besar Bahasa Indonesia (KBBI) *on-line*. Terdapat beberapa definisi tentang udara, yaitu udara merupakan campuran berbagai gas yang tidak berwarna dan tidak berbau (seperti oksigen dan nitrogen) yang memenuhi ruang di atas bumi seperti yang kita hirup apabila kita bernafas. Udara juga merupakan ruang di atas bumi yang berisi hawa atau disebut juga angkasa. Udara juga merupakan keadaan hawa (cuaca, hari).[22]

2.2.6 Pencemaran Udara

Pada intinya pengertian pencemaran udara adalah masuknya, atau tercampurnya, unsur – unsur berbahaya ke dalam *atmosfir* yang dapat mengakibatkan terjadinya kerusakan lingkungan, gangguan pada kesehatan manusia serta secara umum menurunkan kualitas lingkungan. Pencemaran udara dapat terjadi dimana - mana, misalnya, didalam rumah, sekolah, kantor atau yang sering disebut sebagai pencemaran deposisi asam atau salju yang mengakibatkan sifat korosif pada bangunan, tanaman, hutan, di samping itu juga membuat sungai dan danau menjadi suatu lingkungan yang berbahaya bagi ikan – ikan karena nilai pH yang rendah.[23]

2.2.7 Solar Cell

Panel surya adalah alat yang dapat mengubah sinar matahari menjadi listrik, panel surya terbuat dari semikonduktor dengan bahan silikon dan dilapisi dengan bahan khusus. Panel surya menangkap sinar matahari. Setelah panel surya menerima sinar matahari, elektron dilepaskan dari atom silikondan mengalir, membentuk rangkaian listrik untuk menghasilkan listrik. Fungsi solar panel atau (PV) adalah untuk mengubah intensitas sinar matahari menjadi energi listrik. Semakin tinggi intensitas matahari, semakin banyak energi yang dapat dikonversi. Agar kebutuhan listrik motor pompa air dapat dipenuhi oleh aki, diperlukan ukuran yang cukup untuk mengimplementasikannya. Daya PV yang digunakan untuk mengubah sinar matahari menjadi listrik adalah 50 WP. Dalam sehari, daya maksimal bisa mencapai 5 jam.[24]

2.2.8 *Internet of Things*

Internet of Things atau sering disebut IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. misalnya CCTV yang terpasang di sepanjang jalan dihubungkan dengan koneksi internet dan disatukan di ruang kontrol yang jaraknya mungkin puluhan kilometer. atau sebuah rumah cerdas yang dapat dimanage lewat smartphone dengan bantuan koneksi internet. pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan server sebagai pengumpul informasi yang diterima sensor dan untuk analisa.

Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami Internet of Things sebut saja Intel, Microsoft, Oracle, dan banyak lainnya. Banyak yang memprediksi bahwa pengaruh *Internet of Things* adalah “ *the next big thing* ” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa digali. Contoh sederhana manfaat dan implementasi dari *Internet of Things* misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus distok lagi.[25]

2.3 Pemodelan

2.3.1 *Flowchart*

Flowchart merupakan gambaran berbentuk suatu grafik yang disertai langkah – langkah dan urutan suatu prosedur dari suatu program. *Flowchart* dapat membantu proses analisis, perancangan dan pengkodean untuk memecahkan masalah kedalam bagian – bagian yang lebih kecil untuk pengoperasiannya. *Flowchart* biasanya mempermudah penyelesaian suatu masalah pada evaluasi lebih lanjut. Pengetian lain *Flowchart* dapat dikatakan sebuah diagram dengan simbol – simbol grafis yang menyatakan aliran proses yang menampilkan beberapa langkah – langkah yang disimbolkan atau dapat diartikan sebagai penggambaran secara grafik dari langkah – langkah atau urutan – urutan dari suatu prosedur program yang mempunyai fungsi tertentu. Fungsi *Flowchart* digunakan untuk memberikan

gambaran suatu proses produksi agar mudah dipahami dan dilihat berdasarkan urutan langkah – langkahnya dari proses yang satu ke proses yang lainnya. Selanjutnya memberikan kesederhanaan pada rangkaian proses untuk memudahkan pemahaman pengguna terhadap informasi yang dibutuhkan.

Jenis – jenis *Flowchart*

a. *Flowchart* Sistem

Flowchart sistem merupakan diagram yang menggambarkan alur kerja atau kegiatan yang sedang di kerjakan oleh suatu sistem secara keseluruhan. *Flowchart* sistem dapat terdiri atas data yang mengalir pada sistem dan terdapat proses transformasi data. *Flowchart* sistem tersebut dapat digambarkan secara *online* atau *offline*

b. *Flowchart* Program

Flowchart program dihasilkan dari *Flowchart* Sistem. *Flowchart* Program merupakan keterangan yang terinci tentang bagaimana setiap tahapan program atau prosedur sesungguhnya akan atau telah dilaksanakan dan sekaligus menunjukkan setiap tahapan kegiatan pada urutan yang tepat. Seseorang programmer menggunakan *Flowchart* program untuk mendeskripsikan urutan intruksi dari program komputer. Sementara seorang analis sistem menggunakan *Flowchart* program untuk menggambarkan urutan tugas – tugas suatu job pada suatu prosedur.

c. *Flowchart* Skematik

Flowchart Skematik hampir sesuai dengan *Flowchart* sitem yang mendeskrisikan suatu sistem. *Flowchart* Skematic ini tidak sekedar menggunakan simbol – simbol *Flowchart* yang standart, melainkan menggunakan gambar – gambar komputer, perlengkapan lain yang digunakan pada sistem. *Flowchart* Skematic ditujukan untuk digunakan sebagai komunikasi antara seorang analis sistem dengan seseorang yang tidak familiar dengan *Flowchart* konvensional.

d. *Flowchart* Dokumen

Flowchart dokumen menelusuri alur dari data yang ditulis dari sistem. *Flowchart* dokumen mempunyai kegunaan utamanya adalah menelusuri alur

sistem dari suatu bagian ke bagian lainnya yaitu bagaimana alur diproses, dicatat dan disimpan.

e. *Flowchart* Proses

Flowchart Proses merupakan suatu teknik deskripsi rekayasa yang memecahkan masalah dengan langkah – langkah sesuai pada suatu prosedur atau sistem *Flowchart* Proses digunakan oleh perekayasa untuk mempelajari dan mengembangkan proses – proses pembentukan deskripsi. Pada analisis sistem, *Flowchart* proses sangat efektif untuk digunakan menelusuri alur.[26]

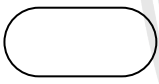
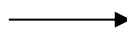
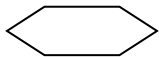

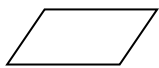
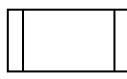
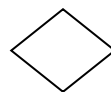
Tabel 2. 1 *Flowchart* Proses

Simbol	Nama	Keterangan
	<i>Input / Output</i>	Mempresentasikan input atau output data yang di proses atau informasi
	Proses	Mempresentasikan operasi
	Penghubung	Keluar atau masuk dari bagian lain dari simbol <i>flowchart</i> khususnya halaman yang sama
	Terminal Point	Menunjukkan awal atau akhir sebuah proses
	Dokumen	I/O dalam format yang di cetak
	<i>Data Store</i>	Tempat menyimpan data
	Anak Panah	Menunjukkan arah aliran proses dari algoritma yang berjalan
	Keputusan	Proses / langkah dimana perlu adanya kondisi tertentu



2.3.2 Diagram Blok

Sebuah sistem pengendalian terdiri dari beberapa komponen hubungan antara komponen ini di nyatakan dalam bentuk blok diagram. Diagram blok sistem merupakan resperensi dari fungsi dari komponen didalam sistem pengendalian dan hubungannya antara satu komponen dengan komponen yang lain. Suatu penyajian bergambar dari fungsi yang dilakukan oleh tiap komponen dan aliran sinyalnya. Dalam suatu diagram blok, semua variable system saling dihubungkan dengan menggunakan blok fungsional. Diagram blok mengandung informasi perilaku dinamik tetapi tidak mengandung informasi mengenai konstruksi fisik dari sistem. Oleh karena itu, beberapa system yang berbeda dan tidak mempunyai relasi satu sama lain dapat dinyatakan dalam diagram blok yang sama. Diagram blok suatu system adalah tidak unik. Suatu system dapat digambarkan dengan diagram blok yang berbeda bergantung pada titik pandang analisis.[27]

Tabel 2. 2 Diagram Blok

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan / akhir program
	GARIS ALIR	Arah aliran program
	<i>PREPARATION</i>	Proses inialisasi / pemberian harga awal
	PROSES	Proses perhitungan / proses pengolahan data
	<i>INPUT / OUTPUT</i>	Proses input / output data, parameter, informasi
	SUB PROGRAM	Permulaan sub program / proses menjalankan sub program
	<i>DECISION</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

Tabel Lanjutan 2.2 Diagram Blok

	<i>ON PAGE CONNECTOR</i>	Penghubung bagian – bagian <i>flowchart</i> yang berada pada satu halaman
	<i>OFF PAGE CONNECTOR</i>	Penghubung bagian – bagian <i>flowchart</i> yang berada pada halaman berbeda

2.3.3 Bahasa C++

C++ adalah bahasa pemrograman komputer yang dibuat oleh Bjarne Stroustrup yang merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standard Institute). Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming). [28]

2.4 Perangkat Lunak yang Digunakan

2.4.1 Arduino Uno



Gambar 2. 1 Logo Arduino Uno

Untuk memulai program Arduino (untuk membuatnya melakukan apa yang kita inginkan) kita menggunakan IDE Arduino (*Integrated Development Environment*), IDE Arduino adalah bagian *Software opensource* yang memungkinkan kita untuk memprogram bahasa Arduino dalam bahasa C. IDE memungkinkan kita untuk menulis sebuah program secara *step by step* kemudian intruksi tersebut di upload ke papan Arduino.[29]

2.4.2 Fritzing



Gambar 2. 2 Logo Fritzing

Fritzing adalah suatu *software* atau perangkat lunak gratis yang digunakan oleh desainer, seniman dan para penghobi elektronika untuk perancangan berbagai peralatan elektronika. Antarmuka *Fritzing* dibuat seinteraktif dan semudah mungkin agar bisa digunakan oleh orang yang minim pengetahuannya tentang simbol dari perangkat elektronika. Didalam *fritzing* sudah terdapat skema siap pakai dari berbagai mikrokontroler arduino serta shieldnya. *Software* ini memang khusus dirancang untuk perancangan dan pendokumentasian tentang produk kreatif yang menggunakan mikrokontroler arduino.[30]

2.4.3 Draw.io



Gambar 2. 3 Logo Draw io

Draw io merupakan sebuah situs yang di desain khusus untuk menggambarkan diagram secara *online*. Untuk mengaksesnya hanya diperlukan *browser* yang mendukung HTML5 dan juga koneksi internet. *Draw.io* sudah terintegrasi dengan *google drive* untuk menyimpan *file* selain mengekspor dalam bentuk JPG/PNG/SVG/XML.[31]

2.4.4 Blynk



Gambar 2. 4 Logo Aplikasi Blynk

Blynk adalah salah satu *platform* IoT yang terkenal karena kemudahahn dalam konfigurasinya. *Platform* ini menawarkan antarmuka intuitif yang memungkinkan pengguna untuk menghubungkan dan mengontrol perangkat IoT dengan cepat melalui aplikasi *mobile* atau *web*. *Blynk* adalah sebuah layanan aplikasi yang digunakan untuk mengontrol mikrokontroler dari jaringan internet.

Dengan dukungan beragam *widget*, *blynk* memudahkan pengguna dalam membuat visualisasi data, mengirim perintah ke perangkat, serta memantau status perangkat IoT dari jauh tanpa memerlukan pemrograman kompleks. Aplikasi *Blynk* memiliki 3 komponen utama yaitu Aplikasi, Server dan *Libraries*. *Blynk server* berfungsi untuk menangani semua komunikasi diantara *smartphone* dan *hardware*. *Blynk* tersedia untuk di unduh secara gratis di

perangkat IOS dan Android. Aplikasi ini memungkinkan pengguna untuk dengan mudah mengembangkan dan mengontrol proyek – proyek IoT langsung dari perangkat seluler mereka. Versi gratisnya menawarkan berbagai fitur dasar yang cukup untuk proyek kecil dan pengujian, sementara versi berbayarnya menyediakan lebih banyak fitur dan kapasitas untuk proyek yang lebih kompleks.[32]



BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Gambaran Umum Objek Penelitian

3.1.1 Objek Penelitian

Kualitas udara dalam ruangan memiliki peran penting dalam menjaga kesehatan dan kenyamanan penghuni suatu bangunan. Udara dalam ruangan yang buruk dapat menyebabkan berbagai masalah kesehatan, seperti gangguan pernapasan, alergi, iritasi mata, hingga penurunan produktivitas kerja. Sumber utama polusi udara dalam ruangan berasal dari asap rokok, gas buangan dari peralatan rumah tangga, debu, jamur, serta bahan kimia dari cat atau furnitur. Oleh karena itu, diperlukan sistem pemantauan yang mampu mengukur kualitas udara secara real-time dan memberikan respon otomatis jika udara dalam ruangan berada dalam kondisi buruk.

Dalam penelitian ini, akan dikembangkan sistem monitoring kualitas udara dalam ruangan berbasis Internet of Things (IoT) yang berfungsi untuk mengukur tingkat polusi udara serta menangani kondisi udara kotor secara otomatis. Sistem ini akan menggunakan sensor untuk mendeteksi berbagai parameter kualitas udara, termasuk kadar karbon monoksida (CO), karbon dioksida (CO₂), total volatile organic compounds (TVOC), kadar oksigen (O₂), kadar partikel debu (PM_{2.5}, PM₁₀), suhu, dan kelembapan. Sistem ini juga dilengkapi dengan mekanisme penanganan udara kotor, seperti aktivasi otomatis kipas ventilasi, air purifier, atau sistem penyaring udara berbasis HEPA dan karbon aktif. Jika kadar polusi udara dalam ruangan melebihi batas aman, sistem akan secara otomatis mengaktifkan perangkat tersebut untuk meningkatkan kualitas udara. Selain itu sistem ini juga memanfaatkan tenaga surya yang bertujuan untuk meningkatkan efisiensi energi, mengurangi ketergantungan pada listrik konvensional dan dapat menciptakan solusi ramah lingkungan.

Metode penelitian yang digunakan mencakup perancangan sistem perangkat keras dan perangkat lunak, pengujian sensor dalam berbagai kondisi ruangan, analisis data hasil pemantauan, serta evaluasi efektivitas sistem penanganan udara kotor dengan bantuan tenaga surya. Pengujian akan dilakukan

di dalam satu ruangan, guna memperoleh data kualitas udara dalam berbagai skenario.

Hasil dari penelitian ini diharapkan dapat memberikan solusi inovatif dalam menjaga kualitas udara dalam ruangan, sehingga dapat meningkatkan kesehatan dan kenyamanan penghuni. Selain itu, sistem ini juga dapat digunakan untuk berbagai kebutuhan, seperti pemantauan kualitas udara di rumah, kantor, sekolah, hingga fasilitas umum lainnya. Dengan penerapan teknologi IoT, sistem ini dapat diintegrasikan dengan smart home dan memberikan manfaat yang lebih luas dalam menciptakan lingkungan dalam ruangan yang lebih sehat dan nyaman.

3.1.2 Keadaan sistem yang bejalan

Analisis system yang berjalan meruapakan gambaran tentang cara kerja monitoring kualitas udara serta penanganan udara kotor berbasis IoT. Berdasarkan hasil wawancara dengan beberapa penghuni di perumahan Dawuhan *Cluster* diketahui bahwa sebelum adanya sistem monitoring dan penanganan udara kotor berbasis IoT, kualitas udara dalam ruangan di Perumahan Dawuhan cenderung tidak terpantau secara *real-time* dan tidak memiliki mekanisme otomatis untuk menangani udara kotor. Berikut adalah gambaran kondisi sistem yang berjalan sebelum inovasi ini diterapkan.

- a. Kualitas udara tidak terkontrol secara *real-time*
 1. Penghuni khususnya di perumahan Dawuhan *Cluster* tidak memiliki alat untuk memantau kadar polusi udara dalam ruangan.
 2. Polusi dari asap dapur, debu, asap rokok, dan bahan kimia (TVOC) sering tidak terdeteksi.
 3. Peningkatan CO₂ akibat kurangnya ventilasi tidak terpantau, yang dapat menyebabkan kantuk dan gangguan kesehatan.
 4. Indeks kualitas udara dalam ruangan tidak diketahui secara akurat.
- b. Ventilasi dan Penyaringan Udara Masih Manual
 1. Sirkulasi udara hanya mengendalikan jendela dan kipas angin tanpa sistem otomatis
 2. Ventilasi sering tertutup dalam kondisi panas atau hujan, menyebabkan penumpukan CO₂ dan polutan.
- c. Pemakaian Energi Tidak Efisien dan Ketergantungan pada Listrik PLN

1. Semua alat elektronik yang digunakan bergantung sepenuhnya pada listrik PLN
 2. Tidak ada sumber energi cadangan jika terjadi pemadaman listrik.
- d. Tidak Ada Sistem Otomatis Untuk Mencegah Udara Buruk
1. Tidak ada sensor atau sistem otomatis yang dapat mendeteksi kenaikan polutan udara di dalam ruangan.
 2. Jika terjadi kebocoran gas atau peningkatan CO₂ penghuni baru menyadari dalam ruangan mencapai tingkat bahaya.
 3. Tidak ada alarm atau peringatan yang memberi tahu penghuni saat udara dalam ruangan mencapai tingkat bahaya

3.1.3 Kelebihan sistem

Dari Hal diatas Adapun kelebihan sistem yang sedang berjalan saat ini yaitu sebagai berikut :

- a. Sederhana dan Mudah Dijalankan
 1. Tidak Memerlukan Teknologi Canggih
Sistem yang berjalan sebelumnya hanya mengandalkan ventilasi alami, kipas angin, atau AC, sehingga tidak memerlukan perangkat tambahan seperti sensor atau kontrol otomatis. Serta tidak ada kebutuhan untuk pemrograman atau pemeliharaan sistem berbasis IoT.
 2. Minim Biaya Awal
Tidak perlu investasi dalam perangkat IoT seperti sensor kualitas udara, modul komunikasi, atau sistem penyaring udara otomatis. Penghuni hanya perlu membuka jendela atau menggunakan kipas untuk meningkatkan kualitas udara.
- b. Tidak Bergantung Pada Jaringan Internet
 1. Tidak Perlu Koneksi Internet
Sistem yang ada sebelumnya tidak membutuhkan jaringan WiFi atau IoT untuk bekerja. Jika terjadi gangguan internet, tidak ada dampak signifikan terhadap ventilasi dan penyaringan udara.
 2. Tidak Terpengaruh Oleh Gangguan Teknis

Tidak adanya resiko error dari sistem otomatis atau sensor yang tidak bekerja dengan baik. Dan tidak ada kemungkinan kegagalan perangkat lunak yang bisa menyebabkan sistem berhenti beroperasi.

c. Pemakaian Energi Relatis Stabil

1. Tidak Membutuhkan Daya Tambahan Untuk Sensor Dan Iot

Pada saat ini sistem belum menggunakan perangkat Iot dan otomatisasi, konsumsi listrik hanya berasal dari perangkat standart seperti kipas, AC, Atau *air purifier* manual. Serta tidak ada kebutuhan untuk mengisi daya perangkat tambahan seperti sensor dan kontrol otomatis.

2. Lebih Stabil Dalam Pemadama Listrik

Sistem ventilasi alami (jendela dan pintu) tetap dapat berfungsi meskipun terjadi pemadaman listrik. Serta tidak ada ketergantungan pada sistem berbasis tenaga surya atau baterai tambahan.

d. Tidak Memerukan Pemeliharaan Rutin Untuk Perangkat Iot

1. Minim Perawatan Teknologi

Tidak perlu kalibrasi sensor udara secara berkala. Tidak ada resiko perangkat IoT mengalami *error*, kehabisan daya atau memerlukan *reset* sistem.

2. Tidak Perlu Penggantian Komponen Elektronik

Sensor dan modul Iot memiliki masa pakai tertentu dan memerlukan penggantian jika rusak, sedangkan sistem ventilasi manual tidak memiliki komponen elektronik yang perlu diganti.

Sebelum adanya sistem monitoring kualitas udara dan penanganan udara kotor berbasis IoT, kelebihanannya terletak pada kesederhanaan, biaya awal yang lebih rendah, tidak bergantung pada jaringan internet, pemakaian energi yang stabil, serta minimnya kebutuhan perawatan teknologi.

3.1.4 Kelemahan sistem

Meskipun sistem manual memiliki beberapa kelebihan, ada juga beberapa kekurangan dalam pengguna sistem sebelum adanya perangkat berbasis IoT adalah memiliki kelamaan sebagai berikut ini :

- a. Tidak ada pemantauan kualitas udara secara *real-time*

Tidak ada alat untuk mengukur tingkat polusi udara, kadar CO₂, debu, asap rokok, TVOC (senyawa organik volatil), dan kelembapan udara dalam ruangan. Penghuni tidak dapat mengetahui kapan udara dalam ruangan mulai memburuk. Tidak ada sistem peringatan dini untuk indikasi udara berbahaya, sehingga penghuni hanya menyadari kualitas udara buruk setelah mengalami efek seperti sesak nafas, pusing atau alergi.

b. Sirkulasi udara tidak optimal dan manual

Ventilasi udara hanya mengandalkan jendela dan kipas angin, tanpa sistem otomatis yang menyesuaikan dengan kondisi udara. Jika udara luar juga tercemar, membuka jendela justru dapat memperburuk kualitas udara dalam ruangan, sehingga udara kotor bisa bertahan lebih lama di dalam ruangan karena tidak ada mekanisme penyaringan otomatis.

c. Tidak ada sistem otomatis untuk menangani udara kotor

Jika ada peningkatan polutan (seperti debu, asap, atau gas beracun), penghuni harus menanganinya secara manual. *Exhaust fan* hanya dinyalakan berdasarkan data kualitas udara. Akibatnya udara kotor tidak ditangani secara efektif, sehingga kualitas udara tetap buruk dalam waktu yang lama.

d. Boros energi dan tidak ramah lingkungan

Peralatan elektronik seperti AC, kipas angin bekerja tanpa kendali otomatis, sehingga sering menyala lebih lama dari yang diperlukan. Tidak ada pemanfaatan tenaga surya untuk menghemat listrik. Dampak hal tersebut menyebabkan tagihan listrik PLN membuat sistem berhenti berfungsi saat pemadaman listrik.

e. Tidak ada notifikasi atau peringatan saat udara mencapai level berbahaya

Tidak ada alarm atau notifikasi untuk memberi tahu penghuni jika udara telah mencapai tingkat berbahaya. Berdampak ketidaknya tindakan tepat untuk mengatasi udara kotor sebelum berdampak buruk pada kesehatan penghuni.

3.2 Alur Proses

Alur proses merupakan sebuah gambaran yang menjelaskan suatu proses yang berjalan pada suatu sistem. Dengan adanya alur proses maka akan lebih mudah dalam memahami dan memaparkan jalannya proses-proses yang ada pada objek penelitian atau sistem yang dibuat.

Pada bagian ini akan dijelaskan identifikasi proses dan analisa proses aplikasi yang akan diusulkan. Disini akan dijelaskan terlebih dahulu identifikasi proses dan dilanjutkan dengan analisis proses bisnis. Hal ini bertujuan untuk memudahkan dan meng-efisienkan dalam hal merinci dan menjelaskan proses-proses yang terjadi dalam memonitoring kualitas udara dan penanganan udara kotor berbasis IoT khusus nya didalam ruangan di Perumahan Dawuhan *Cluster* Situbondo

3.2.1 Identifikasi Dan Analisis Proses Bisnis

Dalam upaya untuk mengetahui hal yang penting sebagai data yang akan digunakan dalam proses pembuatan sistem adalah dengan cara wawancara dan meninjau langsung tempat penelitian, tempat yang akan kami teliti yaitu Perumahan Dawuhan *Cluster*. Berikut adalah identifikasi dan analisis proses bisnis terkait:

a. Identifikasi Proses Bisnis

Mengidentifikasi proses kegiatan bisnis yang terkait dengan monitoring kualitas udara dan penanganan udara kotor berbasis IoT yang akan berjalan adalah :

1. Monitoring kaulitas udara secara *Real-Time*.
2. Penanganan udara kotor berbasis IoT.
3. Penggunaan energi tenaga surya sebagai pengganti listrik konvensional.

b. Analisis Proses Bisnis

Analisis sistem atau analisis proses bisnis adalah penguraian dari suatu sistem aplikasi yang utuh ke dalam bagian bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga di usulkan perbaikan perbaikan, analisis proses bisnis dalam ini adalah sebagai berikut:

1. Monitoring Kualitas Udara Secara *Real-Time*

Sistem ini untuk memperoleh informasi akurat dan terkini mengenai kondisi kualitas udara dalam ruangan, sehingga dapat dilakukan tindakan penanganan secepat mungkin jika terjadi pencemaran udara. Sensor mengukur kualitas udara setiap interval waktu tertentu, setelahnya mikrokontroler membaca nilai sensor dan menentukan apakah nilai tersebut normal atau mencemaskan.

Data dikirim secara *real-time* ke aplikasi atau dashboard berbasis *cloud* melalui koneksi Wi-Fi. Data ditampilkan dalam bentuk grafik, table, atau indicator status pada antarmuka pengguna. Setelah semua prose selesai terakhir adalah sistem akan mengirimkan peringatan otomatis jika nilai kualitas udara melawati ambang batas. Dari hasil proses diatas data dapat dimonitor secara *real-time* tanpa jeda yang signifikan, akurasi dan konsistensi dari pembacaan sensor dapat dikirim dengan cepat dan tepat waktu dan tidak ada kehilangan data selama transmisi.

2. Penanganan Udara Kotor Secara Otomatis

Tujuan dari menangani dan memperbaiki kualitas udara di dalam ruangan secara otomatis melalui sistem yang terintegrasi dengan perangkat IoT, untuk memastikan udara tetap bersih dan aman bagi kesehatan. Sensor akan mendeteksi bahwa kualitas udara berada di bawah standart aman. Data dibandingkan dengan ambang batas, jika melebihi sistem akan merespon. Mikrokontroler akan mengaktifkan alat seperti kipas, air *purifier*, air *humidifier*, atau *exhaust fan*. Jika kualitas udara kembali normal, perangkat secara otomatis dimatikan untuk efisiensi energi. Waktu respons yang cepat terhadap perubahan kualitas udara secara otomatis tanpa interval manual. Kualitas udara meningkat setelah tindakan dilakukan serta penghematan energi melalui sistem penonaktifan otomatis saat kondisi normal.

3. Penggunaan energi Tenaga Surya Sebagai Pengganti Listrik Konvensional

Menyediakan sumber energi alternatif yang ramah lingkungan, berkelanjutan, dan hemat biaya untuk menjalankan sistem monitoring dan penanganan kualitas udara secara mandiri tanpa bergantung penuh pada listrik konvensional. Panel surya menangkap sinar matahari dan mengubahnya menjadi energi listrik (DC), lalu energi listrik disalurkan ke *solar charge controller* untuk mengatur voltase dan arus, lalu disimpan di dalam baterai. Energi dari baterai disalurkan ke perangkat IoT, sistem memantau status pengisian baterai dan konsumsi daya untuk efisiensi. Jika energi energi surya tidak mencukupi (misalnya cuaca mendung), sistem bisa otomatis beralih ke listrik konvensional sebagai cadangan. Sistem dapat berjalan tanpa listrik

PLN selama 6 - 12 jam per hari dikarenakan energidari matahari cukup untuk menyuplai dan menyimpan daya, sehingga baterai terisi penuh secara rutin di siang hari.

3.2.2 Identifikasi dan Analisis Kebutuhan

Setelah identifikasi dan analisis proses bisnis, langkah selanjutnya adalah identifikasi dan analisis kebutuhan – kebutuhan pada objek penelitian terkait dengan Monitoring Kualitas Udara dan Penanganan Udara Kotor Berbasis IoT.

a. Identifikasi Kebutuhan Fungsional

Setelah proses – proses bisnis telah teridentifikasi maka selanjutnya adalah melakukan identifikasi kebutuhan fungsional yang berkaitan dengan sistem yang akan dibuat yaitu sebagai berikut :

1. Monitoring Kualitas Udara Secara *real – time*
 - a. Sistem harus dapat membaca data dari sensor kualitas udara seperti PM2.5, CO₂, Suhu dan Kelembapan secara berkala
 - b. Sistem harus dapat mengirimkan data sensor untuk menentukan kondisi kualitas udara (baik, sedang, buruk).
 - c. Sistem harus dapat mengirimkan data ke *platform cloud* secara *real-time*.
 - d. Sistem harus dapat mengirimkan notifikasi kepada pengguna jika kualitas udara berada pada level tidak sehat.
 - e. Sistem harus mampu memberikan perintah kepada perangkat penanganan udara (seperti kipas, humidify atau purifier) jika kondisi udara buruk
2. Penanganan Udara Kotor Secara Otomatis
 - a. Sistem harus dapat mendeteksi secara otomatis jika kualitas udara berada dibawah ambang batas yang telah ditentukan.
 - b. Sistem harus mampu membandingkan nilai sensor dengan ambang batas kualitas udara untuk menentukan kondisi tidak sehat.
 - c. Sistem harus secara otomatis mengaktifkan perangkat seperti kipas, *exhaust fan*, atau *air purifier* saat kualitas udara buruk.
 - d. Sistem harus terintegrasi penuh dengan proses monitoring *real – time* agar dapat mengambil keputusan secara otomatis berdasarkan data terbaru.
 - e. Sistem harus menonaktifkan perangkat penanganan secara otomatis saat kualitas udara kembali normal.

3. Penggunaan energi tenaga surya sebagai pengganti listrik konvensional
 - a. Sistem harus dapat menangkap energi matahari menggunakan panel surya
 - b. Sistem harus mengubah energi cahaya matahari menjadi listrik (DC) melalui panel surya.
 - c. Sistem harus dapat menyimpan energi listrik yang dihasilkan oleh panel surya ke dalam baterai untuk digunakan kapan saja.
 - d. Sistem harus dapat beralih secara otomatis ke listrik konvensional jika daya dari baterai surya tidak mencukupi
 - e. Sistem harus mampu menyalurkan listrik dari baterai ke komponen – komponen sistem.

Kebutuhan fungsional ini akan membantu dalam merancang monitoring kualitas udara dan penanganan udara kotor berbasis IoT yang efisien, transparan, dan sesuai dengan kebutuhan.

b. Analisis Non Fungsional

Analisis non-fungsional mencakup karakteristik sistem yang tidak berkaitan langsung dengan fungsi utama sistem, tetapi berpengaruh terhadap kualitas, efisiensi, dan keberlangsungan operasional. Berikut ini adalah aspek-aspek non-fungsional yang dianalisis pada sistem Monitoring Kualitas Udara dan Penanganan Udara Kotor Berbasis IoT. Berikut adalah beberapa kebutuhan non-fungsional yang perlu dipertimbangkan :

1. Kinerja

Sistem mampu membaca data kualitas udara secara *real-time* dan mengirimkan data ke server dalam interval yang cepat. Respon dari penanganan udara kotor terhadap nilai ambang batas juga berjalan secara otomatis dan cepat agar memastikan kualitas udara tetap dalam batas aman, serta asupan energi otomatis dari tenaga surya jika tenaga surya tidak memadai akan berpindah otomatis ke listrik konvensional.

2. Keamanan Sistem

Sistem dirancang untuk menjaga keamanan data yang dikirimkan dari sensor ke server IoT. Penggunaan protokol komunikasi yang aman dan autentikasi perangkat diperlukan untuk mencegah akses tidak sah dan manipulasi data.

3. Keandalan

Sistem harus bekerja secara stabil dan mampu beroperasi dalam jangka waktu yang panjang tanpa gangguan. Perangkat keras yang digunakan telah dipilih berdasarkan daya tahan dan performa yang telah teruji di lingkungan serupa.

4. Ketersediaan

Sistem didesain untuk tetap aktif secara *real-time* dengan dukungan sistem tenaga surya dan baterai cadangan. Hal ini memastikan sistem tetap berfungsi bahkan saat terjadi pemadaman listrik.

5. Rencana Pemulihan

Dalam hal terjadi gangguan sistem atau kerusakan perangkat, sistem dilengkapi dengan prosedur pemulihan, seperti *restart otomatis* pada NodeMCU saat terjadi hang, dan pencatatan *log* kesalahan. File konfigurasi disimpan di memori internal agar bisa dimuat ulang tanpa kehilangan data penting.

6. Ketersediaan Suku Cadang

Perangkat keras yang digunakan seperti sensor - sensor modul NodeMCU ESP32, serta fan dan lainnya memiliki ketersediaan yang luas di pasaran, baik secara *online* maupun *offline*, sehingga memudahkan penggantian atau perbaikan saat terjadi kerusakan

7. Skalabilitas

Sistem dirancang modular sehingga dapat ditambahkan sensor tambahan seperti sensor suhu, kelembapan, atau partikel PM2.5 tanpa mengubah struktur utama. Sistem ini juga dapat diterapkan di berbagai ruangan dengan ukuran dan kebutuhan berbeda .

8. Kemudahan Pemeliharaan

Perancangan perangkat keras dilakukan dengan mempertimbangkan kemudahan penggantian komponen. Selain itu, kode program disusun secara terstruktur dan terdokumentasi, sehingga mudah dalam proses debugging atau pengembangan lebih lanjut.

9. Efisiensi Energi

Penggunaan panel surya menjadikan sistem ini hemat energi dan ramah lingkungan. Energi disimpan dalam baterai dan digunakan untuk menghidupkan semua komponen sistem, menjadikan sistem ini efisien bahkan di lokasi tanpa listrik PLN.

10. Dukungan *Multy-Platform*

Dashboard IoT yang digunakan dapat diakses melalui berbagai perangkat seperti smartphone, tablet, maupun laptop melalui browser web (*Blynk* dan *thingspeaks*). Hal ini memudahkan pengguna dalam memantau kualitas udara dari mana saja dan kapan saja.

11. *Usability*

Antarmuka pengguna dibuat intuitif dan mudah dipahami, dengan menampilkan indikator visual seperti grafik, warna status kualitas udara, serta notifikasi peringatan. Hal ini membantu pengguna dalam pengambilan keputusan secara cepat.

Kebutuhan non-fungsional ini harus diperhatikan dengan cermat dalam perancangan sistem untuk memastikan bahwa sistem dapat beroperasi dengan efektif, dan aman dalam memonitoring serta penanganan udara kotor.

3.2.3 Identifikasi dan Analisis Alternatif Solusi

Pada bagian identifikasi dan analisis alternatif solusi ini dijelaskan bagian sistem yang akan dikomputerisasikan, keuntungan, sistem operasi, *webserver*, *browser* dan alat penyimpanan data.

a. Identifikasi alternatif solusi

Pada bagian ini menjelaskan identifikasi dan analisis alternatif solusi yaitu menjelaskan tentang kebutuhan sistem operasi, bahasa program serta aplikasi yang di gunakan.

Tabel 3. 1 Identifikasi alternatif solusi

Karakteristik	Alternatif 1	Alternatif 1
Mikrokontroler	NodeMCU ESP32	NodeMCU ESP8266
Sensor Kualitas Udara	MQ135, MQ7, DHT11	PM2.5, DHT22
Akuator penanganan udara kotor	Kipas DC 12v	Air Purifier mini otomatis
Sumber Daya	Panel Surya + Baterai Litium	<i>Power Suply</i> + PLN
<i>Dashboard</i> IoT	<i>Blynk</i>	<i>Thingspeak</i>

b. Analisis kelayakan alternatif solusi

Pada bagian ini menjelaskan kelayakan alternatif solusi, yaitu berfungsi untuk menganalisa sistem informasi yang akan dibuat dari dokumen-dokumen yang telah di peroleh dari tempat penelitian.

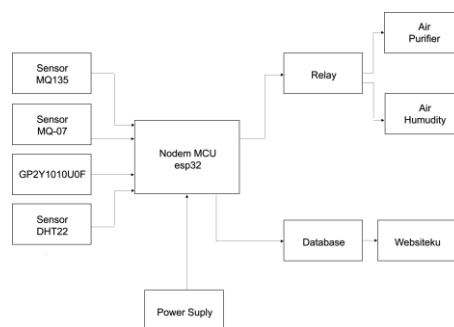
Tabel 3. 2 Analisis Kelayakan Alternatif Solusi

Kriteria Kelayakan	Bobot	Alternatif
Kelayakan operasional fungsionalitas	30%	Mendukung seluruh kebutuhan fungsional
Kelayakan teknis teknologi keahlian	30%	Keahlian personal memadai untuk dapat mengoperasikan dan memelihara sistem yang sudah ada
Kelayakan ekonomis biasa pengembangan	20%	Untuk sistem ini biaya pengembangan hanya dibutuhkan untuk pembiayaan sensor ketika rusak
Kelayakan Perawatan program	20%	Ketika program memiliki permasalahan

3.3 Desain Sistem

3.3.1 Diagram Blok Sistem

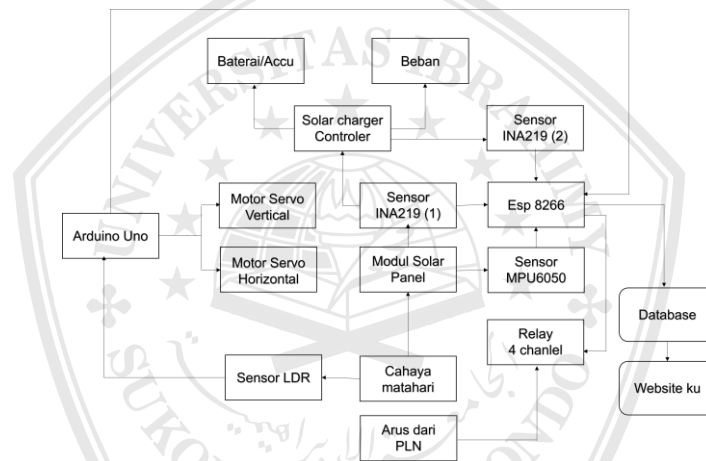
a. Diagram Blok Sistem Pada Pemantauan Kualitas Udara dan Penanganan Udara Kotor



Gambar 3. 1 Diagram Blok Sistem Pada Pemantauan Kualitas Udara dan Penanganan Udara Kotor

Sistem pemantauan kualitas udara beserta dengan penanganannya, menunjukkan proses sistem mulai dari input, proses hingga sampai pada output dari sistem tersebut. Sistem ini bekerja dengan membaca data kualitas udara (gas, debu, suhu, kelembapan), lalu diproses ke ESP32. ESP32 memproses data dan menentukan apakah air purifier atau humidifier perlu dikatiffkan. Jika kondisi udara tidak baik, relay akan mengaktifkan air *purifier* atau *humidifier* secara otomatis. Data sensor dikirim ke database dan dapat diakses melalui website. Pengguna dapat melihat status kualitas udara dan mengontrol perangkat melalui website tersebut.

b. Diagram Blok Sistem Pada *Solar Tracking*



Gambar 3. 2 Diagram Blok Sistem Pada *Solar Tracking*

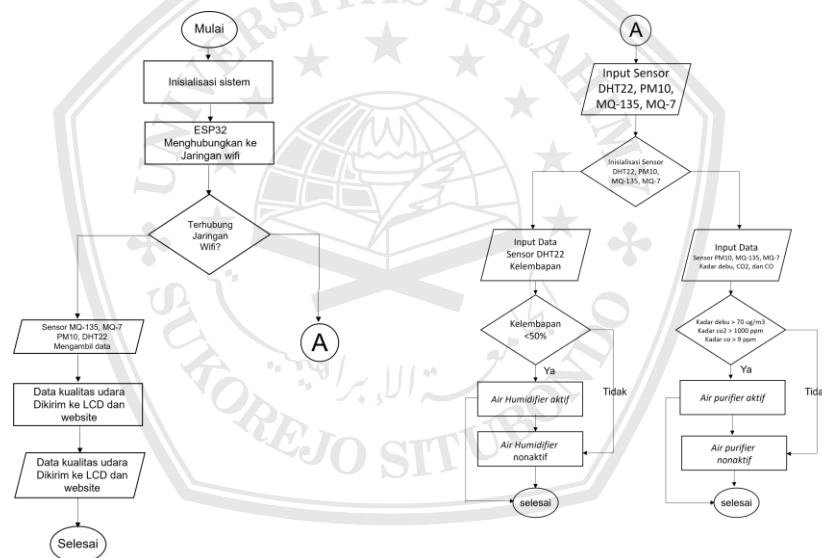
Pada **Gambar 3.2** perancangan blok diagram tersebut yaitu menunjukkan cara kerja sistem tersebut yaitu : Sensor LDR mendeteksi posisi matahari, kemudian Arduino menggerakkan motor servo agar panel surya selalu menghadap ke arah matahari. Panel surya menghasilkan energi listrik yang disalurkan ke *Solar Charge Controller*. *Solar Charge Controller* mengisi baterai 12V dan menyediakan daya ke sistem. Arduino memantau tegangan dan arus melalui sensor ACS712 untuk memastikan efisiensi daya. Jika daya dari baterai tidak mencukupi, sistem akan otomatis beralih ke PLN melalui modul relay. Semua data sistem (tegangan, arus, suhu, status daya) ditampilkan pada LCD 16x2 dan kemungkinan dikirim ke server melalui ESP8266 untuk monitoring jarak jauh. *Solar Tracking dual axis* ini

menggunakan mikrokontroler berbasis *Internet Of Things*. Sedangkan sistem ini merupakan *Hybrid Solar Tracking System*, yang memungkinkan panel surya bergerak mengikuti matahari untuk meningkatkan efisiensi energi. Sistem juga memiliki backup dari PLN, sehingga jika daya dari baterai tidak mencukupi, sistem tetap dapat beroperasi menggunakan listrik konvensional.

3.3.2 Diagram Alir Sistem

Berikut ini merupakan diagram alir (flowchart) yang menggambarkan bagaimana sistem smart housing bekerja dengan sistem solar tracking dan pemantauan kualitas udara dan penanganan udara kotor :

a. Flowchat Diagram Alir Pemantauan Kulaitas udara dan penanganan udara kotor



Gambar 3. 3 *Flowchart* Diagram Alir Pemantauan Kulaitas udara dan penanganan udara kotor

Pada **Gambar 3.3** Merupakan *flowchart* menunjukkan detail proses pada sistem pemantauan dan pembersih saluran air, berikut alur prosesnya :

1) *Start* (Mulai)

Proses sistem diawali dengan menyalakan perangkat dan memulai proses pemantauan.

2) Inisialisasi Sensor dan Komponen

Pada tahap ini, sistem melakukan :

- a. Inisialisasi sensor pada MQ135, MQ7, *Dush sensor*, dan DHT11
- b. Mengaktifkan koneksi ke Wi-Fi untuk ESP32 dan menyiapkan komunikasi dengan *platform* IoT yaitu *Blynk*.
- c. Mengaktifkan LCD, *Buzzer*, dan aktuator yaitu kipas, *air purifier* atau *otomizer*.

3) Baca Data Dari Sensor

Sistem membaca nilai dari berbagai sensor :

- a. MQ-135 untuk gas beracun (amonia, benzena, asap rokok).
- b. MQ-7 untuk karbon monoksida (CO)
- c. *Dush Sensor* untuk debu atau partikulat (PM) di udara
- d. DHT11 untuk sensor suhu dan kelembapan

4) Kirim Data ke *Platform Blynk*

Setelah pembacaan sensor dataa dikirim secara *real-time* ke platform *Blynk* melalui jaringan Wi-Fi

5) Tampilkan Data ke LCD

Nilai sensor akan langsung ditampilkan di layar LCD 16x2 atau LCD I2C agar pengguna bisa memantau langsung.

6) Apakah Nilai Melebihi Ambang Batas ?

Sistem akan membandingkan nilai dari sensor dengan ambang batas yang sudah ditentukan :

- a. Jika tidak melebihi : sistem akan terus memantau (loop ke pembacaan sensor).
- b. Jika melebihi batas: masuk ke proses penanganan.

7) Aktifkan Penanganan Udara Kotor

Jika terdeteksi udara tidak sehat, sistem akan mengaktifkan komponen penanganan :

- a. Aktifkan kipas / filter udara / atomizer.
- b. *Buzzer* menyala sebagai tanda peringatan.
- c. Notifikasi dikirim ke pengguna melalui aplikasi *Blynk* .

8) Apakah Kualitas Udara sudah normal ?

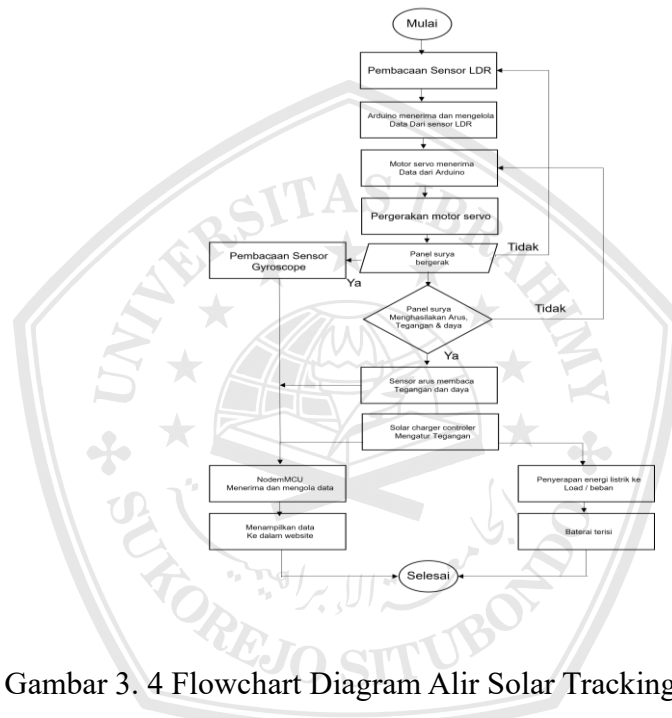
Sistem akan terus memantau setelah penanganan :

- a. Jika sudah normal, maka sistem kembali ke monitoring biasa.
- b. Jika belum normal, maka sistem melanjutkan proses penanganan.

9) Selesai

Jika pengguna mematikan alat atau perangkat berhenti, proses berakhir. Flowchart ini menggambarkan sistem pemantauan kualitas udara real-time, terintegrasi dengan IoT dan penanganan otomatis ketika udara terdeteksi kotor. Setiap langkah saling terhubung dalam satu siklus pemantauan → evaluasi → respons → pemulihan → pemantauan ulang.

b. Flowchart Diagram Alir *Solar Tracking*



Gambar 3. 4 Flowchart Diagram Alir Solar Tracking

Pada Gambar 3.4 Merupakan *flowchart* menunjukkan detail proses pada sistem pemantauan dan pembersih saluran air, berikut alur prosesnya :

- 1) *Start* (Mulai)
 - a. Proses sistem dimulai saat perangkat menyala.
 - b. Ini bisa terjadi secara otomatis ketika perangkat diberi daya (baik dari PLN maupun panel surya).
- 2) Apakah Listrik PLN Hidup ?
 - a. Sistem memeriksa apakah daya dari PLN tersedia.
 - b. Ini adalah logika utama untuk menentukan sumber daya utama yang akan digunakan.

- c. Jika listrik PLN hidup maka sistem akan menggunakan listrik PLN sebagai sumber daya utama. Fungsi dari *solar tracking* No.-aktif (standby) karena tidak diperlukan
 - d. Jika listrik PLN mati maka sistem akan secara otomatis beralih ke sumber listrik dari panel surya.
- 3) Aktifkan Sensor LDR dan Gyroscope
 - a. LDR mendeteksi intensitas cahaya dari berbagai arah
 - b. Gyroscope mendeteksi posisi dan kemiringan panel surya
 - 4) Bandingkan Nilai LDR
Menentukan arah datangnya cahaya paling terang.
 - 5) Bandingkan sudut panel Gyroscope
 - a. Gyroscope akan memastikan Apakah panel masih rata dan berada pada sudut optimal.
 - b. Mendeteksi jika ada kemiringan berlebih atau posisi salah misalnya akibat angin atau getaran.
 - 6) Gerakkan Motor Servo
Berdasarkan hasil perbandingan LDR dan Gyroscope ;
 - a. Motor servo akan memutar panel surya ke arah dengan cahaya paling kuat.
 - b. Hal ini bertujuan untuk mengoptimalkan penyerapan energi matahari.
 - 7) Panel Surya Mengisi Baterai
 - a. Panel surya akan mengisi baterai cadangan (yang digunakan adalah Li-ion) menggunakan energi matahari.
 - b. Energi menjadi cadangan yang digunakan untuk memberi daya pada sistem selama listrik PLN padam.

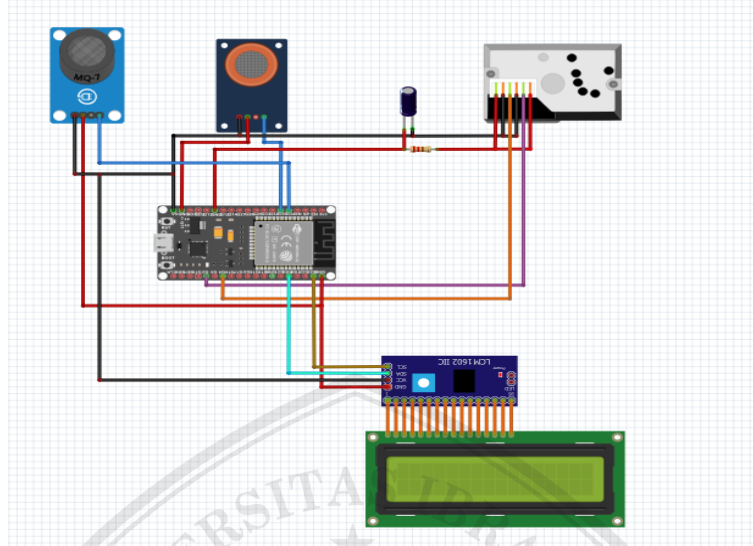
3.4 Ilustrasi Desain

3.4.1 Ilustrasi Perangkat Keras

Perancangan alat atau biasa disebut dengan perancangan perangkat keras adalah beberapa perancangan perangkat keras atau *hardware* yang terkait dengan sistem yang dibuat. Dalam perancangan ini akan dibuat menjadi empat bagian yang pertama adalah ilustrasi pemantaun kualitas udara secara *real-time*, yang ke dua adalah ilustrasi *air purifier* dan *air humidifer*, yang ketiga adalah ilustrasi

perancangan *solar tracking* dan yang ke empat adalah perancangan ilustrasi *solar system*.

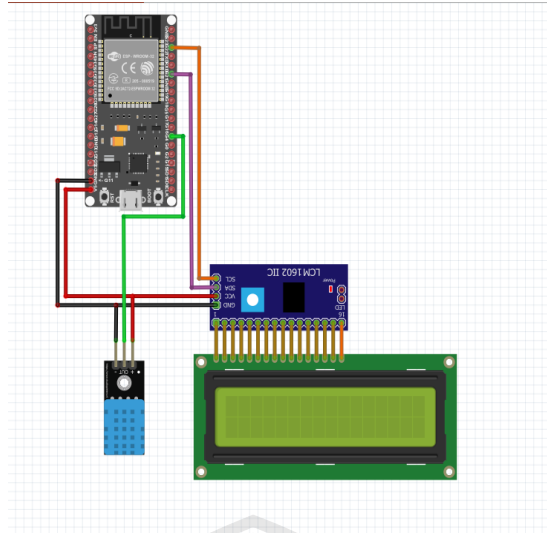
a. *Air Quality* / Kualitas Udara



Gambar 3. 5 Sistem Monitoring Kualitas Udara / *Air Quality*

Rangkaian Pada Gambar 3.5 merupakan sistem monitoring kualitas udara / *air quality* yang terdiri dari dari sensor MQ-135, MQ-7, sensor debu GP2Y1010AU0F, dan LCD 16x2 dengan modul I2C. Sensor MQ-135 berfungsi untuk mendeteksi gas berbahaya seperti amonia, alkohol, dan asap, sedangkan MQ-7 digunakan untuk mendeteksi karbon monoksida (CO). Keduanya dihubungkan ke pin analog ESP32 dan diberi tegangan melalui pin VCC dan GND. Sensor debu GP2Y1010AU0F digunakan untuk mengukur konsentrasi partikel debu di udara, dengan output analog yang juga terhubung ke pin analog ESP32. Sensor ini dilengkapi dengan resistor 150 Ω dan kapasitor 220uF sebagai filter sinyal. LCD 16x2 yang dilengkapi dengan modul I2C berfungsi untuk menampilkan data pembacaan dari ketiga sensor secara langsung, dan terhubung ke ESP32 melalui komunikasi I2C pada pin SDA (GPIO21) dan SCL (GPIO22). Rangkaian ini memungkinkan ESP32 untuk membaca data dari semua sensor dan menampilkannya secara real-time, serta dapat dikembangkan lebih lanjut untuk pengiriman data ke platform IoT seperti Blynk.

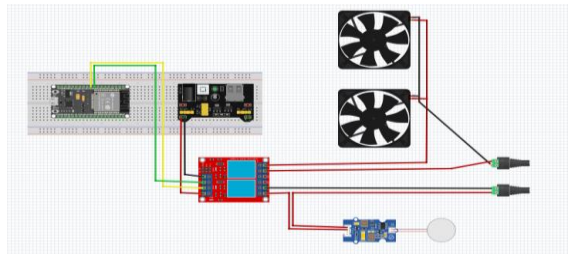
b. Suhu dan Kelembapan



Gambar 3. 6 Monitoring Suhu Dan Kelembapan

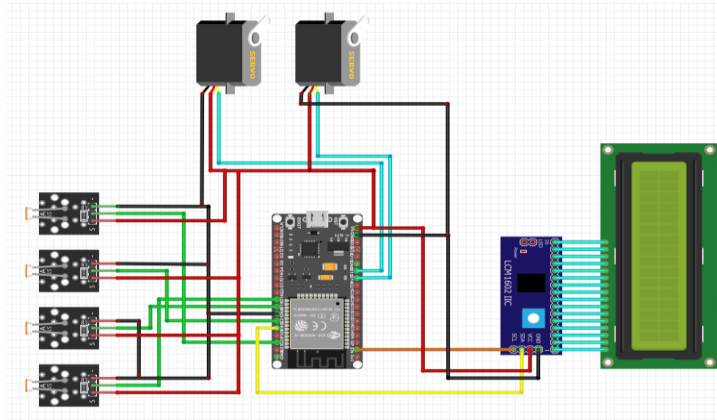
Gambar 3.6 tersebut menunjukkan skematik rangkaian monitoring suhu dan kelembapan menggunakan sensor DHT11 yang terhubung dengan ESP32 dan menampilkan data pada LCD 16x2 melalui modul I2C. Sensor DHT11 digunakan untuk mengukur suhu dan kelembapan udara di sekitar, dengan tiga pin yang terhubung ke ESP32, yaitu VCC (disambungkan ke 3.3V), GND (ke GND), dan Data (ke salah satu pin digital ESP32, dalam gambar ke GPIO4). Modul LCD 16x2 menggunakan antarmuka I2C untuk komunikasi yang efisien, di mana pin SDA dari modul I2C dihubungkan ke pin GPIO21 ESP32, dan pin SCL ke GPIO22. VCC dan GND dari modul I2C juga disambungkan ke 3.3V dan GND ESP32. Dengan konfigurasi ini, ESP32 akan membaca data suhu dan kelembapan dari sensor DHT11 lalu menampilkannya secara real-time ke LCD 16x2.

c. Air purifier dan Air Humidifier

Gambar 3. 7 Sistem *Air Purifier* Dan *Air Humidity*

Gambar 3.7 rangkaian tersebut menunjukkan sistem *air purifier* dan *air humidity* berbasis ESP32 yang terdiri dari dua kipas DC, modul *ionizer* (ion negatif), sensor kelembapan tanah (sebagai pengganti kontrol kelembapan udara), modul relay, *jack barel*, serta satu daya (power supply) eksternal. ESP32 digunakan sebagai otak sistem untuk mengontrol seluruh komponen berdasarkan pembacaan sensor. Dua buah kipas DC dihubungkan melalui modul relay yang memungkinkan ESP32 mengontrol kapan kipas dinyalakan atau dimatikan. Modul *ionizer* juga terhubung ke *relay* untuk mengaktifkan pelepasan ion negatif yang berfungsi membersihkan udara dari partikel berbahaya. Relay mendapatkan sinyal kendali dari ESP32 melalui koneksi pin digital. *Power supply* eksternal yang terhubung melalui *jack barel* digunakan untuk memberikan daya utama ke kipas dan modul lain dengan tegangan yang lebih besar dari 3.3V ESP32. Sensor kelembapan (*humidity*) bertugas mendeteksi tingkat kelembapan dan mengirimkan data ke ESP32 untuk mengatur kipas dan *ionizer* secara otomatis. Rangkaian ini memungkinkan kontrol otomatis sistem pemurnian dan pelembapan udara berdasarkan kondisi lingkungan.

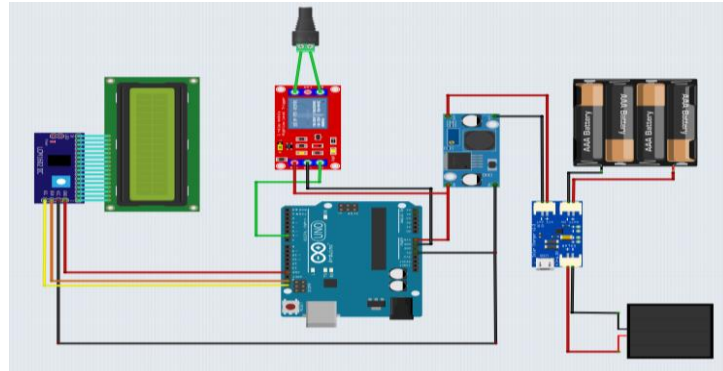
d. Solar Tracking



Gambar 3. 8 Sistem *Solar Tracking* (Pelacak Matahari)

Rangkaian pada Gambar 3. 8 merupakan sistem *solar tracking* (pelacak matahari) berbasis ESP32 yang menggunakan 4 buah sensor LDR (Light Dependent Resistor), 2 buah servo motor, dan LCD I2C 16x2 untuk menampilkan informasi. Keempat sensor LDR dipasang pada keempat arah (atas, bawah, kiri, kanan) dan digunakan untuk mendeteksi intensitas cahaya dari berbagai arah. Masing-masing LDR dihubungkan ke pin analog ESP32 untuk mengukur perbedaan intensitas cahaya. Dua buah servo motor berfungsi untuk menggerakkan panel surya secara horizontal dan vertikal berdasarkan nilai pembacaan dari sensor LDR—satu servo untuk sumbu X (kanan-kiri) dan satu lagi untuk sumbu Y (atas-bawah). ESP32 akan mengatur posisi kedua servo agar panel menghadap ke arah sumber cahaya paling terang (matahari). Selain itu, LCD I2C 16x2 terhubung ke ESP32 melalui pin SDA dan SCL untuk menampilkan informasi seperti nilai intensitas cahaya atau sudut posisi servo. Modul LCD I2C ini memudahkan koneksi hanya dengan 4 kabel (VCC, GND, SDA, dan SCL). VCC dan GND semua sensor dan aktuator terhubung ke jalur *power* (3.3V atau 5V dan GND) dari ESP32. Rangkaian ini sangat ideal digunakan untuk proyek energi terbarukan, khususnya panel surya, agar dapat mengikuti posisi matahari secara otomatis dan memaksimalkan penyerapan energi.

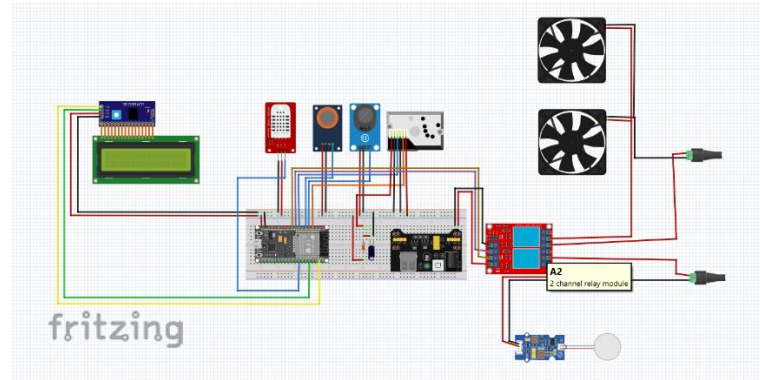
e. Solar System

Gambar 3. 9 Sistem *Solar Panel* Terintegrasi

Rangkaian pada Gambar 3. 9 merupakan sistem solar panel terintegrasi berbasis Arduino Uno, yang dirancang untuk menyuplai dan memonitor daya dari panel surya ke baterai dan mengendalikan beban menggunakan *relay*. Panel surya terhubung ke modul *solar charger* V1.0, yang bertugas mengatur proses pengisian baterai agar aman dan efisien. Output dari baterai kemudian masuk ke modul *step-down* (buck converter) untuk menurunkan tegangan ke level yang sesuai agar dapat digunakan oleh perangkat lain. Arduino Uno menjadi pusat pengendali utama dalam sistem ini. Melalui koneksi dengan *relay*, Arduino dapat mengontrol aliran listrik ke beban (yang disimulasikan melalui jack barel). Sementara itu, LCD I2C 16x2 digunakan untuk menampilkan informasi penting seperti status pengisian, tegangan, atau indikator lainnya. LCD ini dihubungkan melalui komunikasi I2C, yang memudahkan dengan hanya menggunakan dua pin data (SDA dan SCL). Dengan konfigurasi ini, sistem dapat digunakan untuk membangun sistem tenaga surya otomatis, yang memantau status pengisian baterai dari panel surya, mengatur distribusi daya menggunakan relay, dan memberikan informasi *real-time* melalui layar LCD.

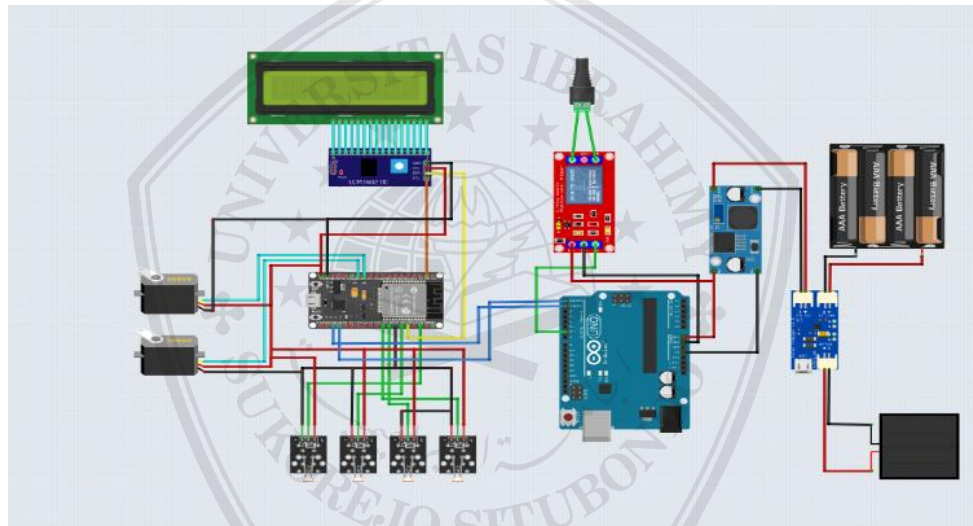
f. Keseluruhan Komponen Perangkat Keras

Skema secara keseluruhan dari pemantauan kualitas udara dan penanganan udara kotor berbasis IoT adalah mengoptimalkan pemantauan kualitas udara secara *real-time* serta penanganan udara kotor dengan bantuan energi surya dan berbasis *Internet of Things*.



Gambar 3. 10 Skema Keseluruhan Pemantauan Kualitas Udara Beserta Penanganannya

Pada Gambar 3.10 adalah skema keseluruhan pemantauan kualitas udara beserta penanganannya dengan berbasis IoT.

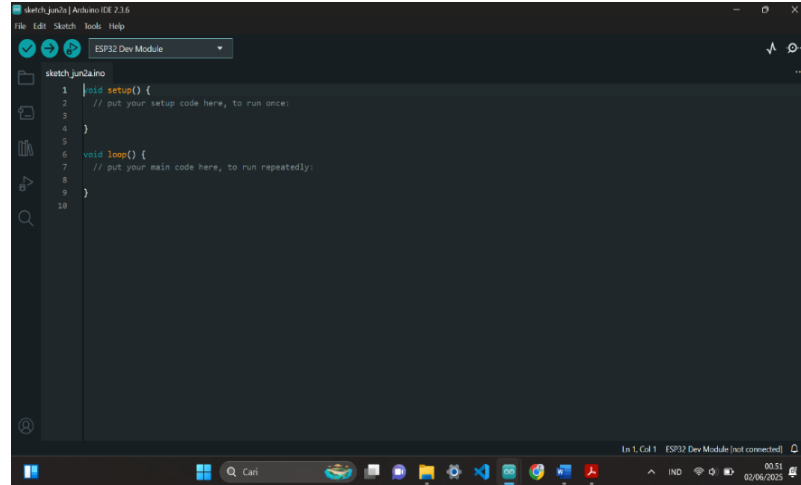


Gambar 3. 11 Skema Keseluruhan *Solar Tracking* Dengan *Solar System*

Pada Gambar 3.11 adalah skema keseluruhan *solar tracking* dengan *Solar system* dengan bantuan tenaga surya agar kinerja dari pemantauan kualitas udara serta penanganannya dapat terpantau secara *real-time* meskipun listrik konvensional mati. Sistem ini akan ber alih kepada *solar system*.

3.4.2 Perancangan Perangkat Lunak

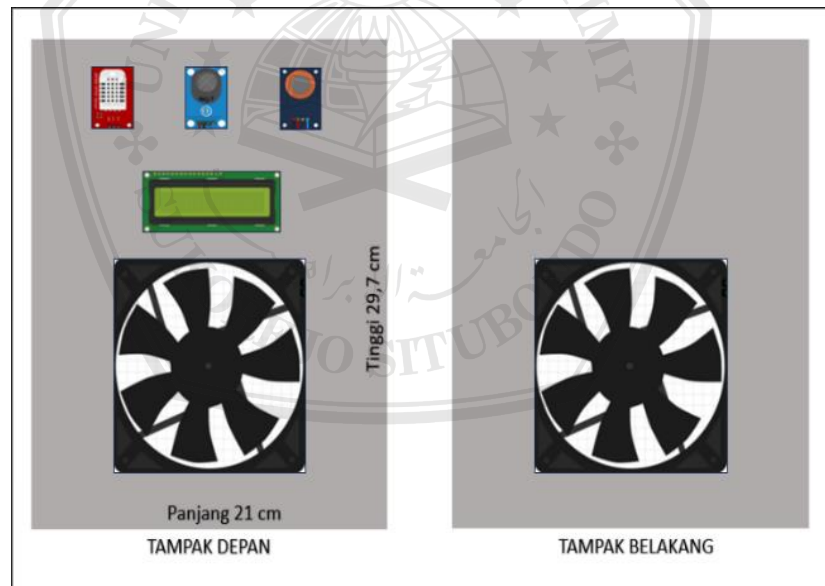
Perancangan perangkat lunak menggunakan aplikasi *Arduino IDE* untuk mengelola program yang nantinya akan di *upload* ke ESP32 sebagai mikrokontroler dari semua komponen program.



Gambar 3. 12 Tampilan Default Ketika Membuka Aplikasi Arduino UNO

Pada Gambar 3. 12 adalah tampilan *Default* ketika membuka aplikasi Arduino UNO yang akan digunakan untuk mengelola program untuk sistem pemantauan kualitas udara dan penanganan udara kotor berbasis IoT.

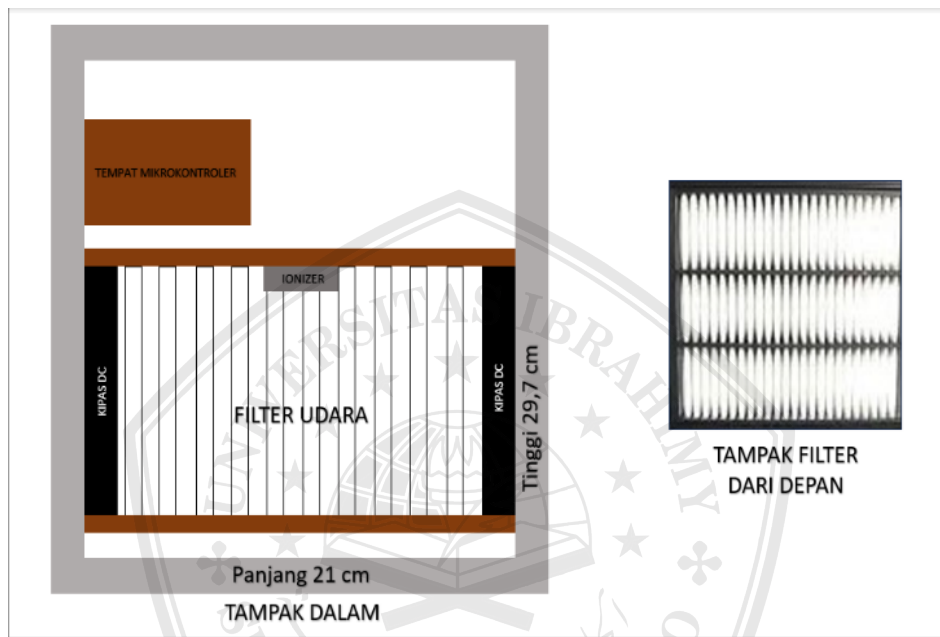
3.4.3 Desain *Case* Seluruh Komponen



Gambar 3. 13 Desain *Case* Seluruh Komponen

Gambar 3. 13 tersebut adalah *case* pada pemantauan kualitas udara dan penanganan udara kotor yang berfungsi sebagai tempat seluruh komponen akan diletakkan. *Case* ini memiliki ukuran panjang 21 cm dan tinggi 29,7 cm, setara dengan ukuran kertas A4, sehingga cukup memuat seluruh komponen. *Case* ini menggunakan bahan akrilik agar tahan terhadap segala keadaan. Pada tampak depan *case*, terdapat beberapa komponen utama. Dibagian atas terdapat tiga

sensor, yaitu sensor DHT11, MQ7, dan MQ135. Dibawah deretan sensor sensor tersebut terdapat modul LCD 16x2 yang digunakan untuk menampilkan data dari sensor-sensor tersebut. Di bagian paling bawah tampak sebuah kipas DC yang berfungsi sebagai alat sirkulasi udara, baik untuk menghisap udara kotor. Pada tampak belakang case, terlihat kipas yang sama, namun dari sisi sebaliknya. Ini menunjukkan bahwa kipas tersebut dipasang menembus case dan memungkinkan aliran udara keluar-masuk dengan efisien.



Gambar 3. 14 Case Alat Pemantau Dan Penjernih Udara Berbasis Iot

Gambar tersebut menunjukkan desain tampak dalam dari *case* alat pemantau dan penjernih udara berbasis IoT. *Case* ini memiliki ukuran panjang 21 cm dan tinggi 29,7 cm. Di bagian atas tampak sebuah ruang khusus yang diperuntukkan bagi *mikrokontroler*, yang berfungsi sebagai pusat kendali sistem. Di bagian tengah *case* terdapat sistem filter udara yang terdiri dari dua kipas DC yang diletakkan di sisi depan dan belakang filter. Kipas-kipas ini berfungsi untuk menghisap dan mendorong udara melewati filter agar partikel-partikel polutan dapat tersaring.

Di antara dua kipas tersebut terdapat filter udara utama yang ditampilkan juga dalam gambar sebelah kanan sebagai "Tampak Filter dari Depan". Filter ini berperan penting dalam menyaring debu, asap, dan partikel-partikel berbahaya dari udara yang masuk. Di atas filter udara terdapat komponen *ionizer*, yang

berfungsi untuk melepaskan ion negatif ke dalam udara guna membantu mengendapkan partikel halus dan membunuh bakteri atau virus di udara. Udara yang masuk akan difilter terlebih dahulu, kemudian diproses oleh *ionizer*; dan akhirnya didorong kembali ke luar melalui kipas sebagai udara bersih.

Secara keseluruhan, desain case ini memperlihatkan sistem pemurnian udara yang efektif dan terintegrasi dengan kontrol otomatis berbasis *mikrokontroler*. Sistem ini dirancang tidak hanya untuk memantau kualitas udara, tetapi juga untuk membersihkannya secara aktif melalui kombinasi filter fisik dan teknologi ionisasi.

3.5 Rencana Alat dan Anggaran

Rencana alat yang ditampilkan pada desain merupakan sistem pemantauan kualitas udara serta penanganan udara kotor berbasis IoT yang bertujuan untuk meningkatkan kualitas udara di dalam ruangan. Rencana alat dan anggaran adalah sebagai berikut :

Tabel 3. 3 Rencana Alat dan Anggaran

No	Nama Alat	Jumlah yang dibutuhkan	Harga
1	ESP32 -Wrom	1	Rp. 65.000
2	DHT11	1	Rp. 32.000
3	Sensor MQ-135	1	Rp. 29.000
4	Sensor MQ-7	1	Rp. 20.500
5	Sensor LDR	4	Rp.5.800 * 4
6	Relay Module	2	Rp. 31.200 * 2
7	Sensor INA219	2	Rp. 30.900 * 2
8	Motor servo	2	Rp. 13.900 * 2
9	Modul solar cell	2	Rp.20.000 * 2
10	Grove-water atomizer v1.0	1	Rp. 23.000
11	Sharp Gp2y1010au0f	1	Rp.215.500
12	Mpu6050 Gyroscope	1	Rp. 15.000
13	Lcd 16x2c	1	Rp.26.000
14	Kipas DC 12 v	2	Rp. 7.800 * 2
15	Jack Dc	4	Rp. 2.000 * 2

Tabel Lanjutan 3. 4 Rencana Alat dan Anggaran

16	Baterai Li Ion	4	Rp. 44.000 * 4
17	1 roll kabel jumper	1	Rp.36.000
18	Akrilik	8	Rp. 8.000 * 8
19	Lem akrilik	3	Rp. 8.500 * 3
Jumlah			Rp. 962.300



BAB IV PERANCANGAN SISTEM

4.1 Kontruksi Sistem

Konstruksi sistem ini merupakan proses perancangan, pembangunan, dan integrasi dari seluruh komponen sistem, baik perangkat keras maupun perangkat lunak hingga sistem ini berfungsi sesuai tujuan.

4.1.1 Kebutuhan sistem

Sistem ini dirancang bertujuan untuk memantau kualitas udara di dalam ruangan serta memberikan penanganan otomatis terhadap udara kotor. Sistem ini berbasis IoT sehingga data kualitas udara dapat di pantau secara *real-time* melalui *platfrom Blynk*, berikut ini bagian – bagian yang digunakan dari sistem adalah sebagai berikut :

a. Hardware

Komponen ini merupakan komponen utama yang sangat di perlukan dalam mewujudkan sistem. Dalam hal spesifikasi *hardware* yang dibutuhkan adalah sebagai berikut :

1. Komputer atau Laptop dengan minimal spesifikasi sebagai berikut :

Tabel 4. 1 Spesifikasi Komputer

Komponen	Spesifikasi Minimal
Prosesor	Intel Core I3 generasi ke – 9 ke atas / AMD Ryzen 3
Ram	8 GB
Storege	128 Gb (Disarankan SSD)
USB Port	Minimal 2 Port USB-A

Selain itu *hardware* yang dibutuhkan pada bagian laptop / komputer yang akan kita gunakan adalah

- a. Monitor
 - b. Mouse (disarankan)
 - c. Keyboard
2. Sensor dan Mikrokontroler
 - a. ESP32 Wrom
 - b. Sensor MQ-135

- c. Sensor MQ-7
 - d. DHT 11
 - e. Relay
 - f. Sensor Mpu6050 Gyroscope
 - g. Sensor LDR
 - h. Sensor INA219
 - i. Grove – water otomization V1.0
 - j. Motor Servo
 - k. Modul Solar Cell
 - l. Sensor Debu Sharp Gp2y1010au0f
 - m. Kipas DC 12v
3. Komponen kelistrikan yang dibutuhkan adalah sebagai berikut :
- a. Kabel USB tipe C
 - b. Kabel Jumper male to female
 - c. Kabel Jumper male to male
 - d. Kabel Jumper female to female
 - e. Jack DC
 - f. Baterai Li Ion
4. Case / Bok Alat untuk tempat rangkaian komponen :
- a. Akrilik
 - b. Lem akrilik
- b. Software

Selain hardware, kebutuhan yang selanjutnya adalah software juga memiliki peran penting yang tidak kalah pentingnya dengan hardware. Berikut ini beberapa software yang dibutuhkan atau digunakan saat pembuatan sistem ini adalah sebagai berikut :

1. Sistem operasi Windows 11 umumnya yang dipakai
2. Arduino IDE
3. Fritzing
4. Blynk IoT
5. Draw.io

4.1.2 Instalasi sistem

Instalasi sistem ini adalah sebagai proses pemasangan, perakitan dan pengaktifan sistem secara menyeluruh agar bisa dijalankan di lingkungan nyata.

a. Instalasi Driver USB to serial pada *Device Manager*

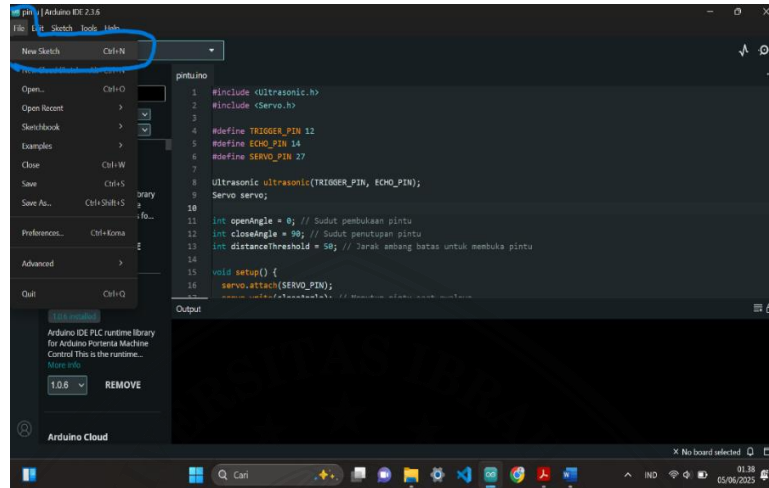
Instalasi driver USB to Serial merupakan Langkah penting agar komputer dapat mengenali perangkat mikrokontroler ESP32 WROOM dan Arduino Uno yang menggunakan chip konverter USB ke Serial. Chip yang digunakan adalah CH340, tanpa driver ini, perangkat tidak akan muncul di port COM dan proses program akan gagal. Untuk menginstal driver, pertama – tama adalah mengidentifikasi jenis chip yang digunakan pada board, kemudian unduh driver resmi dari situs pembuatan chip tersebut, pada sistem ini chip yang digunakan adalah CH340. Setelah itu, jalankan file instalasi sebagai administrator, setelah berhasil di install, port (COM) perangkat akan muncul di Device Manager di bagian Port COM. Nomor COM tersebut kemudian dapat dipilih di Arduino IDE untuk proses upload program. Jika port belum muncul, pastikan kabel USB yang digunakan mendukung transfer data, dan jika perlu, coba *restart* komputer atau ganti kabel USB. Jika terbaca maka port akan muncul di tampilan device manager seperti pada gambar 4.1 di bawah ini :



Gambar 4. 1 Instalasi *Driver USB to serial* pada *Device Manager*

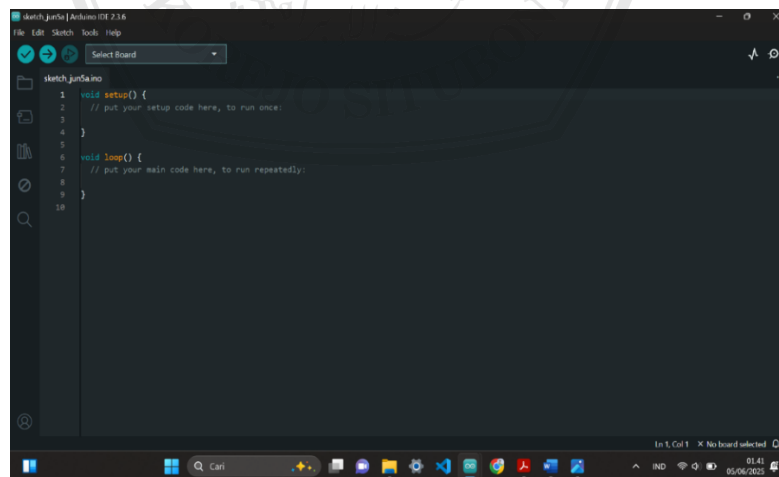
b. Pembuatan *new sketch* Aplikasi Arduino IDE

Sebelum melakukan upload program, pastikan aplikasi arduino IDE sudah terinstall terlebih dahulu. Arduino IDE yang kami gunakan adalah versi 2.3.6. jika sudah terinstall langkah selanjutnya adalah buka aplikasi Arduino IDE. Lalu pergi ke File > klik new skate untuk melakukan input program, Seperti gambar 4. 2 dibawah ini :



Gambar 4. 2 Pembuatan *New Sketch* Aplikasi Arduino IDE

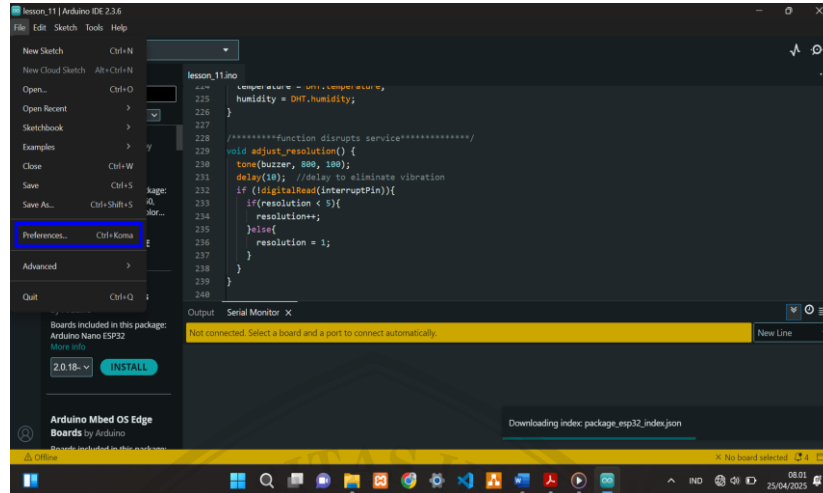
Tampilan awal setelah membuat file baru di aplikasi Arduino IDE. Tampilan awal sebelum malakukan input program seperti gambar 4. 3 dibawah ini :



Gambar 4. 3 Tampilan Awal Sebelum Malakukan Input Program

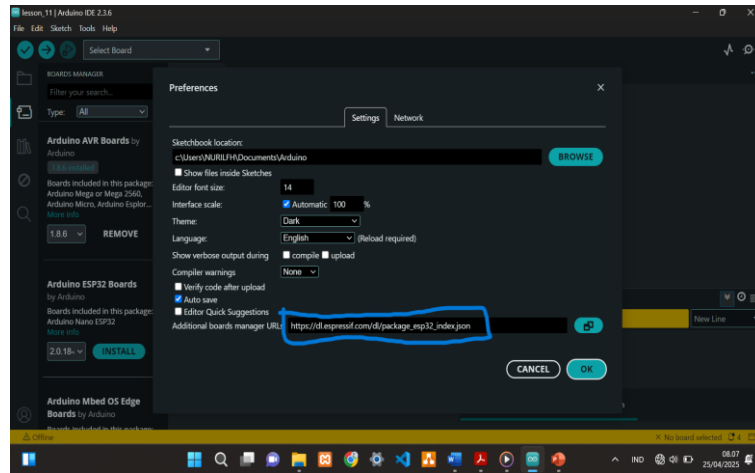
c. Instalasi *Board* ESP32 Wrom pada aplikasi Arduino IDE

Setelah melakukan pembuatan new sketch langkah selanjutnya adalah Instalasi Board ESP32 wrom pada aplikasi Arduino IDE. Langkah pertama yaitu pergi ke File > Preferencer, seperti gambar 4. 4 dibawah ini :



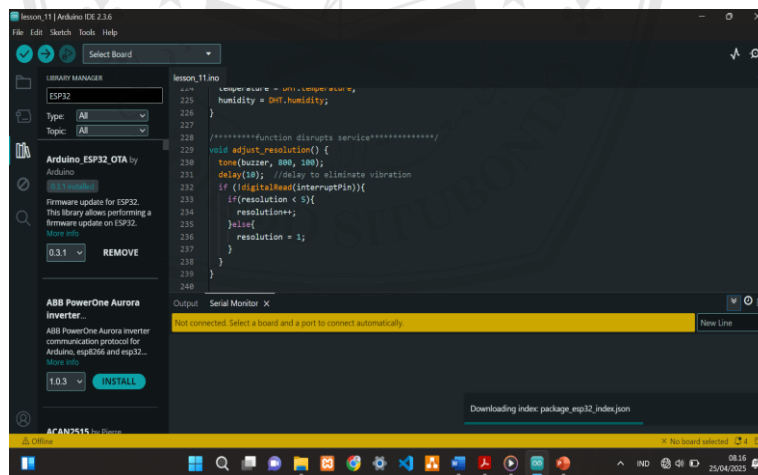
Gambar 4. 4 Tampilan Menu *Preferences*

Setelah File > preferencer terbuka maka langkah selanjutnya adalah memasukkan URL ini https://dl.espressif.com/dl/package_esp32_index.json pada Additional Board manager URL: kemudian langkah selanjutnya adalah menekan tombol OK untuk memulai pengunduhan dan penginstalasian paket Board ESP32 WROM. Proses ini memungkinkan IDE Arduino untuk mengunduh semua file dan pustaka yang diperlukan agar dapat mengenali dan berkomunikasi dengan board ESP32. Selama proses ini, IDE akan mengunduh berkas-berkas yang diperlukan, seperti core, driver, dan konfigurasi khusus untuk board ESP32. Setelah pengunduhan selesai, IDE akan mengkonfigurasi board seperti pada gambar 4. 5berikut:



Gambar 4. 5 Tampilan Proses Konfigurasi pada Board

Jika sudah terinstall, langkah selanjutnya adalah membuka Tools > Library Manager, ketikkan pada kotak pencarian "ESP32" pilih versi terbaru dan install pustaka library ESP32 WROM, serta tak kalah pentingnya selama proses penginstalan harus memastikan koneksi internet berjalan dengan stabil karena jika terputus, maka proses ini dapat gagal dan memerlukan pengulangan. Gambar 4. 6 dibawah ini adalah proses penginstalan library board ESP32 WROM versi terbarunya :

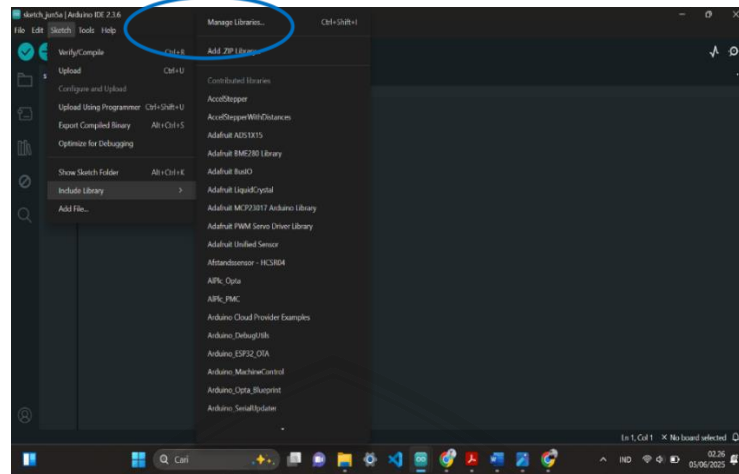


Gambar 4. 6 Proses Penginstalan *Library Board ESP32 WROM*

d. Instlasi Library sesuai dengan sketch yang kita butuhkan

Langkah selanjutnya setelah instalasi board untuk ESP32 pada aplikasi Arduino IDE adalah menginstalasi beberapa library yang kita butuhkan untuk pembuatan sketch pemantauan kualitas udara serta penanganannya langkah pertama adalah pilih menu pada bagian header aplikasi Arduino

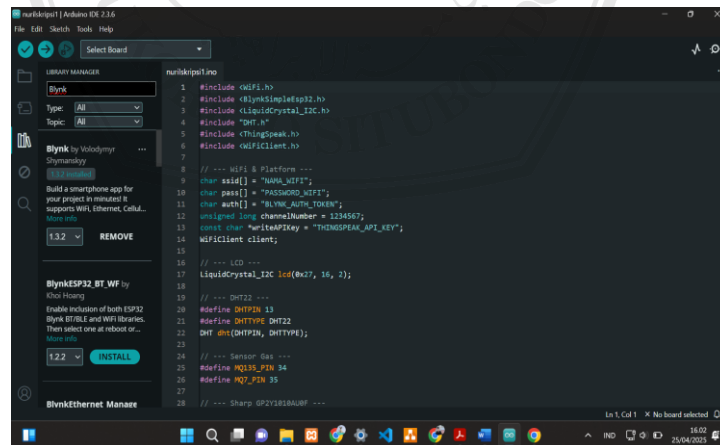
IDE, setelah memilih include library yang akan digunakan dalam program tersebut. Jika pada library yang kita gunakan tidak tersedia maka, pilih menu manage libraries pada tool include library yang nantinya akan diarahkan ke file explore yang sudah kita download seperti gambar 4. 7 :



Gambar 4. 7 Tampilan Instalasi Library Sesuai Dengan Sketch Yang Kita Butuhkan

e. Instalasi *Blynk* pada Arduino IDE

Instalasi *Blynk* pada aplikasi Arduino IDE yaitu ada pada *Board Library Manager*, ketikkan *Blynk* pada filter pencarian, sehingga muncul gambar seperti pada gambar 4. 8 lalu klik **Install**.

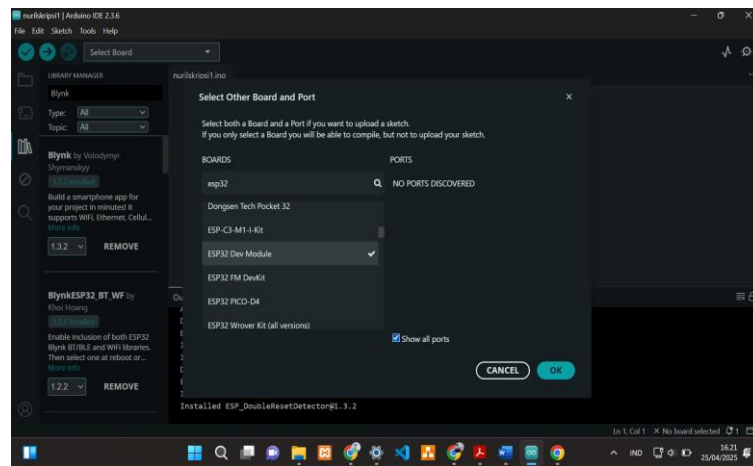


Gambar 4.8 Instalasi *Blynk* pada Arduino IDE

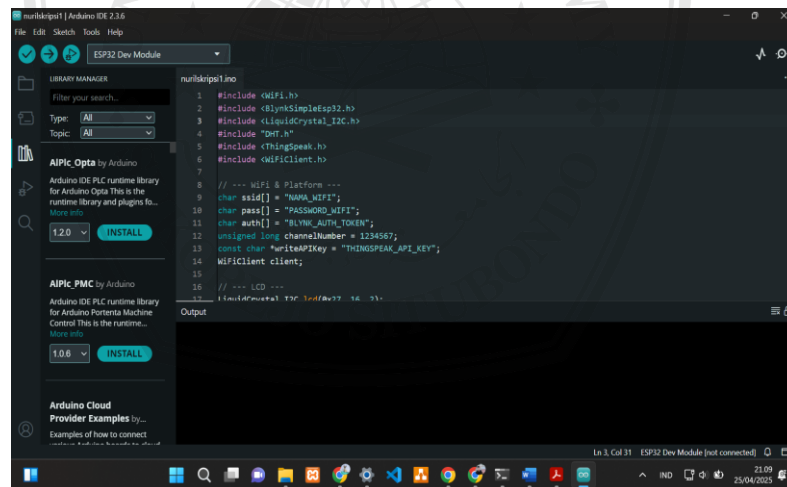
f. Upload Program ke mikrokontroler ESP32 Wrom

Setelah beberapa Pustaka di atas terinstal, selanjutnya adalah memasukan program ke NodeMCU ESP32 menggunakan aplikasi Arduino IDE. Sebelum program di upload pastikan settingan pada *Tools Board and Port*

sudah benar karena disini menggunakan ESP32 maka yang digunakan pada **Board and Port** adalah seperti gambar 4. 9 dibawah ini:



Gambar 4. 9 Tampilan Upload Program ke mikrokontreoler ESP32 Wrom Setelah Board and Port sesuai maka langkah selanjutnya adalah meng upload program ke ESP32 WROM melalui USB yang disesuaikan dengan port di atas seperti gambar 4. 10 :



Gambar 4. 10 Tampilan *Upload* Program ke ESP32

Pilih tanda panah (→) maka proses uploading ke ESP32 akan dimuali tunggu sampai proses peng upload-tan selesai.

4.1.3 Segmen Program

Segmen program akan menampilkan *source code* dari sistem yang dibangun guna menjelaskan bagian – bagian dalam program tersebut.

Segmen program include library

Segmen 1. Library

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
```

Pada segmen satu berisi lima *library* yang digunakan dalam proyek berbasis ESP32. *Library Wire.h* digunakan untuk komunikasi I2C antara ESP32 dengan perangkat lain seperti LCD. *Library LiquidCrystal_I2C.h* digunakan untuk mengendalikan layar LCD yang menggunakan antarmuka I2C, sehingga hanya memerlukan dua pin (SDA dan SCL). Kemudian, *WiFi.h* adalah *library* bawaan ESP32 yang memungkinkan perangkat untuk terhubung ke jaringan WiFi, yang menjadi syarat utama agar ESP32 dapat mengakses internet. *Library BlynkSimpleEsp32.h* digunakan untuk menghubungkan ESP32 dengan platform IoT Blynk, sehingga data dari sensor dapat ditampilkan secara *real-time* di aplikasi *Blynk* pada *smartphone*. Terakhir, *library* *DHT.h* berfungsi untuk membaca data suhu dan kelembaban dari sensor DHT seperti DHT11 atau DHT22. Gabungan dari kelima *library* ini menunjukkan bahwa proyek tersebut bertujuan untuk membaca data suhu dan kelembaban dari sensor DHT, menampilkannya di LCD I2C, dan mengirimkannya ke aplikasi Blynk melalui koneksi WiFi menggunakan *board* ESP32.

Segmen 2. Pendefinisian Blynk

```
// pendefinisian blynk
#define BLYNK_TEMPLATE_ID "Monitor Udara"
#define BLYNK_TEMPLATE_NAME "Monitor"
#define BLYNK_AUTH_TOKEN "0937ynd3y39vdwehk3"
```

Pada segmen program diatas merupakan bagian konfigurasi awal untuk menghubungkan proyek berbasis ESP32 dengan platform IoT *Blynk*. Baris *#define BLYNK_TEMPLATE_ID "Monitor Udara"* berfungsi untuk

mendefinisikan ID *template* yang telah dibuat di *platform Blynk*, yang berguna untuk mengidentifikasi proyek atau perangkat secara spesifik. Selanjutnya, `#define BLYNK_TEMPLATE_NAME "Monitor"` memberikan nama *template* yang bersifat deskriptif dan membantu pengguna mengenali proyek tersebut di *dasbor Blynk*. Terakhir, `#define BLYNK_AUTH_TOKEN "0937ynd3y39vdwehk3"` adalah token otentikasi unik yang diberikan oleh *Blynk* untuk masing-masing perangkat atau proyek. Token ini sangat penting karena berfungsi sebagai kunci penghubung antara aplikasi *Blynk* di *smartphone* maupun laptop/komputer dan perangkat ESP32, sehingga memungkinkan data dapat dikirim dan diterima dengan aman. Ketiga definisi ini wajib disesuaikan dengan pengaturan yang telah dibuat di aplikasi *Blynk* sebelum perangkat dapat berfungsi dengan benar.

Segmen 3. Pendefinisian wifi

```
// pendefinisian wifi
char ssid[] = "Strobo_Jahat";
char pass[] = "fahdazada";
```

Pada segmen program di atas adalah berfungsi untuk menyimpan nama jaringan WiFi (SSID) dan kata sandi (password) yang akan digunakan oleh ESP32 untuk terhubung ke internet. Nilai "Strobo_Jahat" adalah nama WiFi yang tersedia, dan "fahdazada" adalah password-nya. Keduanya disimpan dalam bentuk *array* karakter (string) dengan *tipe char[]*. Kode ini biasanya digunakan bersama dengan *library WiFi.h* agar ESP32 dapat terhubung ke jaringan WiFi yang ditentukan. Tanpa konfigurasi ini, ESP32 tidak akan dapat mengakses internet dan tidak bisa mengirim data ke *platform Blynk*.

Segmen 4. Konfigurasi Pin Sensor MQ135

```
// konfigurasi pin sensor mq135
#define MQ135_PIN 33 // Pin analog
sensor MQ-135
#define MQ135_THRESHOLD_1 1000 // Batas kualitas
udara
#define BLYNK_VPIN V1 // pin virtual di
blynk
```

```

// konfigurasi DHT
#define DHTPIN 5           // pin dht11
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
// konfigurasi pin mq7
const int pinSensorMQ7 = 34; // Input analog MQ-7
const int pinHeater = 25;    // Pemanas MQ-7
// konfigurasi pin GP2Y1010AU0F
const int pinLED = 12;     // Kontrol LED sensor
const int pinVo = 34;     // Output analog sensor
// konfigurasi pin lcd
LiquidCrystal_I2C lcd(0x27, 16, 2);

```

Segmen program tersebut merupakan bagian dari konfigurasi awal yang digunakan dalam skripsi pemantauan kualitas udara serta penanganan udara kotor berbasis ESP32. Bagian pertama mengatur pin untuk sensor gas MQ135, yang digunakan untuk mendeteksi polutan gas seperti amonia dan benzena. Pin sensor tersebut terhubung ke GPIO 33 pada ESP32 dan memiliki ambang batas (MQ135_THRESHOLD_1) sebesar 1000 untuk menentukan tingkat pencemaran udara. Nilai dari sensor ini akan dikirim ke aplikasi *Blynk* melalui virtual pin V1. Selanjutnya, untuk membaca suhu dan kelembaban, digunakan sensor DHT11 yang terhubung ke pin GPIO 5. Objek *dht* dibuat dengan menyebutkan pin dan tipe sensor menggunakan *library* DHT.h.

Kemudian, konfigurasi dilanjutkan dengan sensor MQ7 yang digunakan untuk mendeteksi karbon monoksida (CO). Sensor ini memiliki dua pin penting, yaitu *pinSensorMQ7* yang terhubung ke GPIO 34 untuk membaca output sensor, dan *pinHeater* ke GPIO 25 untuk mengontrol pemanas internal sensor. Selanjutnya, sensor GP2Y1010AU0F yang digunakan untuk mendeteksi debu atau partikel di udara dikonfigurasi dengan dua pin: *pinLED* (GPIO 12) untuk mengendalikan LED internal sensor dan *pinVo* (GPIO 34) untuk membaca nilai *output* sensor debu. Terakhir, layar LCD I2C dikonfigurasi menggunakan alamat I2C 0x27 dengan ukuran 16 kolom dan 2 baris, menggunakan *library* *LiquidCrystal_I2C*, yang berfungsi untuk

menampilkan data seperti suhu, kelembaban, atau tingkat polusi udara secara langsung di perangkat. Semua konfigurasi ini menjadi fondasi utama dalam sistem monitoring kualitas udara yang terintegrasi dengan *platform* IoT.

Segmen 5. Pin Virtual Blynk

```
// Pin Virtual Blynk
#define VPIN_TEMP V0
#define VPIN_HUM V1
```

Segmen program tersebut adalah Baris kode `#define VPIN_TEMP V0` dan `#define VPIN_HUM V1` digunakan dalam pemrograman *mikrokontroler* yang terhubung dengan platform IoT seperti Blynk. Kode ini berfungsi untuk mendefinisikan pin virtual yang digunakan dalam komunikasi antara perangkat (seperti ESP32 atau Arduino) dengan aplikasi Blynk. Virtual pin (pin virtual) adalah jalur komunikasi logis yang tidak terkait langsung dengan pin fisik pada papan mikrokontroler, melainkan digunakan untuk mengirim atau menerima data antara perangkat dan aplikasi. Dalam hal ini, `VPIN_TEMP` merepresentasikan virtual pin `V0` yang digunakan untuk mengirim data suhu, sedangkan `VPIN_HUM` merepresentasikan virtual pin `V1` yang digunakan untuk mengirim data kelembaban. Misalnya, setelah membaca data dari sensor suhu dan kelembaban seperti DHT11 atau DHT22, nilai suhu dikirim ke aplikasi melalui `V0` dan nilai kelembaban melalui `V1`. Di aplikasi *Blynk*, pengguna dapat menampilkan data tersebut secara *real-time* menggunakan *widget* seperti *Gauge* atau *Value Display*.

Segmen 6. Kalibrasi Sensor mq7

```
// kalibrasi sensor mq7
float sensor_reading_clean_air = 620.0;
float reference_resistor_kOhm = 10.0;
// variabel kontrol mq7
float sensor_val = 0;
float last_CO_ppm = 0;
//mq135
unsigned long lastSendTime = 0;
```

```

const unsigned long sendInterval = 25000; // 25
detik
//mq7
unsigned long lastPrint = 0;
unsigned long phaseStart = 0;
int phase = 1; // 0 = measuring, 1 = heating

```

Segmen program tersebut merupakan bagian dari program mikrokontroler yang menggunakan sensor gas MQ-7 dan MQ-135 untuk mendeteksi kualitas udara. Variabel `sensor_reading_clean_air` diset dengan nilai 620.0 sebagai referensi pembacaan sensor saat udara bersih, sedangkan `reference_resistor_kOhm` bernilai 10.0 sebagai nilai hambatan referensi (biasanya bagian dari rangkaian sensor MQ). Variabel `sensor_val` dan `last_CO_ppm` digunakan untuk menyimpan nilai pembacaan sensor MQ-7 dan hasil perhitungan konsentrasi karbon monoksida (CO) dalam satuan ppm. Untuk sensor MQ-135, terdapat pengaturan waktu pengiriman data menggunakan variabel `lastSendTime` dan `sendInterval` dengan interval pengiriman setiap 25 detik (25000 milidetik). Sedangkan untuk pengendalian fase kerja sensor MQ-7, digunakan variabel `lastPrint`, `phaseStart`, dan `phase` yang masing-masing berfungsi untuk mencatat waktu terakhir data dicetak, waktu mulai fase saat ini, dan penanda fase kerja sensor (misalnya fase pemanasan atau pendinginan). Seluruh variabel ini berfungsi sebagai dasar logika pengambilan dan pengolahan data dari sensor dalam sistem monitoring kualitas udara berbasis *mikrokontroler*.

Segmen 7. Inisialisasi Sensor DHT11

```

Serial.begin(115200);
// Inisialisasi DHT
dht.begin();
// Inisialisasi LCD
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Monitoring...");

```

```
delay(2000);  
lcd.clear();  
// Koneksi ke WiFi dan Blynk  
WiFi.begin(ssid, pass);  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("WiFi Connected");  
Blynk.begin(auth, ssid, pass);
```

Segmen program tersebut Fungsi `setup()` pada kode ini digunakan untuk melakukan inisialisasi awal seluruh komponen sistem monitoring kualitas udara berbasis mikrokontroler (seperti ESP32) yang terhubung dengan sensor DHT11, LCD I2C, dan platform IoT Blynk. Pertama, `Serial.begin(115200)` digunakan untuk memulai komunikasi serial antara mikrokontroler dan komputer dengan baud rate 115200, sehingga data dan pesan debug dapat ditampilkan di Serial Monitor. Kemudian, sensor suhu dan kelembaban DHT11 diinisialisasi menggunakan `dht.begin()` agar perangkat siap membaca data lingkungan. Selanjutnya, LCD I2C diinisialisasi melalui `lcd.init()` yang mengaktifkan modul LCD, dan `lcd.backlight()` digunakan untuk menyalakan lampu latar agar tampilan teks lebih jelas. LCD menampilkan teks "Monitoring..." pada baris pertama (`lcd.setCursor(0, 0)` dan `lcd.print("Monitoring...")`) sebagai indikasi bahwa sistem sedang melakukan proses awal, lalu setelah penundaan 2 detik (`delay(2000)`), layar dibersihkan dengan `lcd.clear()`.

Setelah perangkat keras aktif, mikrokontroler mulai menghubungkan diri ke jaringan WiFi menggunakan `WiFi.begin(ssid, pass)`, dengan `ssid` dan `pass` adalah nama dan kata sandi jaringan yang sudah didefinisikan sebelumnya. Proses koneksi ini berlangsung dalam perulangan `while (WiFi.status() != WL_CONNECTED)` yang terus berjalan hingga koneksi ke WiFi berhasil, dengan indikator berupa pencetakan titik-titik secara bertahap di Serial Monitor setiap 500 milidetik sebagai tanda proses koneksi sedang berlangsung. Jika

koneksi berhasil, akan muncul pesan "WiFi Connected" di Serial Monitor untuk memberi tahu bahwa perangkat telah terhubung ke internet. Terakhir, baris `Blynk.begin(auth, ssid, pass)` digunakan untuk menginisialisasi koneksi ke *platform Blynk* menggunakan token autentikasi `auth`, serta SSID dan password WiFi. Dengan koneksi ini, mikrokontroler dapat mengirim dan menerima data melalui aplikasi Blynk, memungkinkan pemantauan sensor secara real-time dari perangkat seluler pengguna.

Segmen 8. Serial Monitor yang digunakan

```
Serial.begin(115200);
  delay(100);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  Serial.println("== ESP32 Mengirim MQ-135 ke Blynk
==");
```

Segmen program tersebut merupakan bagian dari fungsi inisialisasi pada mikrokontroler ESP32 yang digunakan untuk menghubungkan perangkat dengan platform IoT Blynk serta menampilkan informasi pada Serial Monitor. Baris `Serial.begin(115200);` digunakan untuk memulai komunikasi serial antara ESP32 dan komputer dengan kecepatan 115200 baud, yang memungkinkan pengguna melihat output data atau pesan debug melalui Serial Monitor di Arduino IDE. Setelah itu, diberikan jeda selama 100 milidetik dengan `delay(100);` untuk memastikan semua perangkat siap sebelum melanjutkan proses koneksi. Selanjutnya, baris `Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);` berfungsi untuk menginisialisasi koneksi ke server Blynk menggunakan token autentikasi (`BLYNK_AUTH_TOKEN`) serta nama dan kata sandi jaringan WiFi (`ssid` dan `pass`). Dengan koneksi ini, ESP32 dapat mengirim data dari sensor seperti MQ-135 ke aplikasi Blynk secara real-time. Setelah koneksi dimulai, baris `Serial.println("== ESP32 Mengirim MQ-135 ke Blynk ==");` digunakan untuk mencetak pesan ke Serial Monitor sebagai tanda bahwa sistem siap untuk mengirim data dari sensor MQ-135 ke platform Blynk, sekaligus memberi informasi status kepada pengguna bahwa proses telah dimulai.

Segmen 9. Serial Begin MQ7

```
//mq7
Serial.begin(115200);
lcd.init();
lcd.backlight();
pinMode(pinSensorMQ7, INPUT);
pinMode(pinHeater, OUTPUT);
digitalWrite(pinHeater, HIGH); // Mulai dengan
pemanasan
lcd.setCursor(0, 0);
lcd.print("Connect WiFi...");
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Blynk Connected");
delay(2000);
lcd.clear();
phaseStart = millis();
float getCOppm(float analogVal) {
float sensor_resistance = reference_resistor_kOhm
* (1023.0 / analogVal - 1.0);
float sensor_base_resistance =
reference_resistor_kOhm * (1023.0 /
sensor_reading_clean_air - 1.0);
float ratio = sensor_base_resistance /
sensor_resistance;
float ppm = 134 * ratio - 35;
if (ppm < 0) ppm = 0;
return ppm;
}
```

Segmen program tersebut merupakan bagian dari program mikrokontroler ESP32 yang digunakan untuk membaca data dari sensor gas MQ-7, menampilkan informasi di LCD, serta mengirim data ke platform

Blynk. Pada awalnya, komunikasi serial diaktifkan dengan `Serial.begin(115200)`; untuk memungkinkan pengiriman data atau pesan debug ke Serial Monitor dengan baud rate 115200. Selanjutnya, LCD I2C diinisialisasi menggunakan `lcd.init()`; dan lampu latar diaktifkan dengan `lcd.backlight()`; agar teks pada layar terlihat jelas. Pin yang terhubung ke sensor MQ-7 diatur sebagai input melalui `pinMode(pinSensorMQ7, INPUT)`; sedangkan pin untuk pemanas sensor (heater) diatur sebagai output dengan `pinMode(pinHeater, OUTPUT)`; kemudian diberi logika HIGH menggunakan `digitalWrite(pinHeater, HIGH)`; untuk memulai fase pemanasan sensor.

Selanjutnya, LCD menampilkan teks "Connect WiFi..." pada baris pertama (`lcd.setCursor(0, 0); lcd.print("Connect WiFi...");`) untuk menginformasikan bahwa perangkat sedang mencoba menghubungkan diri ke jaringan internet melalui platform Blynk menggunakan perintah `Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass)`; dengan parameter berupa token autentikasi Blynk dan kredensial WiFi. Setelah koneksi berhasil, LCD dibersihkan dan menampilkan pesan "Blynk Connected" selama dua detik untuk memberikan notifikasi keberhasilan koneksi, lalu layar dibersihkan kembali dan variabel `phaseStart` diatur dengan `millis()`; untuk mencatat waktu awal fase kerja sensor, yang akan digunakan dalam manajemen waktu pemanasan dan pendinginan sensor MQ-7.

Setelah itu, terdapat fungsi `getCOppm(float analogVal)` yang digunakan untuk menghitung konsentrasi karbon monoksida (CO) dalam satuan ppm berdasarkan nilai analog yang dibaca dari sensor. Di dalam fungsi ini, pertama-tama dihitung nilai resistansi sensor saat ini (`sensor_resistance`) dengan menggunakan rumus pembagian tegangan berdasarkan nilai referensi resistor. Kemudian dihitung juga nilai resistansi sensor saat udara bersih (`sensor_base_resistance`) menggunakan nilai referensi pembacaan pada udara bersih (`sensor_reading_clean_air`). Selanjutnya, rasio antara resistansi dasar dan resistansi saat ini digunakan untuk menghitung estimasi konsentrasi CO dengan rumus $\text{ppm} = 134 * \text{ratio} - 35$, di mana angka 134 dan 35 merupakan nilai empiris yang diperoleh dari karakteristik sensor. Jika nilai ppm yang dihasilkan kurang dari nol, maka akan disesuaikan menjadi nol (`if (ppm < 0)`

ppm = 0;) agar hasilnya valid. Nilai ppm ini kemudian dikembalikan oleh fungsi dan dapat digunakan untuk ditampilkan atau dikirim ke Blynk.

Segmen 10. Serial Sensor Debu

```
// GP2Y1010AU0F
Serial.begin(115200);
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
pinMode(pinLED, OUTPUT);
digitalWrite(pinLED, LOW);
// Inisialisasi LCD
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Monitoring Debu");
delay(2000);
lcd.clear();
```

segmen program tersebut merupakan bagian dari inisialisasi sistem monitoring partikel debu menggunakan sensor GP2Y1010AU0F dengan mikrokontroler ESP32 yang terhubung ke platform IoT Blynk dan tampilan LCD I2C. Komunikasi serial dimulai dengan `Serial.begin(115200);` untuk memungkinkan pemantauan data dan pesan debugging melalui Serial Monitor. Selanjutnya, koneksi ke jaringan WiFi dan server Blynk dilakukan menggunakan `Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);` dengan parameter berupa token autentikasi dan kredensial jaringan WiFi. Pin LED pada sensor GP2Y1010AU0F diatur sebagai output melalui `pinMode(pinLED, OUTPUT);`, kemudian diset dalam kondisi mati (LOW) agar LED inframerah tidak terus menyala, karena sensor ini membutuhkan pencahayaan LED hanya saat pengambilan data untuk menjaga akurasi dan efisiensi daya. Selain itu, tampilan LCD diinisialisasi menggunakan `lcd.init();` dan lampu latar diaktifkan dengan `lcd.backlight();`. Untuk memberi informasi kepada pengguna, LCD menampilkan pesan “Monitoring Debu” selama dua detik dengan menggunakan `lcd.print()` dan `delay(2000);`, lalu layar dibersihkan menggunakan `lcd.clear();` untuk menyiapkan tampilan selanjutnya. Namun,

agar sistem benar-benar berfungsi secara optimal, perlu ditambahkan inisialisasi pin pembacaan data sensor sebagai input serta prosedur pembacaan data debu yang sesuai dengan timing yang direkomendasikan dalam datasheet sensor GP2Y1010AU0F.

Segmen 11. Void Loop DHT11

```
//dht11
  Blynk.run();
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT
sensor!");
    return;
  }
  // Tampilkan di Serial Monitor
  Serial.print("Temp: ");
  Serial.print(t);
  Serial.print(" *C, Humidity: ");
  Serial.print(h);
  Serial.println(" %");
  // Tampilkan di LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Suhu: ");
  lcd.print(t);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Lembab: ");
  lcd.print(h);
  lcd.print(" %");
  // Kirim ke Blynk
  Blynk.virtualWrite(VPIN_TEMP, t);
```

```
Blynk.virtualWrite(VPIN_HUM, h);
delay(2000);
```

Segmen program tersebut berfungsi untuk membaca data suhu dan kelembaban dari sensor DHT11, lalu menampilkannya melalui Serial Monitor, LCD, serta mengirimkannya ke platform Blynk. Proses diawali dengan `Blynk.run()`; yang menjalankan loop utama dari platform Blynk agar tetap terhubung dengan server dan menjaga sinkronisasi data secara real-time. Kemudian, fungsi `dht.readHumidity()`; dan `dht.readTemperature()`; digunakan untuk membaca nilai kelembaban (h) dan suhu (t) dari sensor DHT11. Jika hasil pembacaan berupa nilai tidak valid (NaN), maka program akan mencetak pesan error “Failed to read from DHT sensor!” ke Serial Monitor dan menghentikan eksekusi fungsi dengan `return`;

Jika pembacaan berhasil, data suhu dan kelembaban ditampilkan ke Serial Monitor untuk kebutuhan debugging atau pemantauan lokal. Selanjutnya, LCD I2C digunakan untuk menampilkan informasi suhu pada baris pertama dengan label “Suhu:” dan kelembaban pada baris kedua dengan label “Lembab:”. Setelah ditampilkan secara lokal, data suhu dan kelembaban juga dikirim ke aplikasi Blynk menggunakan perintah `Blynk.virtualWrite()` ke virtual pin yang telah ditentukan sebelumnya, yaitu `VPIN_TEMP` untuk suhu dan `VPIN_HUM` untuk kelembaban. Seluruh proses ini diulang setiap dua detik sesuai jeda waktu yang diberikan oleh `delay(2000)`;, memungkinkan sistem untuk terus memantau dan memperbarui data lingkungan secara berkala dan real-time.

Segmen 12. Void Loop MQ135

```
//mq135
Blynk.run();
// Kirim setiap 25 detik
if (millis() - lastSendTime > sendInterval) {
  lastSendTime = millis();
  int MQ135_data = analogRead(MQ135_PIN);
  // Tampilkan ke Serial Monitor
  if (MQ135_data < MQ135_THRESHOLD_1) {
```

```
        Serial.print("Kualitas Udara: BERSIH | ");
    } else {
        Serial.print("Kualitas Udara: KOTOR | ");
    }
    Serial.print(MQ135_data);
    Serial.println(" PPM");
    // Kirim ke Blynk
    Blynk.virtualWrite(BLYNK_VPIN, MQ135_data);
}
```

Segmen program tersebut digunakan untuk membaca dan mengirimkan data kualitas udara dari sensor gas MQ-135 secara berkala ke platform Blynk, sekaligus menampilkannya melalui Serial Monitor. Fungsi `Blynk.run()`; dijalankan untuk menjaga koneksi dan komunikasi antara mikrokontroler dan server Blynk tetap aktif. Selanjutnya, dengan menggunakan fungsi `millis()`, sistem melakukan pengecekan apakah waktu saat ini telah melebihi jeda pengiriman data yang ditentukan oleh variabel `sendInterval`, yaitu 25.000 milidetik (25 detik). Jika waktu terpenuhi, maka sistem memperbarui waktu pengiriman terakhir (`lastSendTime`) dan membaca nilai analog dari sensor MQ-135 melalui pin yang telah didefinisikan (`MQ135_PIN`). Hasil pembacaan ini kemudian dievaluasi: jika nilainya di bawah ambang batas tertentu (`MQ135_THRESHOLD_1`), maka udara dikategorikan sebagai "BERSIH", dan jika di atasnya, dikategorikan sebagai "KOTOR". Informasi ini ditampilkan di Serial Monitor bersama dengan nilai gas dalam satuan analog (dalam hal ini dianggap mewakili PPM). Terakhir, nilai sensor dikirim ke aplikasi Blynk menggunakan `Blynk.virtualWrite()` ke virtual pin yang telah ditentukan (`BLYNK_VPIN`) sehingga dapat dipantau secara real-time melalui smartphone atau perangkat lainnya yang terhubung ke Blynk.

Segmen 13. Void Loop MQ7

```
// mq 7
Blynk.run();
unsigned long now = millis();
int raw = analogRead(pinSensorMQ7);
```

```
sensor_val = 0.9 * sensor_val + 0.1 * raw;
if (phase == 1 && now - phaseStart >= 60000) {
    phase = 0;
    phaseStart = now;
    digitalWrite(pinHeater, LOW);
}
if (phase == 0 && now - phaseStart >= 90000) {
    last_CO_ppm = getCOppm(sensor_val);
    phase = 1;
    phaseStart = now;
    digitalWrite(pinHeater, HIGH);
    // Kirim ke Blynk
    Blynk.virtualWrite(V0, last_CO_ppm); // CO ke
Virtual Pin V0
}
if (now - lastPrint >= 2000) {
    lastPrint = now;
    Serial.print("Raw:           ");
Serial.print(sensor_val);
    Serial.print("           |           CO:           ");
Serial.print(last_CO_ppm); Serial.println(" ppm");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("CO: ");
    lcd.print(last_CO_ppm, 1);
    lcd.print(" ppm");
    lcd.setCursor(0, 1);
    lcd.print(phase == 1 ? "Heating..." :
"Measuring...");
}
```

Segmen program tersebut merupakan bagian dari program pemantauan karbon monoksida (CO) menggunakan sensor gas MQ-7 yang dijalankan pada mikrokontroler dan terhubung ke platform Blynk serta layar LCD. Fungsi

Blynk.run()); menjaga koneksi aktif antara perangkat dan server Blynk. Waktu saat ini disimpan dalam variabel now menggunakan millis() untuk membantu mengelola fase pemanasan dan pengukuran sensor. Pembacaan data sensor dilakukan melalui analogRead(pinSensorMQ7); dan hasilnya diolah menggunakan filter eksponensial bergerak ($sensor_val = 0.9 * sensor_val + 0.1 * raw$) guna meredam fluktuasi dan mendapatkan nilai yang lebih stabil.

Sensor MQ-7 bekerja dalam dua fase: fase pemanasan ($phase == 1$) dan fase pengukuran ($phase == 0$). Saat dalam fase pemanasan selama 60 detik ($now - phaseStart \geq 60000$), pemanas diaktifkan ($digitalWrite(pinHeater, HIGH)$), dan setelah waktu tercapai, sistem masuk ke fase pengukuran dengan mematikan pemanas ($digitalWrite(pinHeater, LOW)$) dan memulai timer baru. Setelah fase pengukuran berlangsung selama 90 detik, sistem menghitung konsentrasi CO menggunakan fungsi $getCOppm(sensor_val)$, memperbarui nilai $last_CO_ppm$, kembali ke fase pemanasan, dan mengirimkan data CO tersebut ke Blynk pada virtual pin V0 dengan $Blynk.virtualWrite(V0, last_CO_ppm)$;

Setiap dua detik ($now - lastPrint \geq 2000$), data mentah dan hasil perhitungan CO ditampilkan di Serial Monitor untuk debugging serta ditampilkan di LCD. Baris pertama layar LCD menampilkan kadar CO dalam ppm dengan satu angka desimal, sedangkan baris kedua menunjukkan status fase sensor, yaitu "Heating..." saat pemanasan atau "Measuring..." saat pengukuran. Rangkaian logika ini secara efisien mengelola siklus kerja sensor MQ-7 yang memang memerlukan pengaturan fase panas-dingin agar bisa memberikan pembacaan CO yang akurat.

Segmen 14. Void Loop Sensor Debu

```
// GP2Y1010AU0F
Blynk.run();
float dustDensity = readDustSensor();
// Tampilkan di Serial Monitor
Serial.print("Konsentrasi Debu: ");
Serial.print(dustDensity);
Serial.println(" ug/m3");
```

```
// Tampilkan di LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Debu:");
lcd.setCursor(6, 0);
    lcd.print(dustDensity, 1); // Satu angka di
belakang koma
    lcd.print(" ug/m3");
// Kirim ke Blynk
Blynk.virtualWrite(BLYNK_DUST_PIN, dustDensity);
delay(2000);
}
float readDustSensor() {
    const int samplingTime = 280;
    const int deltaTime = 40;
    const int sleepTime = 9680;
    digitalWrite(pinLED, LOW);
    delayMicroseconds(samplingTime);
    int voMeasured = analogRead(pinVo);
    delayMicroseconds(deltaTime);
    digitalWrite(pinLED, HIGH);
    delayMicroseconds(sleepTime);
    float voltage = voMeasured * (3.3 / 4095.0); //
ADC ESP32
    float dustDensity = (voltage - 0.1) * 1000.0 /
0.5; // µg/m3
    if (dustDensity < 0) dustDensity = 0;
    return dustDensity;
}
```

Segmen program tersebut digunakan untuk membaca konsentrasi debu di udara menggunakan sensor GP2Y1010AU0F, menampilkannya di Serial Monitor dan LCD, serta mengirimkannya ke platform Blynk. Proses dimulai dengan `Blynk.run()`; untuk menjaga koneksi ke server Blynk tetap aktif.

Selanjutnya, fungsi `readDustSensor()` dipanggil untuk membaca nilai sensor debu dan hasilnya disimpan ke dalam variabel `dustDensity`.

Di dalam `readDustSensor()`, sensor bekerja dengan metode optik yang memanfaatkan sinar LED dan deteksi pantulan partikel. Proses pembacaan dimulai dengan mengaktifkan LED inframerah sensor melalui `digitalWrite(pinLED, LOW)` selama waktu `samplingTime` (280 mikrodetik). Setelah itu, nilai tegangan output dibaca melalui pin analog (`analogRead(pinVo)`), yang mencerminkan jumlah partikel debu di udara. Setelah membaca, LED dimatikan kembali dan sistem diberi jeda dengan `delayMicroseconds(sleepTime)` untuk menghindari interferensi antar pembacaan.

Nilai pembacaan analog (dalam rentang 0–4095 untuk ESP32) dikonversi ke tegangan dengan rumus $\text{voltage} = \text{voMeasured} * (3.3 / 4095.0)$. Tegangan ini kemudian dikonversi menjadi nilai konsentrasi debu (dalam mikrogram per meter kubik atau $\mu\text{g}/\text{m}^3$) menggunakan rumus $\text{dustDensity} = (\text{voltage} - 0.1) * 1000.0 / 0.5$. Bila hasilnya negatif, maka disesuaikan menjadi nol karena tidak masuk akal secara fisik.

Nilai `dustDensity` yang diperoleh kemudian ditampilkan pada Serial Monitor dan LCD. Di LCD, baris pertama digunakan untuk label "Debu:", sedangkan hasil ditampilkan di sampingnya dengan satu angka di belakang koma. Akhirnya, data juga dikirim ke aplikasi Blynk menggunakan `Blynk.virtualWrite(BLYNK_DUST_PIN, dustDensity)`; agar dapat dipantau secara jarak jauh melalui smartphone atau perangkat lain. Proses ini dijalankan setiap 2 detik (`delay(2000)`;) sehingga pembaruan data terjadi secara berkala dan real-time.

4.2 Skenario Pengujian

Skenario pengujian ini menggunakan metode pengujian blakbox, yaitu pengujian yang difokuskan pada keluaran sistem berdasarkan masukan tertentu, tanpa memperhatikan struktur internal kode program. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem yang dibangun dapat berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah dirancang sebelumnya. Pengujian dilakukan terhadap seluruh fitur utama, seperti

pemantauan kualitas udara, pembacaan sensor lingkungan, pengiriman data ke platform IoT, dan respon sistem terhadap kondisi udara yang tidak sehat. Masing-masing komponen diuji berdasarkan input yang diberikan untuk memastikan respons sistem terhadap masukan dari sensor dan aktuator dapat ditampilkan dengan benar melalui LCD maupun platform IoT yaitu Blynk.

Setiap data dari sensor diumpankan ke mikrokontroler ESP32, lalu sistem akan memberikan keluaran seperti informasi data sensor (nilai gas, suhu, debu, cahaya, dan lainnya) serta aksi otomatis berupa aktivasi aktuator (servo, atomizer, dll). Berikut adalah tabel skenario pengujian sistem secara rinci.

4.2.1 Pengujian pada pemantauan kualitas udara dan penanganan udara kotor berbasis IoT

Pada skenario pengujian ini pengujian difokuskan pada fungsi-fungsi utama sistem, seperti pemantauan kualitas udara (gas berbahaya, suhu, kelembaban, dan partikel debu), pengendalian aktuator secara otomatis (kipas dan water atomizer), serta tampilan data melalui LCD dan pengiriman data ke platform IoT seperti Blynk. Setiap fitur diuji dengan memberikan input kondisi lingkungan yang berbeda-beda, kemudian diamati output atau respons sistem baik melalui LCD, dashboard IoT, maupun perangkat aktuator.

Tabel 4. 2 Pengujian Pada Pemantauan Kualitas Udara dan Penanganan Udara Kotor Berbasis IoT

No	Nama Proses Data	Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
1	Monitoring Kualitas Udara (Gas Berbahaya)	Asap rokok / polusi / bau menyengat	Nilai MQ-135 dan MQ-7 meningkat	LCD dan Blynk menampilkan nilai ppm gas tinggi	Sensor MQ-135 dan MQ-7 bekerja baik mendeteksi polusi
2	Monitoring Suhu & Kelembaban	Udara panas dan lembab	DHT11 membaca suhu >30°C dan kelembaban >70%	Data suhu dan kelembaban tampil di LCD dan Blynk	Sensor DHT11 bekerja normal

Tabel Lanjutan 4. 2 Pengujian Pada Pemantauan Kualitas Udara dan Penanganan Udara Kotor Berbasis IoT

3	Monitoring Partikel Debu	Debu dari lingkungan /kipas kotor	Nilai sensor GP2Y1010 AU0F meningkat	PM meningkat di LCD dan IoT	Sensor debu mampu mendeteksi partikel dengan baik
4	Aktivasi Kipas Ventilasi Otomatis	Nilai gas atau debu tinggi	Relay mengaktifkan kipas	Kipas menyala otomatis saat kualitas udara buruk	Kendali kipas berfungsi otomatis
5	Aktivasi Water Atomizer Otomatis	Nilai gas tinggi + suhu panas	Relay mengaktifkan atomizer	Atomizer menyala dan mengeluarkan uap air	Penanganan udara berjalan otomatis
6	Tampilan Data pada LCD	Semua sensor terhubung	LCD menampilkan suhu, kelembaban, gas, debu	Tampilan bergantian sesuai logika	LCD bekerja sesuai fungsinya
7	Pengiriman Data ke IoT	Sensor aktif dan terhubung WiFi	Data tampil di dashboard Blynk/ThingSpeak	Semua data berhasil dikirim	Fitur IoT berhasil digunakan
8	Penanganan Otomatis Gabungan	Kondisi buruk secara keseluruhan	Kipas dan atomizer menyala bersama	Penanganan ganda aktif otomatis	Sistem tanggap terhadap kondisi udara buruk

4.2.2 Pengujian Solar System

Pada skenario pengujian sistem tenaga surya ini bertujuan untuk memastikan seluruh rangkaian dapat berfungsi secara optimal dalam menjalankan monitoring dan distribusi daya dari panel surya. Komponen-komponen utama yang diuji meliputi panel surya, baterai, modul step-up (boost converter), modul relay, LCD I2C, serta sensor LDR, sensor arus INA219, dan sensor gyroscope MPU6050. Seperti tabel dibawah ini :

Tabel 4. 3 Pengujian *Solar System*

No	Nama Proses	Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
1	Pengisian Daya Panel Surya	Panel surya terkena sinar	Tegangan dan arus terukur oleh INA219,	INA219 menunjukkan tegangan dan arus, LED	Berhasil

Tabel Lanjutan 4. 3 Pengujian *Solar System*

		matahari langsung	baterai mengisi daya	indikator charging menyala	
2	Pengisian Saat Cahaya Rendah	Panel surya dalam kondisi mendung / cahaya minim	Tidak ada arus masuk ke baterai atau sangat kecil	INA219 mendeteksi arus sangat kecil / nol	Berhasil
3	Deteksi Intensitas Cahaya oleh LDR	Intensitas cahaya bervariasi (ditutup sebagian / diarahkan cahaya)	Nilai ADC LDR berubah sesuai intensitas	Nilai LDR berubah-ubah pada serial monitor/LCD	Berhasil
4	Pelacakan Cahaya (Solar Tracker)	Arah cahaya dipindahkan ke kiri/kanan LDR	Servo bergerak menyesuaikan arah terang	Servo merespons arah cahaya dari LDR	Berhasil
5	Deteksi Sudut Panel (Gyroscope)	Panel dimiringkan secara manual	MPU6050 mendeteksi perubahan sudut (pitch/roll)	Perubahan sudut terpantau di serial monitor	Berhasil
6	Monitoring Tegangan dan Arus	Sistem dalam keadaan aktif siang hari	INA219 membaca tegangan dan arus dengan benar	Data tegangan dan arus tampil di LCD atau serial monitor	Berhasil
7	Distribusi Daya ke Sistem	Beban sistem diaktifkan (ESP32, Arduino, Servo, Sensor)	Semua komponen menyala stabil melalui modul step-up	Sistem bekerja stabil, tidak restart atau drop tegangan	Berhasil

4.3 Pengujian

Pengujian merupakan proses evaluasi menyeluruh terhadap kinerja dan fungsi keseluruhan sistem yang telah dirancang dan dibangun. Dalam konteks ini, pengujian sistem dilakukan untuk memastikan bahwa semua komponen baik perangkat keras (hardware) maupun perangkat lunak (software) dapat

bekerja sesuai dengan fungsinya secara terintegrasi, handal, dan konsisten. Sistem yang di uji adalah sistem pemantauan kualitas udara berbasis ESP32, yang mengumpulkan data dari berbagai sensor seperti MQ-135, MQ-7, DHT11, GP2Y1010AU0F, INA219, LDR, dan MPU6050, serta menjalankan penanganan otomatis udara kotor menggunakan aktuator seperti servo motor, kipas melalui relay, dan Grove Water Atomizer v1.0. Sistem ini ditenagai oleh panel surya, dan dilengkapi dengan tampilan data real-time di LCD serta pemantauan jarak jauh melalui platform Blynk IoT.

4.3.1 Cara Kerja Sistem / *manual book*

Cara kerja sistem ini membantu merakit dan mengoperasikan sistem pemantauan dan penanganan kualitas udara berbasis IoT yang hemat energi.

- a. Monitoring Kualitas Udara
 1. Sensor MQ-135 dan MQ-7 mendeteksi gas – gas berbahaya seperti Co, NH₃ dan asap rokok.
 2. Sensor Sharp GP2Y1010AU0F mendeteksi partikel debu di udara.
 3. Sensor DHT11 mengukur suhu dan kelembapan
 4. Sensor LDR mendeteksi intensitas cahaya sebagai referensi waktu siang/malan
 5. Sensor INA219 memantau tegangan dan arus dari sistem tenaga surya dan baterai.
 6. Sesnor MPU6050 Gyroscope memantau orientasi atau getaran sistem
- b. Penanganan Udara Kotor

Ketika kualitas udara melebihi ambang batas (misalnya: gas CO > 50 ppm, debu > 0.1 mg/m³), maka:

 1. Relay aktif untuk menghidupkan kipas penyaring udara.
 2. Water Atomizer aktif untuk meningkatkan kelembapan dan membantu penanganan udara kotor.
- c. Sumber Daya Tenaga Surya
 1. Panel surya menghasilkan listrik yang disimpan kedalam baterai Li-ion
 2. Jika listrik PLN mati, sistem akan tetap menyala karena mendapatkan daya dari baterai.

3. INA219 memantau performa solar panel dan baterai untuk efisiensi daya.
- d. Tampilan dan Kendali Lokal
LCD 16x2 12c menampilkan secara real-time data seperti :
 1. Kadar gas dan Polutan
 2. Suhu dan kelembapan
 3. Status sistem (normal / bahaya)
 4. Status baterai
- e. Koneksi ESP32 ke Blynk
ESP32 terkoneksi ke wifi lokal dan blynk digunakan untuk :
 1. Memantau data sensor secara real-time melalui smarphone / laptop.
 2. Menerima notifikasi jika kualitas udara buruk
 3. Mengendalikan sistem secara manual semisal mematikan fan dc secara remote
- f. Data yang dikirim ke Blynk
Virtual pin () digunakan untuk mengirim :
 1. CO level (ppm) → V0
 2. Suhu (°C) → V1
 3. Kelembaban (%) → V2
 4. Debu (mg/m³) → V3
 5. Tegangan baterai → V4
 6. Status sistem (Normal/Bahaya) → V5
 7. Status relay/kipas/atomizer → V6, V7,
- g. Notifikasi Blynk
Jika sensor mendeteksi CO melebihi batas → kirim notifikasi "Kualitas Udara Buruk". Jika Suhu terlalu tinggi → kirim notifikasi "Suhu Tinggi"
- h. Kendali jarak jauh via Blynk
Tersedia tombol di blynk dashboard untuk :
 1. menyalakan / mematikan fan dc
 2. reser sistem
 3. monitoring log data sensor

4.3.2 Pengujian Sensor

Pengujian sensor adalah proses evaluasi terhadap performa dan keakuratan sensor dalam mendeteksi dan mengukur parameter tertentu sesuai dengan fungsi masing-masing sensor. Dalam sistem ini, pengujian dilakukan untuk memastikan bahwa seluruh sensor yang digunakan dapat memberikan data yang valid, akurat, dan real-time untuk menunjang pemantauan kualitas udara secara tepat serta merespons kondisi lingkungan dengan penanganan otomatis.

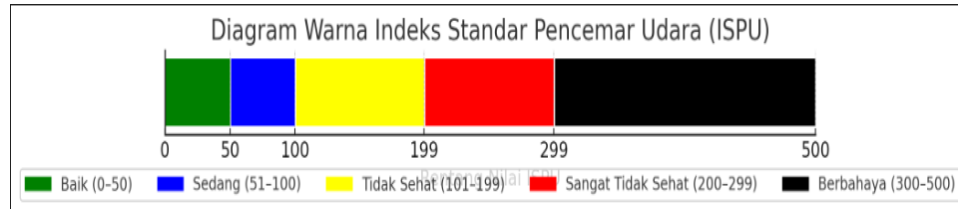
a. Rentang Indeks Standart Pencemaran udara

Indeks standart pemncemaran udara adalah indeks yang menunjukkan tingkat pencemaran udara di suatu wilayah dalam waktu tertentu. Indeks ini membantu masyarakat memahami seberapa bersih atau tercemarnya udara serta dampaknya terhadap kesehatan. Lihat tabel 4. 4 dibawah ini :

Tabel 4. 4 Rentang Indeks Standart Pencemaran udara

Kategori	Rentang	Penjelasan
Baik	0-50	Tingkat kualitas udara yang tidak memberikan efek bagi kesehatan manusia atau hewan dan tidak berpengaruh pada tumbuhan, bangunan atau estetika
Sedang	51-100	Tingkat kualitas udara yang tidak berpengaruh pada kesehatan manusia maupun hewan tetapi berpengaruh pada tumbuhan yang sensitif dan nilai estetika.
Tidak Sehat	101-199	Tingkat kualitas udara yang bersifat merugikan pada manusia ataupun kelompok hewan yang sensitif atau bisa menimbulkan kerusakan pada tumbuhan ataupun nilai estetika
Sangat tidak sehat	200-299	Tingkat kualitas udara yang dapat merugikan kesehatan pada sejumlah segmen populasi yang terpapar
Berbahaya	300- lebih	Tingkat kualitas udara berbahaya yang secara umum dapat merugikan kesehatan yang serius

Berikut adalah diagram warna Indeks standart pencemaran udara yang menunjukkan kategori kualitas udara berdasarkan rentang nilainya, dari "Baik" (Hijau) hingga "Berbahaya (Hitam)"

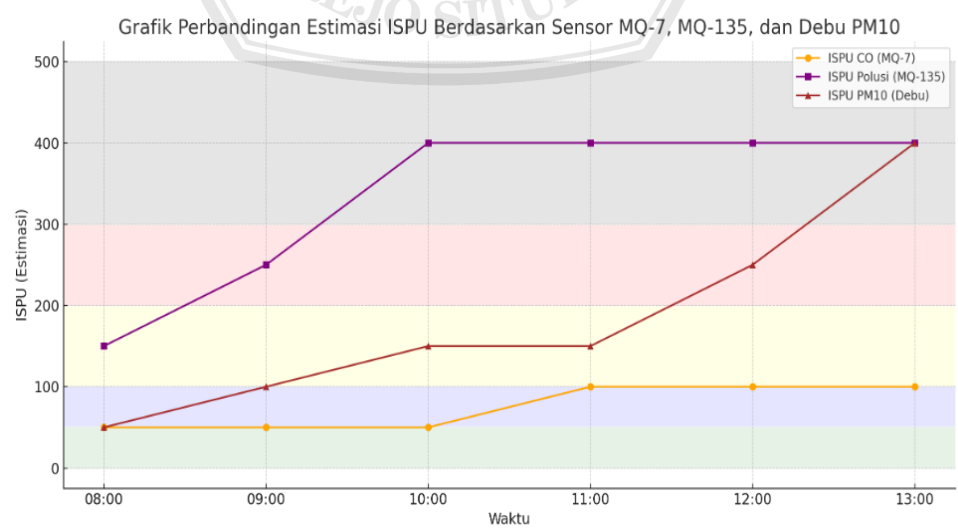


Gambar 4. 11 Diagram Warna Indeks Standart Pencemaran Udara

Grafik perbandingan data sensor MQ135, MQ7 dan PM10 berdasarkan data pembacaan dari beberapa sensor :

- MQ-7 (CO): Mengukur karbon monoksida.
- MQ-135: Mengukur gas polutan umum (NH₃, NO_x, benzena, dll).
- PM10: Mengukur partikel debu di udara (µg/m³).

Setiap kurva memperlihatkan estimasi Indeks standart pencemaran udara berdasarkan hasil pembacaan sensor dari pukul 08.00 sampai 13.00, dengan latar warna menunjukkan kategori kualitas udara dari Baik hingga Berbahaya. Sensor DHT11 tidak ditampilkan sebagai grafik Indeks standart pencemaran udara karena hanya mengukur suhu dan kelembaban, tetapi bisa digunakan hanya untuk korelasi terhadap kenyamanan udara. Berikut grafik indeks standart pencemaran udara dibawah ini :



Gambar 4. 12 Grafik Indeks Standart Pencemaran Udara

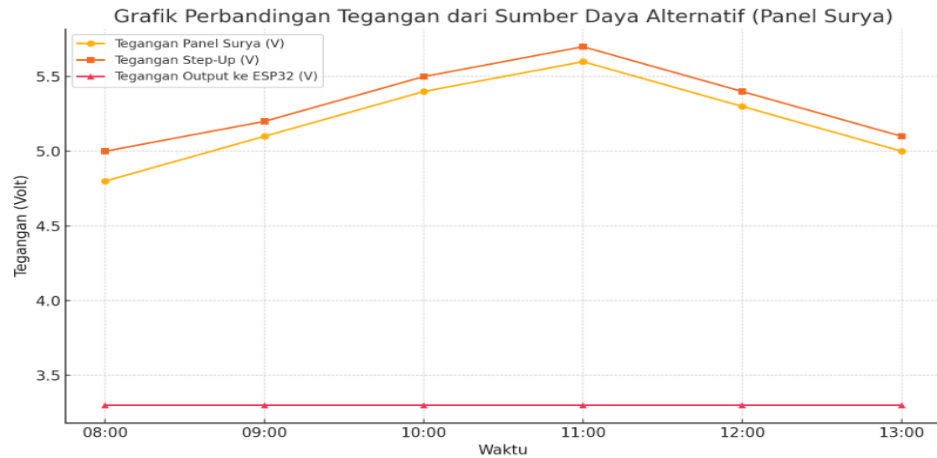
b. Rentang Indeks Daya dan Tegangan

Indeks Sumber Daya dan Tegangan adalah suatu ukuran atau skala yang digunakan untuk menilai seberapa besar dan stabil daya listrik yang tersedia dari sistem tenaga surya, terutama ketika sistem harus mengambil alih fungsi listrik konvensional (PLN) yang sedang padam. Indeks sumber daya dan tegangan seperti tabel 4. 5 di bawah ini:

Tabel 4. 5 Indeks Sumber Daya dan Tegangan

Indeks	Tegangan (V)	Daya (W)	Status Sistem	Tindakan yang Direkomendasikan
81 – 100	12.5 – 13.0 V	> 8 W	Sangat Baik – Semua sistem dapat berjalan penuh.	Semua perangkat dapat dinyalakan normal.
61 – 80	12.0 – 12.4 V	5 – 8 W	Baik – Sistem stabil.	Prioritaskan komponen penting.
41 – 60	11.5 – 11.9 V	3 – 5 W	Cukup – Beban harus dikurangi.	Matikan komponen non-prioritas (misal LCD).
21 – 40	11.0 – 11.4 V	1 – 3 W	Buruk – Daya tidak cukup untuk semua komponen.	Kirim peringatan ke pengguna lewat Blynk.
0 – 20	< 11.0 V	< 1 W	Sangat Buruk – Baterai hampir habis.	Sistem masuk mode hemat daya atau shutdown otomatis.

Tegangan dari panel surya meningkat hingga sekitar pukul 11: 00 dan kemudian menurun, menggambarkan intensitas cahaya matahari yang bervariasi. Tegangan keluaran dari modul step -up yang menstabilkan tegangan untuk sistem. Tegangan output ke esp32 tetap konstan di 3.3v sampai 5v, sesuai kebutuhan mikrokontroler. Berikut grafik tegangan seperti gambar 4. 13 dibawah ini :



Gambar 4. 13 Grafik Perbandingan

c. Persyaratan Kimia

Persyaratan kimia udara *ambien* (lingkungan) atau udara dalam ruangan ditentukan untuk menjamin bahwa kualitas udara tetap aman bagi kesehatan manusia. Berikut adalah penjelasan umum tentang persyaratan kimia udara berdasarkan parameter utama yang diukur oleh sensor seperti MQ-135, MQ-7, dan sensor debu PM10. Berikut tabel 4. 6 persyaratan kimia seperti dibawah ini :

Tabel 4. 6 Persyaratan Kimia

No	Jenis Parameter	Satuan	Kadar maksimal yg dipersyaratkan	Keterangan
1	Sulfur dioksida (SO ₂)	ppm	0,1	24 jam
2	Nitrogen dioksida (NO ₂)	ppm	0,04	24 jam
3	Carbon monosikda (CO)	ppm	9,00	8 jam
4	Carbondioksida (CO ₂)	ppm	1000	8 jam
5	Timbal (Pb)	ppm	1,5	15 menit
6	Asbes	ppm	5	Panjang serat 5μ
7	Formaldehid (HCHO)	ppm	0,1	30 menit

d. Hasil Pengujian Alat Monitoring dan penanganan udara kotor

Hasil pembacaan sensor untuk sistem monitoring kualitas udara dan penanganan udara kotor adalah sebagai berikut :

Tabel 4. 7 Hasil Pengujian Alat Monitoring dan penanganan udara kotor

No	Waktu	Suhu (°C)	Kelembaban (%)	CO (MQ-7) (ppm)	Polusi Udara (MQ-135) (ppm)	Debu (mg/m ³)	Kipas Aktif	Humidifier Aktif	Status Udara
1	08.00	29,1	55	12	180	0,18	Tidak	Tidak	Baik
2	16.48	30,5	49	25	220	0,32	Ya	Tidak	Cukup Bersih
3	10.00	31,7	45	43	315	0,45	Ya	Tidak	Tercemar Ringan
4	11.00	32	39	56	410	0,62	Ya	Ya	Tercemar Sedang
5	12.00	33,4	35	78	500	0,85	Ya	Ya	Tercemar Berat
6	13.00	34,1	32	82	540	0,95	Ya	Ya	Bahaya

Sistem ini menggunakan beberapa sensor untuk memantau kualitas udara dalam ruangan secara menyeluruh. Sensor MQ-7 digunakan untuk mendeteksi karbon monoksida (CO) dengan output dalam satuan ppm berdasarkan hasil kalibrasi. Sensor MQ-135 berfungsi untuk mengukur total gas polutan di udara, seperti amonia, benzena, dan nitrogen dioksida, juga dengan satuan ppm. Sementara itu, sensor debu GP2Y1010AU0F digunakan untuk mengestimasi kepadatan partikel debu (PM10) di udara. Untuk menjaga kualitas udara, kipas akan aktif secara otomatis jika salah satu indikator (CO, polusi, atau debu) melebihi ambang batas yang telah ditentukan. Selain itu, humidifier akan menyala bila kelembaban udara turun di bawah 40%. Berdasarkan kombinasi seluruh parameter tersebut, sistem akan memberikan status kualitas udara secara umum untuk

menunjukkan apakah udara dalam ruangan berada dalam kondisi baik, sedang, atau berbahaya.

e. Hasil Pengujian Solar System

Pengujian sistem tenaga surya dilakukan untuk mengetahui kinerja panel surya, baterai, modul pengatur tegangan, serta beban dalam berbagai kondisi cahaya. Pengujian dilakukan selama beberapa hari dengan kondisi cuaca yang berbeda untuk memastikan stabilitas dan efektivitas sistem.

1. Pengujian Tegangan dan Arus Panel Surya

Tabel 4. 8 Pengujian Tegangan dan Arus Panel Surya

waktu	Intensitas Cahaya (Lux)	Tegangan Panel Surya (V)	Arus Panel Surya (A)	Daya (Waatt)
08:00	13.000	5.1	0.32	1.63
10:00	38.000	6.2	0.40	2.48
12:00	62.000	6.9	0.45	3.11
14:00	55.000	6.4	0.41	2.62
16:00	28.000	5.8	0.35	2.03
18:00	5.000	4.2	0.20	0.84

Tegangan dan arus tertinggi terjadi pada pukul 12.00 saat intensitas cahaya maksimal. Artinya, sistem bekerja optimal saat sinar matahari cukup terang. Saat sore atau cuaca mendung, daya yang dihasilkan turun signifikan.

2. Pengujian Pengisian Baterai

Tabel 4. 9 Pengujian Pengisian Baterai

waktu	Tegangan Baterai Sebelum (V)	Tegangan Setelah (V)	Keterangan
Hari 1	11.8	12.3	Pengisian Stabil
Hari 2	11.6	12.4	Pengisian Optimal

Tabel Lanjutan 4.9 Pengujian Pengisian Baterai

Hari 3	11.9	12.2	Sedikit mendung
Hari 4	11.7	12.0	Cuaca Mendung ringan

Pengisian baterai berjalan dengan baik dan berada dalam batas aman. Pada hari mendung, tegangan naik tidak terlalu tinggi tetapi masih dapat mengisi daya.

3. Pengujian Beban

Tabel 4. 10 Pengujian Beban

Komponen Beban	Konsumsi Daya (Watt)	Waktu Menyala	Status
ESP32	0.6	24 jam	Stabil
Sensor	0.3	24 jam	Stabil
LCD 16x2 c	0.2	24 jam	Stabil
Fan DC	2.0	5 jam siang	Normal

Beban mampu berjalan dengan baik selama suplai dari baterai mencukupi. Saat malam hari, sistem tetap menyala karena cadangan daya baterai.

4. Pengujian Ketahanan saat Listrik Padam

- Saat listrik padam, sistem tetap berfungsi normal hingga 8 – 10 jam tergantung pemakaian beban.
- Lcd, sensor dan mikrokontroler tetap aktif tanpa gangguan
- Sistem *auto switch* (relay) berhasil mengalihkan suplai dari panel / baterai saat listrik utama padam.

f. Perbandingan Pembacaan Sensor IoT dengan alat konvensional

Perbandingan antara alat monitoring kualitas udara berbasis *Internet of Things* dengan alat konvensional secara sistematis dan dalam bentuk tabel adalah sebagai berikut :

Tabel 4. 11 Perbandingan Alat IoT dengan alat konvensional

Aspek	Alat IoT	Alat konvensional
Jenis mikrokontroler	ESP32 (dengan koneksi Wi-Fi & IoT)	Tidak menggunakan mikrokontroler (manual/analog)

Tabel Lanjutan 4. 11 Perbandingan Alat IoT dengan Alat Konvensional

Sensor yang digunakan	DHT11, MQ-7, MQ-135, GP2Y1010AU0F, <i>Humidifier</i>	<i>Termometer</i> , <i>hygrometer</i> , detektor gas portabel
Pemantauan data	Bisa terhubung ke <i>cloud</i> untuk <i>logging</i>	Tidak menyimpan data secara otomatis
Repons otomatis	Kipas dan <i>humidifer</i> aktif otomatis saat kondisi tidak baik	Harus diaktifkan secara manual oleh pengguna
Akurasi dan kompleksitas	Multisensor, akurasi baik indikator	Umumnya hanya satu parameter, seperti suhu atau CO
skalabilitas	Dapat dikembangkan untuk sistem otomatis rumah pintar	Tidak bisa dikembangkan atau diintegrasikan ke sistem lain
konektivitas	Terhubung dengan internet	Tidak memiliki konektivitas
Sumber Energi	Panel Surya + baterai	PLN (terbatas pada jaringan)
Ketahan saat listrik padam	Tetap aktif hingga 10 jam	Tidak dapat digunakan
Efisiensi Energi	Hemat daya, ramah lingkungan	Tidak hemat, bergantung pada listrik konvensional
biaya	Relatif murah dan fleksibel karena berbasis <i>open source</i>	Bisa mahal (alat portable industri)
pemeliharaan	Perlu kalibrasi & <i>update</i> perangkat lunak	Perawatan rutin dan penggantian alat ukur

Perbandingan pembacaan sensor antara alat IoT dengan alat konvensional adalah sebagai berikut :

Tabel 4. 12 Perbandingan pembacaan sensor antara alat IoT dengan alat konvensional

Parameter	Alat IoT	Alat konvensional
Suhu (°C)	DHT11/DHT22, otomatis terbaca dan ditampilkan di LCD/IoT	Termometer analog/digital, dibaca manual
Kelembapan (%)	DHT11, ditampilkan otomatis dan bisa di - <i>log</i>	Hygrometer manual, perlu pembacaan langsung

Tabel Lanjutan 2.2 Perbandingan pembacaan sensor antara alat IoT dengan alat konvensional

Gas Karbon Monoksida (CO)	MQ7, hasil terbaca dalam ppm dan dikirim ke server IoT	CO meter digital/analog, dibaca manual, tidak otomatis
Polusi gas umum	MQ135, membaca polutan seperti amonia, benzena, dll	Tidak tersedia pada alat konvensional umum
Kepadatan debu	<i>Shart GP2Y1010AU0F</i> , memberikan data mg/m ³ secara <i>real-time</i>	Alat ukur partikel khusus (mahal, tidak portabel)
Data <i>Real-Time</i>	Ditampilkan di LCD dan dikirim ke <i>cloud</i> (<i>Blynk/ThingSpeak</i>)	Tidak ada, hanya bisa dilihat langsung saat itu saja
<i>Logging</i> & Riwayat	Bisa disimpan otomatis di <i>database</i> IoT	Tidak ada riwayat, perlu pencatatan manual
<i>Alarm</i> Otomatis	Kipas atau <i>humidifier</i> menyala otomatis berdasarkan data	Tidak ada, pengguna harus merespons manual

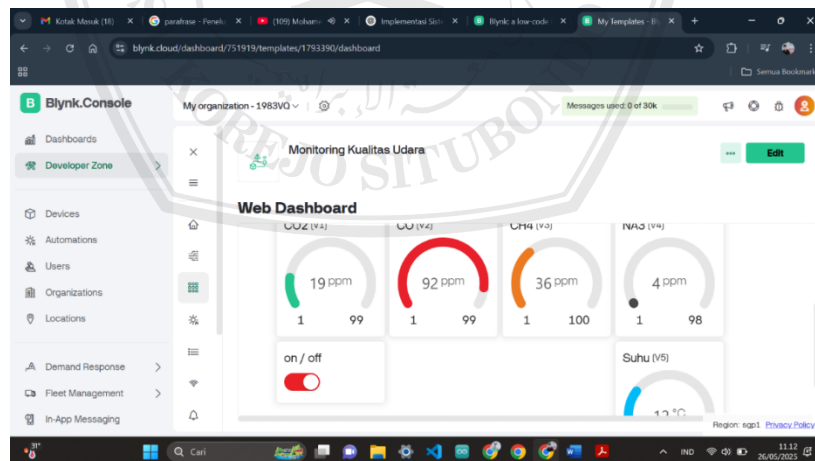
Secara umum alat berbasis ESP32 dan IoT yang dirancang memiliki kemampuan pembacaan sensor yang lebih lengkap, otomatis, dan *real-time* dibandingkan alat konvensional. Setiap data yang diperoleh dari sensor seperti MQ-7, MQ-135, DHT11, dan sensor debu dapat ditampilkan langsung di LCD, dikirim ke server IoT, serta digunakan sebagai pemicu otomatis untuk kipas atau *humidifier*. Sementara itu, alat konvensional hanya memberikan data terbatas, memerlukan pembacaan manual, dan tidak mendukung integrasi data ataupun pemantauan jarak jauh. Hal ini menjadikan sistem Anda lebih unggul dalam hal efisiensi, kecepatan respons, dan skalabilitas sistem monitoring kualitas udara.

4.3.3 Hasil Keseluruhan Pengujian Sistem

a. Implementasi Perangkat Lunak

Pada tahap ini merupakan pembuatan tampilan monitoring kualitas udara dalam ruangan yang akan menampilkan hasil bacaan sensor dan tampilan berbentuk gauge meter. aplikasi Blynk yang digunakan untuk melakukan pemantauan kualitas udara secara real-time menggunakan ESP32 sebagai mikrokontroler utama. Sistem ini mengukur beberapa parameter kualitas

udara dengan memanfaatkan berbagai sensor gas dan suhu, yang hasil pembacaannya dikirimkan ke platform Blynk melalui koneksi Wi-Fi. Sistem monitoring kualitas udara berbasis IoT ini dirancang menggunakan mikrokontroler ESP32 yang terhubung dengan berbagai sensor gas seperti MQ-135 untuk CO₂, MQ-7 untuk CO, dan sensor lain untuk mendeteksi CH₄ serta polutan tambahan (disebut NA3), serta sensor suhu untuk memantau kondisi lingkungan. Data yang diperoleh dari sensor dikirimkan secara real-time ke platform Blynk melalui koneksi Wi-Fi, lalu divisualisasikan dalam bentuk gauge pada dashboard. Setiap parameter memiliki indikator warna sebagai penanda tingkat bahaya, misalnya merah untuk CO yang berbahaya dan hijau untuk CO₂ yang masih aman. Selain memantau, sistem ini juga memungkinkan pengguna untuk mengontrol perangkat penanganan udara seperti kipas atau atomizer secara jarak jauh melalui fitur saklar digital di Blynk. Sistem ini sangat efektif digunakan untuk pemantauan kualitas udara dalam ruangan secara real-time, serta memberikan kemudahan akses dan kendali melalui smartphone atau komputer dari lokasi mana pun. Tampilan Implementasi perangkat lunak seperti gambar di bawah ini :



Gambar 4. 14 Tampilan Implementasi perangkat lunak

b. Implementasi Perangkat Keras

Implementasi perangkat keras terdiri dari keseluruhan pembuatan perangkat keras sistem. Merupakan sistem monitoring dan penanganan udara berbasis IoT dengan mikrokontroler ESP32 sebagai pusat kendali. Sistem ini dilengkapi dengan beberapa sensor untuk mendeteksi kondisi

lingkungan, seperti DHT11 yang berfungsi mengatur suhu dan kelembaban udara, sensor MQ-135 untuk mendeteksi gas-gas berbahaya seperti amonia, CO₂, dan asap, sensor MQ-7 untuk mendeteksi kadar karbon monoksida (CO), serta sensor debu Sharp GP2Y1010AU0F yang mengukur konsentrasi partikel debu di udara. Data dari sensor-sensor tersebut diproses oleh ESP32 dan ditampilkan melalui LCD 16x2 berbasis I2C. Selain itu, sistem ini juga menggunakan modul Grove – Water Atomization sebagai humidifier yang dikendalikan oleh ESP32 melalui 2-channel relay module, bersama dengan kipas DC yang juga diaktifkan saat kualitas udara menurun. Relay digunakan untuk mengontrol ON/OFF kipas dan humidifier berdasarkan ambang batas nilai sensor yang telah ditentukan. Seluruh sistem ini dapat dihubungkan ke platform IoT seperti Blynk untuk pemantauan jarak jauh secara real-time, sehingga memungkinkan pengguna mengetahui kondisi udara dalam ruangan sekaligus mengaktifkan tindakan penanganan secara otomatis untuk menjaga kualitas udara tetap bersih dan sehat. Implementasi perangkat keras seperti gambar dibawah ini :



Gambar 4. 15 Implementasi Perangkat Keras

4.4 Maintenance

Pemeliharaan atau *maintenance* merupakan aspek penting untuk menjaga agar sistem Monitoring Kualitas Udara dan Penanganan Udara Kotor Berbasis *Internet of Things* (IoT) dapat terus berfungsi dengan optimal dan akurat. Berikut ini adalah bentuk kegiatan pemeliharaan atau *maintenance* :

1. Pemeliharaan Perangkat Keras (Hardware)

Tabel 4. 13 Pemeliharaan Perangkat Keras (Hardware)

Komponen	Jadwal Pemeliharaan	Tindakan
Sensor MQ-135, MQ-7	1-2 bulan sekali	Membersihkan sensor dari debu, dan melakukan kalibrasi bila nilai sudah tidak akurat
Sensor debu GP2Y1010AU0F	1-2 bulan sekali	Membersihkan lensa optik sensor dengan udara bertekanan rendah
Sensor DHT11	3 bulan sekali	Memeriksa tingkat kelembapan dan suhu, dan mengganti sensor jika pembacaan sudah tidak stabil
Kipas / Air Purifier	1 bulan sekali	Membersihkan kipas dari kotoran dan debu agar tidak menghambat sirkulasi
Panel Surya	2 minggu sekali	Membersihkan permukaan panel dari debu dan kotoran agar efisiensi tetap tinggi
Baterai Lithium	6 bulan sekali	Memeriksa daya tahan, tegangan outpur dan memastikan tidak terjadi kebocoran daya
Kabel & Konektor	1 bulan sekali	Memeriksa sambungan dan soldering untuk menghindari atau putus koneksi

2. Pemeliharaan Perangkat Lunak (Software)

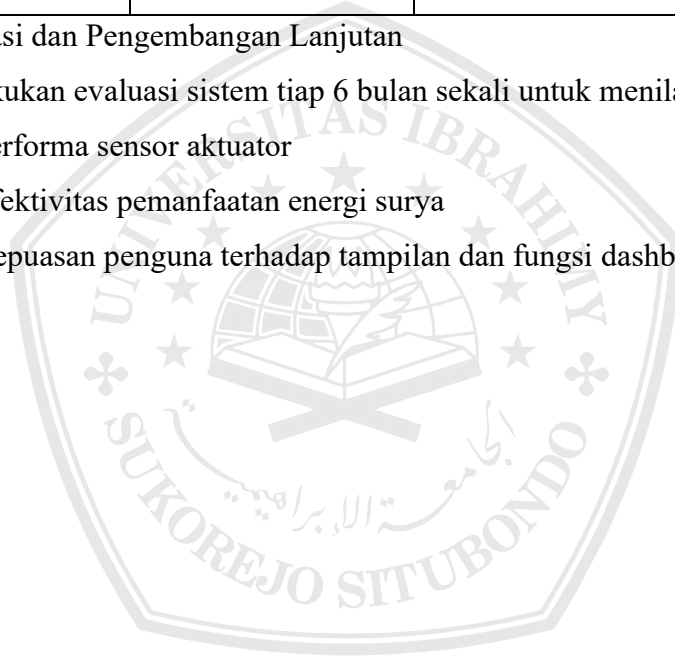
Tabel 4. 14 Pemeliharaan Perangkat Lunak (*Software*)

Komponen	Jadwal Pemeliharaan	Tindakan
Kode ESP32	Bila diperlukan	Update firmware jika terdapat perbaikan bug atau fitur baru.
Dashboard Blynk	3 bulan sekali	Cek kestabilan koneksi ke server, optimalkan UI jika perlu.
Data Logger / IoT Platform	Bulanan	Backup data secara berkala dan pastikan tidak ada kehilangan data.

3. Evaluasi dan Pengembangan Lanjutan

Melakukan evaluasi sistem tiap 6 bulan sekali untuk menilai :

- a. Performa sensor aktuator
- b. Efektivitas pemanfaatan energi surya
- c. Kepuasan pengguna terhadap tampilan dan fungsi dashboard



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut :

1. Sistem berhasil memonitor kualitas udara secara real-time menggunakan sensor MQ-135, MQ-7, dan GP2Y1010AU0F serta menampilkan data melalui LCD dan platform Blynk.
2. Penanganan udara kotor secara otomatis melalui kipas dan *air purifier* dapat diaktifkan saat kualitas udara melebihi ambang batas yang ditentukan.
3. Sistem mampu tetap beroperasi meskipun terjadi pemadaman listrik, karena penggunaan tenaga surya sebagai sumber energi alternatif terbukti efisien.
4. Implementasi IoT mempermudah pemantauan dan pengambilan tindakan terhadap kualitas udara secara jarak jauh dan cepat.
5. Sistem dapat diandalkan dalam meningkatkan kesadaran masyarakat terhadap pentingnya menjaga kualitas udara didalam ruangan maupun di luar ruangan, khususnya di kawasan perumahan seperti dawuhan *cluster* situbondo.

5.2 Saran

1. Sistem ini dapat dikembangkan lebih lanjut dengan menambahkan fitur prediksi polusi menggunakan *machine learning* agar mampu melakukan penanganan lebih proaktif.
2. Diperlukan penelitian lebih lanjut untuk skala ruangan yang lebih besar dan kondisi lingkungan yang lebih bervariasi untuk menguji kestabilan sistem secara menyeluruh.
3. Tampilan antarmuka pengguna di aplikasi *Blynk* dapat ditingkatkan dengan grafik historis dan dashboard visual yang lebih informatif.

4. Untuk efisiensi energi yang lebih tinggi, bisa dipertimbangkan penggunaan sensor dengan konsumsi daya rendah dan optimasi pengaturan sleep mode pada ESP32
5. Disarankan adanya sistem notifikasi tambahan seperti SMS atau email agar pengguna lebih cepat mengetahui kondisi kritis kualitas udara meskipun tidak sedang membuka aplikasi.



DAFTAR PUSTAKA

- [1] M. S. Novelan, "Sistem Monitoring Kualitas Udara Dalam Ruangan Menggunakan Mikrokontroler Dan Aplikasi Android," *Infotekjar J. Nas. Inform. Dan Teknol. Jar.*, Vol. 4, No. 2, Pp. 50-54, 2020, [Online] Available: <https://doi.org/10.30743/infotekjar.v4i2.2306>
- [2] World Health Organization., 2010, "Who Guidelines For Indoor Air Quality: Selected Pollutant", Copenagen Denmark. [Online]. Available: <https://bit.ly/217kiny>.
- [3] Centers For Disease Control And Prevention And U.S. Department Of Housing And Urban Development, "Chapter 2: Basic Principles Of Healthy Housing," *Heal. Hous. Ref. Man.*, Pp. 1–10, 2006, [Online]. Available: www.cdc.gov/healthyhomes/publications.html.
- [4] I. Buana And D. A. Harahap, "Asbestos, Radon Dan Polusi Udara Sebagai Faktor Resiko Kanker Paru Pada Perempuan Bukan Perokok," *Averrous J. Kedokt. Dan Kesehat. Malikussaleh*, Vol. 8, No. 1, Pp. 1–16, 2022
- [5] J. Susilo And F. Tunjungsari, "Resiko Gangguan Pernapasan Pada Industri Besi Baja," *Comphi J. Community Med. Public Heal. Indones. J.*, Vol. 3, No. 2, Pp. 110–121, 2022.
- [6] R. Pratama And L. Parinduri, "Penanggulangan Pemanasan Global," *Bul. Utama Tek.*, Vol. 15, No. 1, Pp. 91–95, 2019.
- [7] W. Setyawan, F. Hadary, And M. Saleh, "Perancangan Sistem Monitoring Kualitas Udara Di Daerah Rural Dengan Karakteristik Gambut Berbasis Teknologi Internet Of Things (Iot)," *J. Tek. Elektro Univ. Tanjungpura*, Vol. 2, No. 1.
- [8] I. Standar And P. Udara, "Index Kualitas Udara," No. 107, Pp. 1–14, 1997.
- [9] A. D. Ramadhani, N. Ningsih, A. Nurcahya, Dan N. Azizah, "Klasifikasi Dan Monitoring Kualitas Udara Dalam Ruangan Menggunakan Thingspeak," *J. Tek. Elektro Dan Komput. Triac*, Vol. 10, No. 1, Pp. 1–5, 2023, Doi: 10.21107/Triac.V10i1.17501.

- [10] N. A. Z. Putri, “Sistem Pendeteksi Kualitas Udara Dalam Ruang Berbasis Internet Of Things (Iot),” *Jurnal Ilmiah Teknologi Informasi*, Vol. 5, No. 1, Pp. 9–17, 2024.
- [11] A. Solechan, J. D. Susatyo, T. W. Ap, And F. Febryantahanuji, “Peluang Bisnis Pada Penerapan Industrial Internet Of Thing (Iiot),” *J. Publ. Ilmu Komput. Dan Multimed.*, Vol. 1, No. 3, Pp. 259–268, 2022.
- [12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, And M. Ayyash, “Internet Of Things: A Survey On Enabling Technologies, Protocols, And Applications,” *Ieee Commun. Surv. Tutorials*, Vol. 17, No. 4, Pp. 2347–2376, 2015, Doi: 10.1109/Comst.2015.2444095
- [13] S. N. Aini, D. N. Munahefi, A. S. Pramasdyahsari, And R. D. Setyowati, “Engineering Design Process Stem : Projek Miniatur Gazebo Joglo,” *Prism. Pros. Semin. Nas. Mat.* , Vol. 7, Pp. 37–43, 2024, [Online]. Available: <https://proceeding.unnes.ac.id/prisma>.
- [14] T. F. Schubert Jr., F. G. Jacobitz And E. M. Kim, Student Perceptions And Learning Of The Engineering Design Process: An Assessment At The Freshmen Level, *Research In Engineering Design*, Vol. 23, Pp. 177-190, 2012.
- [15] N. A. Z. Putri, “Sistem Pendeteksi Kualitas Udara Dalam Ruang Berbasis Internet Of Things (Iot),” Vol. 5, No. 1, Pp. 9–17, 2024.
- [16] Z. Mukhammad, “Perancangan Sistem Pembersih Udara Menggunakan Metode Fuzzy Mamdani Untuk Kontrol Kipas Berbasis Iot (Internet Of Things),” Pp. 1–59, 2020.
- [17] R. Purbakawaca And S. A. Fauzan, “Rancang Bangun Sistem Pemantauan Kualitas Udara Dalam Ruang Berbiaya Rendah Berbasis Iot,” *J. Talent. Sipil*, Vol. 5, No. 1, P. 118, 2022, Doi: 10.33087/Talentsipil.V5i1.104.
- [18] A. Surahmat, “Rancang Bangun Aplikasi Sistem Penjualan Pada Percetakan Cubic Art,” *Jati (Jurnal Mhs. Tek. Inform.)*, Vol. 7, No. 1, Pp. 81–86, 2023, Doi: 10.36040/Jati.V7i1.6064.
- [19] N. I. Widiastuti And R. Susanto, “Kajian Sistem Monitoring Dokumen Akreditasi Teknik Informatika Unikom,” *Maj. Ilm. Unikom*, Vol. 12, No. 2, Pp. 195–202, 2014, Doi: 10.34010/Miu.V12i2.28.

- [20] P. Akhir And T. Syawal, “Bangka Belitung Tahun 2020,” 2020.
- [21] R. U. Abshar, Tinjauan Fiqh Siyasa Terhadap Partisipasi Masyarakat Dalam Penanganan Covid-19 (Studi Di Kelurahan Labuhan Ratu Bandar Lampung), Skripsi, Fakultas Syariah, Universitas Islam Negeri Raden Intan Lampung, Lampung, Indonesia, 2022.
- [22] B. A. Sugiarto, A. S. M. Lumenta, B. S. Narasiang, And A. M. Rumagit, “Aplikasi Sensor Polusi Udara,” *J. Tek. Elektro Dan Komput.*, Vol. 8, No. 3, Pp. 193–200, 2019.
- [23] Y. Ristia, “Pengendalian Pencemaran Udara,” *J. El-Thawalib*, Vol. 3, No. 2, Pp. 375–386, 2022, Doi: 10.24952/El-Thawalib.V3i2.5331.
- [24] Mardianto, A. Akmal, A. Hafid, And Adriani, “Perancangan Solar Cell Untuk Sumber Energi Listrik Mesin Pompa Air,” *Tek. Elektro Unismuh*, Vol. 15, Pp. 48–56, 2023.
- [25] Y. Efendi, “Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile,” *J. Ilm. Ilmu Komput.*, Vol. 4, No. 2, Pp. 21–27, 2018, Doi: 10.35329/Jiik.V4i2.41.
- [26] Malabay, “Pemanfaatan Flowchart Untuk Kebutuhan Deskripsi Proses Bisnis,” *J. Ilmu Komput.*, Vol. 12, No. 1, Pp. 21–26, 2016, [Online]. Available: [https://Digilib.Esaunggul.Ac.Id/Pemanfaatan-Flowchart-Untuk-Kebutuhan-Deskripsi-Proses-Bisnis-9347.Html](https://digilib.esaunggul.ac.id/pemanfaatan-flowchart-untuk-kebutuhan-deskripsi-proses-bisnis-9347.html).
- [27] T. Keselamatan, D. A. N. Kesehatan, And T. Kontrol, “Program,” No. 051604007 1, 2019.
- [28] F. T. Elektro, “Pengenalan Bahasa C++, Algoritma Pemrograman, Integrated Development Equipment (Ide) Visual C++ Dan Dasar Dasar Bahasa C++,” Univ. Malang, Pp. 1–17, 2018, [Online]. Available: [Http://Elektro.Um.Ac.Id/Wpcontent/Uploads/2016/04/Dasar-Pemrograman-Modul-1-Pengenalan-Pemrograman.Pdf](http://elektro.um.ac.id/wpcontent/uploads/2016/04/Dasar-Pemrograman-Modul-1-Pengenalan-Pemrograman.Pdf).
- [29] F. T. Elektro, “Pengenalan Bahasa C++, Algoritma Pemrograman, Integrated Development Equipment (Ide) Visual C++ Dan Dasar Dasar Bahasa C++,” Univ. Malang, Pp. 1–17, 2018, [Online]. Available: [Http://Elektro.Um.Ac.Id/Wpcontent/Uploads/2016/04/Dasar-Pemrograman-Modul-1-Pengenalan-Pemrograman.Pdf](http://elektro.um.ac.id/wpcontent/uploads/2016/04/Dasar-Pemrograman-Modul-1-Pengenalan-Pemrograman.Pdf).

- [30] A. I. Ahmad Fatoni, Dhany Dwi Nugroho, “Rancang Bangun Alat Pembelajaran Microcontroller Berbasis Atmega 328 Di Universitas Serang Raya,” Jurnaljournal Prosisko Vol. 2 No. 1 Maret 2015, Vol. 2, No. 1, Pp. 10–18, 2015.
- [31] S. Eko, “Perancangan Aplikasi Pengenalan Budaya Nusantara Berbasis Android Dengan Metode Rad,” J. Ilmu Komput. Jik, Vol. 5, No. 01, Pp. 30–39, 2022.
- [32] P. U. Rakhmawati, “Analisis Komunikasi Platform Internet Of Things Aplikasi Blynk,” Vol. 9, No. 2502, Pp. 40–46, 2024.



LAMPIRAN

- A. Lampiran Code Pemrograman
- B. Lampiran Penelitian

