

**OPTIMASI AKURASI INSPEKSI KUALITAS SUHU UDARA
BERBASIS IOT MENGGUNAKAN DETEKSI GANGGUAN**

*Optimization of IoT-Based Air Temperature Quality Inspection Accuracy Using
Intrusion Detection*

SKRIPSI

**Disusun sebagai syarat untuk menyelesaikan mata kuliah Skripsi program
studi Teknik Elektro Universitas Sangga Buana YPKP**

**Disusun oleh:
BRILIANTO WIBISONO
2114237024**



**PROGRAM STUDI SARJANA TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SANGGA BUANA YPKP
BANDUNG
2025**

	UNIVERSITAS SANGGA BUANA		FORMULIR LEMBAR PENGESAHAN	
	Jl. P.H.H Mustofa No. 68 Bandung, 40124		No Revisi	
			Berlaku Efektif	

LEMBAR PENGESAHAN

SKRIPSI

OPTIMASI AKURASI INSPEKSI KUALITAS SUHU UDARA BERBASIS IOT MENGGUNAKAN DETEKSI GANGGUAN

*Optimization of IoT-Based Air Temperature Quality Inspection Accuracy Using
Intrusion Detection*

Disusun oleh:

BRILIANTO WIBISONO

2114237024

Telah disetujui dan disahkan sebagai Skripsi Program S1 Teknik Elektro Fakultas
Teknik Universitas Sangga Buana YPKP
Bandung, Januari 2025

Disahkan oleh:

Pembimbing



Nina Lestari, S.T., M.T

	UNIVERSITAS SANGGA BUANA	FORMULIR LEMBAR PENGESAHAN	
	JL PH.H Mustofa No 68 Bandung 40124	No. Revisi	
		Berlaku Efektif	

Dosen Penguji 1



Ketut Abimanyu Munastha, ST., M.T.

Dosen Penguji 2



Ivany Sarief, S.T., M.T.

Mengetahui,

Ketua Program Studi



Ivany Sarief, S.T., M.T.

NIK. 432.200.122

	UNIVERSITAS SANGGA BUANA	FORMULIR LEMBAR PENGESAHAN	
	JL PH.H Mustofa No 68 Bandung 40124	No. Revisi	
		Berlaku Efektif	

LEMBAR PERNYATAAN ORISINALITAS

Nama : BRILIANTO WIBISONO
 NPM : 2114237024
 Alamat : Sambiegi Lor RT06 RW54 Maguwoharjo Depok Sleman
 E-mail : brilliantowibisono@gmail.com
 Menyatakan bahwa Skripsi ini merupakan karya orisinal saya sendiri, dengan judul:

OPTIMASI AKURASI INSPEKSI KUALITAS SUHU UDARA BERBASIS IOT MENGGUNAKAN DETEKSI GANGGUAN

Optimization of IoT-Based Air Temperature Quality Inspection Accuracy Using Intrusion Detection

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila di kemudian hari ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian.

Bandung, Februari 2025.



BRILIANTO WIBISONO

NPM. 2114237024

ABSTRAK

Pemantauan suhu udara berbasis Internet of Things (IoT) telah banyak digunakan di berbagai sektor seperti industri, pertanian, dan kesehatan. Akurasi data suhu sangat penting untuk memastikan sistem bekerja dengan optimal. Namun, perangkat IoT sering mengalami gangguan akibat faktor lingkungan, interferensi sinyal, atau manipulasi data, yang dapat menurunkan akurasi pemantauan. Oleh karena itu, diperlukan metode untuk mendeteksi dan mengatasi gangguan guna meningkatkan keandalan sistem. Penelitian ini bertujuan untuk mengoptimalkan akurasi inspeksi kualitas suhu udara pada sistem IoT dengan menerapkan deteksi gangguan menggunakan sensor deteksi gangguan. Sensor deteksi gangguan adalah perangkat yang dirancang untuk mendeteksi manipulasi atau kerusakan fisik pada perangkat keras atau sistem. Dalam penelitian ini, sensor PIR digunakan untuk deteksi gangguan fisik pada perangkat sensor sehingga data yang dihasilkan akurat sesuai dengan pembacaan sensor yang seharusnya. Hasil penelitian menunjukkan bahwa dengan adanya perangkat sensor ini, deteksi gangguan menjadi lebih cepat, sehingga dapat mengurangi kesalahan pembacaan suhu dan meningkatkan akurasi inspeksi suhu udara secara signifikan. Implementasi metode ini juga meningkatkan efisiensi pemantauan, karena sistem dapat memberikan peringatan dini terhadap potensi gangguan. Dengan demikian, penelitian ini berkontribusi dalam pengembangan sistem IoT yang lebih andal, aman, dan akurat untuk pemantauan suhu udara di berbagai aplikasi.

Kata kunci: Pemantauan suhu udara, deteksi gangguan, keandalan sistem

ABSTRACT

Air temperature monitoring based on the Internet of Things (IoT) has been widely used in various sectors such as industry, agriculture, and healthcare. The accuracy of temperature data is crucial to ensure optimal system operation. However, IoT devices often experience disturbances due to environmental factors, signal interference, or data manipulation, which can decrease monitoring accuracy. Therefore, a method is needed to detect and address disturbances to enhance system reliability. This study aims to optimize the accuracy of air temperature quality inspection in IoT systems by implementing disturbance detection using disturbance detection sensors. Disturbance detection sensors are devices designed to detect manipulation or physical damage to hardware or systems. In this study, PIR sensors are used to detect physical disturbances on sensor devices so that the data produced is accurate according to the supposed sensor readings. The research findings show that with this sensor device, disturbance detection is faster, thereby reducing temperature reading errors and significantly improving air temperature inspection accuracy. The implementation of this method also increases monitoring efficiency, as the system can provide early warnings against potential disturbances. Thus, this research contributes to the development of more reliable, secure, and accurate IoT systems for air temperature monitoring in various applications

Keywords: *Air Temperature Monitoring, Intrusion Detection, System Reliability.*

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, yang mana atas berkat, rahmat dan karunia-Nya, penulis dapat menyelesaikan Skripsi yang berjudul “Optimasi Akurasi Inspeksi Kualitas Suhu Udara Berbasis Iot Menggunakan Deteksi Gangguan”. Penulis menyusun Skripsi ini adalah syarat untuk menyelesaikan program Sarjana Teknik Elektro Universitas Sangga Buana YPKP. Dalam proses pengerjaan tugas akhir ini, banyak tantangan yang harus dihadapi penulis, Oleh karena itu penulis menyadari masih banyak kekurangan akan penyelesaian Tugas Akhir ini, karena keterbatasan ilmu pengetahuan yang dimiliki. Penulis berharap Tugas Akhir ini dapat bermanfaat bagi semua orang, baik yang membaca dan mempelajarinya terutama untuk penulis itu sendiri, sehingga penulis dapat mengimplementasikan dan meningkatkan kemampuan pada setelah membuat Tugas Akhir dan menulis Skripsi ini.

Dengan ini, penulis ingin mengucapkan terima kasih kepada semua orang yang telah membantu menyelesaikan skripsi ini, baik secara materi, do’a dan motivasi tanpa henti.

Bandung, Februari 2025

BRILIANTO WIBISONO

NPM. 2114237024

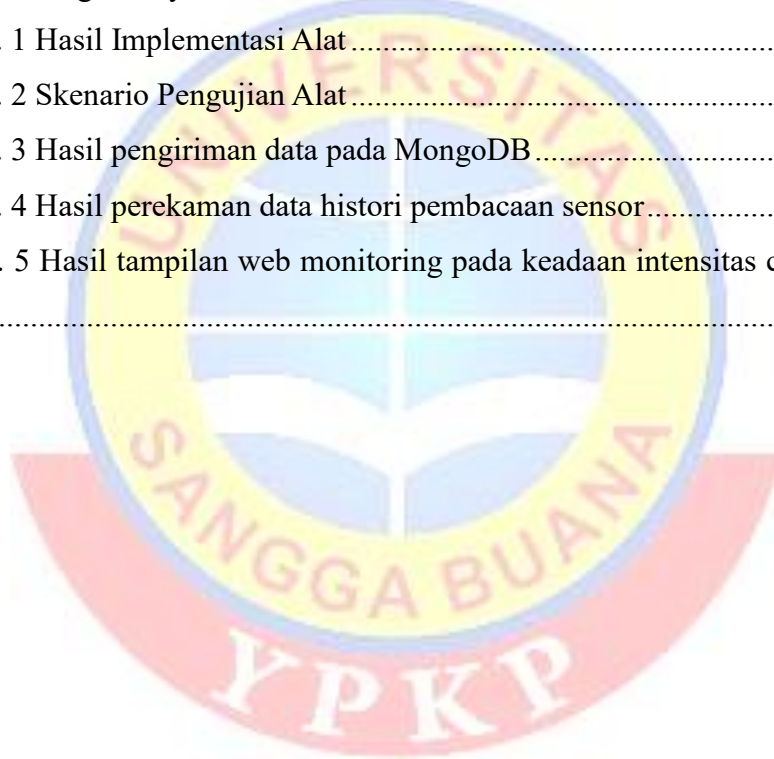
DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Penelitian	3
1.6 Sistematis Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Sistem Pemantauan Keamanan	5
2.2 Sensor Suhu	6
2.3 Sensor PIR (Passive Infrared)	7
2.4 Deteksi Perubahan Tiba-tiba dan Tampering	7
2.5 Penelitian Terdahulu	8
BAB III METODA PENELITIAN	10
3.1 Rancangan Sistem	10

3.1.1 Konfigurasi Perangkat Keras	10
3.1.2 Alur Kerja Alat (Flowchart)	11
3.2 Alat dan Bahan Perancangan	12
3.3 Rancangan Perangkat Lunak (<i>Software</i>)	13
3.3.1 <i>Coding Sketch</i> Arduino IDE	13
3.3.2 Python ScrSipt di Visual Studio Code	15
3.3.3 Penerimaan Data Pada MongoDB	16
BAB IV PENGUJIAN DAN HASIL	17
4.1 Implementasi Sistem	17
4.2 Pengujian Sistem	17
4.2.1 Pengujian Sensor Suhu dan Kelembaban	18
4.2.2 Pengujian Pemantauan Melalui Web	20
4.3 Hasil Pengujian Sensor PIR	21
BAB V PENUTUP	23
5.1 Kesimpulan	23
5.2 Saran	23
DAFTAR PUSTAKA	24
LAMPIRAN A	27
LAMPIRAN B	30

DAFTAR GAMBAR

Gambar 2. 1 PIR Motion Detector Sensor Module.....	7
Gambar 3. 1 Diagram skematik ESP32.....	10
Gambar 3. 2 Flowchart Alur Kerja Alat	12
Gambar 3. 3 Tampilan Awal Arduino IDE	13
Gambar A 1 Program dalam Visual Studio Code.....	14
Gambar A 2 Program Python dalam Visual Studio Code	15
Gambar B 1 Program dalam Visual Studio Code.....	14
Gambar B 2 Program Python dalam Visual Studio Code	15
Gambar 4. 1 Hasil Implementasi Alat.....	17
Gambar 4. 2 Skenario Pengujian Alat.....	18
Gambar 4. 3 Hasil pengiriman data pada MongoDB.....	19
Gambar 4. 4 Hasil perekaman data histori pembacaan sensor.....	19
Gambar 4. 5 Hasil tampilan web monitoring pada keadaan intensitas cahaya yang berbeda	20



DAFTAR TABEL

Tabel 3. 1 Alat dan Bahan Perancangan Alat	13
Tabel 3. 2 Software Pendukung	13
Tabel 4. 1 Hasil Pengujian Sensor Suhu dan Kelembaban	18
Tabel 4. 2 Hasil Pengujian Sensor PIR dari Berbagai Sudut Gangguan	21



DAFTAR LAMPIRAN

LAMPIRAN A

Program ESP8266

LAMPIRAN B

Program Python



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, teknologi Internet of Things (IoT) telah berkembang pesat dan banyak diterapkan di berbagai bidang, termasuk dalam sistem pemantauan suhu udara. Sistem pemantauan berbasis IoT memungkinkan pengumpulan, analisis, dan pengiriman data suhu secara real-time, sehingga memberikan manfaat besar dalam berbagai sektor seperti industri, pertanian, kesehatan, dan lingkungan. Salah satu aspek utama dalam pemantauan suhu udara adalah akurasi data, karena ketidaktepatan pengukuran dapat menyebabkan dampak yang merugikan, seperti kesalahan dalam proses produksi, penurunan kualitas produk, hingga risiko kesehatan bagi manusia. Oleh karena itu, menjaga keakuratan data suhu menjadi hal yang sangat penting.

Namun, dalam implementasinya, sistem pemantauan suhu berbasis IoT sering mengalami berbagai tantangan yang dapat mengurangi akurasi pengukuran. Gangguan dapat berasal dari faktor lingkungan, seperti perubahan suhu ekstrem, gangguan elektromagnetik, atau paparan debu dan kelembaban yang memengaruhi performa sensor. Selain itu, gangguan juga dapat terjadi akibat kesalahan perangkat keras (hardware failure), seperti kerusakan pada sensor, pergeseran posisi sensor, atau daya baterai yang melemah. Faktor lainnya adalah interferensi sinyal dan serangan siber, yang dapat menyebabkan data suhu yang dikirim menjadi tidak akurat atau bahkan mengalami manipulasi [4].

Salah satu bentuk gangguan yang sering terjadi adalah gangguan fisik pada sensor yang dapat menyebabkan pembacaan data yang tidak sesuai dengan kondisi sebenarnya. Misalnya, dalam lingkungan industri, sensor suhu dapat mengalami getaran yang kuat, paparan bahan kimia, atau perpindahan lokasi tanpa disengaja, yang dapat mengganggu stabilitas pengukuran. Selain itu, dalam beberapa kasus, terdapat potensi manipulasi data yang disengaja, seperti dalam sistem penyimpanan makanan atau farmasi, di mana pihak tertentu mungkin mencoba mengubah data suhu untuk menghindari sanksi regulasi.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan optimasi akurasi inspeksi kualitas suhu udara menggunakan deteksi gangguan berbasis sensor tamper. Sensor tamper adalah sensor tambahan yang mampu mendeteksi adanya gangguan fisik, manipulasi, atau perubahan kondisi sensor utama. Dengan adanya sensor ini, sistem dapat memberikan peringatan dini jika terjadi anomali atau gangguan pada sensor suhu, sehingga memungkinkan respons cepat untuk memperbaiki atau menyesuaikan sistem pemantauan [5].

Penelitian ini akan fokus pada pengembangan metode deteksi gangguan yang lebih akurat dan responsif dalam sistem IoT untuk pemantauan suhu udara. Metode yang digunakan meliputi identifikasi jenis gangguan, penerapan algoritma deteksi gangguan, serta evaluasi kinerja sistem dengan dan tanpa sensor tamper. Dengan pendekatan ini, diharapkan sistem pemantauan suhu berbasis IoT dapat bekerja lebih optimal, menghasilkan data yang lebih akurat, serta lebih andal dalam berbagai kondisi lingkungan.

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan teknologi pemantauan suhu yang lebih cerdas dan adaptif. Dengan adanya optimasi akurasi melalui deteksi gangguan, sistem pemantauan suhu dapat digunakan secara lebih luas dalam berbagai sektor yang membutuhkan kontrol suhu ketat, seperti penyimpanan vaksin, industri makanan, sistem HVAC (Heating, Ventilation, and Air Conditioning), serta pemantauan lingkungan dalam skala besar.

Dengan demikian, penelitian ini berupaya menjawab tantangan dalam pemantauan suhu berbasis IoT, sekaligus menghadirkan solusi inovatif untuk meningkatkan keandalan dan ketepatan pengukuran suhu udara di berbagai aplikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka diperoleh rumusan masalah sebagai berikut:

1. Bagaimana gangguan fisik dan manipulasi data dapat mempengaruhi akurasi pemantauan suhu udara pada sistem IoT?
2. Bagaimana penerapan sensor tamper dapat mendeteksi dan mengurangi gangguan pada sensor suhu?

3. Bagaimana efektivitas sistem yang dioptimalkan dalam mendeteksi dan menangani gangguan dibandingkan dengan sistem tanpa sensor tamper?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah:

1. Mengembangkan metode deteksi gangguan yang efektif untuk meningkatkan akurasi pemantauan suhu udara dalam sistem IoT menggunakan sensor tamper.
2. Menilai pengaruh gangguan fisik dan manipulasi data terhadap akurasi data suhu dan menentukan sejauh mana sensor tamper dapat mengurangi dampak tersebut.
3. Membandingkan efektivitas sistem yang dioptimalkan dengan sistem konvensional tanpa sensor tamper dalam hal responsivitas dan keandalan deteksi gangguan.

1.4 Manfaat Penelitian

Beberapa manfaat penelitian ini adalah:

1. Penelitian ini dapat meningkatkan keandalan sistem pemantauan suhu berbasis IoT dengan mengintegrasikan sensor tamper untuk mendeteksi dan mengurangi gangguan. Dengan demikian, sistem lebih tahan terhadap faktor eksternal yang bisa merusak akurasi data.
2. Peningkatan Respon Darurat: Dengan deteksi gangguan yang lebih cepat dan akurat, sistem dapat memberikan respons darurat yang lebih efektif untuk mengatasi anomali sebelum menyebabkan kerusakan lebih lanjut atau bahaya keamanan.
3. Dasar untuk Penelitian Lanjutan: Hasil dari penelitian ini dapat dijadikan dasar untuk pengembangan lebih lanjut dalam teknologi IoT, terutama dalam peningkatan algoritma deteksi gangguan dan desain sensor yang lebih canggih.

1.5 Batasan Penelitian

1. Penggunaan sensor tamper spesifik untuk deteksi gangguan fisik pada sensor suhu.

2. Analisis dilakukan dalam konteks laboratorium dengan simulasi gangguan yang dikontrol untuk memastikan repeatabilitas dan validitas hasil.
3. Penerapan dan evaluasi sensor hanya terfokus pada sistem pemantauan suhu udara tanpa memasukkan variabel lingkungan lain seperti kelembaban atau tekanan.

1.6 Sistematis Penulisan

Sistematika yang terdiri dari Langkah-langkah berikut, digunakan untuk memberikan gambaran yang jelas dan mempermudah penyampaian laporan skripsi dan merupakan penjabaran dari setiap bab yang ditulis dalam skripsi secara keseluruhan:

BAB I PENDAHULUAN

Bab ini meliputi latar belakang, maksud dan tujuan, manfaat penelitian, rumusan masalah, dan sistematika penulisan.

BAB II LANDASAN TEORI

Ulasan komprehensif tentang literatur yang relevan, termasuk penggunaan IoT dalam pemantauan suhu, pengaruh gangguan pada sistem sensor, dan penerapan sensor tamper. Bab ini juga akan mengidentifikasi gap yang ada dalam penelitian sebelumnya yang penelitian ini berusaha untuk alamat.

BAB III METODA PENELITIAN

Deskripsi metodologi yang digunakan untuk pengumpulan data, termasuk desain sistem, pemilihan dan pengaturan sensor, serta prosedur eksperimental untuk menguji efektivitas sensor tamper.

BAB IV DATA DAN ANALISIS

Presentasi dan analisis data yang dikumpulkan dari eksperimen. Bab ini akan mengevaluasi efektivitas sensor tamper dalam mengurangi kesalahan akibat gangguan fisik dan manipulasi data.

BAB V PENUTUP

Bab ini mencakup hasil kesimpulan dan saran (rekomendasi) atas penelitian yang dilakukan penulis agar dapat dikembangkan suatu saat nanti.

BAB II

LANDASAN TEORI

2.1 Sistem Pemantauan Keamanan

Sistem pemantauan keamanan adalah sistem yang dirancang untuk memantau kondisi lingkungan dan mendeteksi adanya ancaman atau gangguan dalam suatu area tertentu. Tujuan utamanya adalah untuk meningkatkan tingkat keamanan dengan memanfaatkan teknologi sensor yang mampu memberikan informasi secara real-time. Dalam implementasinya, berbagai jenis sensor digunakan, di antaranya sensor suhu dan sensor PIR (Passive Infrared). Sensor suhu digunakan untuk mengukur dan memonitor suhu lingkungan, sementara sensor PIR digunakan untuk mendeteksi pergerakan yang menunjukkan potensi perubahan kondisi yang tidak diinginkan (tampering). Sistem ini umumnya terintegrasi dengan perangkat lainnya, seperti kamera pengawas, alarm, dan perangkat IoT lainnya, untuk memberikan respons yang cepat dan otomatis ketika ancaman terdeteksi.

Dengan semakin berkembangnya teknologi, sistem pemantauan keamanan kini dapat diintegrasikan dengan platform berbasis cloud atau aplikasi smartphone, memungkinkan pengawasan jarak jauh secara real-time. Hal ini sangat berguna untuk pemantauan area yang luas atau di lokasi yang sulit dijangkau secara fisik, seperti dalam pengawasan rumah pintar, gedung perkantoran, atau fasilitas industri. Melalui sistem berbasis sensor, informasi yang diperoleh tidak hanya berbentuk data statis, tetapi juga dapat diproses secara analitik untuk mendeteksi pola perilaku yang mencurigakan dan memberikan peringatan lebih awal terkait potensi ancaman.

Selain itu, penerapan sistem pemantauan keamanan dapat membantu dalam pengelolaan dan penghematan energi di area yang dipantau. Misalnya, sensor PIR dapat digunakan untuk mendeteksi aktivitas manusia, sehingga sistem pencahayaan dapat diatur untuk menyala hanya ketika ada pergerakan, mengurangi konsumsi energi listrik. Demikian pula, sensor suhu dapat digunakan untuk mengontrol sistem pemanas atau pendingin udara secara otomatis, menjaga kestabilan suhu lingkungan tanpa perlu pengaturan manual yang terus-menerus.

Sistem pemantauan keamanan bertujuan untuk memberikan pengawasan yang efektif terhadap objek atau area tertentu, sehingga dapat mendeteksi potensi

gangguan seperti kebakaran, kerusakan perangkat, atau bahkan kegiatan yang mencurigakan seperti pencurian. Penggunaan teknologi sensor dalam sistem ini juga bertujuan untuk memberikan informasi yang lebih cepat dan akurat, meminimalkan risiko kerusakan, dan meningkatkan ketahanan sistem terhadap ancaman eksternal.

Dalam konteks peralatan elektronik dan rumah pintar, pengawasan yang terintegrasi antara sensor-sensor dapat meningkatkan efisiensi dan efektivitas pengelolaan energi serta mencegah terjadinya kerusakan atau pelanggaran keamanan [6]. Penerapan teknologi pemantauan yang cerdas dan terhubung ini tidak hanya memperkuat tingkat keamanan tetapi juga memberikan nilai tambah dalam hal pengelolaan sumber daya dan pengurangan biaya operasional.

2.2 Sensor Suhu

Sensor suhu adalah perangkat yang digunakan untuk mengukur suhu dalam suatu lingkungan, baik dalam ruang terbuka maupun tertutup. Sensor ini mengubah suhu yang terdeteksi menjadi sinyal listrik yang dapat diproses oleh mikrokontroler untuk memberikan respons terhadap perubahan suhu. Sensor suhu bekerja dengan mengukur perubahan fisik atau elektrik yang terjadi seiring perubahan suhu. Berbagai jenis sensor suhu digunakan dalam aplikasi pemantauan, di antaranya adalah termistor, termokopel, dan sensor suhu digital seperti DS18B20, atau DHT 11 dan DHT22. Setiap jenis sensor memiliki karakteristik yang berbeda dalam hal sensitivitas dan akurasi pengukuran suhu.

Dalam sistem pemantauan, sensor suhu berperan penting dalam mendeteksi kenaikan suhu yang tidak biasa, yang bisa menjadi indikasi kebakaran, overheating pada perangkat elektronik, atau adanya tampering yang mengubah kondisi suhu secara tiba-tiba. Penggunaan sensor suhu memungkinkan akurasi yang lebih tinggi dalam pemantauan suhu dalam jangka waktu panjang [7].

Pada sistem keamanan, sensor suhu digunakan untuk memberikan peringatan apabila ada lonjakan suhu yang melebihi ambang batas yang telah ditentukan. Misalnya, pada perangkat elektronik atau sistem pendingin, kenaikan suhu secara drastis bisa menandakan adanya masalah atau kegagalan yang membutuhkan perhatian segera [8].

2.3 Sensor PIR (Passive Infrared)

Passive Infrared Receiver (PIR) merupakan sebuah sensor berbasis infrared. Akan tetapi, tidak seperti sensor infrared kebanyakan yang terdiri dari LED dan fototransistor. PIR tidak memancarkan apapun seperti IR LED. Sesuai dengan namanya "*Passive*" sensor ini hanya merespon energi dari pancaran sinar inframerah pasif yang dimiliki oleh setiap benda yang terdeteksi olehnya. Benda yang bisa dideteksi oleh sensor ini biasanya adalah tubuh manusia.[9]



Gambar 2. 1 PIR Motion Detector Sensor Module

Dalam konteks keamanan, sensor PIR dapat digunakan untuk mendeteksi perubahan yang tiba-tiba, seperti pergerakan yang tidak terduga di area yang seharusnya tidak ada aktivitas. Ini dapat memberikan indikasi adanya tindakan tampering yang dilakukan oleh pihak yang tidak berwenang, seperti merusak perangkat atau merusak sistem pengawasan.

Sensor PIR sering digunakan dalam sistem alarm dan deteksi intrusi. Misalnya, sensor PIR dapat mengaktifkan alarm ketika mendeteksi gerakan yang mencurigakan, terutama pada malam hari atau saat tidak ada aktivitas yang diharapkan [11]. Sistem ini sangat berguna untuk memberikan notifikasi dini mengenai ancaman yang terjadi.

2.4 Deteksi Perubahan Tiba-tiba dan Tampering

Deteksi perubahan suhu yang mendalam dapat dilakukan dengan menggunakan algoritma pemrograman yang memantau perbedaan suhu dalam waktu tertentu. Misalnya, jika suhu meningkat lebih dari 2°C dalam waktu kurang dari 5 menit, sistem bisa memicu alarm atau mengirimkan notifikasi kepada administrator untuk tindakan lebih lanjut [12]. Untuk mendeteksi perubahan suhu yang signifikan, berbagai metode pengolahan data digunakan. Salah satu metode yang sering digunakan adalah perbandingan data suhu saat ini dengan suhu rata-rata dalam jangka waktu tertentu. Jika perbedaan antara suhu normal dan suhu saat

ini melebihi ambang batas yang telah ditentukan, sistem akan merespons dengan alarm atau pemberitahuan.

Salah satu metode yang sering digunakan adalah algoritma perbandingan berbasis ambang batas (threshold-based comparison). Algoritma ini memonitor suhu dalam waktu tertentu dan memberikan peringatan saat suhu yang terdeteksi melewati ambang batas yang ditentukan. Selain itu, penggunaan filter dan teknik pemrosesan sinyal dapat membantu mengurangi noise dan meningkatkan akurasi deteksi perubahan suhu [13]. Setiap perubahan yang terdeteksi, baik dari sensor suhu maupun PIR, bisa memicu sistem alarm yang memberitahukan pengguna atau administrator. Sistem alarm dapat berupa suara atau visual, sementara notifikasi dapat dikirim melalui aplikasi berbasis IoT ke perangkat pengguna [14].

2.5 Penelitian Terdahulu

Penelitian mengenai sistem pemantauan keamanan berbasis sensor suhu dan PIR telah banyak dilakukan dalam beberapa tahun terakhir, dengan fokus pada peningkatan efisiensi deteksi dan respons terhadap potensi ancaman. Beberapa penelitian terkait yang relevan dalam konteks ini antara lain:

- **Sistem Keamanan Berbasis IoT**

Beberapa penelitian telah mengembangkan sistem keamanan menggunakan sensor suhu dan PIR yang terintegrasi dengan teknologi Internet of Things (IoT) untuk pemantauan real-time. Salah satu penelitian oleh Baghel, Khyati mengembangkan sistem keamanan berbasis IoT yang mengintegrasikan sensor suhu dan PIR dengan perangkat pemantauan lainnya, seperti kamera pengawas dan alarm. Sistem ini mampu memberikan peringatan dini terkait ancaman di lingkungan yang dipantau [6].

- **Deteksi Suhu dan Pergerakan**

Penelitian lain yang dilakukan oleh Hasibuan, Arnawan berfokus pada desain dan pengembangan sistem pemantauan suhu yang menggunakan sensor suhu berbasis Arduino. Dalam sistem tersebut, sensor suhu digunakan untuk mendeteksi suhu yang melebihi ambang batas yang dapat mengindikasikan potensi kebakaran atau kerusakan perangkat [7].

Penelitian ini menggarisbawahi pentingnya akurasi dalam mendeteksi suhu yang tidak biasa untuk memberikan respons yang tepat waktu.

- Integrasi Sensor Suhu dan PIR untuk Keamanan

Penelitian Hidayat, Rizky mengembangkan sistem pemantauan yang menggabungkan sensor suhu dan PIR dalam satu platform berbasis IoT. Penelitian ini menunjukkan bahwa integrasi kedua sensor ini dapat meningkatkan ketepatan deteksi, terutama dalam mendeteksi pergerakan yang berhubungan dengan perubahan suhu yang signifikan, misalnya, ketika seseorang memasuki area yang dipantau [15].

- Keamanan Data dalam Sistem IoT

Srhir, Ahmed meneliti tentang tantangan dan solusi terkait dengan keamanan data dalam sistem IoT yang menggunakan sensor suhu dan PIR. Mereka menyoroti pentingnya penggunaan enkripsi dan protokol komunikasi yang aman untuk melindungi data yang dikumpulkan oleh sensor dan mencegah akses tidak sah yang dapat merusak integritas sistem pemantauan [16].

- Penggunaan AI dalam Deteksi Anomali

Salah satu studi oleh Cadet, Emmanuel mengeksplorasi penggunaan kecerdasan buatan (AI) dalam sistem pemantauan keamanan. Mereka menunjukkan bahwa algoritma pembelajaran mesin dapat digunakan untuk mendeteksi pola anomali dalam data suhu dan gerakan, sehingga dapat mengidentifikasi ancaman yang lebih sulit terdeteksi oleh sensor secara langsung. Penggunaan AI ini memperkuat kemampuan sistem untuk memberikan peringatan lebih awal dan lebih akurat dalam menghadapi potensi ancaman [17].

Penelitian-penelitian ini menunjukkan bahwa meskipun teknologi sensor sudah cukup maju dalam mendeteksi ancaman secara otomatis, pengembangan lebih lanjut dalam hal integrasi sensor, pengolahan data, dan penggunaan teknologi keamanan canggih seperti AI dan enkripsi data masih diperlukan untuk meningkatkan kinerja dan ketahanan sistem pemantauan keamanan berbasis sensor.

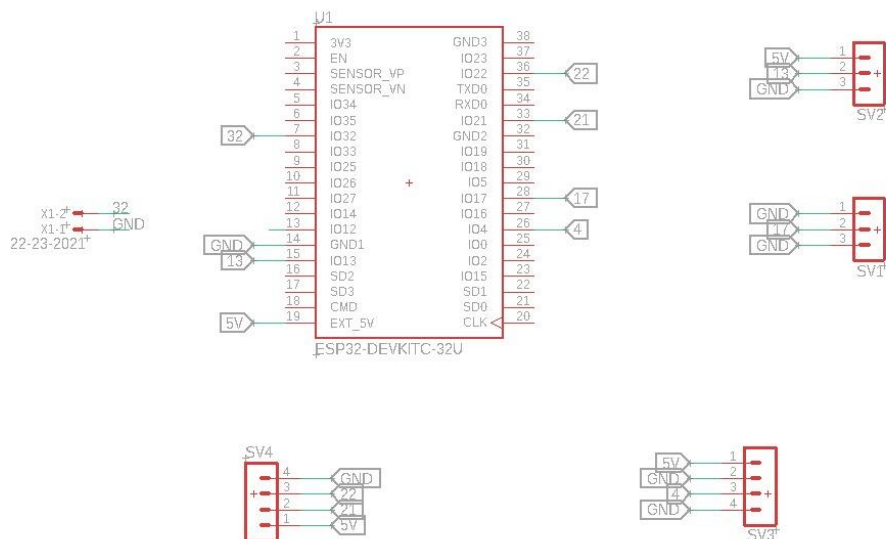
BAB III

METODA PENELITIAN

3.1 Rancangan Sistem

Dalam pengembangan sistem pemantauan suhu udara berbasis IoT, digunakan mikrokontroler ESP32-DEVKITC-32U sebagai unit pengendali utama. ESP32 dipilih karena kemampuannya dalam konektivitas Wi-Fi dan Bluetooth serta dukungan untuk berbagai I/O digital dan analog yang memungkinkan pengumpulan data dari berbagai sensor.

3.1.1 Konfigurasi Perangkat Keras



Gambar 3. 1 Diagram skematik ESP32

Diagram koneksi ESP32 pada Gambar 3.1 mengilustrasikan komponen dan terminal dihubungkan ke modul ESP32-DEVKITC-32U. Diagram ini menunjukkan detail sebagai berikut:

- **Power Supply:** ESP32 diberi daya melalui pin 3V3 dan GND untuk ground. Ada konektor tambahan yang digunakan untuk menghubungkan sensor suhu dan kelembaban, PIR, Buzzer dan LED (SV1, SV2, SV3, dan SV4) yang menunjukkan koneksi untuk power eksternal dan ground, memastikan stabilnya supply daya ke modul selama operasi.

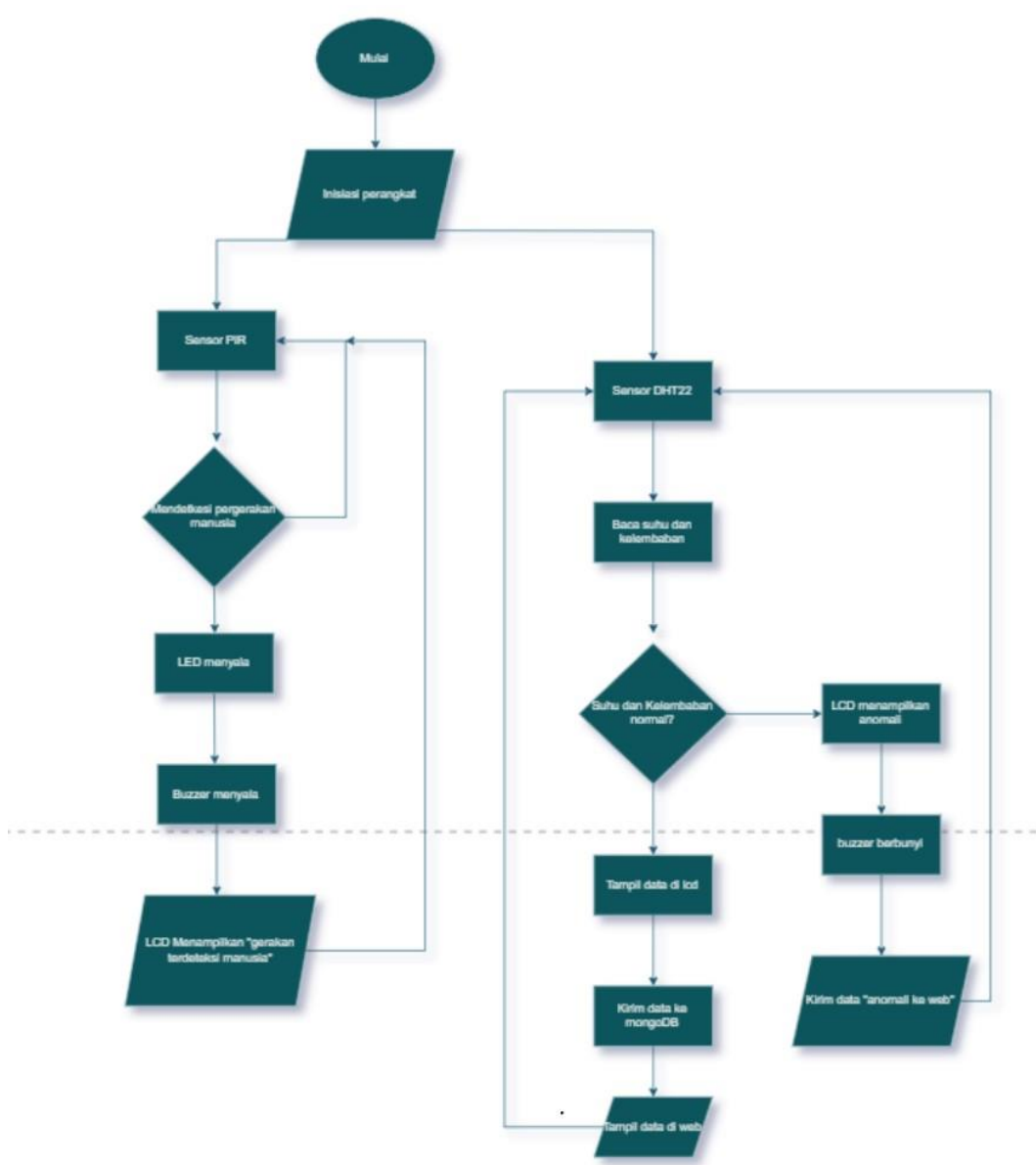
- **Koneksi I/O:** Pin-pin seperti `SENSOR_VP` dan `SENSOR_VN` digunakan untuk menghubungkan sensor suhu, sedangkan pin `IO34` hingga `IO35` diatur sebagai input atau output sesuai kebutuhan fungsi sistem.
- **Komunikasi Serial:** Pin `TX0` dan `RX0` digunakan untuk komunikasi serial, yang penting untuk transmisi data antara ESP32 dan komputer atau perangkat jaringan lainnya.

Setiap koneksi pada diagram dirancang untuk memaksimalkan efisiensi dan keakuratan data yang diperoleh dari sensor suhu serta memastikan responsifitas sistem terhadap kondisi lingkungan yang berubah.

3.1.2 Alur Kerja Alat (Flowchart)

Berikut akan alur kerja dari penelitian sistem pemantauan berbasis sensor yang menggunakan IoT untuk mendeteksi pergerakan manusia dan membaca suhu serta kelembaban lingkungan. Flowchart pada Gambar 3.2 ini terbagi menjadi dua jalur utama, masing-masing menunjukkan proses yang dijalankan oleh sistem berdasarkan input dari sensor PIR (Passive Infrared Sensor) dan sensor DHT22 (Digital Humidity and Temperature sensor). Tahapan pada sistem ini adalah:

1. **Inisiasi Sistem:** Sistem diinisiasi dengan mengaktifkan sensor dan output yang terlibat.
2. **Deteksi dan Respon:** Melalui sensor PIR, sistem mendeteksi pergerakan dan memberikan respon melalui LED dan buzzer, serta menampilkan pemberitahuan pada LCD. Paralel dengan itu, sensor DHT22 terus memantau suhu dan kelembaban.
3. **Evaluasi Data Lingkungan:** Sistem mengevaluasi data suhu dan kelembaban. Jika terdeteksi dalam kisaran normal, data ditampilkan di LCD dan dikirim ke database MongoDB. Jika abnormal, sistem mengaktifkan buzzer dan mengirim peringatan melalui web.
4. **Integrasi Data:** Semua data diproses dan dapat diakses melalui antarmuka web untuk monitoring real-time dan analisis lebih lanjut.



Gambar 3. 2 Flowchart Alur Kerja Alat

3.2 Alat dan Bahan Perancangan

Daftar bahan dan peralatan yang digunakan untuk membuat alat Sistem Sensor Deteksi Koneksi dan Pengukuran Rangkaian Modul Praktikum Laboratorium Elektronika Hibrid disajikan pada tabel dibawah ini.

1. Alat dan Bahan

Tabel 3. 1 Alat dan Bahan Perancangan Alat

No	Nama Alat	Jumlah
1	ESP32-DEVKITC-32U	1 buah
2	Sensor PIR (<i>Passive Infrared Sensor</i>)	1 buah
3	Sensor DHT22	1 buah
4	D	1 buah
5	Buzzer	1 buah
6	Kabel dan Konektor	1 paket

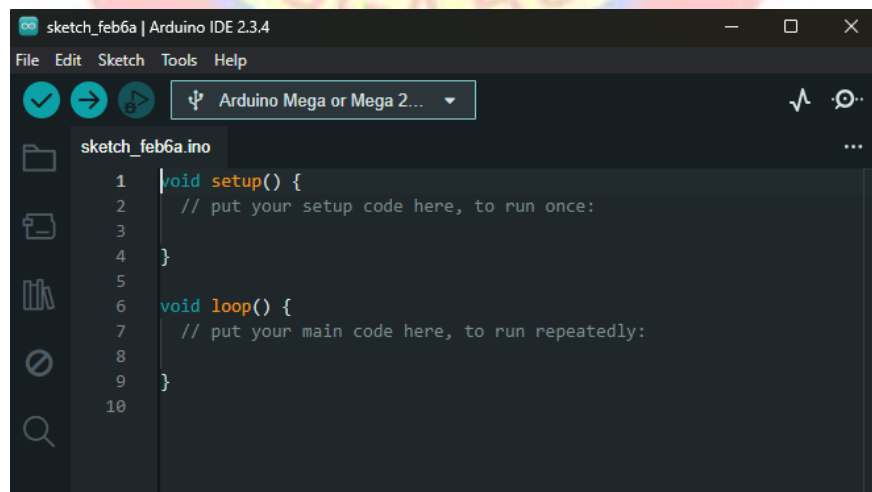
2. Bahan Perancangan (*Software Bantu*)

Tabel 3. 2 *Software* Pendukung

No	Nama	Keterangan
1	Arduino IDE	Versi 1.8.19
2	MQTT Broker	HiveMQ
3	MongoDB	<i>Database</i>
4	Visual Studio	<i>python</i>

3.3 Rancangan Perangkat Lunak (*Software*)

3.3.1 *Coding Sketch* Arduino IDE



Gambar 3. 3 Tampilan Awal Arduino IDE

Untuk membuat program pada ESP32 diperlukan instalasi software Arduino IDE (versi 2.3.4) serta library yang mendukung komunikasi WiFi dan protokol MQTT. Selain itu membuat program ESP32 sebagai pengkoneksi dengan Wifi yang nantinya akan mengkoneksi ke MQTT. Berikut merupakan program yang digunakan:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <BH1750.h>
#include <ArduinoJson.h>

// Konfigurasi WiFi
const char* ssid = "TEKIDO";
const char* password = "1234567890";

// Konfigurasi MQTT
const char* mqtt_server = "broker.hivemq.com";
const int mqtt_port = 1883;
const char* device_id = "Device 2"; // Atau "Device 2" untuk sensor kedua
const char* mqtt_topic = "sensor/luxmeter";
// Pin LED indikator WiFi
const int wifi_ledPin = D5;

// Variabel untuk monitoring WiFi
unsigned long previousMillis = 0;
unsigned long interval = 30000; // Interval pengecekan WiFi (30 detik)
// Variabel untuk pengiriman data
unsigned long lastSendTime = 0;
const long sendInterval = 2000; // Kirim data setiap 2 detik

// Inisialisasi objek
BH1750 lightMeter;
WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
```

Gambar A 1 Program dalam Visual Studio Code

```
WiFi.begin(ssid, password);

int attempt = 0;
while (WiFi.status() != WL_CONNECTED && attempt < 20) {
  delay(500);
  Serial.print(".");
  attempt++;
}

if (WiFi.status() == WL_CONNECTED) {
  digitalWrite(wifiLedPin, HIGH);
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Signal Strength (RSSI): ");
  Serial.println(WiFi.RSSI());
  Serial.println(" dBm");
} else {
  digitalWrite(wifiLedPin, LOW);
  Serial.println("");
  Serial.println("Failed to connect WiFi. Will try again later.");
}
}

*lanjut pada lampiran.
```

Gambar B 1 Program dalam Visual Studio Code

3.3.2 Python ScrSipt di Visual Studio Code

Pembuatan program Python menggunakan Visual Studio Code untuk memproses data dari broker MQTT. Instal pustaka seperti *paho-mqtt* untuk menangani komunikasi dengan broker MQTT dan *pymongo* untuk integrasi dengan *database* MongoDB.

```
from flask import Flask, jsonify, render_template
from flask_cors import CORS
from pymongo import MongoClient
import os
from dotenv import load_dotenv

load_dotenv()

app = Flask(__name__)
CORS(app, resources={
    r"/get-data": {"origins": ["http://localhost:5000", "http://127.0.0.1:5000"]}
})

try:
    MONGO_URI = os.getenv("MONGO_URI")
    MONGO_DB_NAME = os.getenv("MONGO_DB_NAME")
    MONGO_COLLECTION = os.getenv("MONGO_COLLECTION")

    client = MongoClient(MONGO_URI)
    db = client[MONGO_DB_NAME]
    collection = db[MONGO_COLLECTION]
except Exception as e:
    print(f"Error koneksi MongoDB: {e}")

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/get-data")
def get_data():
    try:
        response_data = {
            'device1': {'values': [], 'timestamps': []},
            'device2': {'values': [], 'timestamps': []}
        }

        # Ambil data terbaru untuk setiap device
        for device_id in ["Device 1", "Device 2"]: # Perubahan di sini
            latest_data = collection.find_one(
                {'device_id': device_id}) # Menggunakan device_id dengan spasi
            if latest_data:
                # Mengubah "Device 1" menjadi "device1" untuk response
                device_key = f"{device(device_id.split()[0])}" # Perubahan di sini
                response_data[device_key]['values'].append(float(latest_data['lux']))
                response_data[device_key]['timestamps'].append(
                    latest_data['timestamp'].strftime("%H:%M:%S"))
            else:
                response_data[device_key]['values'].append(0)
                response_data[device_key]['timestamps'].append(0)

        return jsonify(response_data), 200

    except Exception as e:
        print(f"Error: {e}") # Tambahkan print error untuk debugging
        return jsonify(response_data), 400

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

Gambar A 2 Program Python dalam Visual Studio Code

```
# Ambil data terbaru untuk setiap device
for device_id in ["Device 1", "Device 2"]: # Perubahan di sini
    latest_data = collection.find_one(
        {'device_id': device_id}) # Menggunakan device_id dengan spasi
    if latest_data:
        # Mengubah "Device 1" menjadi "device1" untuk response
        device_key = f"{device(device_id.split()[0])}" # Perubahan di sini
        response_data[device_key]['values'].append(float(latest_data['lux']))
        response_data[device_key]['timestamps'].append(
            latest_data['timestamp'].strftime("%H:%M:%S"))
    else:
        response_data[device_key]['values'].append(0)
        response_data[device_key]['timestamps'].append(0)

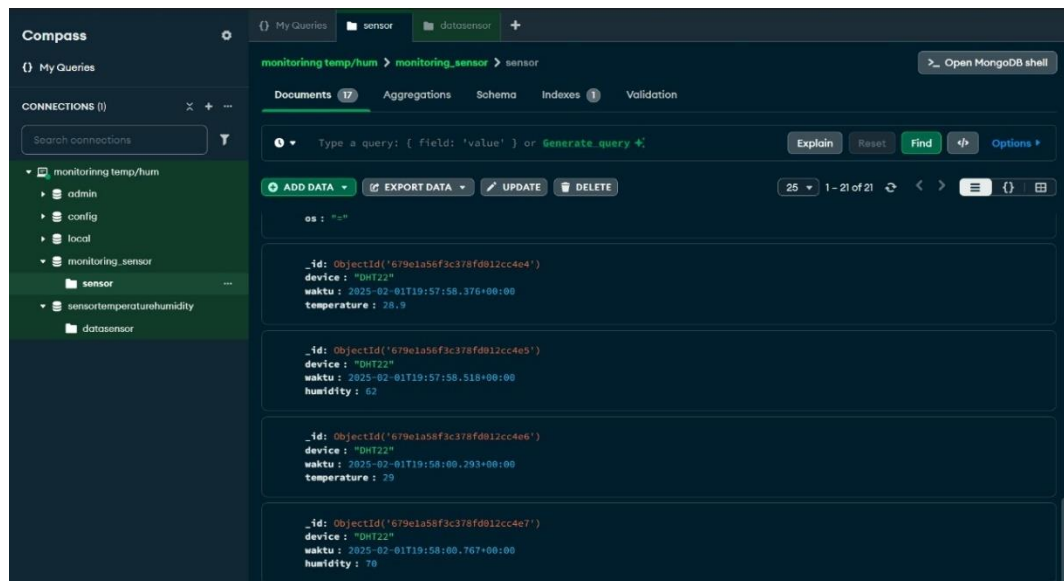
return jsonify(response_data), 200

except Exception as e:
    print(f"Error: {e}") # Tambahkan print error untuk debugging
    return jsonify(response_data), 400

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

Gambar B 2 Program Python dalam Visual Studio Code

3.3.3 Penerimaan Data Pada MongoDB



Gambar 3. 3 Tampilan awal MongoDB

Instal MongoDB compass pada web resmi MongoDB. Setelah MongoDB Compass terhubung, buat database dan koleksi yang akan digunakan. Contoh:

Database: data sensor

Koleksi: sensor

Selanjutnya, di dalam program Python, nama database (*MONGO_DB_NAME*) dan koleksi (*MONGO_COLLECTION*) harus disesuaikan dengan yang sudah dibuat di MongoDB Compass.

BAB IV

PENGUJIAN DAN HASIL

4.1 Implementasi Sistem

Gambar 4.1 menunjukkan hasil implementasi alat untuk sensor pemantauan suhu dengan sensor deteksi gangguan.



Gambar 4. 1 Hasil Implementasi Alat

4.2 Pengujian Sistem

Pada penelitian ini penulis melakukan pengujian pada sistem deteksi koneksi dan pengukuran tegangan untuk mengetahui apakah alat tersebut berjalan dengan tepat dan memiliki hasil pengukuran yang akurat. Skenario pengujian dilakukan secara terus menerus untuk memastikan sensor dapat bekerja dengan baik dan sesekali diberikan gangguan pada alat untuk mengetahui alarm berbunyi ketika akan terjadi ancaman gangguan pada alat.



Gambar 4. 2 Skenario Pengujian Alat

4.2.1 Pengujian Sensor Suhu dan Kelembaban

Pengujian sensor suhu dan kelembaban selama 30 menit, dengan pengambilan data 1 kali per menit . Tabel 4.1 menunjukkan hasil pengujian sensor tersebut

Tabel 4. 1 Hasil Pengujian Sensor Suhu dan Kelembaban

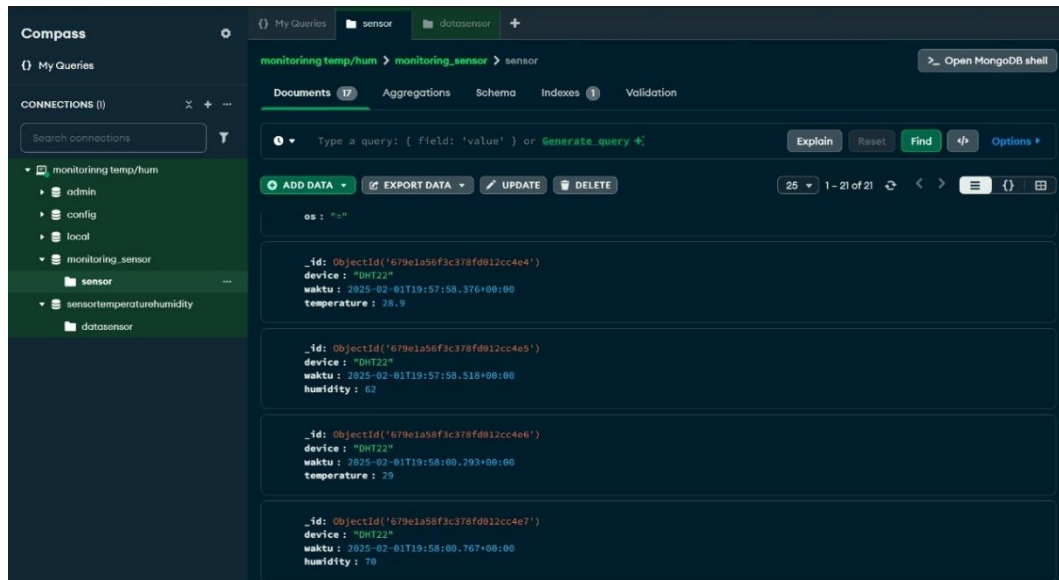
No	Suhu	Kelembaban	No	Suhu	Kelembaban
1	30.1	61.9%	16	31.6	56.3%
2	30.9	60.2%	17	31.8	55.8%
3	31.1	59%	18	32.0	55.4%
4	31.2	58.3%	19	31.3	57.4%
5	31.3	57.6%	20	30.6	58.5%
6	31.3	57.7%	21	30.6	58.4%
7	31.2	58.3%	22	30.5	58.5%
8	31.1	58.8%	23	30.5	58.3%
9	31.4	58.8%	24	30.8	58.2%
10	31.5	58.0%	25	30.9	57.9%
11	31.6	57.4%	26	31.3	57.5%
12	31.2	57.7%	27	31.0	57.0%
13	31.2	57.5%	28	30.3	58.2%
14	31.4	56.9%	29	30.3	59.3%
15	31.3	57.2%	30	29.9	60.0%

Dari hasil pengujian ini, beberapa poin dapat disimpulkan terkait keakuratan dan keandalan sensor:

1. Sensor menunjukkan kestabilan dalam pembacaan suhu dan kelembaban tanpa adanya outlier atau data yang tidak wajar.

2. Tidak ada lonjakan mendadak pada data suhu dan kelembaban, menandakan bahwa sensor bekerja dengan baik dan memberikan data yang dapat diandalkan.

Pada gambar 4.3, terlihat sistem sudah terhubung ke MongoDB dan sistem akan mengirimkan setiap data hasil pengkoneksian dan pengukuran yang terbaca oleh sistem seperti gambar di bawah ini.



Gambar 4. 3 Hasil pengiriman data pada MongoDB

Waktu	Suhu (°C)	Kelembaban (%)	Status
21-02-2025 17:20	29.6 °C	59.5 %	NORMAL
21-02-2025 17:20	29.7 °C	59.4 %	NORMAL
21-02-2025 17:20	29.8 °C	59.5 %	NORMAL
21-02-2025 17:20	30.0 °C	59.8 %	NORMAL
21-02-2025 17:20	30.1 °C	60.4 %	NORMAL
21-02-2025 17:19	30.2 °C	61.4 %	NORMAL
21-02-2025 17:19	30.4 °C	62.2 %	NORMAL
21-02-2025 17:19	30.6 °C	63.7 %	NORMAL
21-02-2025 17:19	30.8 °C	65.4 %	NORMAL
21-02-2025 17:19	31.1 °C	68.0 %	NORMAL

Gambar 4. 4 Hasil perekaman data histori pembacaan sensor

Gambar 4.4 merupakan History Data yang mencatat riwayat pembacaan suhu dan kelembaban dari sistem pemantauan berbasis IoT. Dari data tersebut diperoleh bahwa:

1. Tren Suhu dan Kelembaban

Berdasarkan data yang diperoleh, suhu lingkungan bervariasi antara 29.6°C hingga 31.1°C, sedangkan kelembaban berkisar antara 59.5% hingga 68.0%. Nilai ini tetap berada dalam batas normal dan tidak menunjukkan adanya perubahan yang signifikan selama periode pengukuran.

2. Validasi Data

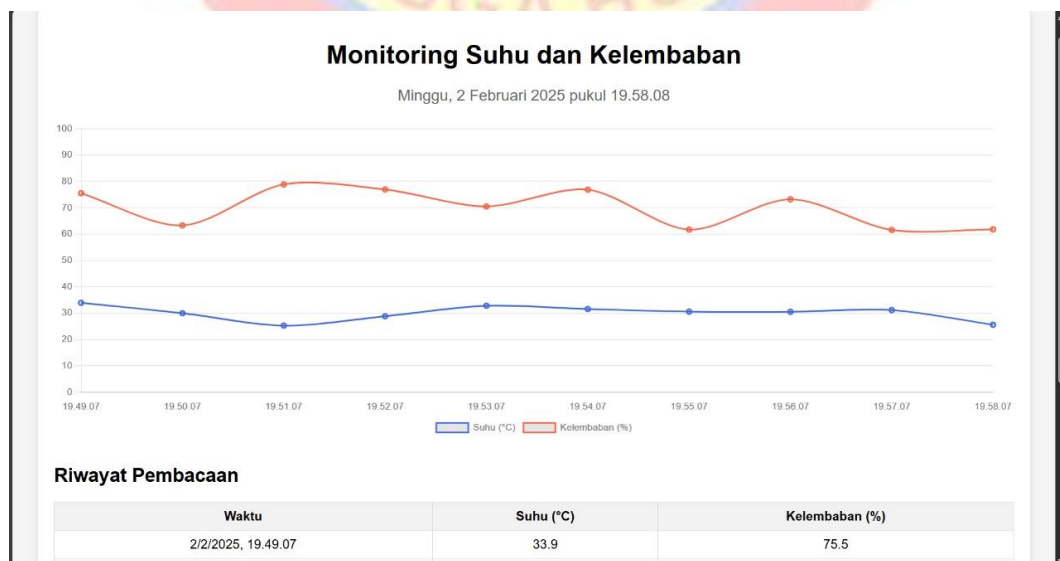
Semua data yang direkam memiliki status NORMAL, yang berarti tidak ada kondisi yang memerlukan intervensi sistem. Hal ini mengindikasikan bahwa sensor bekerja dengan baik dan tidak mengalami gangguan lingkungan seperti interferensi elektromagnetik atau perubahan suhu yang ekstrem.

3. Keandalan Sistem

Data menunjukkan bahwa sistem mampu melakukan pemantauan suhu dan kelembaban secara real-time dengan akurasi yang baik. Tidak ditemukan data kosong atau nilai yang menyimpang secara drastis, yang menandakan bahwa koneksi antara sensor dan database berjalan stabil.

Pada penelitian ini, sistem pemantauan suhu dan kelembaban berbasis IoT telah diuji dalam kondisi lingkungan yang terkendali. Data diperoleh secara real-time menggunakan sensor DHT22 yang dikendalikan oleh mikrokontroler ESP32 dan dikirim ke basis data MongoDB sebelum ditampilkan dalam antarmuka web

4.2.2 Pengujian Pemantauan Melalui Web



Gambar 4. 5 Hasil tampilan web monitoring pada keadaan intensitas cahaya yang berbeda

Gambar 4.5 menunjukkan hasil pemantauan suhu dan kelembaban. Data yang ditampilkan meliputi:

1. Grafik perubahan suhu dan kelembaban dalam satuan waktu tertentu. Suhu ($^{\circ}\text{C}$) ditunjukkan oleh garis biru. Kelembaban (%) ditunjukkan oleh garis merah.
2. Tabel riwayat pembacaan yang mencatat data suhu dan kelembaban pada waktu spesifik.

Hasil pengukuran menunjukkan bahwa suhu berada pada kisaran 30°C - 35°C , sedangkan kelembaban berada dalam rentang 70% - 80%. Perubahan data ini mencerminkan dinamika kondisi lingkungan yang dapat dipengaruhi oleh faktor eksternal, seperti sirkulasi udara atau aktivitas manusia di sekitar sensor.

Dari grafik yang ditampilkan, dapat dianalisis bahwa:

1. Suhu relatif stabil, dengan sedikit fluktuasi sekitar $2-3^{\circ}\text{C}$ selama periode pemantauan.
2. Kelembaban mengalami perubahan lebih signifikan dibandingkan suhu, dengan kenaikan dan penurunan yang lebih sering. Hal ini menunjukkan bahwa kelembaban lebih sensitif terhadap perubahan lingkungan dibandingkan suhu.

4.3 Hasil Pengujian Sensor PIR

Pengujian sensor PIR sebagai pendeteksi adanya gangguan dilakukan pada jarak kurang dari 30 cm. Pada saat ini, buzzer akan menyala selama 1 detik. Sementara itu pengujian sumber gangguan dilakukan pada berbagai sudut dari titik 0° sd 315° . Menurut penelitian yang sudah dilakukan oleh [18], sudut pengukuran PIR maksimum adalah 120°

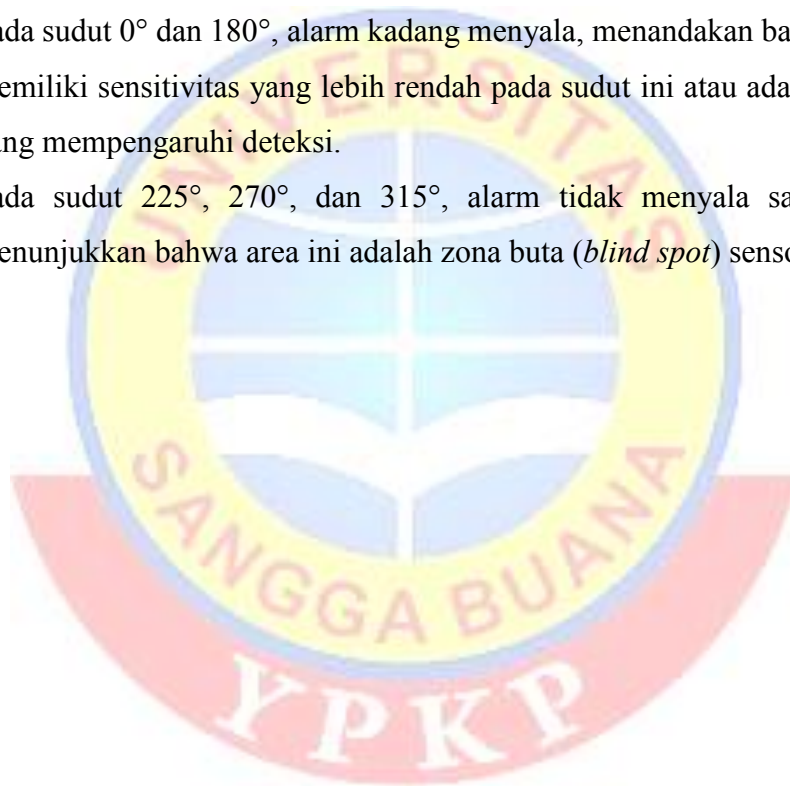
Tabel 4. 2 Hasil Pengujian Sensor PIR dari Berbagai Sudut Gangguan

No	Arah Sumber Gangguan dari Sensor	Alarm
1	0°	Kadang Menyala
2	45°	Selalu Menyala
3	90°	Selalu Menyala
4	135°	Selalu Menyala

5	180°	Kadang Menyala
6	225°	Tidak Menyala
7	270°	Tidak Menyala
8	315°	Tidak Menyala

Hasil yang ditampilkan dalam Tabel 4.2, yang menunjukkan sensor PIR mendeteksi pergerakan berdasarkan arah sumber gangguan. Dari hasil pengujian, ditemukan bahwa:

1. Sensor PIR selalu menyala pada sudut 45°, 90°, dan 135°, menunjukkan bahwa area ini adalah zona deteksi optimal.
2. Pada sudut 0° dan 180°, alarm kadang menyala, menandakan bahwa sensor memiliki sensitivitas yang lebih rendah pada sudut ini atau ada faktor lain yang mempengaruhi deteksi.
3. Pada sudut 225°, 270°, dan 315°, alarm tidak menyala sama sekali, menunjukkan bahwa area ini adalah zona buta (*blind spot*) sensor.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa sistem pemantauan suhu, kelembaban, dan deteksi pergerakan berbasis IoT yang dikembangkan mampu berfungsi dengan baik.

1. Sensor DHT22 berhasil membaca suhu dan kelembaban secara real-time dengan tingkat akurasi yang tinggi. Data menunjukkan bahwa suhu berkisar antara 29.9°C hingga 32.0°C, sementara kelembaban berada pada rentang 55.4% hingga 61.9%. Hasil pengujian menunjukkan pola hubungan invers antara suhu dan kelembaban, di mana peningkatan suhu cenderung menyebabkan penurunan kelembaban.
2. Pengujian sensor PIR menunjukkan bahwa deteksi pergerakan optimal terjadi pada sudut 45° hingga 135°, sedangkan pada sudut 0° dan 180° deteksi terjadi secara tidak konsisten. Sensor tidak dapat mendeteksi pergerakan pada sudut 225° hingga 315°, menunjukkan adanya zona buta dalam sistem deteksi.
3. Sistem pemantauan yang dikembangkan berhasil mengirimkan data suhu, kelembaban, dan status deteksi pergerakan ke server secara real-time.
4. Tampilan LCD 16x2 dapat menampilkan informasi penting terkait suhu, kelembaban, dan status deteksi. Sistem tetap berjalan stabil meskipun terjadi fluktuasi koneksi WiFi, dengan mekanisme reconnect otomatis jika koneksi terputus.

5.2 Saran

Setelah melakukan penelitian ini, maka dapat dilakukan pengembangan sebagai berikut:

1. Untuk meningkatkan efektivitas deteksi pergerakan, disarankan menggunakan lebih dari satu sensor PIR atau mengombinasikannya dengan sensor lain seperti ultrasonik atau kamera.
2. Menambahkan fitur notifikasi ke perangkat seluler melalui layanan cloud, serta mengintegrasikan kecerdasan buatan untuk analisis tren suhu, kelembaban, dan pola pergerakan

DAFTAR PUSTAKA

- [1] A. Nurcahya, H. P. Nurba, Kusmadi, K. A. Munastha, I. Sarief, and N. Lestari, "Enhancing Seedbeds Climate Control System with Internet of Things-Based Remote Automation," in *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/TSSA59948.2023.10367000.
- [2] N. Lestari, K. A. Munastha, I. H. Setyo, and D. Hadian, "Rancang Bangun Pengatur Suhu Kandang Ayam Otomatis Untuk Perternakan Ayam Skala Kecil," *TECHNO-SOCIO EKONOMIKA*, vol. 13, no. 1, pp. 1–14, Apr. 2020, doi: 10.32897/techno.2020.13.1.307.
- [3] A. Y. Rangan, Amelia Yusnita, and Muhammad Awaludin, "Sistem Monitoring berbasis Internet of things pada Suhu dan Kelembaban Udara di Laboratorium Kimia XYZ," *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, vol. 4, no. 2, pp. 168–183, Dec. 2020, doi: 10.37339/e-komtek.v4i2.404.
- [4] P. Liu and T. Liu, "Physical Intrusion Detection for Industrial Control System," in *2018 IEEE Conference on Communications and Network Security (CNS)*, IEEE, May 2018, pp. 1–2. doi: 10.1109/CNS.2018.8433194.
- [5] H. I. Islam *et al.*, "SISTEM KENDALI SUHU DAN PEMANTAUAN KELEMBABAN UDARA RUANGAN BERBASIS ARDUINO UNO DENGAN MENGGUNAKAN SENSOR DHT22 DAN PASSIVE INFRARED (PIR)," in *PROSIDING SEMINAR NASIONAL FISIKA (E-JOURNAL) SNF2016 UNJ*, Pendidikan Fisika dan Fisika FMIPA UNJ, 2016, pp. SNF2016-CIP-119-SNF2016-CIP-124. doi: 10.21009/0305020123.
- [6] K. Baghel¹ and D. Sharma², "Home Automation Security System with IoT and Embedded System," 2018. [Online]. Available: www.rsisinternational.org
- [7] A. Hasibuan, A. Qodri, and M. Isa, "Temperature Monitoring System using Arduino Uno and Smartphone Application," *Bulletin of Computer Science and Electrical Engineering*, vol. 2, no. 2, pp. 46–55, 2021, doi: 10.25008/bcsee.v2i2.1139.

- [8] V. Mane, "Environmental Monitoring Using Internet of Things," *International Journal of Electrical and Computer Engineering*, vol. 11, p. 149, Feb. 2022, doi: 10.15662/IJAREEIE.2022.1101015.
- [9] M. Reza Hidayat, B. Septiana Sapudin, T. Elektro Universitas Jenderal Achmad Yani, and T. Elektro Sekolah Tinggi Teknik-PLN, "PERANCANGAN SISTEM KEAMANAN RUMAH BERBASIS IoT DENGAN NodeMCU ESP8266 MENGGUNAKAN SENSOR PIR HC-SR501 DAN SENSOR SMOKE DETECTOR," vol. 7, no. 2, 2018.
- [10] FLYROBO, "PIR Motion Detector Sensor Module."
- [11] V. Kalbande, G. Tikale, R. Agrawal, S. Patil, L. Thakre, and G. Khekare, "Design and Implementation of Motion Sensing Security System," Feb. 2022, pp. 517–612. doi: 10.1109/ICESC54411.2022.9885628.
- [12] J. Zhou, L. Li, A. Vajdi, X. Zhou, and Z. Wu, "Temperature-Constrained Reliability Optimization of Industrial Cyber-Physical Systems Using Machine Learning and Feedback Control," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 20–31, 2023, doi: 10.1109/TASE.2021.3062408.
- [13] R. Casinillo, A. L. So, M. Mandaya, S. Dabalos, M. Enriquez, and J. F. Cane, "Development of Arduino-based high heat detector temperature control prototype for household appliances," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, pp. 140–159, Feb. 2024, doi: 10.11591/ijra.v13i2.pp140-159.
- [14] Taryudi, D. B. Adriano, and W. A. Ciptoning Budi, "IoT-based Integrated Home Security and Monitoring System," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Dec. 2018. doi: 10.1088/1742-6596/1140/1/012006.
- [15] R. Hidayat and S. Syahputra, "Journal of Artificial Intelligence and Engineering Applications Design of a Temperature Control and Object Motion Detection System in the Server Room Using IOT-Based Wemos D1," 2023. [Online]. Available: <https://ioinformatic.org/>
- [16] A. Srhir, T. Mazri, and M. BENBRAHIM, "Security in the IoT: State-of-the-Art, Issues, Solutions, and Challenges," *International Journal of Advanced*

Computer Science and Applications, vol. 14, Feb. 2023, doi: 10.14569/IJACSA.2023.0140507.

- [17] E. Cadet, O. Osundare, H. Ekpobimi, Z. Samira, and Y. Weldegeorgise, “AI-powered threat detection in surveillance systems: A real-time data processing framework,” *Open Access Research Journal of Engineering and Technology*, vol. 7, pp. 31–45, Feb. 2024, doi: 10.53022/oarjet.2024.7.2.0057.
- [18] A. Haidar, E. Elisma, M. Yamin, and W. P. Pertiwi, “Sistem keamanan pintu rumah berbasis aplikasi Telegram dan Internet of Things (IoT),” *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga)*, vol. 4, no. 2, pp. 137–144, Aug. 2024, doi: 10.35313/jitel.v4.i2.2024.137-144.



LAMPIRAN A

Program ESP32 untuk Pemantauan Suhu

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>

// Konfigurasi pin dan tipe sensor
#define DHTPIN 5
#define DHTTYPE DHT22
#define BUZZER_PIN 33
#define LED_PIN 2
#define PIR_PIN 14

// Konfigurasi batas suhu dan kelembaban
#define TEMP_MAX 35.0
#define HUMIDITY_MAX 90.0

// Konfigurasi WiFi
const char* ssid = "TP-Link_E322";
const char* password = "28082576";

// Konfigurasi MQTT
const char* mqtt_server = "broker.hivemq.com";
const int mqtt_port = 1883;
const char* mqtt_topic = "sensor/DHT22";

// Inisialisasi objek
WiFiClient espClient;
PubSubClient client(espClient);
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); //
Alamat I2C LCD dan ukuran (16x2)

// Variabel untuk timing
unsigned long lastMsg = 0;
const long interval = 10000; // Interval
pengiriman 10 detik
unsigned long lastPageSwitch = 0;
const long pageInterval = 3000; // Interval
pergantian halaman LCD (3 detik)

// Variabel untuk data sensor dan status
float temperature = 0.0;
float humidity = 0.0;
bool anomaly = false;
bool peopleDetected = false;
int currentPage = 0; // Halaman saat ini (0
= suhu/kelembapan, 1 = status)

void setup_wifi() {
    delay(10);

    Serial.println();
    Serial.print("Menghubungkan ke ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() !=
WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi terhubung");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Menghubungkan ke
MQTT...");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff),
HEX);

        if (client.connect(clientId.c_str())) {
            Serial.println("terhubung");
        } else {
            Serial.print("gagal, rc=");
            Serial.print(client.state());
            Serial.println(" coba lagi dalam 5
detik");
            delay(5000);
        }
    }
}

void checkAndAlertAnomaly(float
temperature, float humidity) {
    if (temperature > TEMP_MAX || humidity
> HUMIDITY_MAX) {
        anomaly = true;
        digitalWrite(LED_PIN, HIGH);
        digitalWrite(BUZZER_PIN, HIGH);
        delay(1000);
        digitalWrite(LED_PIN, LOW);
        digitalWrite(BUZZER_PIN, LOW);
    } else {
        anomaly = false;
    }
}
```

```

bool sendSensorData() {
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();

    // Tunggu sebentar untuk memastikan
    pembacaan sensor valid
    delay(100);

    // Cek apakah pembacaan berhasil
    if (isnan(humidity) ||
    isnan(temperature)) {
        Serial.println("Gagal membaca sensor
        DHT22!");
        return false;
    }

    // Buat objek JSON
    StaticJsonDocument<200> doc;
    doc["device_id"] = "ESP32_DHT22";
    doc["temperature"] =
    round(temperature * 10.0) / 10.0; //
    Pembulatan 1 desimal
    doc["humidity"] = round(humidity *
    10.0) / 10.0; // Pembulatan 1 desimal

    // Serialize JSON ke string
    char jsonBuffer[100];
    serializeJson(doc, jsonBuffer);

    // Publish ke MQTT
    if (client.publish(mqtt_topic,
    jsonBuffer)) {
        Serial.print("Data terkirim: ");
        Serial.println(jsonBuffer);

        // Cek anomali setelah pengiriman
        berhasil
        checkAndAlertAnomaly(temperature,
        humidity);
        return true;
    } else {
        Serial.println("Gagal mengirim data");
        return false;
    }
}

void displayPage() {
    lcd.clear();
    if (currentPage == 0) {
        // Halaman 1: Tampilkan suhu dan
        kelembapan
        lcd.setCursor(0, 0);
        lcd.print("Temp: ");
        lcd.print(temperature, 1);
        lcd.print("C");
        lcd.setCursor(0, 1);

        lcd.print("Hum: ");
        lcd.print(humidity, 1);
        lcd.print("%");
    } else if (currentPage == 1) {
        // Halaman 2: Tampilkan status
        deteksi dan anomali
        lcd.setCursor(0, 0);
        if (peopleDetected) {
            lcd.print("People Detected");
        } else {
            lcd.print("No Movement ");
        }
        lcd.setCursor(0, 1);
        if (anomaly) {
            lcd.print("Status: Anomali");
        } else {
            lcd.print("Status: Normal ");
        }
    }
}

void setup() {
    Serial.begin(115200);

    // Inisialisasi pin buzzer, LED, dan PIR
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(LED_PIN, OUTPUT);
    pinMode(PIR_PIN, INPUT);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED_PIN, LOW);

    // Inisialisasi sensor DHT
    dht.begin();

    // Inisialisasi LCD
    lcd.begin();
    lcd.backlight();
    lcd.print("Inisialisasi...");
    delay(2000);
    lcd.clear();

    // Setup WiFi
    setup_wifi();

    // Setup MQTT
    client.setServer(mqtt_server,
    mqtt_port);

    Serial.println("Sistem monitoring suhu
    dan kelembaban siap!");
}

void loop() {
    // Cek koneksi MQTT
    if (!client.connected()) {
        reconnect();
    }
}

```

```

client.loop();

// Kirim data setiap interval
unsigned long now = millis();
if (now - lastMsg > interval) {
    lastMsg = now;
    sendSensorData();
}

// Cek koneksi WiFi
if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Koneksi WiFi terputus.
Menghubungkan kembali...");
    setup_wifi();
}

// Cek sensor PIR
int pirState = digitalRead(PIR_PIN);
if (pirState == HIGH) {
    digitalWrite(LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH);
    peopleDetected = true;
    delay(1000); // Buzzer berbunyi
    selama 1 detik
    digitalWrite(BUZZER_PIN, LOW);
} else {
    digitalWrite(LED_PIN, LOW);
    peopleDetected = false;
}

// Ganti halaman LCD setiap interval
if (now - lastPageSwitch > pageInterval)
{
    lastPageSwitch = now;
    currentPage = (currentPage + 1) % 2;
    // Berpindah antara halaman 0 dan 1
    displayPage();
}
}

```

LAMPIRAN B

Program Python

app.py

```
from flask import Flask, jsonify,
render_template
from flask_cors import CORS
from pymongo import MongoClient
from datetime import datetime

app = Flask(__name__,
    static_folder='static',
    template_folder='templates')
CORS(app)

# Koneksi MongoDB
try:
    client =
MongoClient('mongodb://localhost:2701
7/')
    db = client['monitoring_sensor']
    collection = db['sensor']
except Exception as e:
    print(f"Error koneksi MongoDB: {e}")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/api/data')
def get_data():
    try:
        # Ambil data terbaru dengan sort
        data = list(collection.find({}, {'_id':
0}).sort('waktu', -1).limit(10))

        # Konversi waktu ke format yang bisa
di-serialize
        for item in data:
            if isinstance(item.get('waktu'),
datetime):
                item['waktu'] =
item['waktu'].isoformat()

        return jsonify(data)
    except Exception as e:
        print(f"Error mengambil data: {e}")
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

web.py

```
from flask import Flask, jsonify,
render_template
from flask_cors import CORS
from pymongo import MongoClient
from datetime import datetime

app = Flask(__name__,
    static_folder='static',
    template_folder='templates')
CORS(app)

# Koneksi MongoDB
try:
    client =
MongoClient('mongodb://localhost:2701
7/')
    db = client['monitoring_sensor']
    collection = db['sensor']
except Exception as e:
    print(f"Error koneksi MongoDB: {e}")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/api/data')
def get_data():
    try:
        # Ambil data terbaru dengan sort
        data = list(collection.find({}, {'_id':
0}).sort('waktu', -1).limit(10))

        # Konversi waktu ke format yang bisa
di-serialize
        for item in data:
            if isinstance(item.get('waktu'),
datetime):
                item['waktu'] =
item['waktu'].isoformat()

        return jsonify(data)
    except Exception as e:
        print(f"Error mengambil data: {e}")
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

mqttmongo.py

```
import paho.mqtt.client as mqtt
```

```

import json
from pymongo import MongoClient
from datetime import datetime

# Koneksi MongoDB
try:
    mongo_client =
MongoClient('mongodb://localhost:2701
7/')
    db = mongo_client['monitoring_sensor']
    collection = db['sensor']
    print("Terhubung ke MongoDB!")
except Exception as e:
    print(f"Error koneksi MongoDB: {e}")

# Callback saat menerima pesan
def on_message(client, userdata,
message):
    try:
        payload =
message.payload.decode('utf-8')
        print(f"Pesan diterima: {payload}")

        try:
            data = json.loads(payload)
            # Tambahkan timestamp
            data['waktu'] = datetime.now()
            collection.insert_one(data)
            print(f"Data berhasil disimpan:
{data}")
        except json.JSONDecodeError as e:
            print(f"Error parsing JSON: {e}")

    except Exception as e:
        print(f"Error dalam pengolahan
pesan: {e}")

def on_connect(client, userdata, flags, rc):
    print("Terhubung ke MQTT Broker!")
    client.subscribe("sensor/DHT22")
    print("Subscribe ke topic:
sensor/DHT22")

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

print("Mencoba hubungkan ke
broker.hivemq.com...")
client.connect("broker.hivemq.com",
1883, 60)

client.loop_forever()

```