

SISTEM KENDALI DAN MONITORING KONDISI RUANGAN UNTUK PENDERITA SLEEP APNEA BERBASIS IOT

M Febriansyah, Fazryan Dwicahya
Program Studi Teknik Elektro, Fakultas Teknik,
Institut Sains Dan Teknologi Nasional Jakarta
E_mail: m.febriansyah@istn.ac.id

ABSTRAK

Sleep apnea adalah gangguan tidur serius yang ditandai dengan berhentinya pernapasan secara periodik selama tidur, yang dapat menyebabkan berbagai masalah kesehatan. Diperlukan suatu sistem untuk memantau dan mengontrol kondisi ruangan yang dapat mempengaruhi sleep apnea. Sistem akan memantau suhu, kelembaban, dan cahaya ruangan berbasis IoT dengan MQTT. Mikrokontroler ESP32 mengumpulkan data sensor dan mengirimkannya ke server pusat melalui protokol MQTT untuk dianalisis dan dipantau secara real-time. Protokol MQTT memastikan transmisi data yang andal dan efisien melalui jaringan berbandwidth rendah, yang membuatnya ideal untuk pemantauan terus-menerus. Sistem pengendali bertujuan untuk meningkatkan kualitas tidur dengan memastikan kondisi ruangan tetap nyaman dan optimal dengan mengaktifkan perangkat seperti kipas, humidifier, dan lampu secara otomatis sehingga tercipta lingkungan yang ideal bagi penderita sleep apnea.

Kata kunci: Sleep Apnea, MQTT, IoT.

ABSTRACT

Sleep apnea is a serious sleep disorder characterized by periodic cessation of breathing during sleep, which can lead to various health problems. A system is needed to monitor and control room conditions that can affect sleep apnea. The system will monitor the temperature, humidity, and lighting of the room based on IoT with MQTT. The ESP32 microcontroller collects sensor data and sends it to a central server via the MQTT protocol for real-time analysis and monitoring. The MQTT protocol ensures reliable and efficient data transmission over a low-bandwidth network, making it ideal for continuous monitoring. The control system aims to improve sleep quality by ensuring that room conditions remain comfortable and optimal by automatically activating devices such as fans, humidifiers, and lights to create an ideal environment for sleep apnea sufferers.

Keywords: Sleep Apnea, MQTT, IoT

1. PENDAHULUAN

Gangguan tidur serius yang ditandai dengan berhentinya napas secara berulang selama tidur (*Sleep apnea*) yang dapat mengganggu kualitas tidur dan menyebabkan berbagai masalah kesehatan seperti hipertensi, penyakit jantung, dan diabetes [1]. Oleh karena itu, menjaga kondisi ruangan yang optimal bagi penderita sleep apnea sangat penting untuk membantu mereka mendapatkan tidur yang lebih baik [2]. Internet of Things (IoT) memungkinkan integrasi berbagai sensor dan aktuator untuk menciptakan lingkungan yang saling terhubung. MQTT (Message Queueing Telemetry Transport) dirancang untuk komunikasi yang ringan dan efisien, sehingga cocok untuk penerapan teknologi IoT. Dengan model Subscriber-Publisher memungkinkan perangkat untuk berkomunikasi dengan broker pusat dan bertukar data secara real-time.

Implementasi IoT dalam sistem kontrol ruangan otomatis melibatkan penggunaan sensor seperti sensor kelembaban, suhu, cahaya dan penggunaan aktuator yang mengontrol perangkat seperti AC, kipas dan humidifier. Broker MQTT bertindak sebagai perantara antara sensor, aktuator, dan aplikasi pemantauan. Aplikasi monitoring memungkinkan pengguna untuk memantau kondisi ruangan secara real-time dan mengontrol perangkat yang terhubung [3]. Keuntungan sistem IoT dengan MQTT

untuk sleep apnea termasuk respons cepat dan real-time, komunikasi efisien, monitoring berkelanjutan, dan peningkatan kualitas tidur. Dengan menjaga kelembaban ruangan pada tingkat optimal, dapat membantu penderita sleep apnea mendapatkan tidur yang lebih baik [4].

2. LANDASAN TEORI

Beberapa komponen pendukung fungsionalitas sistem kendali ruangan untuk penderita sleep apnea, yaitu :

2.1 Obstructive Sleep Apnea

Obstructive Sleep Apnea (OSA) adalah gangguan tidur di mana aliran udara terhambat atau terhalang meskipun ada usaha untuk bernafas. Gangguan ini menghasilkan berhenti napas sebagian (hipopnea) atau sepenuhnya (apnea) selama setidaknya 10 detik yang menyebabkan penurunan tajam dalam kadar oksigen darah.



Gambar 2.1 Terjadi Obstructive Sleep Apnea

Respons otak terhadap kekurangan oksigen adalah dengan memberikan sinyal pada tubuh untuk bangun

sejenak dari tidur sehingga dapat mengembalikan pernapasan menjadi normal. Pola ini dapat terjadi berulang-ulang kali dalam satu malam, mengakibatkan tidur yang terfragmentasi dan rasa kantuk berlebihan di siang hari. Sleep apnea dapat dipengaruhi oleh berbagai faktor, yaitu faktor lingkungan, faktor fisiologis dan faktor gaya hidup [6].

▪ Faktor Lingkungan

Mencakup kondisi eksternal yang dapat memengaruhi kualitas tidur dan kesehatan secara keseluruhan. Contoh seperti kebisingan, cahaya, kualitas udara.

▪ Faktor Fisiologis

Melibatkan kondisi tubuh dan kesehatan yang memengaruhi kemungkinan terjadinya sleep apnea, antara lain usia, jenis kelamin, Obesitas, anatomi saluran pernapasan.

▪ Faktor Gaya Hidup

Faktor kebiasaan yang melibatkan pola perilaku dan gaya hidup yang dapat memengaruhi sleep apnea, seperti merokok dan mengkonsumsi alkohol.

2.2 Aplikasi Web

Aplikasi web atau Web Apps adalah program atau software yang berbasis web serta bisa diakses hanya dengan menggunakan browser saja. Hal ini memungkinkan adanya interaksi dalam website dari pengguna (user) dengan server web tersebut.

Aplikasi web sering dimanfaatkan untuk situs e-commerce, online banking dan juga pengiriman formulir tertentu. Terdapat banyak sekali fitur yang tersedia di dalam Web Apps ini yang memungkinkan interaksi bisa dilakukan dengan lebih baik.



Gambar 2.2 Cara Kerja Aplikasi Web

Web Apps tidak mewajibkan pengguna untuk mengunduhnya, menjadikannya mudah diakses sekaligus mengeliminasi kebutuhan perawatan bagi pengguna akhir.

2.3 MQTT

MQTT (Message Queueing Telemetry Transport) adalah protokol komunikasi ringan yang dirancang untuk perangkat dengan sumber daya terbatas dan jaringan yang tidak stabil. Protokol ini dioptimalkan untuk pengiriman pesan yang cepat dan efisien, menjadikannya sangat cocok untuk aplikasi IoT (Internet of Things). MQTT berjalan di atas protokol TCP/IP, yang memastikan pengiriman data yang handal antara perangkat yang terhubung [7].

MQTT menggunakan model komunikasi Subscribe-Publish, melibatkan 4 komponen utama, yaitu :

1. Publisher

Entitas yang mengirimkan pesan ke sistem MQTT.

Pesan ini dikirimkan dengan topik tertentu, yang berfungsi sebagai pengidentifikasi atau kategori untuk pesan tersebut.

2. Subscriber

Entitas yang berlangganan ke satu atau lebih topik untuk menerima pesan terkait.

Subscriber hanya menerima pesan yang relevan dengan topik yang mereka langgani.

3. Broker

Bertindak sebagai perantara yang menerima pesan dari Publisher, mendistribusikannya pada Subscriber yang berlangganan topik tersebut.

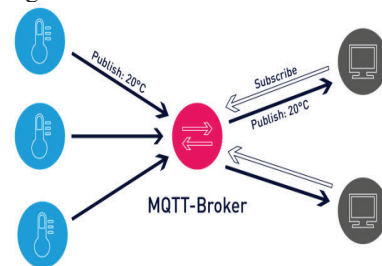
Broker juga memastikan keamanan, mengelola koneksi klien, dan mengelola lalu lintas pesan.

4. Topic

Sebuah string yang digunakan untuk mengategorikan pesan.

Topik bisa berupa "sensor/temperature/livingroom" yang menunjukkan data suhu dari ruang tamu.

Sistem topik yang terstruktur ini memungkinkan pengaturan dan distribusi pesan yang sangat efisien dalam jaringan yang besar.



Gambar 2.3 Cara Kerja MQTT

Dalam gambar 2.3 dapat terlihat bagaimana proses Subscribe-Publish terjadi. Publisher mengirimkan pesan (payload) dengan Topic tertentu ke Broker. Broker berperan sebagai perantara, menerima pesan dari Publisher dan meneruskannya kepada Subscriber yang berlangganan Topic terkait. Subscriber menerima pesan dari Broker sesuai dengan topik yang mereka langgani (Subscribe).

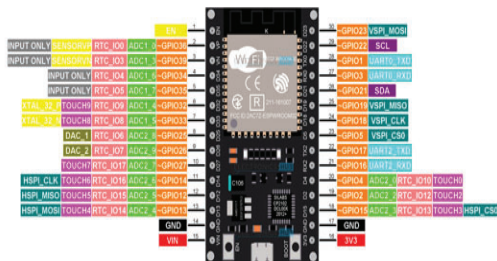
MQTT dirancang untuk komunikasi yang cepat dan efisien. Pesan-pesan dalam MQTT sangat kecil dan hanya membutuhkan bandwidth yang minim, menjadikannya ideal untuk perangkat dengan sumber daya terbatas atau koneksi jaringan yang lambat. Protokol ini didesain untuk meminimalkan penggunaan daya, sehingga cocok untuk perangkat yang menggunakan baterai. MQTT juga mampu menangani ribuan klien yang terhubung secara bersamaan, membuatnya ideal untuk implementasi IoT berskala besar.

Karena desainnya yang ringan, MQTT tidak ideal untuk mentransfer file besar atau data yang tidak terstruktur seperti video atau gambar. Jika broker mengalami kegagalan, seluruh sistem MQTT bisa terhenti, menjadikan broker sebagai titik kegagalan

tunggal yang memerlukan solusi seperti redundansi atau high-availability.

2.4 Mikrokontroler ESP32

ESP atau Espressif Systems Processors adalah mikrokontroler yang dikembangkan oleh perusahaan Espressif Systems, yang berbasis di China. Dikenal karena kemampuannya yang kuat dan harganya yang terjangkau, menjadikannya populer di kalangan pengembang IoT (Internet of Things). ESP32 merupakan mikrokontroler penerus ESP8266 yang telah terintegrasi dengan prosesor dual-core, memori flash, Wi-Fi, Bluetooth Low Energy (BLE), dan berbagai antarmuka perangkat keras lainnya. ESP32 memperoleh popularitas yang tinggi dalam pengembangan berbagai aplikasi IoT karena kemampuannya yang luas dan fleksibilitas yang tinggi [8].



Gambar 2.4 Mikrokontroler ESP32

ESP32 terdiri dari 18 pin ADC (Analog Digital Converter) yang berfungsi untuk mengubah sinyal analog ke digital, 2 pin DAC (Digital Analog Converter) yang berfungsi untuk mengubah sinyal digital ke analog, 16 pin PWM (Pulse Width Modulation), 10 pin sensor sentuh, 2 pin jalur antarmuka UART, serta pin antarmuka I2C, I2S, dan SPI.

Tabel 2.1 Spesifikasi Mikrokontroler ESP32

Parameter	Spesifikasi
Tegangan	2.2V – 3.6V
Prosesor	Xtensa dual-core 32-bit
Jumlah IO pin	36 Digital 18 Analog Input
Konektivitas	Wifi 802.11 b/g/n Bluetooth BLE
Frekuensi Operasi	~240 MHz
Interfaces	UART, SPI, I2C, I2S, PWM, ADC

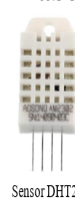
dilengkapi dengan prosesor Xtensa dual-core 32-bit, yang memberikan kemampuan pemrosesan yang kuat dan efisien untuk menjalankan berbagai tugas secara bersamaan. Dalam hal konektivitas, ESP32 mendukung jaringan WiFi 802.11 b/g/n dan Bluetooth Low Energy (BLE), memungkinkan dengan mudah terhubung ke internet atau perangkat lainnya, dengan frekuensi operasi hingga 240 MHz sehingga mampu menjalankan aplikasi yang memerlukan pemrosesan cepat dan responsif. Mikrokontroler ESP32 juga mendukung berbagai antarmuka komunikasi seperti UART, SPI, I2C, I2S, PWM, dan ADC.

2.5 Sensor DHT22

Sensor digital yang mampu mengukur suhu dan kelembaban dengan tingkat akurasi cukup tinggi yang merupakan penerus dari DHT11 [9]. Sensor ini sering dipergunakan untuk pemantauan kondisi lingkungan. DHT22 menggabungkan sensor kapasitif kelembaban dan termistor untuk suhu, yang hasil pengukurannya dikonversi menjadi sinyal digital yang dapat dibaca oleh mikrokontroler seperti Arduino, Raspberry Pi, dan ESP. Karena Sensor ini sudah dikalibrasi dari pabrik, banyak yang menggunakannya karena mempermudah pengguna dalam implementasi sensor tanpa khawatir dengan akurasi yang berubah-ubah. Berikut ini merupakan spesifikasi dari Sensor DHT22.

Tabel 2.2 Spesifikasi Sensor DHT22

Parameter	Spesifikasi
Tegangan	3.3V ~ 6V
Arus Operasi	0.3μA (Standby) 5.5μA (Operasional)
Rentang Sensor Suhu	-40°C ~ 80°C ± 0.5°C
Rentang Sensor Kelembaban	0 ~ 100% RH ± 2%

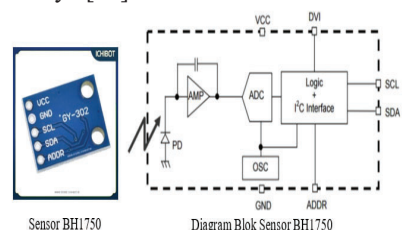


Tabel 2.2 menunjukkan bahwa sensor ini beroperasi pada rentang tegangan 3.3V hingga 6V, memungkinkan fleksibilitas dalam berbagai lingkungan daya. Memiliki arus operasi yang sangat rendah, yaitu 0.3μA dalam mode standby dan 5.5μA dalam mode operasional, sehingga sangat efisien dalam penggunaan energi.

Dalam hal pengukuran suhu, sensor DHT22 memiliki rentang yang luas dari -40°C hingga 80°C dengan akurasi ±0.5°C. Untuk kelembaban, sensor ini mampu mengukur dari 0 hingga 100% RH dengan akurasi ±2%. Spesifikasi ini menjadikan DHT22 pilihan yang ideal untuk aplikasi yang memerlukan pemantauan kondisi lingkungan dengan akurasi yang tinggi dan konsumsi daya yang rendah.

2.6 Sensor BH1750

Sensor cahaya yang menghasilkan keluaran sinyal digital, sehingga tidak memerlukan perhitungan yang rumit. BH1750 lebih akurat dan lebih mudah digunakan dibandingkan dengan sensor lain seperti fotodiode dan LDR yang menghasilkan keluaran sinyal analog serta memerlukan perhitungan untuk mendapatkan data intensitas cahaya [10].



Sensor BH1750

Diagram Blok Sensor BH1750

Gambar 2.5 Sensor BH1750

Pada gambar 2.5 terlihat komponen utama dari sensor BH1750, yaitu :

- PD : Fotodioda adalah komponen sensor yang sensitif terhadap cahaya. Komponen ini mengubah cahaya menjadi arus listrik.
- AMP : Penguat (amplifier) memperkuat arus lemah dari fotodioda ke tingkat yang dapat diproses ADC.
- ADC : Pengonversi ADC mengubah sinyal analog dari penguat menjadi nilai digital.
- Logic : Unit logika memproses sinyal digital dari ADC untuk menghitung tingkat cahaya sekitar. Unit ini juga mencakup antarmuka I2C untuk komunikasi dengan mikrokontroler.
- OSC : Osilator internal menyediakan sinyal jam untuk operasi internal sensor.

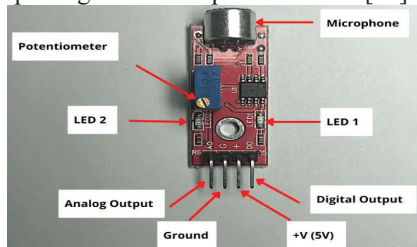
Tabel 2.3 Spesifikasi Sensor BH1750

Parameter	Spesifikasi
Catu Daya	3V ~ 5V
Arus Operasi	10 μ A (Standby) 120 μ A (Operasional)
Rentang Sensor	0 ~ 65335 Lux $\pm$ 20%

Tabel 2.3 menampilkan spesifikasi sensor BH1750. Dalam mode standby, sensor BH1750 hanya membutuhkan arus sebesar 10 μ A, sehingga sangat hemat energi ketika tidak aktif. Saat dalam mode operasional, sensor ini menggunakan arus sebesar 120 μ A. Memiliki rentang pengukuran yang sangat luas, yaitu dari 0 hingga 65335 lux, dengan akurasi $\pm$ 20%. Hal ini menjadikannya sangat ideal untuk aplikasi yang memerlukan pengukuran tingkat cahaya dari kondisi sangat gelap hingga sangat terang.

2.7 Sensor KY037

Berfungsi mendeteksi intensitas suara dan mengubahnya menjadi sinyal listrik untuk diproses oleh mikrokontroler. Sensor ini beroperasi berdasarkan prinsip bahwa gelombang suara yang mengenai membran sensor akan menyebabkan kumparan kecil di balik membran tersebut bergerak naik dan turun [11]. Gerakan kumparan ini dipicu oleh kekuatan gelombang suara yang mengenainya. Perubahan posisi kumparan menghasilkan sinyal listrik dengan intensitas yang bervariasi tergantung pada kecepatan gerakan kumparan tersebut [12].



Gambar 2.6 Sensor KY037

Pada gambar 2.6 terlihat komponen utama sensor KY037 sebagai berikut :

- Q1 : Mikrophone - Menangkap gelombang suara dan mengubahnya menjadi sinyal listrik.
- VR1: Potentiometer-Mengatur sensitivitas deteksi suara.

- R1-R6: Resistors - Mengatur arus dan membagi tegangan dalam rangkaian.
- L1 dan L2: LED - Menunjukkan indikator visual dari deteksi suara.
- U1.1 dan U1.2: Operational Amplifiers (LM393) - Memperkuat sinyal dari mikrofon dan membandingkan dengan referensi.
- A0: Analog Output-Menghasilkan sinyal kontinu yang merepresentasikan intensitas suara.
- D0: Digital Output-Menghasilkan sinyal ON/OFF berdasarkan ambang batas suara.

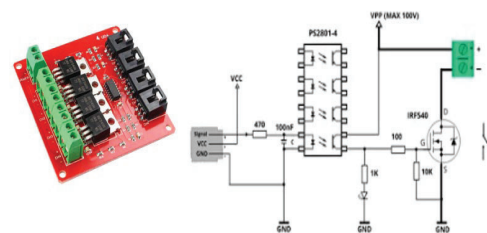
Tabel 2.4 Spesifikasi Sensor KY037

Parameter	Spesifikasi
Catu Daya	3.3V ~ 5.5V
Arus Operasi	~0.5 mA
Sensitivitas Mikrophone	-42 $\pm$ 3 dB
Output	Digital(DO) Analog(AO)

KY037 memiliki sensitivitas mikrofon sebesar -42 $\pm$ 3 dB, yang memungkinkan sensor ini mendeteksi suara dengan tingkat presisi yang tinggi. Sensor ini menyediakan dua jenis output: digital (DO) dan analog (AO), yang memungkinkan integrasi yang mudah dengan berbagai jenis mikrokontroler dan perangkat lain. Output digital dapat digunakan untuk mendeteksi adanya suara di atas ambang batas tertentu, sementara output analog memberikan representasi kontinu dari tingkat suara yang terdeteksi. Spesifikasi ini menjadikan KY037 ideal untuk aplikasi yang memerlukan deteksi suara dengan akurasi tinggi dan fleksibilitas dalam penggunaan daya dan jenis output.

2.8 Modul MOSFET IRF540 4-Channel

Sistem ini menggunakan modul MOSFET IRF540 yang dapat mengontrol beban DC hingga 100V/10A dan sudah dilengkapi dengan Optocoupler yang berfungsi untuk melindungi MOSFET dari gangguan sinyal atau lonjakan tegangan yang dapat merusak MOSFET.



Gambar 2.7 Modul MOSFET IRF540 4-Channel (kiri) dan Skema Diagram Channel Modul (kanan)

Modul ini pada dasarnya adalah "Solid-State Relay" untuk arus DC. Maksimal 4 beban yang dapat dikendalikan secara individu dapat terhubung ke output terputus (+) dan (-), semuanya diberi daya dari catu daya yang sama. Komponen yang digunakan antara lain :

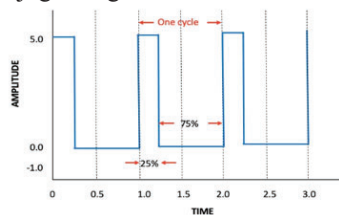
- IRF540 : Sebagai saklar elektronik untuk mengontrol aliran arus dalam rangkaian.
- PS2801-4 : Memungkinkan kontrol yang efisien dan isolasi listrik antara sinyal kontrol dan MOSFET,

memastikan bahwa sinyal yang mengendalikan MOSFET tidak mengganggu sirkuit lainnya.

- Resistor (470 Ω , 1 k Ω , 10 k Ω) : Digunakan dalam rangkaian untuk membatasi arus dan membagi tegangan.
- Kapasitor (100 nF) : Berfungsi sebagai filter untuk menekan noise atau transien dalam rangkaian.

2.11 PWM

PWM (Pulse Width Modulation) merupakan teknik modulasi yang merubah lebar pulsa sinyal dengan nilai frekuensi dan amplitudo yang tetap. Sinyal tinggi PWM disebut juga dengan istilah on time dan sinyal rendah PWM disebut juga dengan istilah off time.



Gambar 2.8 Periode PWM

Gambar 2.8 menunjukkan sinyal PWM selama 3 (tiga) detik dengan 25% on time dan 75% off time, dengan 0V sebagai amplitudo rendah dan 5V sebagai amplitudo tinggi. Rumus untuk mencari lebar pulsa dan Duty Cycle.

- Rumus Mencari Lebar Pulsa

$$T_{\text{total}} = T_{\text{on}} + T_{\text{off}} \quad (1)$$

- Rumus Mencari Duty Cycle

$$\text{Duty Cycle} = (T_{\text{on}} / T_{\text{total}}) \times 100\% \quad (2)$$

- Rumus Mencari Tegangan Output

$$V_{\text{out}} = \text{Duty Cycle} \times V_{\text{in}} \quad (3)$$

Dimana :

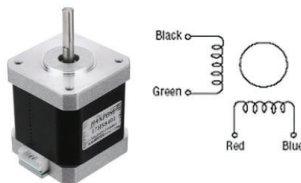
T_{total} = Waktu satu siklus pwm

T_{on} = Waktu positif, T_{off} = Waktu negative

Duty Cycle = lebar Siklus kerja PWM

2.12 Motor Stepper

Motor stepper adalah alat elektromekanis yang mengubah pulsa elektronik menjadi gerakan mekanis yang bersifat diskrit. Pergerakan motor stepper dipengaruhi oleh urutan pulsa yang diterimanya, sehingga memerlukan kontrol khusus untuk mengatur pergerakan dengan pulsa-pulsa yang periodik.



Gambar 2.9 Motor Stepper Nema 17

Berbeda dengan motor DC konvensional, motor stepper menawarkan beberapa keunggulan, seperti sudut rotasi yang sesuai dengan pulsa masukan, torsi penuh saat motor mulai bergerak, serta kemampuan untuk

menentukan posisi dan pergerakan dengan akurasi tinggi [13].

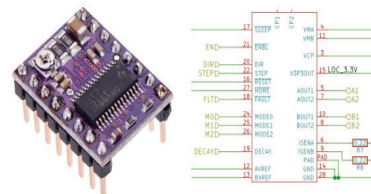
Tabel 2.5 Spesifikasi Motor Stepper Nema 17

Parameter	Spesifikasi
Catu Daya	12V
Arus Operasi	1.2A @ 4V
Sudut Step	1.8°
Jumlah Fasa	2

Tabel 2.5 memperlihatkan spesifikasi dari motor stepper Nema 17. Motor ini membutuhkan catu daya sebesar 12V untuk operasionalnya. Arus operasi motor ini adalah 1.2A pada tegangan 4V, yang menunjukkan kebutuhan arus untuk memastikan motor berfungsi secara optimal pada tegangan yang lebih rendah. Dengan sudut step sebesar 1.8 derajat, motor ini dapat bergerak dalam langkah-langkah kecil yang memungkinkan kontrol yang presisi dan ketepatan posisi, menghasilkan 200 langkah dalam satu putaran penuh. Motor ini juga memiliki dua fasa, yang memberikan kontrol yang lebih baik dan memungkinkan motor untuk bergerak dengan presisi tinggi.

2.13 Driver DRV8825

DRV8825 merupakan driver motor terintegrasi yang cocok untuk berbagai aplikasi seperti printer, scanner, dan peralatan otomatisasi lainnya. Perangkat ini dilengkapi dengan dua driver H-bridge dan satu pengatur mikrostepping yang dirancang untuk mengoperasikan motor stepper bipolar. Blok output driver terdiri dari beberapa MOSFET daya N-channel yang dikonfigurasi sebagai full H-bridges untuk menggerakkan kumparan motor stepper.



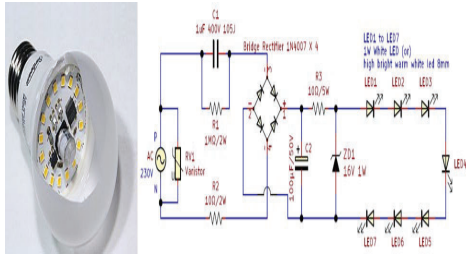
Gambar 2.10 Driver DRV8825

Driver ini mampu menangani arus hingga 2,5 ampere pada setiap output motor (dengan heat sink yang tepat, pada tegangan 24 Volt dan suhu 25°C). DRV8825 dapat diberi daya dengan tegangan suplai antara 8,2 V hingga 45 V dan menyediakan arus keluaran hingga 2,5 A pada skala penuh. Fitur STEP/DIR yang sederhana memudahkan interfacing dengan rangkaian mikrokontroler [14].

Fasilitas internal indexer pada driver ini memungkinkan eksekusi mikrostepping yang sangat akurat tanpa memerlukan kontrol arus dari mikrokontroler. Driver ini juga dilengkapi dengan 3 (tiga) pin MODE yang dapat digunakan untuk mengkonfigurasi mode motor stepper dari full-step hingga 32 mikrostep.

2.14 Lampu LED

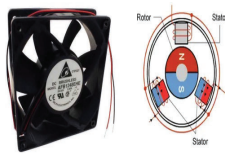
Lampu LED (Light Emitting Diode) adalah jenis lampu yang memanfaatkan dioda semikonduktor untuk memproduksi cahaya. Keunggulan lampu LED tidak hanya pada efisiensinya yang tinggi dalam penggunaan energi, tetapi juga pada umur pakainya yang lebih panjang, yang secara signifikan dapat mengurangi biaya listrik dan pemeliharaan. Lampu LED kini telah banyak digunakan di berbagai tempat, seperti di jalan raya, toko, dan bangunan komersial, menggantikan lampu pijar konvensional.



Gambar 2.11 Diagram Rangkaian Lampu LED

2.15 Kipas DC

Kipas DC mempunyai fungsi mengubah energi listrik menjadi energi elektromagnetik melalui tegangan DC dan induksi elektromagnetik, kemudian energi elektromagnetik menjadi energi mekanik dan terakhir menjadi energi kinetik sehingga menyebabkan saling-baling berputar.



Gambar 2.12 Kipas dan Motor DC

Motor dalam Kipas DC dilengkapi dengan sepasang magnet yang terletak pada bagian yang bergerak (Rotor), serta beberapa besi dengan lilitan kumparan pada bagian yang diam (Stator). Ketika arus listrik mengalir melalui kumparan besi, kumparan tersebut menjadi magnet. Sifat tarik-menolak magnet antara kumparan besi dan sepasang magnet menyebabkan magnet pada bagian Rotor akan berputar secara periodik. Hal ini menghasilkan gerakan yang diterjemahkan menjadi energi mekanik yang digunakan untuk menggerakkan bilah kipas.

2.16 Humidifier

Alat yang berfungsi untuk meningkatkan kelembaban udara dan kualitas udara dalam ruangan serta meredakan gejala kekeringan dan masalah pernapasan.



Gambar 2.13 Humidifier menyebarkan uap air

Humidifier bekerja dengan mengubah air menjadi kabut atau uap dan menyebarkan ke udara. Humidifier

dapat membantu meredakan gejala kulit kering, bibir pecah-pecah, dan masalah pernapasan, serta mengurangi penyebaran virus dan bakteri yang terbawa udara. Namun, penting untuk menjaga tingkat kelembaban yang tepat, karena kelembaban berlebih dapat mendorong pertumbuhan jamur dan lumut. Dianjurkan untuk menjaga tingkat kelembaban dalam ruangan antara 30-50%, dan membersihkan serta mendisinfeksi humidifier secara teratur untuk mencegah penumpukan bakteri dan deposit mineral [15].

3. PERANCANGAN SISTEM

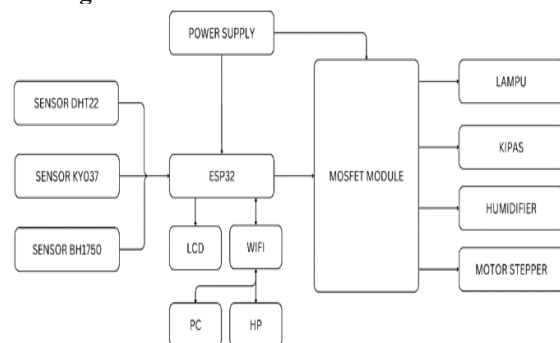
Perancangan sistem kendali ruangan untuk penderita sleep apnea meliputi pembuatan hardware dan juga software yang digunakan untuk memantau dan mengendalikan kondisi ruangan. Meliputi perancangan pada ESP32 dengan sensor DHT22, KY037 dan BH1750 serta aktuator-aktuator seperti LCD, lampu, kipas, motor stepper dan humidifier sedangkan perancangan *software* arduino menggunakan bahasa pemrograman dan *compiler* Arduino IDE.

3.1 Konsep Sistem Kendali Ruangan Untuk Penderita Sleep Apnea

Sensor DHT22 dan BH1750 diletakkan di dalam ruangan sebagai fungsi mendeteksi kondisi dan situasi ruangan penderita Sleep Apnea. Data suhu ruangan, tingkat kelembaban relatif, dan iluminasi ruangan akan dikirim ke broker MQTT secara real-time melalui ESP32. Selain itu, ESP32 juga akan melakukan Subscribe ke broker MQTT untuk mendapatkan data serta mengontrol lampu, kipas dan humidifier. Server juga akan Subscribe topik yang sama untuk mendapatkan semua data sensor untuk menghasilkan tampilan data-data sensor yang dapat diakses melalui aplikasi web. Server ini juga dapat melakukan Publish data kontrol untuk mengendalikan lampu, kipas dan humidifier secara manual.

Saat Sensor KY037 mendeteksi suara dengkur, maka ESP32 akan mengirim sinyal ke driver DRV8825 untuk mengontrol motor stepper yang akan menaikkan posisi bantal dalam upaya untuk menghentikan suara dengkur.

3.2 Diagram Blok Sistem



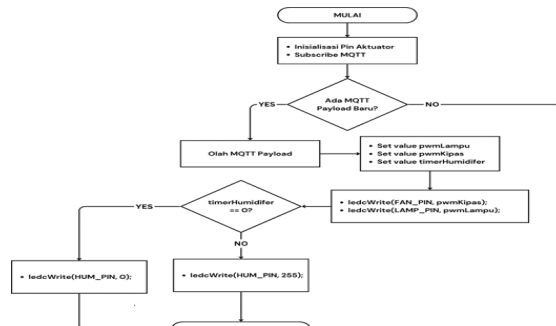
Gambar 3.1 Diagram Blok Sistem

Gambar 3.1 menunjukkan diagram blok dari sistem. Sensor DHT22, Sensor BH1750 dan sensor KY037 akan mengukur kebisingan, iluminasi, temperatur dan kelembapan ruangan. Data-data tersebut kemudian dikirim ke Mikrokontroler ESP32 yang akan mengolah data tersebut untuk mengendalikan aktuator berupa lampu, kipas, humidifier dan motor stepper melalui modul MOSFET. Data yang telah diolah akan dikirim melalui WiFi dan dapat diakses menggunakan Handphone atau komputer. Data tersebut juga ditampilkan di layar LCD.

3.3 Flowchart Kerja Sistem

Flowchart tidak hanya mempermudah proses pemrograman, tetapi juga membantu dalam memahami cara kerja dari sistem sehingga dapat menggambarkan alur kerja sistem secara keseluruhan.

3.3.1 Flowchart Proses Kontrol Manual Aktuator



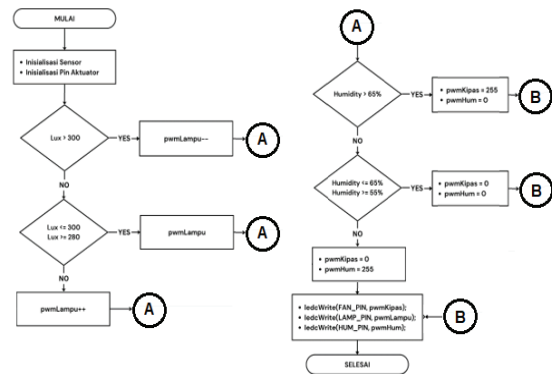
Gambar 3.2 Flowchart Kontrol Manual Aktuator

Dalam gambar 3.2 dapat dilihat proses loop kendali lampu, kipas dan humidifier secara manual. Proses akan melakukan inisialisasi pin pada mikrokontroler ESP32 yang digunakan untuk lampu, kipas dan humidifier, dan juga melakukan Subscribe ke topic “SENSOR/KONTROL” agar mikrokontroler ESP32 dapat menerima payload berupa nilai PWM dan nilai “auto mode” untuk mengendalikan aktuator. Proses kontrol manual hanya akan berjalan saat nilai “auto mode” yang diterima bernilai nol. Untuk mengendalikan lampu dan kipas, proses akan mengirimkan sinyal ke pin aktuator masing-masing sesuai dengan nilai PWM yang diterima. Sedangkan untuk mengendalikan humidifier, proses ini menggunakan nilai timer Humidifier untuk menentukan lama menyalanya humidifier dengan menggunakan sistem interval.

3.3.2 Flowchart Proses Kontrol Otomatis Kondisi Ruangan.

Cara mengendalikan kondisi ruangan yang menggunakan lampu, kipas dan humidifier secara otomatis dengan menggunakan variable global Lux dan Humidity diperlihatkan flowchart 3.3.

Dalam gambar 3.3 dapat terlihat proses ini melakukan inisialisasi pin ESP32 yang digunakan untuk mengendalikan lampu, kipas dan humidifier. Proses ini juga memerlukan payload dari topic “SENSOR/KONTROL” untuk menentukan aktif tidaknya mode otomatis. Proses ini hanya akan berjalan saat nilai autoMode adalah 1.



Gambar 3.3 Flowchart Kontrol Otomatis Lampu, Kipas dan Humidifier

Untuk mengendalikan tingkat pencahayaan ruangan, proses ini menggunakan 3 logika sederhana untuk mengendalikan cahaya lampu. Logika tersebut antara lain :

1. Jika nilai Lux > 300, maka nilai pwmLampu akan terus berkurang.
2. Jika nilai Lux < 280, maka nilai pwmLampu akan terus bertambah.
3. Jika nilai Lux ≥ 280 dan ≤ 300 , maka nilai pwmLampu adalah tetap.

Sedangkan untuk mengendalikan tingkat kelembapan relatif ruangan, proses ini menggunakan 3 logika sederhana untuk mengendalikan kipas dan humidifier. Logika tersebut antara lain :

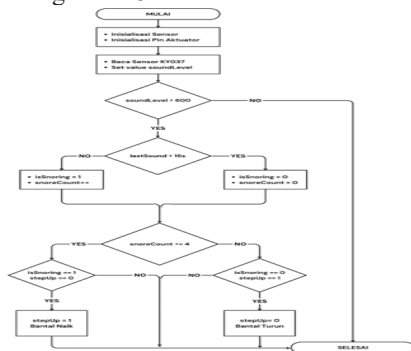
1. Jika nilai Humidity > 65, maka nilai pwmKipas adalah 255 dan nilai pwmHum adalah 0.
2. Jika nilai Humidity < 55 maka nilai pwmKipas adalah 0 dan nilai pwmHum adalah 255.
3. Jika nilai Humidity ≥ 55 dan ≤ 65 , maka nilai pwmKipas adalah 0 dan nilai pwmHum adalah 0.

Dimana Lux dan Humidity adalah nilai tingkat pencahayaan dan kelembapan ruangan yang dibaca sensor BH1750 dan DHT22. Setelah memperoleh nilai PWM lampu, kipas dan humidifier, proses lalu mengirimkan sinyal melalui pin masing-masing aktuator sesuai dengan nilai yang diperoleh.

3.3.3 Flowchart Proses Kontrol Otomatis Motor Stepper.

Pengaturan kontrol motor stepper melalui penggunaan data-data tingkat kebisingan dari sensor KY037. Proses ini akan menginisialisasi sensor KY037, pin DIR dan STEP untuk mengendalikan arah rotasi motor stepper dan hanya akan berjalan saat nilai autoMode adalah 1. Saat terdeteksi nilai soundLevel > 600, sistem akan memulai hitungan mundur. Apabila sensor kembali mendeteksi soundLevel > 600 dalam kurun waktu 15 detik, maka nilai *snoreCount* akan bertambah satu dan nilai *isSnoring* berubah menjadi 1. Sebaliknya, saat tidak ada soundLevel > 600 terdeteksi dalam kurun waktu 15 detik, maka *snoreCount* akan reset menjadi 0 dan nilai *isSnoring* berubah menjadi 0. Saat nilai *snoreCount* ≥ 4 , selanjutnya sistem akan memeriksa

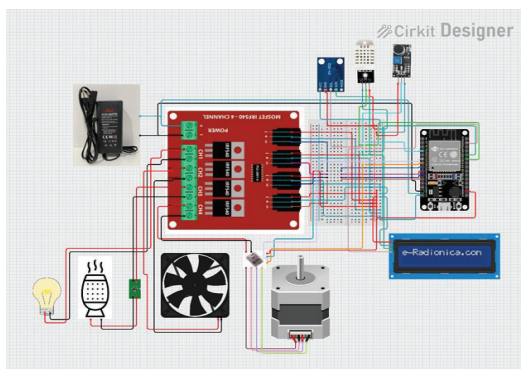
nilai *isSnoring* dan *stepUp* untuk mencegah terjadinya debounce atau pengulangan rotasi motor stepper. Jika nilai *stepUp* adalah 0 (nol) dan *isSnoring* adalah 1, maka motor stepper akan berputar searah jarum jam dan merubah nilai *stepUp* menjadi 1. Saat nilai *stepUp* adalah 1, proses memutuskan bahwa motor stepper telah menaikkan bantal sehingga tidak perlu menaikkan bantal lagi. Saat nilai *snoreCount* < 4, selanjutnya sistem akan memeriksa nilai *isSnoring* dan *stepUp*. Jika nilai *stepUp* adalah 1 dan *isSnoring* adalah 0, maka motor stepper akan berputar berlawanan arah jarum jam dan merubah nilai *stepUp* menjadi 0. Saat nilai *stepUp* adalah 0, proses memutuskan bahwa posisi bantal telah turun, sehingga motor stepper tidak perlu menurunkan bantal kembali. Cara kerja proses kontrol otomatis motor stepper diperlihatkan gambar 3.4.



Gambar 3.4 Flowchart Kontrol Motor Stepper

3.4 Wiring Diagram Sistem Keseluruhan

Berdasarkan wiring diagram dari Gambar 3.5, Mikrokontroler ESP32, MOSFET dan Aktuator mendapatkan daya langsung dari Adaptor 12V. Karena ESP32 menggunakan Base Board yang dilengkapi dengan Voltage Regulator, sehingga aman saat terhubung dengan daya hingga 16V. Sistem ini menggunakan 3 sensor yang terhubung ke ESP32 untuk memantau kondisi ruangan, yaitu sensor DHT22 untuk memantau tingkat kelembaban relatif dan suhu ruangan, sensor BH1750 untuk mengukur tingkat luminasi cahaya dan sensor KY037 untuk mendeteksi kebisingan suara dalam ruangan.



Gambar 3.5 Wiring Diagram Sistem Keseluruhan

Data-data kondisi ruangan diolah oleh ESP32 untuk mengendalikan aktuator berupa lampu, kipas, humidifier dan motor stepper secara otomatis. Lampu akan menyala sesuai dengan tingkat pencahayaan ruangan, humidifier dan kipas bekerja untuk menjaga nilai kelembaban ruangan menggunakan sensor DHT22 agar berada dibatas yang ditentukan, sedangkan motor stepper akan menaikkan bantal apabila suara dengkur terdeteksi. Semua Aktuator tersebut dikendalikan oleh ESP32 menggunakan MOSFET sebagai saklar elektronik.

Sensor BH1750 akan memantau pencahayaan ruangan secara realtime dan menghasilkan nilai pencahayaan dengan satuan lux. Karena sensor ini memiliki rentang pembacaan dari 0 hingga 65335 lux yang terbilang luas, maka hasil pembacaan Sensor BH1750 dalam sistem ini akan dibatasi dari 0 hingga 500 lux saja untuk menjaga simplifikasi data dan menghemat penggunaan memori ESP32.

Saat mode otomatis diaktifkan, sistem ini akan menjaga tingkat pencahayaan ruangan sekitar 280 hingga 300 lux [5] dengan mengatur daya yang diperoleh oleh lampu menggunakan PWM dan MOSFET sebagai saklar elektrik.

Sensor DHT22 akan membaca tingkat kelembaban dan suhu ruangan secara *realtime*. Pada saat nilai kelembaban yang diperoleh oleh sensor DHT22 < 55%, ESP32 akan menyalakan humidifier dan mematikan kipas hingga kelembaban ruangan mencapai 55%. Saat nilai kelembaban yang diperoleh oleh sensor DHT22 > 65%, ESP32 akan mematikan humidifier dan menyalakan kipas untuk mengurangi kelembaban ruangan hingga mencapai 65%. Apabila nilai kelembaban yang diperoleh oleh sensor DHT22 diantara 55% hingga 65%, maka humidifier dan kipas akan mati. Ambang bawah 55% dan ambang atas 65% ditetapkan untuk mencegah terjadinya siklus mati-nyala terlalu cepat yang dapat merusak ESP32, MOSFET ataupun kipas dan humidifier. Kipas dan humidifier dikendalikan menggunakan sinyal PWM dan MOSFET sebagai saklar elektronik yang dikendalikan oleh ESP32.

Sensor KY037 akan memantau tingkat kebisingan secara realtime. Sensor ini akan menghasilkan output berupa sinyal analog yang berbanding lurus dengan besarnya suara yang terdeteksi. Sensor KY037 diatur sensitifitasnya menggunakan potensiometer yang terdapat di sensor tersebut, sehingga sensor ini akan memperoleh sinyal antara 490 hingga 520 untuk tingkat kebisingan normal.

Pada saat suara menghasilkan sinyal > 600 terdeteksi, sistem akan memulai timer sebesar 15 detik [16]. Apabila dalam jangka waktu tersebut sensor kembali menghasilkan sinyal > 600, maka sistem memutuskan bahwa suara yang terdeteksi adalah dengkur, kemudian timer akan reset kembali menjadi 15 detik dan sistem mulai menghitung jumlah dengkur yang terdeteksi. Proses ini terus dilakukan secara berulang hingga jumlah

dengkuran yang terdeteksi mencapai lebih dari 3x, yang kemudian ESP32 akan mengendalikan motor stepper untuk menaikkan posisi bantal.

Pada saat sensor tidak mendeteksi suara dengan sinyal > 600 dalam 15 detik, sistem memutuskan dengkuran telah berhenti dan mereset jumlah dengkuran yang terdeteksi dan mengendalikan motor stepper untuk menurunkan bantal ke posisi semula.

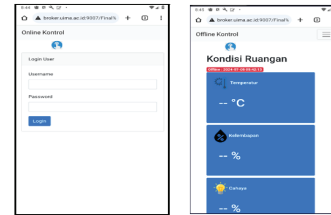
Adapun konfigurasi pin pada sensor BH1750, DHT22, KY037, Lampu, kipas, humidifier dan motor stepper adalah sbb :

- a) Sensor BH1750
 - Pin SCL sensor terhubung dengan pin 21 ESP32.
 - Pin SDA LCD terhubung dengan pin 22 ESP32.
- b) Sensor DHT22
 - Pin S (Sinyal) sensor terhubung pin 19 ESP32.
- c) Sensor KY037
 - Pin AO (Analog Output) pada sensor terhubung dengan pin 34 pada ESP32.
- d) Lampu
 - Kutub Positif dan Kutub Negatif socket lampu terhubung dengan pin output positif dan pin output negatif pada channel 1 modul MOSFET.
- e) Kipas
 - Pin positif kipas terhubung dengan pin output positif channel 2 modul MOSFET.
 - Pin negatif kipas terhubung dengan pin output negatif channel 2 modul MOSFET.
- f) Humidifier
 - Kutub (+) step down converter terhubung pin output positif channel 3 modul MOSFET.
 - Kutub (-) step down converter terhubung pin output negatif channel 3 modul MOSFET.
 - Menghubungkan humidifier dengan step down converter menggunakan kabel USB.
- g) Motor Stepper
 - Pin positif fasa satu pada motor stepper terhubung dengan pin A1 pada DRV8825.
 - Pin negatif fasa satu pada motor stepper terhubung dengan pin A2 pada DRV8825.
 - Pin positif fasa dua pada motor stepper terhubung dengan pin B1 pada DRV8825.
 - Pin negatif fasa dua pada motor stepper terhubung dengan pin B2 pada DRV8825.

3.5 Perancangan User Interface

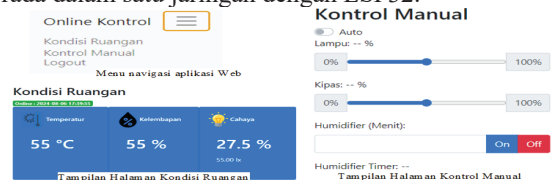
User interface yang digunakan untuk kontrol dan monitoring sistem adalah aplikasi Web yang dibuat menggunakan HTML, tetapi menggunakan protokol pengiriman data yang berbeda. Untuk Online kontrol, protokol yang digunakan adalah MQTT Over Websocket, yaitu dengan melakukan handshake dengan broker, Online kontrol akan terus mendapat payload yang dikirim oleh ESP32. Sedangkan untuk Offline kontrol, akan digunakan protokol HTTP request. Protokol ini digunakan karena Offline kontrol berada dalam jaringan yang sama dengan ESP32, sehingga HTTP request dapat dilakukan

dengan cepat (spam) tanpa gangguan keamanan, seperti yang terjadi saat melakukan spam HTTP request melalui internet.



Gambar 3.6 Tampilan awal Online kontrol (kiri) dan Offline kontrol (kanan)

Dari gambar 3.6 dapat terlihat bahwa Online kontrol memiliki halaman login karena aplikasi Web dapat diakses oleh siapapun yang mengetahui Web Address aplikasi tersebut. Sedangkan, Offline kontrol tidak memerlukan karena aplikasi tersebut hanya bisa diakses saat alat yang digunakan untuk mengakses aplikasi berada dalam satu jaringan dengan ESP32.



Gambar 3.7 Menu navigasi aplikasi Web, Halaman Kondisi Ruang dan Halaman Kontrol Manual

Tampilan Online Kontrol pada aplikasi Web menampilkan Menu Kondisi Ruang, Kontrol Manual dan Logout. Halaman kondisi ruangan menampilkan data temperatur, kelembaban dan tingkat pencahayaan. Halaman ini juga menampilkan status konektivitas ESP32 dengan server. Untuk halaman Kontrol Manual memiliki 3 parameter input untuk mengendalikan lampu, kipas dan humidifier secara manual. Jika input lampu dan kipas menggunakan slider dengan jarak 0 sampai 100, untuk input humidifier berupa angka dan tombol On dan Off. Saat tombol On ditekan, aplikasi akan mengirim status on untuk menyalakan humidifier dan input angka sebagai countdown untuk mematikan humidifier secara otomatis dalam satuan menit. Tombol Off digunakan untuk mematikan humidifier secara manual dan mengabaikan countdown yang sedang berjalan. Selain itu, di halaman Kontrol Manual juga terdapat toggle untuk mengaktifkan mode otomatis. Saat toggle dalam keadaan menyala, itu menandakan bahwa sistem sedang berjalan dalam mode otomatis. Saat mode otomatis menyala, semua kontrol manual tidak dapat digunakan karena lampu, kipas dan humidifier dapat mengatur kelembaban dan cahaya ruangan secara otomatis.

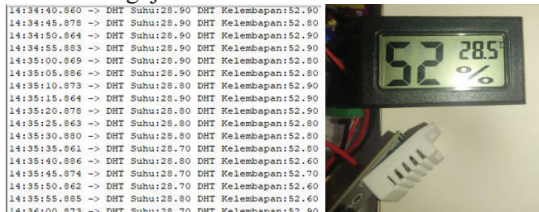
4. HASIL DAN ANALISIS

Hasil pengujian digunakan untuk menganalisis kinerja setiap komponen sehingga sistem kontrol dan monitoring ruangan penderita sleep apnea bekerja optimal. Pengujian seluruh sistem membantu mengevaluasi kinerja dan keberhasilan sistem.

4.1 Pengujian Sensor-Sensor

Pada tahap ini, nilai pembacaan sensor DHT22 akan dibandingkan dengan Hygrometer sedangkan sensor BH1750 akan dibandingkan dengan Lux Meter atau Light Meter.

➤ Hasil Pengujian Sensor DHT22



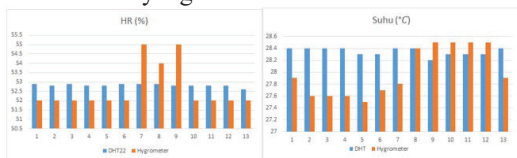
Gambar 4.1 Pembacaan DHT22 & Hygrometer

Gambar 4.1 menunjukkan hasil pengukuran nilai suhu dan kelembapan relative (HR) dimana nilai DHT22 akan dibandingkan dengan nilai Hygrometer. Kedua sensor diletakkan berdekatan agar dapat mengukur kondisi yang sama.

Tabel 4.1 Tabel hasil DHT22 & Hygrometer

t	Suhu (°C)		HR (%)	
	DHT22	Hygrometer	DHT22	Hygrometer
0	28,4	27,9	52,9	52
5	28,4	27,6	52,8	52
10	28,4	27,6	52,9	52
15	28,4	27,6	52,8	52
20	28,3	27,5	52,8	52
25	28,3	27,7	52,9	52
30	28,4	27,8	52,9	55
35	28,4	28,4	52,9	54
40	28,2	28,5	52,8	55
45	28,3	28,5	52,8	52
50	28,3	28,5	52,8	52
55	28,3	28,5	52,8	52
60	28,4	27,9	52,6	52
Rata-Rata	28,35	28	52,82	52,62

Hasil perbandingan diambil setiap 5 detik untuk memastikan data yang dihasilkan konsisten.



Gambar 4.2 Chart Perbandingan Pengukuran Suhu dan Kelembapan

Berdasarkan dari grafik perbandingan pada gambar 4.2, dapat diketahui nilai error HR dan selisih rata-rata suhu antara sensor DHT22 dan Hygrometer dengan menggunakan rumus sebagai berikut :

$$\text{Percent Error HR} = \left| \frac{52,82 - 52,62}{52,62} \right| \times 100\% = 0,38\%$$

$$\text{Selisih Rata - rata Suhu} = |28,35 - 28| = 0,35\text{ }^{\circ}\text{C}$$

Berdasarkan datasheet DHT22, sensor tersebut memiliki rentang error pengukuran suhu sebesar $\pm 0,5^{\circ}\text{C}$ dan pengukuran HR sebesar 2%. Karena hasil perbandingan masih dalam rentang error pengukuran, maka dapat disimpulkan bahwa sensor DHT22 bekerja dengan baik.

➤ Hasil Pengujian Sensor BH1750

13:36:28.319 -> BH1750 Cahaya:30.00
13:36:28.819 -> BH1750 Cahaya:30.00
13:36:29.292 -> BH1750 Cahaya:30.00
13:36:29.840 -> BH1750 Cahaya:30.83
13:36:30.343 -> BH1750 Cahaya:30.00
13:36:30.813 -> BH1750 Cahaya:30.00
13:36:31.331 -> BH1750 Cahaya:30.00
13:36:31.847 -> BH1750 Cahaya:30.00
13:36:32.320 -> BH1750 Cahaya:30.00
13:36:32.836 -> BH1750 Cahaya:30.00
13:36:33.305 -> BH1750 Cahaya:30.83
13:36:33.852 -> BH1750 Cahaya:30.83
13:36:34.311 -> BH1750 Cahaya:30.00
13:36:34.828 -> BH1750 Cahaya:30.00
13:36:35.348 -> BH1750 Cahaya:30.83



Gambar 4.3 Perbandingan hasil pembacaan BH1750 dengan Lux Meter

Penempatan sensor BH1750 dan Lux Meter saat pengukuran harus berdekatan agar didapat hasil pengukuran kondisi pencahayaan yang sama.

Tabel 4.2 Tabel hasil BH1750 dan Lux Meter

NO	Lux	
	BH1750	Lux Meter
1	29,17	30
2	34,17	35
3	40,83	40
4	44,17	45
5	51,67	50
6	57,50	55
7	62,50	60
8	64,17	65
9	70,83	70
10	77,50	75
11	82,50	80
12	86,67	85
13	95,83	90
Rata-rata	61,35	60,00

Tabel 4.2 menampilkan perbandingan tingkat pencahayaan yang terdeteksi sensor BH1750 dan Lux Meter. Keragaman intensitas cahaya digunakan untuk memastikan bahwa sensor BH1750 dan Lux Meter dapat memberikan hasil pengukuran yang konsisten dan tepat dalam berbagai kondisi pencahayaan.

Berdasarkan hasil dari perbandingan tersebut, dapat diketahui nilai error sensor BH1750 dan Lux meter dengan menggunakan rumus sbb :

$$\text{Percent Error} = \left| \frac{61,35 - 60}{60} \right| \times 100\% = 2,25\%$$

Karena nilai error hasil pengujian masih berada di bawah 20% yang merupakan rentang error sensor BH1750, maka dapat disimpulkan sensor BH1750 bekerja dengan baik.

➤ Hasil Pengujian Sensor KY037

Pengujian dan kalibrasi sensor KY037 menggunakan beberapa data suara dengkur yang diperoleh dari sumber online, seperti soundsnap.com [16]. *SnoreCount* bertambah saat ada 2 (dua) suara terdeteksi yang melebihi threshold di bawah 15 detik. Jika suara terdeteksi kurang dari threshold atau waktu antara dua suara lebih 15 detik, maka *snorecount* akan reset menjadi 0 (nol).

Suara A, B dan D berhasil dideteksi oleh sensor secara baik dari jarak 4 cm dan 7 cm, tetapi gagal mendeteksi beberapa dengkur dalam jarak 10 cm. Suara C hanya berhasil terdeteksi 1 suara dengkur karena jarak antara dengkur yang melebihi threshold lebih dari

15 detik yang menyebabkan *snoreCount* dihitung ulang kembali dari 0.

Tabel 4.3 Kalibrasi Sensor KY037

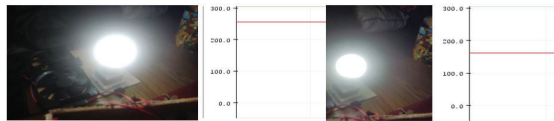
NO	Data Suara	Durasi (detik)	Jarak (cm)	snoreCount
1	Suara_A	34	4	5
2	Suara_A	34	7	5
3	Suara_A	34	10	3
4	Suara_B	32	4	5
5	Suara_B	32	7	5
6	Suara_B	32	10	4
7	Suara_C	53	4	1
8	Suara_C	53	7	1
9	Suara_C	53	10	0
10	Suara_D	61	4	8
11	Suara_D	61	7	8
12	Suara_D	61	10	5

4.2 Pengujian Kontrol Manual

Pengujian lampu, kipas, dan humidifier secara manual dilakukan untuk memastikan bahwa ketiga perangkat ini berfungsi dengan baik dan sesuai dengan kebutuhan. Hasil dari pengujian ini dapat digunakan untuk memastikan bahwa sistem kontrol manual yang melibatkan ketiga perangkat ini dapat bekerja secara optimal.

➤ Hasil Pengujian Kontrol Lampu Manual

Untuk melakukan pengujian lampu secara manual, dilakukan dengan menggerakkan slider pada input lampu dan mengamati perubahan cahaya lampu serta besar sinyal PWM yang dikirim oleh ESP32 melalui serial monitor Arduino IDE.

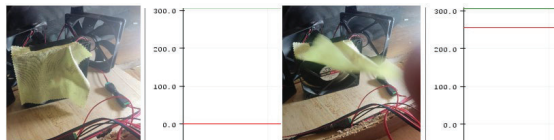


Gambar 4.4 Kontrol lampu dengan input 100% (Atas) dan 50% (Bawah)

Perbedaan terang cahaya lampu terlihat pada Gambar 4.4. Tampilan serial plotter menunjukkan perubahan besar sinyal PWM saat input dari kontrol berubah. Pada saat slider input Lampu digerakkan ke 100%, lampu menyala dengan terang dengan sinyal sebesar 255 PWM sehingga lampu menerima tegangan penuh sebesar $\pm 12V$. Saat slider input Lampu digerakkan 50%, menyebabkan cahaya lampu mulai meredup karena sinyal PWM yang dikirim menurun hingga mencapai 128 PWM sehingga lampu hanya menerima tegangan sebesar $\pm 6V$.

➤ Hasil Pengujian Kontrol Manual Kipas

Pengujian kipas dilakukan dengan menggerakkan slider pada input kipas dan mengamati perubahan kecepatan putaran dan besar sinyal PWM yang dikirim oleh ESP32 melalui serial monitor Arduino IDE.



Gambar 4.5 Kipas Input 0% (kiri) dan 100% (kanan)

Gambar 4.5 menampilkan perbedaan kecepatan putaran kipas dengan input 100% dan 0%. Tampilan serial plotter menunjukkan perubahan besar sinyal PWM

yang dikirim ESP32 saat input dari kontrol berubah. Saat slider input kipas digerakkan ke value 100%, kipas berputar dengan kecepatan penuh ESP32 mengirimkan sinyal sebesar 255 PWM sehingga kipas menerima tegangan penuh sebesar $\pm 12V$. Pada saat slider digerakkan ke value 0% secara perlahan, kecepatan kipas akan menurun karena sinyal PWM yang dikirim juga menurun hingga mencapai 0 PWM sehingga kipas tidak menerima tegangan 0 volt.

➤ Hasil Pengujian Kontrol Manual Humidifier

Pengujian humidifier dilakukan dengan memasukkan durasi menyala humidifier yang diinginkan dalam satuan menit dan menekan tombol On untuk memulai timer. Tombol Off digunakan untuk mematikan humidifier secara manual.



Gambar 4.6 Humidifier dalam keadaan menyala (kiri) dan mati setelah timer selesai (kanan)

ESP32 mengirimkan sinyal PWM sebesar 255 (nilai maksimum) agar humidifier aktif dan bekerja. Ketika timer menghitung mundur hingga mencapai 0 atau tombol OFF ditekan oleh *user*, maka ESP32 mengubah sinyal PWM yang dikirimkan menjadi 0, menyebabkan humidifier berhenti beroperasi atau tidak aktif.

4.3 Hasil Pengujian Otomatis Lampu, kipas dan Humidifier

Perhatikan tabel 4.4, yang merupakan pengujian terhadap otomatis kipas dan humidifier.

Tabel 4.4 Pengujian Otomatis Kipas dan Humidifier

NO	Target HR	UI HR	Kipas	Humidifier
1	55% - 65%	76.84%	On	Off
2	55% - 65%	60.31%	Off	Off
3	55% - 65%	45.84%	Off	On

Saat kelembaban relatif (HR) aktual bernilai 76.84% (melebihi target) dari 55%-65%, maka kipas nyala untuk menurunkan kelembaban dan humidifier dimatikan. Setelah (HR) aktual bernilai 60.31% (dalam rentang target) dari 55%-65%, maka kipas dan humidifier dimatikan karena kondisi sudah sesuai target. Saat HR aktual bernilai 45.84% (di bawah rentang target) dari 55%-65%, maka kipas dimatikan dan humidifier dinyalakan agar kelembaban meningkat hingga mencapai target.

Saat tingkat pencahayaan aktual (17 lux) jauh di bawah rentang target yang diinginkan (280-300 lux), terang cahaya lampu akan meningkat. Setelah tingkat pencahayaan aktual (283 lux) mencapai rentang target yang diinginkan (280-300 lux), status lampu tetap karena pencahayaan sudah sesuai dengan target. Saat tingkat pencahayaan aktual (500 lux) jauh di atas rentang target yang diinginkan (280-300 lux), sehingga lampu

diredupkan untuk menurunkan tingkat pencahayaan ruangan hingga mencapai rentang target.

Tabel 4.5 Pengujian Lampu Otomatis

NO	Target Lux	UI Lux	Lampu
1	280-300	17	<i>Menerang</i>
2	280-300	283	<i>Tetap</i>
3	280-300	500	<i>Meredup</i>

4.4 Hasil Pengujian Motor Stepper

Motor Stepper untuk pengujian adalah Nema 17 yang dilengkapi dengan lead screw sebesar 8 mm. motor bergerak sebesar 8 mm untuk setiap putaran dengan sudut langkah 1.8°, sehingga memerlukan 200 step untuk melakukan satu putaran [17]. Jumlah putaran dan step untuk menggerakkan nut sebesar 100 mm atau 10 cm dapat ditemukan dengan rumus :

$$\text{Step} = \frac{360^\circ}{1.8^\circ} = 200 \text{ step}$$

$$\text{Putaran} = \frac{100\text{mm}}{8\text{mm}} = 12.5 \text{ putaran}$$

$$\text{Total Step} = \text{Putaran} \times \text{Step} = 12.5 \times 200 = 2500 \text{ step}$$



Gambar 4.7 SnoreCount mencapai nilai 4 (Kiri) dan pengujian motor stepper (kanan)

LCD menampilkan nilai *snoreCount* lebih besar dari 3, sehingga stepper mulai berputar searah jarum jam sebanyak 12.5 putaran dan nut akan naik sebesar 10 cm. Jika suara dengkur tidak terdeteksi dalam jangka waktu 15 detik, *snoreCount* akan me-reset nilai menjadi 0 dan stepper berputar berlawanan arah jarum jam dan nut akan turun sebesar 10 cm ke posisi awal.

5. SIMPULAN

Hasil dari perancangan, pembuatan, dan pengujian alat menunjukkan :

1. Selisih rata-rata suhu pada sensor DHT22 dan Hygrometer adalah 0,35°C dan persentase error HR adalah 0,38% sedangkan Nilai error pada sensor BH1750 dan sensor Lux meter sebesar 2,25%.
2. Data-data dari sensor-sensor dikirim ke database Server menggunakan protokol MQTT.
3. Pengendalian Online menggunakan protokol MQTT Over Websocket dengan melakukan handshake dengan broker, sehingga akan terus mendapat payload yang dikirim oleh ESP32. Sedangkan pengendalian Offline menggunakan protokol HTTP request.

DAFTAR PUSTAKA

1. A. M. Osman, S. G. Carter, J. C. Carberry, and D. J. Eckert, "Obstructive sleep apnea: current perspectives,".

2. A. Lax, S. Colamartino, P. Banfi, and A. Nicolini, "Sleep-related breathing disorders and non-invasive ventilation,".
3. E. Supriyadi and S. Dinaryati, "Rancang Bangun System Monitoring dan Kendali Listrik Rumah Tangga Berbasis ESP8266 NodeMCU," Jurnal Elektro, vol. 4, no. 4, 2020.
4. C. Lou et al., "Human respiratory monitoring based on schottky resistance humidity sensors," Materials (Basel).
5. S. Lappharat, N. Taneepanichskul, S. Reutrakul, and N. Chirakalwasan, "Effects of Bedroom Environmental Conditions on the Severity of Obstructive Sleep Apnea,".
6. M. E. Billings, L. Hale, and D. A. Johnson, "Physical and Social Environment Relationship With Sleep Health and Disorders,".
7. E. Rustami, R. Fitria Adiati, M. Zuhri, and A. Arif Setiawan, "Uji Karakteristik Sensor Suhu dan Kelembaban Multi-Channel Menggunakan Platform Internet of Things (IoT),".
8. C. W. Hew and M. Ramasamy, "Development of a IoT Based Low Cost Wearable Smart Health Monitoring System for Elderly,".
9. F. Marwita, A. Ariman, M. Febriansyah, and I. Iswoko, "Rancang Bangun Alat Ukur Kondisi Ruang Inkubator Bayi berbasis Komputer PC dan Aplikasi Android,".
10. H. Rahman, A. F. Adziima, and S. F. Mujiyanti, "Otomatisasi Lampu Selasar Departemen Instrumentasi Menggunakan Light Intensity Detector Bh1750 Berbasis Expert System,".
11. A. A. Al-Farzaq, H. Harmadi, and E. Elvaswer, "Wireless Data Logger Instrument for Indoor Acoustic Quality Measurement Based on Noise Background, Sound Distribution and Reverberation Time,".
12. A. Muis and M. I. Thirafi, "Rancang Bangun Robot Pembersih Lantai dengan Sensor Suara,".
13. Ooznest.co.uk, "Nema 17 stepper motor," Ooznest.Co.Uk.
14. R. Kusmayanto, G. Heryana, and Jatira, "Perancangan Sistem Mekanik Penggerak Sumbu X Dan Y Pada Mesin CNC (Computer Numerical Control) CO2 Laser Cutting,".
15. W.-T. Liu et al., "The impacts of ambient relative humidity and temperature on supine position-related obstructive sleep apnea in adults,".
16. T. Khan, "A deep learning model for snoring detection and vibration notification using a smartwearable gadget,".
17. S. Romadhon and A. Multi, "Design and Development of Real-Time Monitoring & Controlling Infant Incubator with Tilt Stabilizer Using Raspberry Pi Remotely Controlled via PC and Smartphone to Reduce Tilt during Baby Transfer,".