

# IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR PADA SISTEM PEMANTAU SUHU DAN KELEMBAPAN RUANG SERVER MENGGUNAKAN PROTOKOL MQTT BERBASIS IOT

Aditya Aziz Fikhri<sup>1</sup>, Nurdin Nurdin<sup>2</sup>

<sup>1,2</sup>Prodi Magister Teknologi Informasi Universitas Maikussaleh, Jl. Kampus Bukit Indah, Lhokseumawe, Aceh, 24355.

Received: 25 September 2024

Accepted: 5 Oktober 2024

Published: 12 Oktober 2024

## Keywords:

*Internet of Things*, KNN, MQTT, Suhu, Kelembapan.

## Correspondent Email:

aditfreedom11@gmail.com

**Abstrak.** Pemantauan kondisi ruang server sangat penting untuk mencegah kegagalan sistem, terutama terkait suhu dan kelembapan. Penelitian ini mengacu pada standar ruang server di Indonesia, yaitu suhu 21°C - 23°C dan kelembapan 45% - 60%. Menggunakan mikrokontroler ESP32 dan sensor DHT22, data dikirim melalui protokol MQTT dengan broker Mosquitto pada server lokal, dan dashboard pengguna dibangun dengan Node-Red. Data latih mencakup referensi standar dan data dummy untuk meningkatkan akurasi klasifikasi. Hasil pengujian menunjukkan selisih 7.13°C pada suhu dan 21.22% pada kelembapan setelah pendingin ruangan dimatikan selama 1 jam. Klasifikasi menunjukkan 100% Normal saat pendingin hidup dan 100% Tidak Normal saat mati. Pengujian waktu eksekusi dengan *access point* dan hotspot Wi-Fi pada *smartphone* menunjukkan tidak ada perubahan. Waktu pembacaan sensor 4 ms, waktu eksekusi KNN 22 ms, dan pengiriman ke server lokal MQTT 14 ms, semuanya kurang dari 1 detik.

**Abstract.** Monitoring the condition of the server room is very important to prevent system failure, especially related to temperature and humidity. This study refers to the server room standards in Indonesia, namely a temperature of 21°C - 23°C and a humidity of 45% - 60%. Using an ESP32 microcontroller and a DHT22 sensor, data is sent via the MQTT protocol with a Mosquitto broker on the local server, and a user dashboard is built with Node-Red. Training data includes standard references and dummy data to improve classification accuracy. The test results showed a difference of 7.13°C in temperature and 21.22% in humidity after the air conditioner was turned off for 1 hour. The classification showed 100% Normal when the cooler was on and 100% Abnormal when it was off. Testing the execution time with an *access point* and Wi-Fi hotspot on a *smartphone* showed no change. The sensor reading time was 4 ms, the KNN execution time was 22 ms, and the sending to the local MQTT server was 14 ms, all less than 1 second.

## 1. PENDAHULUAN

Kondisi ruang *server* sangat penting untuk dilakukan pemantauan, karena proses penyimpanan data dan informasi pada layanan Teknologi Informasi (TI) dilakukan pada sebuah *server*. Apabila hal tersebut diabaikan maka hal ini dapat menyebabkan kegagalan

perangkat *server* yang implikasinya berdampak pada layanan TI pada sebuah bisnis. Adapun hal mendasar yang menjadi tantangan utama dalam pengelolaan *server* agar tidak terjadi hal yang tidak diinginkan yaitu pada aspek suhu dan kelembapan pada ruang *server*. Suhu dan kelembapan merupakan parameter kritis yang

harus dipantau dan dilakukan manajemen secara cermat dalam operasional pusat data dan informasi. Suhu rata-rata *server* di Indonesia adalah 21°C -23°C (70°F -74°F), dengan kisaran kelembapan rata-rata pada 45%-60% [1]. Suhu yang terlalu rendah dapat menyebabkan kinerja *server* melambat atau bahkan berhenti, sebaliknya juga suhu yang terlalu tinggi akan menyebabkan kinerja *server* menjadi terlalu panas (*overheat*) dan akhirnya mati. Sementara kelembapan yang tinggi dapat menyebabkan korosi atau korsleting Listrik pada perangkat. Dampak dari kondisi ruang *server* yang tidak optimal ini bukan hanya berdampak pada kinerja server, tetapi juga dapat merusak data, mengganggu ketersediaan layanan, dan berujung pada peningkatan biaya operasional pada *server*.

Suatu konsep yang dapat digunakan untuk melakukan monitoring terhadap kondisi suhu dan kelembapan pada ruang *server* adalah *Internet of Things* (IoT), yaitu sebuah proses otomatisasi dimana data dapat dikirim melalui jaringan internet tanpa memerlukan interaksi manusia ke komputer [2]. Kehadiran IoT menjadi solusi kepada sistem administrator untuk melakukan pemantauan suhu dan kelembapan ruang *server* dari jarak jauh yang dapat dipantau secara *real-time*.

Salah satu protokol yang sering digunakan dalam komunikasi IoT adalah MQTT berdasarkan the Eclipse 2023 IoT & Edge Developer Survey. MQTT (*Message Queue Telemetry Transport*) merupakan protokol komunikasi yang bersifat *publish-subscribe* yang merupakan protokol yang ringan dalam melakukan untuk berkomunikasi [3]. Protokol ini dirancang untuk berfokus pada meminimalkan penggunaan *bandwidth* dan sumber daya yang terbatas. Sebuah studi yang dilakukan dengan menguji beban pada MQTT server yang menggunakan 10.000 payload dengan 50 client membutuhkan 22.63% penggunaan CPU dibandingkan dengan protokol HTTP yang membutuhkan 87% penggunaan CPU, kemudian dari aspek kecepatan pengiriman pesan, protokol MQTT lebih unggul hanya dengan membutuhkan 3.24 detik saja pesan terkirim ke server sedangkan HTTP membutuhkan 117.344 detik lebih lama dibandingkan dengan MQTT [4]. Maka atas dasar hal itu protokol MQTT digunakan dalam penelitian ini.

Dalam memberikan keputusan dari hasil deteksi suhu dan kelembapan dari sensor yang digunakan, dibutuhkan sebuah algoritma untuk mendukung hasil tersebut. K-Nearest Neighbor (KNN) merupakan algoritma klasifikasi yang populer dalam *machine learning* yang bekerja dengan mencari k atau tetangga terdekat yang identik pada data *training* [5]. Dalam hal suhu dan kelembapan ruang *server*, standar suhu dan kelembapan dapat dijadikan data *training* dalam penelitian ini. KNN memiliki keunggulan akan kesederhanaan dan cukup efektif dalam melakukan klasifikasi jika data *training* yang diberikan cukup banyak.

Penelitian ini bertujuan untuk mengimplementasi pemantauan jarak jauh pada ruang *server* dengan menggunakan konsep IoT. Secara khusus, tujuan penelitian ini adalah: (1) Mengeksplorasi kemampuan protokol MQTT dalam kecepatan mengirimkan data hasil deteksi sensor suhu dan kelembapan. (2) Menguji keakuratan metode KNN pada keadaan ruang *server* dalam beberapa kondisi. (3) Mengembangkan dashboard visualisasi data untuk memudahkan pemantauan kondisi ruang *server* secara *real-time*.

## 2. TINJAUAN PUSTAKA

### 2.1. Internet of Things (IoT)

Internet of Things adalah konsep yang bertujuan untuk memaksimalkan keuntungan dari konektivitas yang terus-menerus terhubung ke internet. Konsep ini beroperasi melalui pemrograman berbasis argumen, di mana setiap instruksi yang diberikan membentuk koneksi yang memungkinkan mesin menjalankan tugas secara otomatis tanpa intervensi manusia [6]. Perkembangan teknologi IoT menawarkan berbagai manfaat bagi kehidupan manusia, seperti mempermudah pengawasan dan pengelolaan pekerjaan secara jarak jauh [7].

Dalam konteks pemantauan suhu dan kelembapan, IoT memiliki peran penting seperti penggunaan sensor terintegrasi yang terhubung ke jaringan. Sensor-sensor ini secara *real-time* memantau kondisi lingkungan, mengumpulkan data yang kemudian dikirim ke sistem untuk dianalisis. Hal ini memungkinkan pengelola untuk memantau kondisi secara terus-menerus. Jika suhu atau kelembapan melebihi batas yang ditentukan, sistem IoT dapat mengirimkan peringatan kepada pengguna,

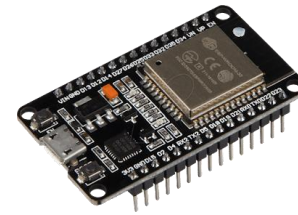
sehingga mereka dapat mengambil tindakan cepat untuk mencegah kerusakan pada peralatan. Data yang terkumpul juga dapat dianalisis untuk melihat tren jangka panjang, membantu dalam perencanaan dan pengambilan keputusan terkait infrastruktur TI. Dengan demikian, IoT meningkatkan efisiensi, keamanan, dan keandalan ruang server.

## 2.2. Protokol MQTT

MQTT adalah protokol komunikasi publish-subscribe yang khusus dirancang untuk aplikasi IoT, dengan pendekatan *machine-to-machine* (M2M) [3]. Fokus utama dari protokol ini adalah meminimalkan penggunaan bandwidth jaringan dan sumber daya perangkat untuk memastikan pengiriman pesan yang handal [8]. MQTT pertama kali dikembangkan pada tahun 1999 oleh Andy Stanford-Clark dari IBM dan Arlen Nipper dari Arcom, dengan tujuan mengurangi penggunaan sumber daya pemrosesan dan bandwidth. Dalam sistem MQTT, terdapat dua peran utama: publisher, yang bertanggung jawab untuk mengirim pesan, dan subscriber, yang menerima pesan tersebut [9]. Selain itu, terdapat broker yang berfungsi sebagai perantara, mengirimkan pesan dari publisher ke subscriber yang telah berlangganan topik yang sama.

## 2.3. ESP32

ESP32 adalah mikrokontroler yang dikembangkan oleh Espressif Systems dan merupakan pengembangan dari mikrokontroler ESP8266. Dengan modul WiFi on-chip, ESP32 mendukung pembuatan aplikasi Internet of Things tanpa perlu menambahkan modul WiFi terpisah [10]. Sebagai salah satu modul mikrokontroler yang populer dalam pengembangan perangkat keras dan IoT, ESP32 menawarkan berbagai kemampuan dan fitur canggih. Fitur utamanya mencakup konektivitas nirkabel yang mendukung standar Wi-Fi 802.11 b/g/n, memungkinkan perangkat terhubung ke jaringan internet atau lokal. Selain itu, ESP32 juga mendukung Bluetooth *Classic* dan Bluetooth Low Energy (BLE), yang memudahkan komunikasi dengan perangkat lain seperti smartphone atau sensor.



Gambar 1 : ESP32

Berdasarkan referensi dari datasheetnya, mikrokontroler ini dilengkapi dengan prosesor dual-core Tensilica LX6 yang dapat beroperasi hingga 240 MHz, memberikan performa yang cukup untuk berbagai aplikasi. Dalam hal memori, ESP32 memiliki memori flash internal dan Random Access Memory (RAM) sebesar 520 KB, yang cukup untuk menjalankan berbagai program dan aplikasi. Terdapat juga 34 pin GPIO (*General Purpose Input/Output*) yang dapat digunakan untuk menghubungkan berbagai sensor, aktuator, dan komponen elektronik lainnya. Selain itu, ESP32 mendukung berbagai antarmuka seperti SPI, I2C, UART, ADC, DAC, dan PWM, memungkinkan integrasi dengan perangkat eksternal. Di sisi keamanan, ESP32 memiliki fitur yang kuat, termasuk enkripsi hardware, secure boot, dan flash encryption untuk melindungi data dan firmware.

## 2.4. DHT22

Sensor DHT-22 adalah perangkat yang digunakan untuk mendeteksi suhu dan kelembaban, dengan keluaran berupa sinyal digital yang dikonversi dan dihitung oleh mikrokontroler terintegrasi 8-bit [11]. Sensor ini bekerja berdasarkan prinsip kapasitansi untuk mengukur kelembaban relatif dan menggunakan termistor untuk mengukur suhu. Kapasitansi elemen sensor berubah seiring dengan variasi kelembaban, dan perubahan ini dikonversi menjadi sinyal digital oleh sirkuit internal sensor. Untuk pengukuran suhu, DHT-22 memanfaatkan termistor yang resistansinya berubah sesuai suhu, dan resistansi ini kemudian diubah menjadi sinyal digital. Sensor ini berkomunikasi dengan mikrokontroler melalui protokol One-Wire, yang hanya memerlukan satu pin untuk transmisi data, ditambah dengan pin power dan ground. Data yang dikirimkan mencakup dua byte untuk kelembaban, dua byte untuk suhu, dan satu byte untuk checksum, yang berfungsi memastikan

integritas data dengan memverifikasi nilai checksum yang diterima.



Gambar 2 : DHT22

### 2.5. K-Nearest Neighbor

*K-Nearest Neighbor* (KNN) merupakan kelompok algoritma *supervised learning*. Algoritma KNN digunakan untuk mencari kelompok K objek dalam data latih yang paling dekat (mirip) dengan objek pada data baru atau pengujian [12]. Pencarian kelompok pada KNN umumnya menggunakan fungsi jarak euclidian yang digunakan untuk menghitung jarak ketetanggaan, fungsi jarak Euclidian dapat dilihat pada Persamaan 1.

$$euc = \sqrt{(\sum_{i=1}^n (p_i - q_i)^2)} \quad (1)$$

dimana :

- pi = sampel data / data latih
- qi = data uji
- i = variabel data
- n = dimensi data

Metode ini melakukan klasterisasi dengan menggunakan jarak terpendek antara data uji dan data latih. Sebagian besar hasil klasterisasi didapatkan dari jumlah tetangga terbanyak yang diambil sebanyak jumlah K yang ditentukan, jarak euclidian umumnya digunakan untuk menentukan seberapa dekat atau jauh tetangga. Proses penghitungan metode KNN antara lain sebagai berikut:

1. Menentukan jumlah K (tetangga).
2. Menghitung jarak antara data latih dan data uji dengan menggunakan fungsi jarak euclidian.
3. Mengurutkan nilai jarak yang terbentuk dari yang terkecil hingga nilai jarak terbesar.
4. Menentukan kelas pada data uji sejumlah nilai K berdasarkan jumlah kelas dari data latih yang sering muncul.

## 3. METODE PENELITIAN

### 3.1. Pengumpulan Data

Dalam penelitian ini, data yang dibutuhkan terdiri dari dua jenis, yaitu data uji dan data latih. Data uji diperoleh dari hasil pembacaan sensor suhu dan kelembapan yang akan diuji menggunakan metode KNN. Mengingat KNN merupakan metode *supervised learning*, adanya data latih sangat penting untuk meningkatkan akurasi hasil pengujian.

Data latih diperoleh melalui dua pendekatan. Pertama, dengan melakukan studi literatur mengenai standar suhu dan kelembapan normal pada ruang server, yang menunjukkan suhu dalam rentang 21°C-23°C dan kelembapan antara 45%-60% [1]. Data dengan kondisi normal ini diberi klasifikasi label dengan nilai 1. Kedua, dilakukan pembuatan *dummy data* yang mencerminkan suhu dan kelembapan dalam kondisi tidak normal, yang akan diberi klasifikasi label dengan nilai 0. Data latih dengan label 0 yang menunjukkan kondisi tidak normal dapat dilihat pada Tabel 1. Dengan pendekatan ini, diharapkan model KNN dapat berfungsi secara optimal dalam menganalisis data uji yang akan diperoleh.

Tabel 1 : Data latih dengan label 0

Suhu (°C)	Kelembapan (%)	Label
27	67	0
27	68	0
27	69	0
28	70	0
28	71	0
28	72	0
29	73	0
29	74	0
29	75	0
30	76	0
30	77	0
30	78	0
31	79	0
31	80	0
31	81	0
16	30	0
16	31	0
16	32	0
17	33	0
17	34	0
17	35	0
18	36	0

18	37	0
18	38	0
19	39	0
19	40	0
19	41	0
35	81	0
35	82	0
35	83	0

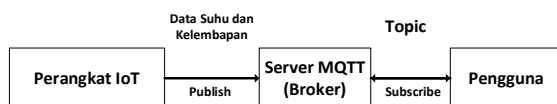
### 3.2. Arsitektur Pengujian

Pada bagian ini dijelaskan tentang arsitektur yang digunakan dalam penelitian ini yaitu perangkat keras dan lunak yang digunakan pada server MQTT, kemudian alur pengiriman dan penerimaan data juga dibahas dalam bagian ini, Tabel 2 merupakan spesifikasi perangkat keras dan lunak yang digunakan dalam penelitian ini.

Tabel 2 : Spesifikasi perangkat keras dan lunak

No	Item	Rincian
1	CPU	4 Core
2	RAM	16 GB
3	SSD	256 GB
4	OS	Ubuntu 20.04 LTS
5	Broker	Mosquitto
6	Dashboard	Node-Red

Penelitian ini menggunakan Ubuntu 20.24 LTS dengan CPU 4 Core, RAM 16Gb, dan SSD 256 GB. Dengan spesifikasi dan sistem operasi yang digunakan sudah sangat mendukung untuk membangun server MQTT, Mosquitto digunakan dalam penelitian ini sebagai broker MQTT. Dalam merancang dashboard yang nantinya dapat diakses oleh pengguna, Node-Red digunakan dalam pengembangan dashboard IoT dalam penelitian ini. Adapun proses pengiriman dan penerimaan data hasil pembacaan sensor suhu dan kelembapan pada penelitian ini digambarkan dengan blok diagram yang dapat dilihat pada Gambar 3.



Gambar 3 : Blok Diagram

Pada Gambar 3 terdapat tiga entitas yang ada pada sistem yaitu Perangkat IoT, Server yang didalamnya terdapat broker, dan Pengguna. Perangkat IoT mengirimkan (*publish*) data suhu

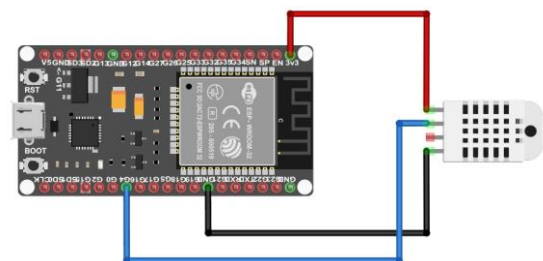
dan kelembapan ke server MQTT. Penggunaan broker berperan dalam mengirimkan kedua data tersebut pada pengguna yang mengakses dashboard secara *real-time* dengan syarat harus berlangganan (*subscribe*) topic yang sama agar data suhu dan kelembapan dapat diterima di dashboard. Variabel topic yang digunakan pada setiap sensor pada protokol MQTT dalam penelitian ini dapat dilihat pada Tabel 3.

Tabel 3 : Parameter Variabel Topic

No	Parameter	Topic
1	Suhu	server/suhu
2	Kelembapan	server/kelembapan
3	Klasifikasi	server/klasifikasi

### 3.3. Perancangan Sistem

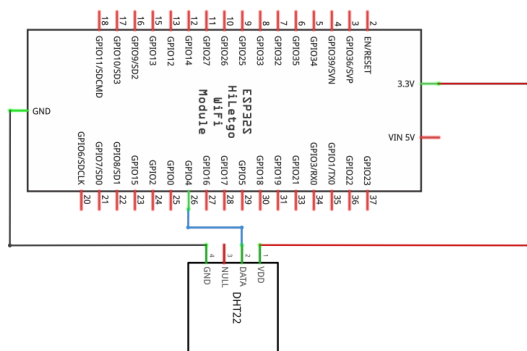
Perancangan perangkat IoT pada penelitian ini menggunakan mikrokontroler ESP32, dan sensor DHT22. ESP32 merupakan mikrokontroler yang sudah terintegrasi dengan modul Wi-Fi sehingga dalam penelitian ini tidak perlu ditambahkan modul Wi-Fi seperti ESP8266. Sensor DHT22 dapat bekerja pada tegangan 3.3V dan dapat memiliki keakuratan pembacaan data yang lebih baik dibandingkan dengan DHT11 [13]. Perancangan perangkat IoT pada penelitian ini dapat dilihat pada Gambar 4.



Gambar 4 : Perancangan Perangkat IoT

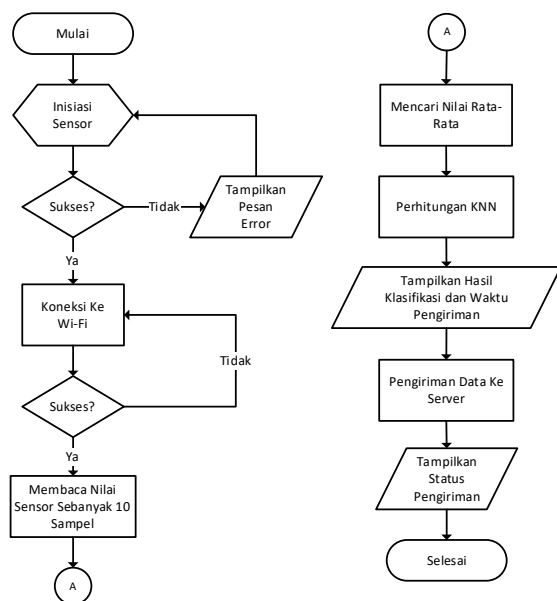
Pada pin VCC/VDD pada DHT22 dihubungkan ke 3.3V pada ESP32 yang digunakan sebagai daya sensor, kemudian pin data dihubungkan pada pin GPIO4 pada ESP32 untuk menerima hasil pembacaan sensor DHT22. Kemudian pada pin GND sensor DHT22 dihubungkan pada pin GND pada ESP32 sebagai ground. *Wiring diagram* pada perancangan perangkat IoT pada penelitian ini dapat dilihat pada Gambar 5.





Gambar 5 : Wiring Diagram

Berdasarkan perancangan perangkat IoT yang telah dilakukan, seperti yang ditunjukkan pada Gambar 5, langkah selanjutnya adalah merancang sistem yang dapat mendukung operasional perangkat IoT serta memberikan antarmuka yang intuitif bagi pengguna melalui dashboard. Gambar 6 menunjukkan flowchart perancangan sistem yang akan dibangun, yang menggambarkan alur proses dan interaksi antara berbagai komponen sistem.



Gambar 6 : Flowchart Sistem

Pada Gambar 6 menggambarkan bagaimana sistem pemantau suhu dan kelembapan pada ruang server bekerja. Proses diawali dengan kondisi “Mulai”, ketika sistem sudah mulai berjalan maka akan dilakukan inisiasi sensor DHT22, jika gagal dilakukan inisiasi maka akan dilakukan inisiasi ulang hingga sensor berhasil diinisiasi. Kemudian jika berhasil, ESP32 akan

melakukan koneksi internet menggunakan Wi-Fi yang sudah dikonfigurasi. Jika gagal dilakukan koneksi maka akan melakukan koneksi ulang sampai ESP32 benar-benar dapat melakukan koneksi internet. Kemudian sensor DHT22 akan melakukan pembacaan sebanyak 10 sampel. Lalu sampel yang sudah dikumpulkan kemudian dilakukan pencarian nilai rata-rata yang mana nantinya nilai rata-rata tersebut yang akan menjadi data uji yang akan dilakukan klasifikasi dengan algoritma KNN berdasarkan data latih yang ada. Setelah itu sistem akan menampilkan hasil klasifikasi suhu dan kelembapan ruang server dan melakukan pengiriman ke server MQTT sehingga nantinya apabila sudah selesai, maka akan ditampilkan status pengiriman.

### 3.4. Pengujian Klasifikasi KNN

Tujuan dari pengujian klasifikasi KNN dalam penelitian ini adalah untuk menganalisis data hasil pembacaan sensor suhu dan kelembapan secara akurat. Sebelum proses klasifikasi dilakukan, pembacaan data suhu dan kelembapan akan dilakukan sebanyak 10 kali untuk memastikan keandalan dan konsistensi hasil. Hasil dari setiap pembacaan ini kemudian dirata-rata untuk mendapatkan nilai yang representatif, proses rata-rata ditunjukkan dalam Persamaan 2.

$$\text{Rata - rata} = \frac{\sum xi}{n} \quad (2)$$

dimana :

xi = jumlah data  
n = total data

Setelah nilai rata-rata diperoleh, klasifikasi dilakukan menggunakan algoritma KNN dengan parameter K = 3 berdasarkan data latih yang telah disiapkan sebelumnya. Dengan pendekatan ini, diharapkan model KNN dapat memberikan klasifikasi yang tepat dan akurat untuk kondisi suhu dan kelembapan yang terukur.

Pengujian klasifikasi KNN pada penelitian ini dilakukan pada dua kondisi berbeda yaitu pada kondisi pendingin ruangan hidup, dan kondisi pendingin ruangan dimatikan. Tujuannya agar menguji keakuratan KNN yang diperoleh dari sensor suhu dan kelembapan,

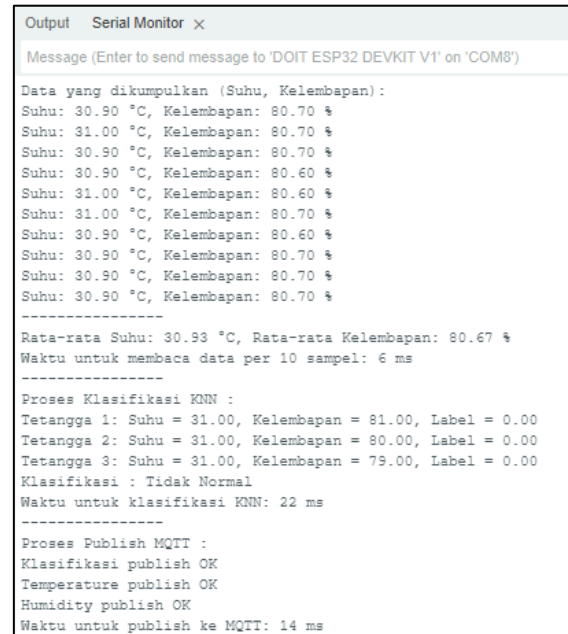
hasil dari pengujian ini disajikan dalam bentuk tabel.

### 3.5. Pengujian Waktu

Pengujian ini bertujuan untuk menghitung waktu yang dibutuhkan untuk transfer data hasil deteksi suhu dan kelembapan dengan menggunakan protokol MQTT. Dalam pengujian ini, data akan dikirim dari sensor ke server lokal MQTT, dan waktu transfer akan diukur dalam berbagai kondisi jaringan. Pengujian kecepatan transfer juga dilakukan dengan menggunakan *access point*, dan hotspot Wi-Fi yang diaktifkan pada ponsel pintar dengan frekuensi 2.4GHz. Selain melakukan pengujian waktu transfer data, penelitian ini juga melakukan pengujian waktu eksekusi algoritma KNN yang dijalankan pada mikrokontroler ESP32.

Dengan demikian, kedua pengujian menjadi langkah penting dalam meningkatkan keandalan dan responsivitas pada sistem pemantau, hasil pengujian ini disajikan dalam bentuk tabel.

serial monitor dengan *baud rate* sebesar 115200 pada aplikasi Arduino IDE ketika program dijalankan.



```

Output Serial Monitor x
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM8')

Data yang dikumpulkan (Suhu, Kelembapan):
Suhu: 30.90 °C, Kelembapan: 80.70 %
Suhu: 31.00 °C, Kelembapan: 80.70 %
Suhu: 30.90 °C, Kelembapan: 80.70 %
Suhu: 30.90 °C, Kelembapan: 80.60 %
Suhu: 31.00 °C, Kelembapan: 80.60 %
Suhu: 31.00 °C, Kelembapan: 80.70 %
Suhu: 30.90 °C, Kelembapan: 80.60 %
Suhu: 30.90 °C, Kelembapan: 80.70 %
Suhu: 30.90 °C, Kelembapan: 80.70 %
Suhu: 30.90 °C, Kelembapan: 80.70 %
-----
Rata-rata Suhu: 30.93 °C, Rata-rata Kelembapan: 80.67 %
Waktu untuk membaca data per 10 sampel: 6 ms
-----
Proses Klasifikasi KNN :
Tetangga 1: Suhu = 31.00, Kelembapan = 81.00, Label = 0.00
Tetangga 2: Suhu = 31.00, Kelembapan = 80.00, Label = 0.00
Tetangga 3: Suhu = 31.00, Kelembapan = 79.00, Label = 0.00
Klasifikasi : Tidak Normal
Waktu untuk klasifikasi KNN: 22 ms
-----
Proses Publish MQTT :
Klasifikasi publish OK
Temperature publish OK
Humidity publish OK
Waktu untuk publish ke MQTT: 14 ms
  
```

Gambar 8 : Serial Monitor

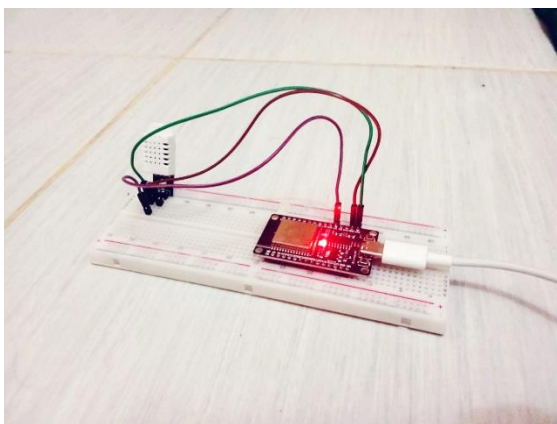
Gambar 8 merupakan tampilan serial monitor pada *baud rate* 115200, yang berarti data akan ditransfer antara perangkat ESP32 dan komputer dengan kecepatan 115200 bit per detik. Kecepatan ini penting untuk memastikan bahwa data dapat dikirim dan diterima dengan cepat, terutama saat mencetak informasi ke konsol serial. Output data yang ditampilkan berupa deteksi suhu dan kelembapan, rata-rata nilai, waktu eksekusi data per 10 sampel rata-rata, proses klasifikasi KNN, waktu eksekusi KNN, proses publish data ke MQTT broker, dan waktu untuk publish data ke MQTT.

Data hasil pembacaan suhu dan kelembapan pada sensor DHT22 dikirimkan ke server MQTT dengan menggunakan broker Mosquitto, broker ini bertujuan sebagai penyedia topic yang akan diakses oleh pengguna nantinya. Proses mengakses topic pada pengguna agar dapat melihat data hasil pembacaan suhu, kelembapan, dan hasil klasifikasi dibuat dengan menggunakan Node-Red. Gambar 9 merupakan tampilan konfigurasi pada Node-Red untuk menampilkan data data hasil pembacaan suhu, kelembapan, dan hasil klasifikasi secara *real-time*.

## 4. HASIL DAN PEMBAHASAN

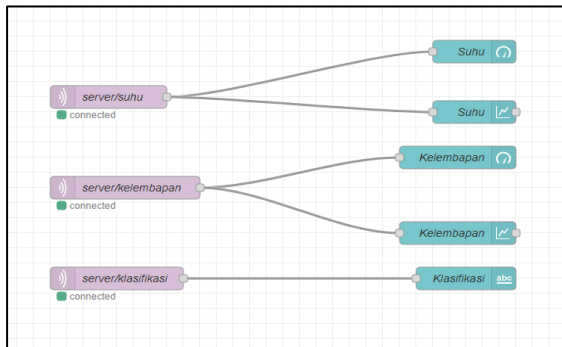
### 4.1. Perangkat IoT dan Sistem

Hasil implementasi perangkat yang telah dilakukan perancangan dapat dilihat pada Gambar 7.



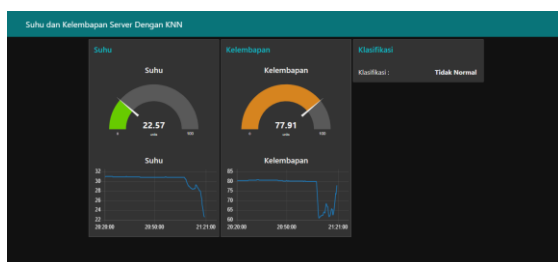
Gambar 7 : Perangkat IoT

Gambar 7 merupakan perangkat IoT yang sudah dilakukan perancangan, perangkat IoT yang telah dirancang bangun dilakukan instalasi pada ruang *server* tepatnya berdekatan dengan mesin *server*. Gambar 8 menunjukkan tampilan



Gambar 9 : Konfigurasi Node-Red

Pada Gambar 9, terdapat tiga topic yang ada pada sistem sebagaimana telah dijelaskan pada Tabel 3. Ketiga topic ini mewakili tiga parameter, topic server/suhu digunakan untuk menampilkan data suhu, topic server/kelembapan digunakan untuk menampilkan data kelembapan, kemudian topic server/klasifikasi digunakan untuk menampilkan data hasil klasifikasi dengan algoritma KNN. Suhu dan kelembapan ditampilkan dengan menggunakan grafik garis dan gauge, sedangkan hasil klasifikasi ditampilkan dalam bentuk teks. Hasil perancangan dashboard pengguna dengan menggunakan Node-Red ditampilkan pada Gambar 10.



Gambar 10 : Dashboard Node-Red

Hasil dari pengujian sensor DHT22 untuk membaca suhu dan kelembapan pada ruang server dilakukan sebanyak 10 kali pengambilan sampel data dan kemudian dilakukan rata-rata. Data hasil pengujian sensor DHT22 ditampilkan pada Tabel 4.

Tabel 4 : Nilai Rata-Rata Pengujian

No	Suhu (°C)	Kelembapan (%)
1	22.83	60.58
2	23.12	60.34
3	22.89	60.34

4	22.80	60.34
5	22.73	60.34
6	22.73	60.34
7	22.72	60.33
8	22.72	60.33
9	22.72	60.33
10	22.71	60.33
<b>Rata-rata</b>	<b>22.79</b>	<b>60.36</b>

Hasil yang ditampilkan pada Tabel 4 merupakan hasil perhitungan rata-rata yang diambil dari 10 sampel ketika pendingin ruangan server dalam keadaan hidup. Tabel 5 menunjukkan perhitungan dari nilai rata-rata suhu dan kelembapan ruang server yang dilakukan pengujian ketika dalam keadaan pendingin ruangan setelah dipadamkan selama 1 jam.

Tabel 5 : Nilai Rata-Rata Pengujian Ketika Pendingin Ruangan Padam

No	Suhu (°C)	Kelembapan (%)
1	29.80	81.90
2	29.80	81.90
3	29.90	81.70
4	29.90	81.70
5	29.90	81.60
6	29.90	81.50
7	30.00	81.50
8	30.00	81.40
9	30.00	81.30
10	30.00	81.30
<b>Rata-rata</b>	<b>29.92</b>	<b>81.58</b>

Hasil yang ditampilkan pada tabel 5 terlihat perbedaan yang signifikan ketika dilakukan pengujian dengan kondisi pendingin ruangan dipadamkan selama 1 jam. Terdapat selisih sebesar 7.13 °C pada suhu dan 21.22 % pada kelembapan. Jika dibandingkan dengan standar suhu dan kelembapan ruang server di Indonesia, maka ketika pendingin dilakukan pemadaman selama 1 jam nilai yang dihasilkan diluar standar suhu dan kelembapan.

#### 4.2. Klasifikasi KNN

Hasil dari pengujian klasifikasi menggunakan algoritma KNN dengan jumlah tetangga  $K = 3$  dilakukan pada kondisi ketika pendingin ruangan dihidupkan, dan dipadamkan. Nilai suhu dan kelembapan yang dilakukan klasifikasi yaitu berupa nilai rata-rata



hasil pengambilan 10 sampel data. Tabel 6 merupakan hasil dari klasifikasi KNN 10 data dengan kondisi pendingin ruangan dihidupkan.

Tabel 6 : Pengujian Klasifikasi KNN Kondisi Pendingin Ruangan Hidup

Suhu (°C)	Kelembapan (%)	Klasifikasi
22.83	60.58	Normal
23.12	60.34	Normal
22.89	60.34	Normal
22.80	60.34	Normal
22.73	60.34	Normal
22.73	60.34	Normal
22.72	60.33	Normal
22.72	60.33	Normal
22.72	60.33	Normal
22.71	60.33	Normal

Hasil yang diperoleh dari Tabel 6 pada klasifikasi KNN ketika kondisi pendingin ruangan hidup menunjukkan bahwa data yang dihasilkan 100% menunjukan kondisi Normal. Tabel 7 merupakan hasil klasifikasi KNN yang diuji 1 jam setelah kondisi pendingin ruangan dilakukan pemadaman. Hal ini menunjukkan perubahan yang signifikan dalam klasifikasi, dimana kondisi suhu dan kelembapan mengalami peningkatan signifikan dibandingkan ketika kondisi pendingin ruangan hidup.

Tabel 7 : Pengujian Klasifikasi KNN Kondisi Pendingin Ruangan Padam

Suhu (°C)	Kelembapan (%)	Klasifikasi
27.83	92.30	Tidak Normal
28.55	87.37	Tidak Normal
29.07	85.13	Tidak Normal
29.44	83.50	Tidak Normal
29.73	82.39	Tidak Normal
29.92	81.58	Tidak Normal
30.07	81.02	Tidak Normal
30.20	80.53	Tidak Normal
30.28	80.25	Tidak Normal
30.32	80.05	Tidak Normal

Berdasarkan data yang disajikan, 100% hasil klasifikasi suhu dan kelembapan menunjukkan kondisi Tidak Normal. Rentang suhu antara 27.83°C hingga 30.32°C dengan kelembapan yang bervariasi antara 80.05% hingga 92.30% menunjukkan bahwa kondisi lingkungan berada

di luar parameter normal. Hal ini jika dibiarkan akan mengakibatkan potensi masalah dalam *server* yang berujung dapat mengalami kerugian sistem.

#### 4.3. Waktu Eksekusi

Hasil pengujian waktu eksekusi dilakukan dengan menggunakan koneksi Wi-Fi pada *access point*, dan hotspot pada posel pintar (*smartphone*). Tabel 8 merupakan hasil 10 data waktu eksekusi yang diperoleh dengan menggunakan *access point*.

Tabel 8 : Waktu Eksekusi Dengan Access Point

Suhu (°C)/ Kelembapan (%)	Waktu Pembacaan Sensor	Waktu Eksekusi KNN	Waktu Publish MQTT
22.83 / 60.58	6 ms	22 ms	14 ms
23.12 / 60.34	6 ms	22 ms	14 ms
22.89 / 60.34	6 ms	22 ms	14 ms
22.80 / 60.34	6 ms	22 ms	14 ms
22.73 / 60.34	6 ms	22 ms	14 ms
22.73 / 60.34	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.71 / 60.33	6 ms	22 ms	14 ms

Hasil pengujian yang dihasilkan dengan menggunakan *access point* terdapat waktu rata-rata pembacaan sensor selama 6 ms, waktu eksekusi KNN selama 22 ms, dan waktu publish MQTT selama 14 ms. Dari ketiga hasil yang diperoleh, semua eksekusi dilakukan kurang dari 1 detik, dalam hal ini ESP32 dengan jaringan *access point* sangat cepat dalam melakukan eksekusi baik program maupun pengiriman data. Tabel 9 merupakan hasil pengujian waktu eksekusi dengan menggunakan hotspot Wi-Fi pada *smartphone*

Tabel 9 : Waktu Eksekusi Dengan Hotspot WiFi Smartphone

Suhu (°C)/ Kelembapan (%)	Waktu Pembacaan Sensor	Waktu Eksekusi KNN	Waktu Publish MQTT
22.71 / 60.58	6 ms	22 ms	14 ms
22.71 / 60.34	6 ms	22 ms	14 ms
22.71 / 60.34	6 ms	22 ms	14 ms
22.71 / 60.34	6 ms	22 ms	14 ms

22.71 / 60.34	6 ms	22 ms	14 ms
22.72 / 60.34	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.72 / 60.33	6 ms	22 ms	14 ms
22.71 / 60.33	6 ms	22 ms	14 ms

Hasil yang diperoleh pada waktu eksekusi yang dilakukan dengan menggunakan hotspot Wi-Fi pada *smartphone* menunjukkan tidak ada perubahan yang terjadi. Hal ini kemungkinan disebabkan karena server yang digunakan masih berupa server lokal, sehingga pengujian dapat dikembangkan berupa *server public* agar dapat ditemukan perbedaannya.

## 5. KESIMPULAN

- Sistem pemantau suhu dan kelembapan ruang *server* bekerja sesuai seperti yang diharapkan. Terdapat selisih sebesar 7.13 °C pada suhu dan 21.22% pada kelembapan. Ketika dilakukan pengujian dengan pendingin ruangan yang telah dipadamkan selama 1 jam.
- Klasifikasi dengan menggunakan algoritma KNN pada penelitian ini yang dilakukan pada 10 data, terdapat 100% klasifikasi Normal pada keadaan pendingin ruangan yang hidup. Namun ketika dilakukan pengujian setelah pendingin ruangan dipadamkan, terdapat 100% hasil klasifikasi menunjukkan Tidak Normal.
- Pengujian waktu eksekusi dengan menggunakan *access point* dan hotspot Wi-Fi pada *smartphone* tidak ada perubahan dalam eksekusinya. Waktu pembacaan sensor dihasilkan selama 4ms, waktu eksekusi KNN selama 22ms, dan proses pengiriman ke server lokal MQTT selama 14ms yang mana semua hasil diperoleh kurang dari 1 detik.
- Agar mendapatkan hasil yang lebih variatif, disarankan agar melakukan pengujian dengan server publik atau dapat diakses secara *online*.

- Penelitian ini dapat dilakukan pengembangan dengan menganalisa kinerja dari 2 jenis mikrokontroler yang berbeda dengan menggunakan algoritma KNN, misalkan ESP32 dengan Arduino Uno agar mendapatkan hasil yang lebih representatif.
- Dari segi algoritma, penelitian ini juga dapat dikembangkan dengan menganalisa kinerja dari dua algoritma klasifikasi dengan menggunakan mikrokontroler ESP32.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada bapak Dr. Nurdin, S.Kom., M.Kom. selaku ketua prodi Magister Teknologi Informasi Universitas Malikussaleh yang telah memberikan motivasi serta dukungan sehingga artikel penelitian ini terbit.

## DAFTAR PUSTAKA

- E. B. Raharjo dkk., "Rancangan Sistem Monitoring Suhu Dan Kelembapan Ruang Server Berbasis Internet of Things," *TEKNIKA*, vol. 6, no. 2, hlm. 61–68, 2019.
- Y. Nur, I. Fathulrohman, A. Saepuloh, dan M. Kom, "Alat Monitoring Suhu Dan Kelembapan Menggunakan Arduino Uno" *JUMANTAKA*, vol. 02, no. 01, hlm. 161–170, 2018.
- M. Bender, E. Kirdan, M. O. Pahl, dan G. Carle, "Open-source MQTT evaluation," dalam *2021 IEEE 18th Annual Consumer Communications and Networking Conference, CCNC 2021*, Institute of Electrical and Electronics Engineers Inc., Jan 2021. doi: 10.1109/CCNC49032.2021.9369499.
- I. R. Nugraha, W. H. N. Putra, dan E. Setiawan, "A Comparative Study of HTTP and MQTT for IoT Applications in Hydroponics," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 1, hlm. 119–126, Feb 2024, doi: 10.29207/resti.v8i1.5561.
- F. A. Kusuma, "Pemodelan Klasifikasi Anemia Aplastik Menggunakan Teknik Oversampling dan K-Nearest Neighbors," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Agu 2024, doi: 10.23960/jitet.v12i3.4326.

- [6] Z. Zulkifli, M. Muhallim, dan H. Hasnahwati, "Pengembangan Sistem Alarm dan Pemadam Kebakaran Otomatis Menggunakan Internet of Things," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Agu 2024, doi: 10.23960/jitet.v12i3.4774.
- [7] R. Firdaus dan E. Mulyana, "Smart Building Lighting System," dalam *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jul 2019, hlm. 1–9. doi: 10.1088/1757-899X/384/1/012071.
- [8] B. Mishra dan A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," *IEEE Access*, vol. 8, hlm. 201071–201086, 2020, doi: 10.1109/ACCESS.2020.3035849.
- [9] A. Kurnianto, J. Dedy Irawan, F. X. Ariwibisono, dan A. Wardhana, "Penerapan IoT (Internet Of Things) Untuk Controlling Lampu Menggunakan Protokol MQTT Berbasis Web," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, hlm. 1153, 2022, [Daring]. Tersedia pada: <https://www.embedded.com/>
- [10] H. Dhien Chandra dan Amrizal, "Sistem Informasi Absensi RFID Berbasis Web Menggunakan ESP32 di PT Dharma Sentosa Marindo," *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, vol. 6, no. 1, hlm. 2621–4962, 2023.
- [11] A. Rusadi dan Z. Ardian, "Implementasi Radio Frekuensi Untuk Memonitoring Suhu Tumbuhan Hidroponik Berbasis IoT," *Journal of Informatics and Computer Science*, vol. 10, no. 1, 2024.
- [12] D. Cahyanti, A. Rahmayani, dan S. Ainy Husniar, "Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," *Indonesian Journal of Data and Science*, vol. 1, no. 2, hlm. 39–43, 2020.
- [13] B. Adi dan B. R. Saputra, "System Control Dan Monitoring Suhu Pada Sub Distribution Panel (SDP) Berbasis IoT MENGGUNAKAN SENSOR DHT22 Untuk Mengurangi Terjadinya Overheat Pada PT.XYZ," *Jurnal Teknik Elektro*, vol. 6, no. 2, 2022.