

# Otomatisasi Prediksi Kebutuhan Pemeliharaan AC Berbasis IoT Menggunakan Algoritma Random Forest

Angga Sebastian<sup>1\*</sup>, Moh. Muhtarom<sup>2</sup>, Hanifah Permatasari<sup>3</sup>

<sup>1</sup>Teknik Informatika/Fakultas Ilmu  
Komputer

Universitas Duta Bangsa Surakarta

<sup>1</sup>\*230103217@mhs.udb.ac.id

<sup>2</sup>Teknik Informatika/Fakultas Ilmu  
Komputer

Universitas Duta Bangsa Surakarta

<sup>2</sup> muhtarom@udb.ac.id

<sup>3</sup>Teknik Informatika/Fakultas Ilmu  
Komputer

Universitas Duta Bangsa Surakarta

<sup>3</sup> hanifah\_permatasari@udb.ac.id

**Abstrak**— Pengelolaan ruang server di Disdukcapil Kota Surakarta memerlukan kestabilan suhu dan kelembapan guna menjaga keandalan sistem informasi. Saat ini, pemeliharaan *air conditioner* (AC) sebagai pendingin utama masih dilakukan berdasarkan perkiraan manual atau interval waktu tetap yang belum mencerminkan kebutuhan aktual. Hal ini dapat menyebabkan pemborosan biaya dan penurunan umur perangkat. Meski sensor suhu dan kelembapan telah tersedia, pemanfaatannya masih terbatas pada pemantauan tanpa analisis lanjutan. Untuk mengatasi hal tersebut, dikembangkan sistem prediksi kebutuhan servis AC berbasis algoritma *Random Forest*. Sistem ini menganalisis data suhu, kelembapan, dan histori servis AC selama satu tahun untuk menentukan kebutuhan servis secara akurat. Jika kondisi dinilai memerlukan perawatan, sistem secara otomatis mengirimkan notifikasi pemesanan layanan melalui *Telegram* kepada teknisi atau pengelola barang. Solusi ini diharapkan mampu meningkatkan efisiensi perawatan AC serta menjaga performa ruang server secara berkelanjutan.

**Kata kunci**— Ruang server, Servis AC, *Random Forest*, Pemantauan suhu dan kelembapan, Otomatisasi notifikasi.

**Abstract**— Server room management at the Disdukcapil Kota Surakarta requires stable temperature and humidity to ensure the reliability of its information systems. Currently, air conditioner (AC) maintenance is conducted based on manual estimation or fixed intervals that do not necessarily reflect actual needs. This may result in cost inefficiencies and reduced equipment lifespan. Although temperature and humidity sensors are already in use, their function is limited to monitoring without further analysis. To address this limitation, a predictive system using the Random Forest algorithm is developed to accurately forecast AC service needs. The system analyzes temperature, humidity, and AC maintenance history over the past year to identify service requirements. When maintenance is deemed necessary, the system automatically sends a service request notification via Telegram to technicians or asset managers. This solution is expected to enhance the efficiency of AC maintenance and ensure sustained server room performance.

**Keywords**— Server room, AC maintenance, Random Forest, Temperature and humidity monitoring, Notification automation.

## I. PENDAHULUAN

Pengelolaan ruang server menjadi bagian krusial dalam menjaga keandalan sistem informasi, terutama di lingkungan Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil) Kota Surakarta. Salah satu faktor utama yang harus diperhatikan dalam pengelolaan ruang server adalah kestabilan suhu dan kelembapan ruangan [1]. Untuk menjaga kondisi tersebut, Disdukcapil Kota Surakarta mengandalkan perangkat *air conditioner (AC)* sebagai sistem pendingin utama. *AC* bekerja dengan cara memindahkan panas dari dalam ruangan ke luar, sehingga suhu dan kelembapan tetap berada pada level ideal guna melindungi perangkat server dari risiko kerusakan [2].

Agar *AC* dapat terus bekerja secara optimal dan efisien, dibutuhkan perawatan rutin, terutama pada kebersihan komponen-komponennya [3]. Komponen yang kotor dapat menghambat proses pelepasan panas, memaksa *AC* bekerja lebih keras, yang berujung pada peningkatan konsumsi energi dan risiko kerusakan [4]. Saat ini, penjadwalan servis *AC* masih mengandalkan perkiraan manual atau interval waktu tertentu yang belum tentu mencerminkan kebutuhan aktual. Akibatnya, servis bisa terlambat dilakukan atau justru dilakukan terlalu dini, keduanya berpotensi menyebabkan pemborosan biaya dan penurunan umur perangkat. Meskipun sudah tersedia sensor suhu dan kelembapan yang merekam data secara *realtime*, pemanfaatannya masih terbatas pada fungsi pemantauan saja, tanpa ada proses analisis atau tindakan lanjutan.

Melihat keterbatasan sistem yang ada, diperlukan solusi yang mampu memprediksi kebutuhan servis *AC* secara tepat waktu dan otomatis melakukan pemesanan layanan ke teknisi atau pihak pengelola barang. Solusi ini dirancang dengan memanfaatkan algoritma *Random Forest*, yang akan menganalisis data suhu, kelembapan, dan riwayat servis *AC* selama satu tahun terakhir. Algoritma ini membentuk sejumlah pohon keputusan dari data acak, lalu menentukan hasil melalui proses voting [5]. Dari hasil prediksi tersebut, sistem dapat mengenali kondisi yang membutuhkan servis, dan secara otomatis mengirimkan notifikasi pemesanan

melalui pesan Telegram kepada teknisi atau pihak pengelola barang, sehingga perawatan *AC* menjadi lebih tepat sasaran dan efisien.

## II. METODOLOGI PENELITIAN

### A. Data Penelitian

Pada penelitian ini, data yang digunakan adalah data suhu dan kelembapan yang bersumber dari sensor *DHT22* di ruang server Dinas Kependudukan dan Pencatatan Sipil Kota Surakarta dan dikombinasikan data selisih waktu servis terakhir yang bersumber dari riwayat servis *AC* selama satu tahun terakhir. Algoritma machine learning yang digunakan dalam prediksi pada penelitian ini adalah algoritma *Random Forest*. Pemilihan algoritma *Random Forest* dalam sistem prediksi servis *AC* didasarkan pada keunggulannya yang terbukti mampu menghasilkan model klasifikasi yang akurat dan stabil tanpa memerlukan penyesuaian parameter kompleks, sebagaimana juga ditunjukkan dalam penelitian sejenis yang menggunakan data iklim seperti suhu dan kelembapan sebagai fitur utama [6].

### B. Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan yaitu metode *Prototyping*, yang memiliki lima tahapan sebagai berikut [7]:

- 1) *Communication* (Komunikasi)  
Langkah pertama dalam metode ini adalah menentukan kebutuhan apa saja yang digunakan untuk membuat sistem ini mulai dari *hardware* dan *software*.
- 2) *Quick Plan* (Perencanaan Cepat)  
Pada tahap ini, membangun prototipe dengan membuat perencanaan perancangan dari sistem yang akan dibangun.
- 3) *Modeling Quick Design* (Melakukan Pemodelan)  
Pada tahap ini, berfokus pada aspek *software* dan *hardware*, dilakukan dengan membuat sebuah skema sistem.
- 4) *Construction of Prototype* (Membangun Prototipe)

Pada tahap ini, membangun rancangan prototipe dengan melakukan perakitan komponen-komponen *hardware*, kemudian komponen *hardware* tersebut diintegrasikan dengan *software* yang telah dibuat.

#### 5) *Deployment, Delivery & Feedback* (Implementasi dan Evaluasi)

Dalam tahapan ini dilakukan pengujian terhadap prototipe sistem, apakah sistem yang dirancang sudah sesuai dengan kebutuhan atau belum.

### C. *Internet of Things (IoT)*

*Internet of Things (IoT)* adalah suatu konsep yang memungkinkan perangkat fisik seperti sensor suhu dan kelembapan terhubung ke jaringan internet untuk mengirim dan menerima data secara otomatis [8]. Dalam konteks pengelolaan ruang server, *IoT* tidak hanya digunakan untuk pemantauan lingkungan secara *real-time*, tetapi juga dapat dikembangkan menjadi sistem cerdas yang mampu melakukan analisis dan pengambilan keputusan secara mandiri, seperti memprediksi kebutuhan servis AC berdasarkan kondisi aktual dan histori pemeliharaan, serta mengirimkan notifikasi otomatis kepada teknisi.

### D. *Single Board Computer (SBC)*

*Single Board Computer (SBC)* adalah komputer mini yang seluruh komponen utamanya—seperti prosesor, memori, dan antarmuka input/output—terintegrasi dalam satu papan sirkuit tunggal [9]. Perangkat ini umum digunakan dalam sistem tertanam karena bentuknya yang ringkas, hemat daya, dan cukup andal untuk menjalankan aplikasi komputasi ringan. Salah satu contoh *SBC* yang relevan untuk implementasi sistem pemantauan suhu dan kelembapan ruang server adalah *Orange Pi Zero*. Perangkat ini menggunakan prosesor *Allwinner H2+ Quad-core Cortex-A7* dengan kecepatan hingga 1.2 GHz dan *RAM* sebesar 512 MB DDR3. *Orange Pi Zero* efisien dari segi konsumsi daya dan mampu menjalankan sistem pemrosesan data citra maupun komunikasi jaringan secara *real-time*. Hal ini menjadikannya sangat cocok untuk

digunakan dalam sistem berbasis *Internet of Things (IoT)* yang menuntut efisiensi energi, termasuk pada aplikasi monitoring lingkungan ruang server untuk menjaga kestabilan operasi perangkat digital.

### E. *Sensor DHT22*

Sensor *DHT22* adalah perangkat digital yang digunakan untuk mengukur suhu dan kelembapan udara secara akurat [10], yang dalam penelitian ini berperan penting sebagai sumber data utama untuk menganalisis kondisi lingkungan ruang server Disdukcapil Kota Surakarta. Sensor ini bekerja menggunakan teknologi kapasitif untuk mendeteksi kelembapan relatif dan termistor presisi untuk mengukur suhu, serta menghasilkan data dalam format digital melalui satu jalur komunikasi (*single-wire interface*) yang kompatibel dengan berbagai mikrokontroler berbasis *IoT*. *DHT22* mampu mengukur suhu dari  $-40^{\circ}\text{C}$  hingga  $+80^{\circ}\text{C}$  dengan akurasi  $\pm 0,5^{\circ}\text{C}$ , dan kelembapan dari 0% hingga 100% RH dengan akurasi sekitar  $\pm 2-5\%$  RH. Keunggulan seperti konsumsi daya rendah, ukuran yang kompak, dan kemampuannya melakukan pemantauan secara *real-time* menjadikan sensor ini sangat sesuai untuk kebutuhan sistem monitoring berbasis prediksi.

### F. *Algoritma Random Forest*

*Random Forest (RF)* pertama kali diperkenalkan oleh Leo Breiman (2001). RF merupakan salah satu metode yang dapat meningkatkan hasil akurasi dalam membangkitkan atribut untuk setiap node yang dilakukan secara acak. RF terdiri dari sekumpulan *decision tree*, dimana kumpulan pohon keputusan ini digunakan untuk mengklasifikasi data ke suatu kelas. Pohon keputusan dibuat dengan menentukan node akar dan berakhir dengan beberapa node daun untuk mendapatkan hasil akhir [11].

Sebagai contoh kita memiliki data berikut dan sekaligus dilakukan pelabelan perlu atau tidaknya servis:

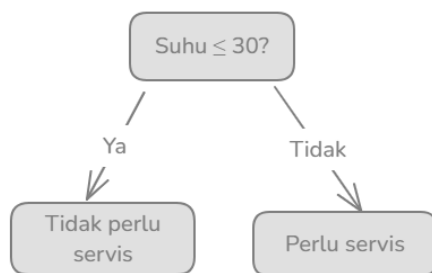
Tabel 1. Sampel Dataset

No	Suhu (°C)	Kelembapan (%)	Hari Sejak Servis	Perlu Servis
1	28	55	3	Tidak
2	31	70	7	Ya
3	29	60	5	Tidak
4	33	75	9	Ya

Dari dataset di atas, kita buat simulasi 1 pohon keputusan, dengan mengambil fitur suhu & dikelompokkan menjadi 2, maka:

- Kelompok A, suhu  $\leq 30 = 1, 3 \rightarrow$  Tidak perlu servis
- Kelompok B, suhu  $> 30 = 2, 4 \rightarrow$  Perlu servis

Terbentuk satu pohon keputusan seperti berikut:



Gambar 1. Ilustrasi 1 buah pohon keputusan dengan fitur suhu.

Setelah itu kita uji dengan data berikut:

- Suhu = 32
- Kelembapan = 65
- Hari Sejak Servis = 6

Karena suhu  $32 > 30$ , maka hasil prediksi adalah Perlu servis. Selain satu contoh tersebut, *Random Forest* membangun banyak pohon lagi, tapi mengacak data & fitur, dan melakukan voting.

Misalkan ada 3 pohon:

- Pohon 1 = Perlu servis
- Pohon 2 = Tidak perlu servis
- Pohon 3 = Perlu servis

Maka setelah diambil suara mayoritas, hasil akhirnya adalah Perlu servis. Begitu dan seterusnya.

### G. Telegram Bot

*Telegram Bot* adalah antarmuka berbasis perintah otomatis yang disediakan oleh platform Telegram, memungkinkan pengembang untuk membangun sistem komunikasi dua arah antara aplikasi dan pengguna melalui *Application*

*Programming Interface (API)*. Bot ini dapat digunakan untuk mengirim maupun menerima pesan secara otomatis dan *real-time*, serta mudah diintegrasikan dengan berbagai sistem karena bersifat *open license* dan multiplatform. Teknologi ini juga mendukung metode pengiriman pesan seperti *webhook* dan *long polling*, menjadikannya fleksibel untuk digunakan pada server lokal maupun *cloud-based* [12].

Dalam penelitian ini, *Telegram Bot* digunakan sebagai media penyampaian informasi hasil analisis prediksi kebutuhan servis AC. Ketika sistem mendeteksi bahwa kondisi suhu dan kelembapan ruang server di Disdukcapil Kota Surakarta memerlukan perawatan, bot akan mengirimkan notifikasi pemesanan layanan secara otomatis kepada teknisi atau pengelola barang.

## III. HASIL DAN PEMBAHASAN

### A. Komunikasi

Pada tahap ini, dilakukan identifikasi kebutuhan yang diperlukan, meliputi:

#### 1) Kebutuhan Fungsional

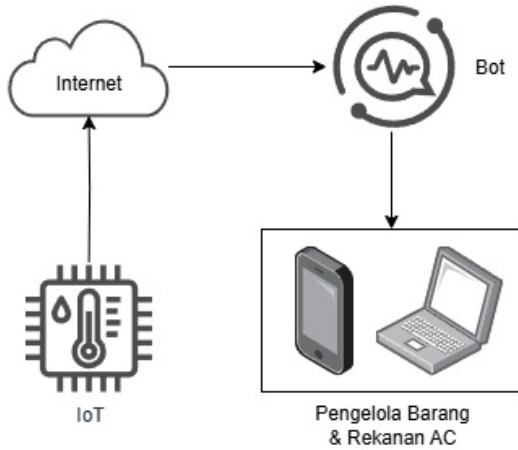
- Sistem yang mampu memprediksi kebutuhan pemeliharaan AC ruang server berdasarkan suhu, kelembapan dan riwayat servis terakhir.
- Pengumpulan data suhu, kelembapan dan riwayat servis terakhir.
- Implementasi algoritma *random forest* untuk memprediksi kebutuhan pemeliharaan AC.
- Permohonan perawatan AC otomatis kepada pegawai pengelola barang dan rekanan pemeliharaan AC.

#### 2) Kebutuhan Non Fungsional

- Mini PC Orange Pi Zero*, dengan spesifikasi:
  - Prosesor *Allwinner H2+*
  - RAM 512MB
  - Penyimpanan 32GB
  - Sistem Operasi *Armbian* berbasis *Debian Bullseye*
- Sensor suhu dan kelembapan *DHT22*
- Jaringan Internet
- Telegram Bot*

**B. Perencanaan Cepat**

Pada tahap ini, dibuat sebuah alur sistem sebagai berikut:



Gambar 2. Alur Sistem Prediksi

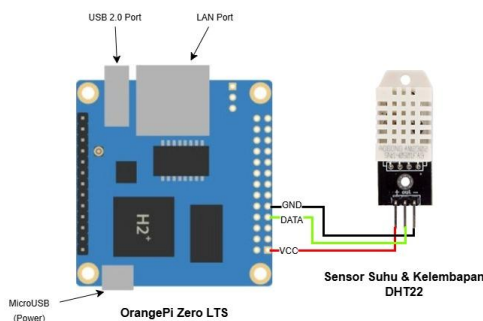
Alur sistem meliputi pembacaan suhu dan kelembapan oleh sensor *IoT*. Data dari sensor dibaca, kemudian diolah dan dilakukan prediksi pemeliharaan *AC*. Jika memenuhi kriteria tertentu maka server secara otomatis akan mengirimkan notifikasi pemesanan pemeliharaan kepada pengelola barang dan rekanan *AC* melalui Telegram *bot*.

**C. Melakukan Pemodelan**

Pemodelan mencakup pembuatan diagram rangkaian dan *flowchart* program.

**1) Diagram Rangkaian *IoT***

Berikut adalah diagram rangkaian *IoT*.



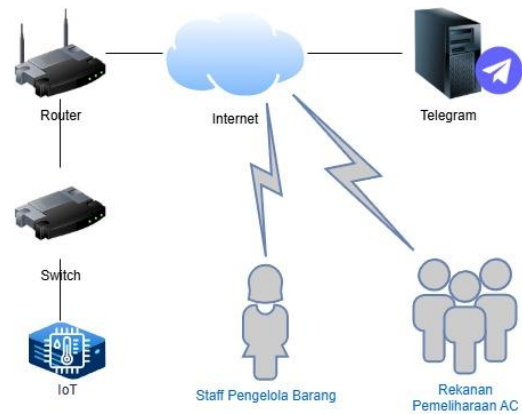
Gambar 3. Diagram Rangkaian *IoT*

Diagram rangkaian di atas menunjukkan koneksi antara *Orange Pi Zero* dan sensor suhu & kelembapan *DHT22*. Sensor *DHT22* memiliki tiga pin utama: VCC

(daya), DATA (sinyal), dan GND (*ground*). Pin VCC dihubungkan ke pin 3.3V pada *header* GPIO *Orange Pi* untuk menyuplai daya ke sensor, DATA dihubungkan ke salah satu pin GPIO digital untuk membaca data suhu dan kelembapan, sedangkan GND disambungkan ke pin *ground* pada *Orange Pi*.

**2) Topologi Jaringan**

Berikut ini adalah topologi jaringan yang digunakan.



Gambar 4. Topologi Jaringan

Sensor suhu dan kelembapan yang terdapat pada perangkat *IoT* (*Orange Pi Zero*) terhubung ke *switch*. *IoT* sekaligus mengolah data dari sensor dan mengirimkan hasil analisis melalui router ke internet. Selanjutnya, sistem dapat mengirim notifikasi ke layanan Telegram, yang akan diteruskan ke staf pengelola barang dan rekanan pemeliharaan *AC*.

**3) Diagram Alir Program**

Berikut ini adalah diagram alir dari program yang diimplementasikan.



Gambar 5. Diagram Alir Program

Proses dimulai dengan pengambilan data dari sensor, dilanjutkan dengan perhitungan rata-rata harian suhu dan kelembapan. Kemudian sistem menghitung jumlah hari sejak terakhir kali dilakukan servis. Berdasarkan data tersebut, sistem akan mengevaluasi apakah kondisi saat ini memenuhi kriteria kebutuhan servis menggunakan model *machine learning Random Forest* yang telah dilatih sebelumnya. Jika hasil evaluasi menunjukkan bahwa AC perlu diservis, maka sistem akan secara otomatis mengirimkan order atau notifikasi ke pihak terkait. Jika tidak, sistem akan kembali ke awal untuk terus memantau kondisi secara berkala.

D. Pembangunan Prototipe

Pada tahap ini, dilakukan pembangunan prototipe sistem. Dimulai dengan merangkai *Orange Pi Zero* dan *DHT22* sesuai dengan

diagram rangkaian pada Gambar 3, kemudian dilakukan instalasi jaringan sesuai topologi Gambar 4. *Library* yang digunakan adalah *DHT* dan *pyA20* yang sudah dimodifikasi. Hasil bacaan suhu dan kelembapan bisa dibaca oleh server atau perangkat lain melalui protokol *SNMP*.

Pada saat yang sama, juga dilakukan *training* model algoritma *random forest* pada dataset dengan fitur suhu, kelembapan, hari sejak servis sebagai input dan sekaligus label perlu tidaknya servis sebagai target. Pelabelan dibutuhkan karena *random forest* termasuk dalam algoritma *supervised learning* yang membutuhkan input fitur dan label target. Dataset diambil dalam rentang satu tahun mulai tanggal 24 April 2024 sampai dengan 24 April 2025. Data lengkap sebagai berikut:

Tabel 2. Dataset Suhu, Kelembapan, Hari Sejak Servis dan Label Perlu Servis

Minggu Ke	Rata-rata Suhu (°C)	Rata-rata Kelembapan (%)	Hari Sejak Servis	Perlu Servis
1	20,75	56,26	17	Tidak
2	21,25	55,10	23	Tidak
3	20,90	55,17	30	Tidak
4	21,21	53,87	37	Tidak
5	21,32	54,36	44	Tidak
6	21,32	53,42	51	Tidak
7	21,35	52,58	58	Tidak
8	20,81	55,70	65	Tidak
9	20,58	54,97	72	Tidak
10	20,17	54,38	79	Tidak
11	20,25	54,76	86	Tidak
12	20,13	55,33	93	Ya
13	21,78	50,44	100	Ya
14	21,93	48,78	107	Ya
15	21,96	48,31	114	Ya
16	22,00	47,81	121	Ya
17	21,97	47,59	128	Ya
18	21,33	49,43	135	Ya
19	21,88	47,44	142	Ya
20	22,05	46,33	149	Ya
21	21,57	54,28	156	Ya
22	20,59	56,32	25	Tidak
23	20,44	57,55	9	Tidak
24	20,46	56,60	16	Tidak
25	20,60	56,91	23	Tidak
26	20,92	56,06	30	Tidak
27	21,03	52,46	37	Tidak
28	20,79	51,45	44	Tidak
29	20,37	52,30	51	Tidak
30	21,31	55,44	58	Tidak
31	21,52	62,76	65	Ya
32	21,82	64,40	72	Ya
33	21,53	63,27	79	Ya
34	21,48	63,73	86	Ya
35	21,62	63,86	93	Ya
36	22,01	64,82	100	Ya
37	21,54	63,63	107	Ya
38	21,14	62,92	114	Ya

Minggu Ke	Rata-rata Suhu (°C)	Rata-rata Kelembapan (%)	Hari Sejak Servis	Perlu Servis
39	21,86	61,43	121	Ya
40	21,65	62,25	128	Ya
41	20,75	63,42	135	Ya
42	20,70	63,56	142	Ya
43	21,49	61,78	149	Ya
44	21,25	62,44	156	Ya
45	21,08	62,59	163	Ya
46	20,82	62,76	170	Ya
47	21,13	62,18	177	Ya
48	21,20	62,26	184	Ya
49	21,21	63,04	191	Ya
50	22,49	54,84	198	Ya
51	22,59	51,91	205	Ya
52	22,28	51,39	3	Ya
53	22,39	50,22	8	Ya

Label perlu servis dikonversi menjadi biner untuk keperluan klasifikasi dan dibagi menjadi data latih dan uji dengan rasio 80:20 (42 data latih dan 11 data uji)

Setelah data siap, dilakukan pelatihan model *Random Forest* menggunakan *script* dengan bahasa pemrograman *Python* dengan memanfaatkan *library Random Forest Classifier*. Hasil pengujian pada data uji menunjukkan akurasi mencapai 90,91% yang artinya dari seluruh data uji (11 data), model berhasil mengklasifikasikan 10 data dengan benar dan 1 data salah.

Dari proses pelatihan tersebut, model diekspor dan ditanam pada perangkat *IoT*. Selanjutnya *IoT* akan memuat model tersebut dan disandingkan dengan data hasil bacaan suhu, kelembapan dan hari sejak servis, dan pada akhirnya menentukan perlu atau tidaknya servis dengan output pesan *Telegram*.

#### E. Implementasi dan Evaluasi

Pada tahap ini, sistem diimplementasikan di lingkungan yang sebenarnya. *IoT* diletakkan di dalam rak server agar bisa mendapatkan hasil bacaan suhu dan kelembapan yang benar-benar sesuai di sekitar perangkat server.



Gambar 6. Perangkat IoT di Rak Server Disdukcapil Kota Surakarta

Dibuat sebuah program *Python* yang dijalankan secara otomatis setiap hari menggunakan *cronjob* untuk membaca suhu dan kelembapan secara *realtime*, menyandingkan dengan model yang sebelumnya sudah ditanamkan dan mengirimkan hasil prediksi melalui *Telegram* kepada user.

Berikut data hasil evaluasi fungsionalitas perangkat IoT.

Tabel 3. Hasil Evaluasi Fungsionalitas *IoT* Sistem Prediksi Servis AC

No	Komponen	Parameter	Hasil Evaluasi	Keterangan
1	Pembacaan Sensor Suhu dan Kelembapan	Interval pembacaan data	5 detik	Sesuai
2	Pembacaan Sensor Suhu dan Kelembapan	Akurasi sensor terhadap alat ukur standar	Suhu $\pm 0.5^{\circ}\text{C}$ , RH $\pm 2\%$	Sesuai
3	Ketepatan Prediksi Servis AC	Sesuai dengan histori servis	90,91% akurat	Sesuai
4	Notifikasi Otomatis via Telegram Bot	Waktu pengiriman dari prediksi ke notifikasi	$\pm 2.5$ detik	Sesuai
5	Keberhasilan Pengiriman Notifikasi	Persentase notifikasi sukses dikirim	100% (10/10)	Sesuai

#### IV. KESIMPULAN

Berdasarkan pembahasan dan hasil yang telah dipaparkan, dapat disimpulkan bahwa sistem

otomasi prediksi kebutuhan pemeliharaan AC berbasis IoT dengan menerapkan algoritma *Random Forest*, yang memanfaatkan data suhu, kelembapan, dan hari sejak servis terakhir untuk menentukan kondisi perawatan secara akurat. Sistem ini menggantikan metode manual berbasis interval tetap dengan pendekatan berbasis data aktual, serta terintegrasi dengan fitur notifikasi otomatis melalui *Telegram* untuk mempercepat respons teknisi. Hasil evaluasi menunjukkan bahwa model memiliki akurasi dan performa klasifikasi yang tinggi, sehingga efektif dalam mendukung perawatan preventif AC secara berkelanjutan dan menjaga kestabilan ruang server di lingkungan Disdukcapil Kota Surakarta.

#### UCAPAN TERIMA KASIH

Ucapan terimakasih diberikan kepada Dinas Kependudukan dan Pencatatan Sipil Kota Surakarta yang telah memberikan izin penelitian dan Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta yang telah memberikan kesempatan penulis mengembangkan pengetahuan.

#### REFERENSI

- [1] D. Debby, Alita, and A. Rahman, "Pendeteksian Sarkasme Pada Proses Analisis Sentimen Menggunakan Random Forest Classifier," *Jurnal Komputasi*, vol. 8, no. 2, pp. 50–58, 2020.
- [2] R. F. Fambudi and S. M. Isa, "Predictive Maintenance Air Conditioner Using Machine Learning," *KESATRIA Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 5, no. 3, pp. 885–894, 2024.
- [3] N. A. Pambudi, A. Sarifudin, R. A. Firdaus, D. K. Ulfa, I. M. Gandidi, and R. Romadhon, "The immersion cooling technology: Current and future development in energy saving," *Alexandria Engineering Journal*, vol. 61, no. 11, pp. 9509–9527, 2022, doi: 10.1016/j.aej.2022.02.059.
- [4] S. M. Siagian, Analisis Kinerja Evaporator pada AC Split Menggunakan R-22 dan R-32, Skripsi, Univ. Medan Area, Medan, 2024.
- [5] I. M. M. Parwita, "Sistem Monitoring Kondisi AC untuk Menentukan Waktu Pemeliharaan Menggunakan Algoritma K-Nearest Neighbor," *Jurnal Sistem dan Informatika (JSI)*, vol. 18, no. 2, pp. 148–158, 2024.
- [6] N. A. P. Indaryono, R. R. Saedudin, dan F. Hamami, "Analisa Perbandingan Algoritma Random Forest dan Naïve Bayes untuk Klasifikasi Curah Hujan Berdasarkan Iklim di Indonesia," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 1, pp. 158–167, Mar. 2024, doi: 10.29100/jupi.v9i1.4421.
- [7] J. M. S. Waworundeng, "Desain Sistem Deteksi Asap dan Api Berbasis Sensor, Mikrokontroler dan IoT," *CogITo Smart Journal*, vol. 6, no. 1, pp. 117–127, 2020, doi: 10.31154/cogito.v6i1.239.117-127.
- [8] H. E. P. M. Jamil and S. Lutfi, "Smart Akuarium Berbasis IOT Menggunakan Raspberry Pi 3," *J. Pendidik. Sains dan Komput.*, vol. 2, no. 02, pp. 333–336, 2022, doi: 10.47709/jpsk.v2i02.1742.
- [9] W. Dewanto, A. Fathurrahman, dan A. Bejo, "Evaluasi Platform Perangkat Keras Sistem Tertanam untuk Unit Kontrol Parkir Otomatis," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 12, no. 4, pp. 287–292, Nov. 2023.
- [10] A. Widodo, A. Sumaedi, dan E. Hendrawati, "Memanfaatkan Teknologi Sensor DHT22 pada Sistem Manajemen Gudang Modern," *TENSILE: Jurnal Pengabdian Kepada Masyarakat*, vol. 2, no. 3, pp. 138–146, Nov. 2024.
- [11] M. A. Rizkiawan, H. Ramza, Nuroji, and A. Sofwan, "Pemantauan Ruang Data Center Berdasarkan Suhu dan Kelembaban dengan Internet of Things," *Jambura Journal of Electrical and Electronics Engineering*, vol. 6, no. 2, Jul. 2024.
- [12] E. Andika, L. Nurlani, S. Rahayu, dan I. Yasirudin, "Pemanfaatan Telegram Bot pada Sistem Informasi Pembayaran SPP," *Jurnal Teknologi Rekayasa (JTERA)*, vol. 7, no. 1, pp. 127–134, Jun. 2022, doi: 10.31544/jtera.v7.i1.2022.127-134.