

## Alat Deteksi Kesehatan AC Berbasis Internet of Things Menggunakan Algoritma K-Nearest Neighbor

Ilham Fadli Surbakti<sup>1</sup>, Aji Gautama Putrada<sup>2</sup>, Catur Wirawan Wijiutomo<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>ilhamsurbakti@students.telkomuniversity.ac.id, <sup>2</sup>ajigps@telkomuniversity.ac.id,

<sup>3</sup>caturwijiutomo@telkomuniversity.ac.id

---

### Abstrak

Air Conditioner (AC) merupakan sebuah alat yang sangat umum digunakan untuk mendinginkan ruangan. Proses pendinginan oleh AC tidak selalu berjalan normal apalagi seiring bertambahnya usia perangkat tersebut. Ketika AC sudah berada pada usia tertentu, maka akan ada perubahan perilaku AC. Hal ini terjadi terutama pada AC tipe lama. Keluaran yang diberikan akan berbeda saat masih dalam usia penggunaan yang wajar. Tapi meskipun begitu, AC tidak perlu langsung diganti karena dapat dilakukan perawatan yang baik untuk memberikan keluaran yang baik lagi seperti sebelumnya. Penelitian ini membuat alat yang dapat mendeteksi perilaku AC yang sudah tidak sehat, dimana keluarannya tidak normal. Perhitungan ini dilakukan dengan menggunakan algoritma yang mempelajari pola dari AC. Cara kerja alat ini adalah dengan menangkap suhu keluaran AC dan menghitung polanya dengan menggunakan algoritma K-Nearest Neighbor. Setelah didapat hasil perhitungan dengan algoritma, maka alat akan mengirim sinyal kepada OLED sebagai output yang dapat menunjukkan bahwa perilaku AC normal atau tidak. Jurnal ini berisi mengenai bagaimana sistem alat yang dibangun sehingga mendapatkan hasil dari AC yang diuji.

**Kata kunci :** K-Nearest Neighbor, inframerah, SmartAC, Monitoring dan Penyesuaian Suhu

---

### Abstract

Air Conditioner (AC) is a very common tool used to cool the room. The cooling process by the AC does not always run normally especially with the age of the device. When AC is at a certain age, there will be a change in AC's behavior. This happens mainly on old type AC. The given output will be different with the output that given before it reaches a certain age. In this case, AC does not need to be replaced immediately because we can give it a maintenance to provide good output again. This research makes a tool that can detect behavior of air conditioner that is not healthy. This calculation is done by using an K-Nearest Neighbor algorithm that studies the pattern of the AC. The way this tool works is to catch the AC output temperature and calculate the pattern using the algorithm. After the calculation results obtained, then the tool will send a signal to the OLED as an output that can indicate that the AC behavior is normal or not. This journal contains how the system tools are built so as to get the results of the AC being tested.

**Keywords:** K-Nearest Neighbor, infrared, SmartAC, Monitoring and Temperature Adjustment

---

## 1. Pendahuluan

### Latar Belakang

Air Conditioner (AC) merupakan sebuah alat yang sangat umum digunakan untuk mendinginkan ruangan. Proses pendinginan oleh AC tidak selalu berjalan normal apalagi seiring bertambahnya usia perangkat tersebut. Bertambahnya usia pada AC mempengaruhi kinerja pada pendinginan dan energy yang dikonsumsi[2]. Tentunya semakin tua AC maka semakin sulit untuk memberikan keluaran suhu dan konsumsi energy yang baik, maka adanya Smart AC memberi penerangan untuk mengatasi masalah penuaan pada AC ini.

Pada kenyataannya, tentunya banyak didapati AC yang bisa mengatur suhu secara otomatis, namun hal ini tidak berlaku pada AC tipe lama. Pada tipe AC lama, masih memerlukan inputan dari pengguna. Meskipun diberikan input, belum tentu juga AC tersebut dapat mengeluarkan suhu sesuai inputan. Misal bila kita nyalakan pada suhu 24, seiring berjalannya waktu AC tidak selamanya dapat mengeluarkan suhu tersebut secara normal.

Untuk tahu antara normal dan tidak normal, kita perlu mempelajari bagaimana nilai AC yang baik dan optimal[4]. AC yang normal adalah AC yang mengeluarkan suhu sesuai dengan inputan remote oleh user dan memiliki kelembaban yang tinggi. Kelembaban yang tinggi ini dikarenakan ruangan yang diisi oleh udara dari AC sehingga partikel air dalam ruangan mengering, jadilah nilai kelembaban tinggi pada ruangan tersebut. Algoritma K-Nearest Neighbor (KNN) yang diterapkan pada penelitian ini mampu membaca pola AC untuk mengetahui apakah AC tersebut mengeluarkan suhu secara normal atau tidak. Algoritma KNN adalah algoritma dalam menentukan klasifikasi. Cara kerja algoritma ini adalah dengan membaca dataset dan mengklasifikasikannya pada suatu state tertentu[8].

Berdasarkan penelitian sebelumnya, algoritma KNN ini memiliki tingkat akurasi yang bagus bila diterapkan pada alat Internet of Things (IoT), itulah alasan mengapa algoritma ini dipilih[7]. Selain itu, parameter perhitungan adalah dua fitur, yaitu: temperatur dan kelembaban dengan satu label yaitu normal atau tidak normal yang dimana algoritma ini cocok untuk digunakan. Algoritma KNN ini memerlukan dua data untuk dijadikan percobaan memahami pola AC, yaitu data *train* dan data *test*. Dengan kedua data ini, algoritma akan dipelajari bagaimana pola dari AC yang mengeluarkan suhu secara normal dan tidak normal untuk nantinya dijadikan sebagai parameter pengukur dalam pengujian. Semakin banyak data *train* maka semakin bagus tingkat akurasinya.

Data sebelum diproses akan disimpan ke *thingspeak*, sebuah *platform* API dan *web service* yang disediakan secara gratis untuk mendukung pembuatan IoT[10]. Penyimpanan ini bertujuan untuk kita dapat memonitor nilai secara realtime. Penyimpanan data ke *thingspeak* dilakukan secara otomatis dengan menggunakan alat SmartAC. Selain dapat mengirim data ke *thingspeak*, alat juga dapat mengirim perintah kepada AC ketika sudah dapat hasil perhitungan KNN. Perintah tersebut berupa sebuah sinyal inframerah yang akan menaik-turunkan suhu dengan memberi inputan kepada AC.

### Topik dan Batasannya

Dalam pengerjaannya, masalah yang diangkat adalah bagaimana rancangan dan implementasi alat SmartAC yang menerapkan algoritma K-Nearest Neighbor dimana dapat mengadaptasi pengaturan suhu berdasarkan kinerja AC dalam mengklasifikasikan state dari keluaran suhu AC tersebut serta bagaimana kinerja algoritma tersebut setelah diterapkan pada alat. Dalam hal ini pengujian penelitian dilakukan pada dua AC yang pengambilan datanya hanya dilakukan pada satu AC dalam satu waktu.

### Tujuan

Tujuan dari penelitian ini adalah untuk merancang dan mengimplementasi alat SmartAC yang menerapkan algoritma K-Nearest Neighbor dan bagaimana kinerja algoritma tersebut pada alat yang diterapkan.

### Organisasi Tulisan

Penulisan jurnal Tugas Akhir ini disusun dalam beberapa bagian yang dari setiap bagian, berisi data-data berikut :Bagian 1 - Pendahuluan, Bagian 2 - Studi Terkait, Bagian 3 - Sistem yang dibangun, Bagian 4 - Evaluasi, Bagian 5 - Kesimpulan.

## 2. Studi Terkait

Penelitian sebelumnya banyak mengangkat topic pengaturan temperatur suatu ruangan namun tidak terlalu banyak yang berkaitan dengan AC. Salah satu diantara penelitian mengenai pengaturan temperatur adalah penerapan "*Fuzzy Logic Control*" untuk menghitung suhu optimal yang dikeluarkan AC jika suatu ruangan dalam kondisi tertentu[4]. Pada penelitian tersebut, tujuannya adalah untuk mendapatkan nilai optimal dari suatu AC namun tidak dilakukan pada AC secara langsung. Penelitian hanya dilakukan dengan melakukan perhitungan secara matematis dimana hasil keluaran adalah nilai atau angka. Ada penelitian yang melakukan pemodelan atau simulasi, namun tidak menggunakan algoritma melainkan hanya menggunakan timer[1].

Selain penelitian mengenai pengaturan suhu tersebut, terdapat penelitian mengenai penerapan algoritma K-Nearest Neighbor untuk perangkat *Internet of Things* atau IoT. Penelitian tersebut tidak menerapkan algoritma pada AC, namun kepada alat lain yaitu *Field Programming Gate Arrays* (FPGA) yang mana merupakan perangkat IoT semacam RFID *tapping*[7]. Adapula penelitian tentang perbandingan hasil dari algoritma-algoritma yang ada ketika diterapkan pada alat IoT yang mana menjadi salah satu alasan mengapa penulis memilih algoritma K-Nearest Neighbor dalam penelitiannya[5].

Untuk dapat mengetahui nilai keluaran AC, diperlukan alat monitoring yang dapat bekerja secara *realtime*. Dalam penelitian ini digunakan *Thingspeak*. *Thingspeak* merupakan *platform* gratis untuk memonitoring perangkat IoT. Penyimpanan datanya dapat dilakukan secara *realtime* selama alat dapat terhubung ke jaringan internet untuk melakukan unggah data. *Thingspeak* dapat mengakomodasi banyak data untuk disimpan dalam bentuk file yang berbeda[10].

### 3. Sistem yang Dibangun

#### 3.1 Spesifikasi Alat

Alat SmartAC terdiri dari tiga komponen, yaitu: NodeMCU, sensor DHT11 dan IR. Sensor DHT11 adalah sensor temperatur dan kelembaban yang digunakan untuk menangkap keluaran AC sedangkan sensor *Infrared*(IR) digunakan untuk menerima dan mengirim perintah dari *remote* ke AC. NodeMCU adalah mikrokontroler jenis ESP8266 yang didalamnya sudah terdapat modul wifi sehingga tidak diperlukan tambahan modul untuk dapat terhubung kedalam jaringan[6]. Hal ini yang menjadi salah satu alasan pemilihan alat tersebut untuk digunakan. Mikrokontroler inilah yang mengirim hasil tangkapan nilai dari sensor DHT11 untuk dikirim ke *thingspeak*, begitu juga untuk mengirim sinyal infrared ke AC.

#### 3.2 Spesifikasi Ruangan

Ruangan yang digunakan untuk pengambilan data adalah ruangan berukuran 5 x 5 meter dengan jumlah AC adalah satu buah. Ruangan yang digunakan tidak hanya satu karena dibutuhkan pengambilan data yang banyak dari AC yang berbeda. Kegiatan pengambilan data ini meliputi: pengambilan data training, data test serta pengujian data pada dua AC yang berbeda. Pada setiap kegiatan yang melibatkan pengambilan data dilakukan pada ruangan yang berbeda sehingga dijumlahkan ada 4 ruangan yang dipakai selama kegiatan penelitian ini.

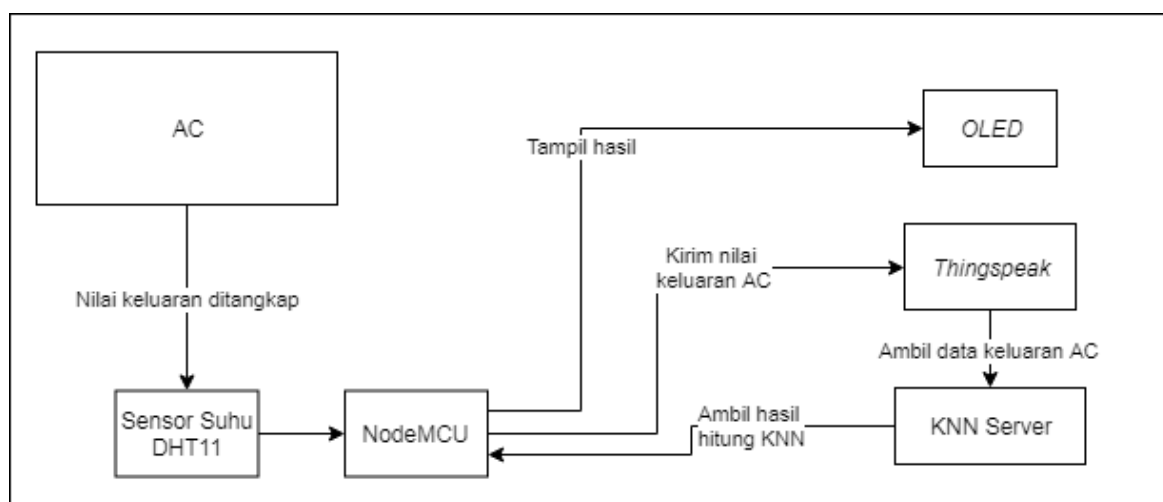
#### 3.3 Perbedaan KNN dan Fuzzy

Fuzzy merupakan algoritma yang sangat umum ditemui pada penelitian mengenai Internet of Things, namun tidak dalam penelitian ini. Fuzzy memiliki keunggulan dimana data tidak perlu dilakukan training, sehingga dapat dilakukan klasifikasi secara langsung ketika data didapat. Kegiatan klasifikasi ini melalui proses *ruling*, dimana kita harus membuat *rule* sendiri untuk membuat klasifikasi tertentu. Namun ada masalah dalam hal ini yaitu, pembuatan *rule* yang harus benar dan teliti. Ketika *rule* benar, tentunya data yang diolah akan memberikan hasil yang benar pula dan sebaliknya bila *rule* salah maka hasil akan salah.

Akibat dari *ruling* ini, Fuzzy hanya dapat mengklasifikasikan data dengan *rule* dan parameter yang sebelumnya telah diatur. Pada kenyataannya, tidak selamanya suatu parameter selalu sama, misal pada temperatur dan kelembaban. Data yang didapat berubah secara terus menerus. Sebagai contoh, pada suatu ruangan yang nilai temperatur 24<sup>0</sup>C kita mendapat kelembaban senilai 78%. Pada kondisi ini kita dapat mendefinisikan ruangan tersebut adalah kering, sehingga udara keluaran AC bersifat normal. Namun bila nilai kelembaban berubah tapi temperatur tetap sama, tentu bisa saja hasilnya berbeda. Selain itu, apabila kita menggunakan Fuzzy maka akan terlalu banyak *rule* yang dimasukkan dimana setiap temperatur memiliki *rule* masing-masing. Untuk itu penulis menggunakan algoritma KNN yang dapat menutupi kekurangan tersebut.

#### 3.4 Arsitektur Sistem dan Flowchart

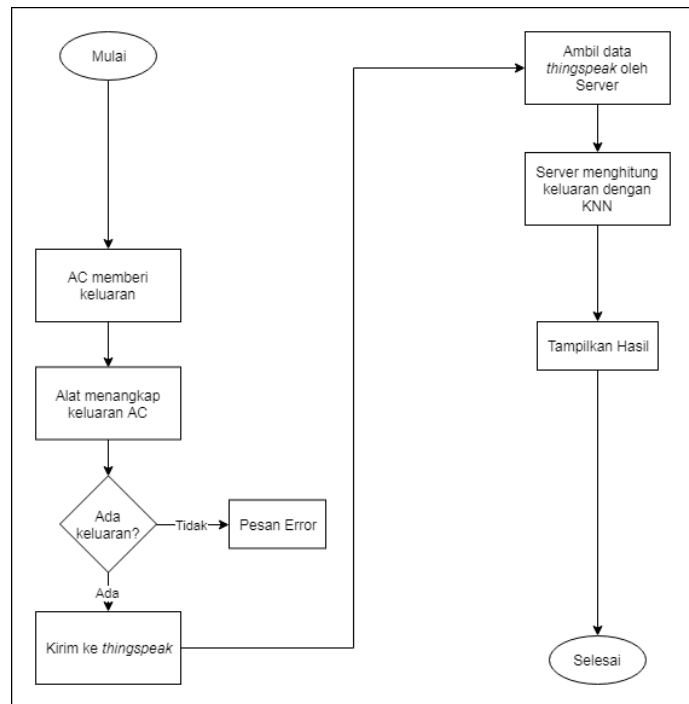
Adapun arsitektur sistem yang diajukan adalah seperti gambar 1:



**Gambar 1.** Gambaran Umum Sistem SmartAC

Pada arsitektur tersebut terdapat empat bagian yaitu: *remote* AC, AC, alat SmartAC, *Buzzer*, *thingspeak* dan API. *Remote* bertugas mengirimkan sinyal perintah kepada AC, dapat berupa perintah menyala atau mengatur suhu dll. Kemudian ketika AC sudah bekerja setelah diberi perintah oleh *remote*, alat SmartAC akan melakukan tugasnya. Alat akan menangkap suhu keluaran AC dengan sensor dan mikrokontroler yang terdapat didalamnya, kemudian mengirimnya ke *thingspeak*. Setelah sampai di *thingspeak*, akan diambil data menggunakan API yang

kemudian dihitung nilai keluarannya dengan K-Nearest Neighbor apakah termasuk kategori normal apa tidak. Bila hasil perhitungan algoritma adalah **tidak normal**, maka alat mikrokontroler akan mengirim perintah untuk menyalakan OLED dan memberikan pemberitahuan. Alur sistem dapat dilihat pada gambar 2 dibawah ini:



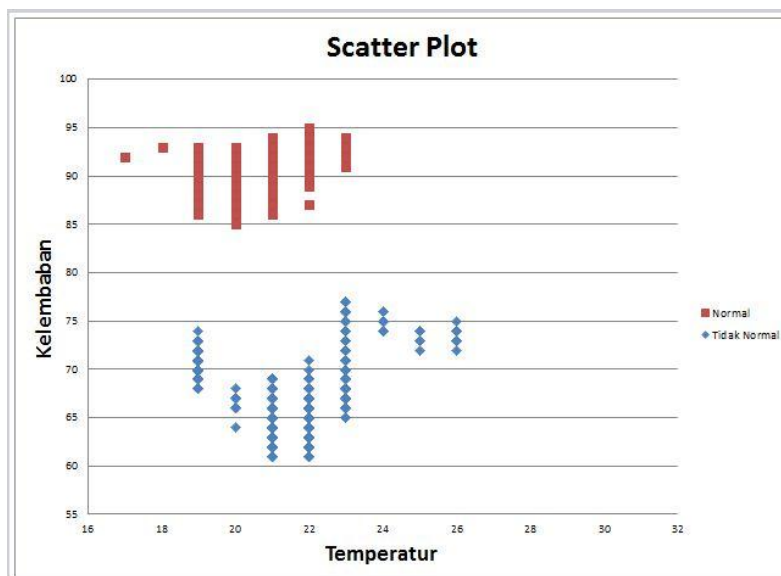
Gambar 2. Alur sistem SmartAC

#### 4. Evaluasi

##### 4.1 Skenario Pengujian

Pengujian dilakukan dengan persiapan sebagai berikut: pengambilan *data train* dan *data test*, pengujian akurasi algoritma KNN, pengambilan data pengujian dan unggah data ke thingspeak serta menguji data pengujian dengan algoritma KNN.

Pengambilan data *train* dan *data test* keduanya dilakukan dengan memasang sensor suhu DHT11 diruangan yang terdapat AC didalamnya. Pengambilan data ini bertujuan untuk menjadi pembelajaran pola AC oleh algoritma KNN. AC yang diambil datanya ada dua jenis, yaitu normal dan tidak normal. Total jumlah data yang diambil adalah 2700 data dimana data *test* yang digunakan sebanyak 500 atau 18% dari total data dan sisanya adalah data *train*. Scatter plot dari data dapat dilihat pada gambar 3.



Gambar 3. Scatter plot data normal dan tidak normal

Pengambilan data pengujian mirip dengan pengambilan data *train* dan data *test*. Perbedaannya adalah, dalam pengambilan data ini kita tidak tahu apakah AC dalam keadaan normal atau tidak normal. Hal ini akan kita ketahui setelah dilakukan proses klasifikasi dengan algoritma.

Cara kerja pengujian akurasi algoritma KNN adalah dengan membaca data yang ada di data *test* serta di data *train*. Setiap satu *record* di data *test* akan dibandingkan dengan seluruh *record* yang ada di data *train*. Dari kegiatan tersebut, akan dicari masing-masing jaraknya (nilai euclidenya) untuk diambil sebanyak k-buah terdekat. Rumus perhitungan jarak:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

Dimana  $a = a_1, a_2, \dots, a_n$  dan  $b = b_1, b_2, \dots, b_n$  mewakili n nilai atribut dari dua record.

4.2 Analisis Hasil Pengujian

4.2.1 Pengujian Akurasi Algoritma KNN

Hasil Pengujian Akurasi Algoritma dapat dilihat pada tabel 1.

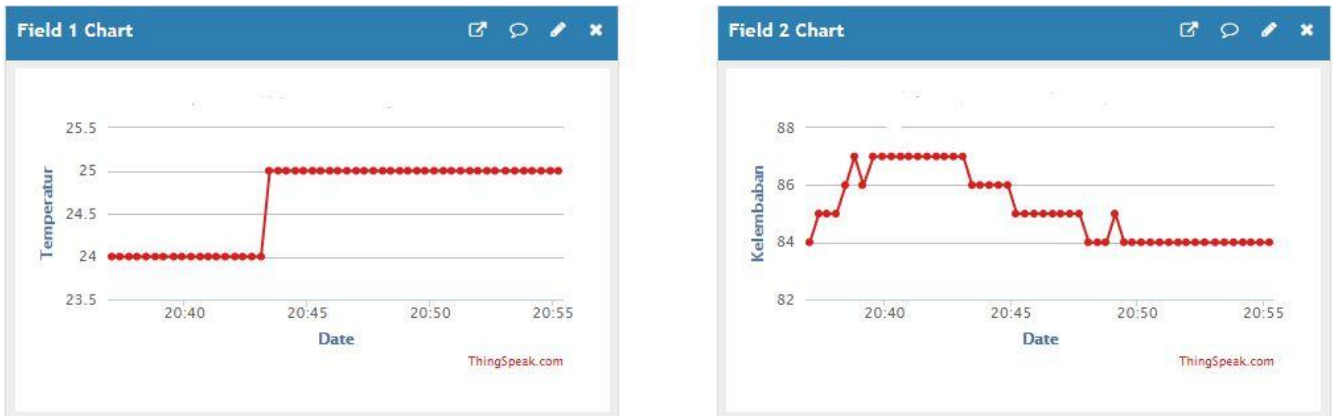
Tabel 1. Hasil Pengujian Akurasi KNN

Nilai K	Akurasi
3	100%
5	100%
7	100%

Dari hasil diatas, akurasi dari algoritma KNN yang diterapkan pada masalah ini sangat baik. Baik dengan nilai k yang 3, 5 atau 7, ketiganya memberi akurasi 100%. Hal ini terjadi karena bila dilihat pada gambar 3 sebelumnya, nilai dari temperatur dan kelembaban antara AC normal dan tidak normal memiliki rentang yang cukup jelas terutama pada fitur kelembaban. Sehingga ketika ada data baru yang ingin dihitung nilai euclidean-nya akan menjadi lebih mudah dan hasilnya jelas mendekati pada titik normal atau tidak.

4.2.2 Unggah Data AC ke Thingspeak

Hasil unggahan data pengujian ke thingspeak dapat dilihat pada gambar 4 dan 5.



Gambar 4. Grafik pengujian AC A



Gambar 5. Grafik pengujian AC B

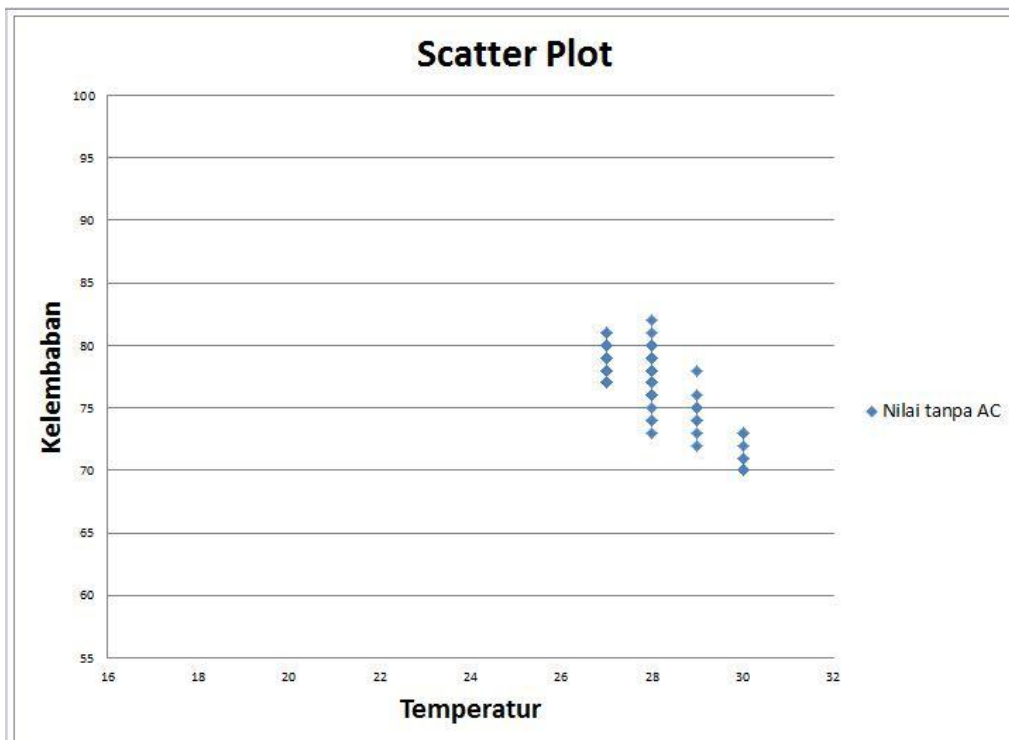
Unggahan yang dilakukan oleh perangkat NodeMCU ke *thingspeak* berjalan dengan benar. Hal ini dapat dilihat dari munculnya grafik ketika kita buka API *thingspeak* yang diset pada Arduino. Data dari kedua gambar diatas adalah data dari AC pengujian. Data ini yang akan diklasifikasikan oleh algoritma untuk dihitung nilainya apakah normal atau tidak. Seperti halnya yang dilihat dari gambar 3, pada grafik *thingspeak* juga memiliki perbedaan rentang data. Pada grafik AC A kita lihat dengan suhu 24 dan 25 memberikan kelembaban pada sekitaran nilai 84-87, sedangkan grafik AC B dengan suhu yang sama memberikan rentang kelembaban dari 75-84. Selain grafik, *thingspeak* juga dapat memberikan format penyimpanan data dalam bentuk tabel yang dapat dilihat pada gambar 6.

created_at	ID	Temperatur	Kelembaban	created_at	ID	Temperatur	Kelembaban
2018-07-19 13:37:03 UTC	1	24	84.00	2018-07-19 08:57:07 UTC	21	28	77.00
2018-07-19 13:37:24 UTC	2	24	85.00	2018-07-19 08:57:30 UTC	22	28	77.00
2018-07-19 13:37:45 UTC	3	24	85.00	2018-07-19 08:57:51 UTC	23	28	78.00
2018-07-19 13:38:06 UTC	4	24	85.00	2018-07-19 08:58:12 UTC	24	28	77.00
2018-07-19 13:38:28 UTC	5	24	86.00	2018-07-19 08:58:33 UTC	25	28	77.00
2018-07-19 13:38:49 UTC	6	24	87.00	2018-07-19 08:58:55 UTC	26	28	77.00
2018-07-19 13:39:10 UTC	7	24	86.00	2018-07-19 08:59:16 UTC	27	28	77.00
2018-07-19 13:39:34 UTC	8	24	87.00	2018-07-19 08:59:37 UTC	28	28	77.00
2018-07-19 13:39:55 UTC	9	24	87.00	2018-07-19 09:00:00 UTC	29	28	78.00
2018-07-19 13:40:17 UTC	10	24	87.00	2018-07-19 09:00:21 UTC	30	28	78.00

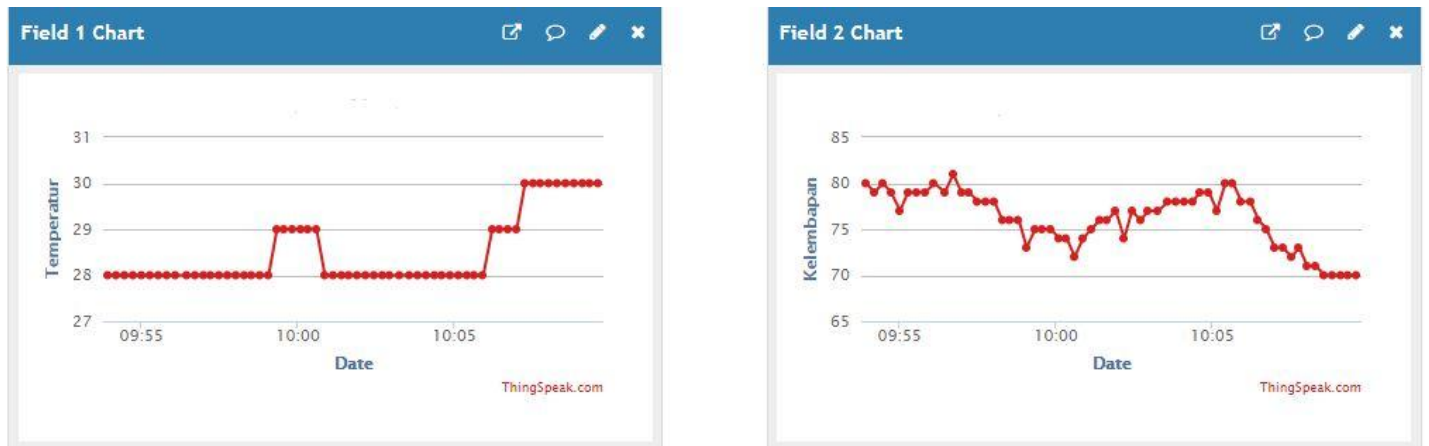
Gambar 6. Hasil export ke format csv dari *thingspeak*

#### 4.2.3 Uji Hasil Pada Ruangan Tanpa AC

Grafik temperatur menunjukkan nilai yang tinggi, yaitu pada rentang 28 sampai 30 derajat. Pengujian ini diambil di ruangan yang tidak dipasang AC sehingga nilai berada pada pada rentang tersebut. Sebenarnya nilai tersebut terhitung baik mengingat data diambil pada ruangan yang tidak dipasang AC. Karena nilai temperatur dan kelembaban berada pada nilai kewajaran untuk ruangan tanpa AC. Beberapa penelitian sebelumnya mengatakan bahwa kelembaban ruangan sewajarnya adalah dibawah 80%. Grafik dan scatter plot hasil penangkapan temperatur dan kelembaban pada ruangan tanpa AC dapat dilihat pada gambar 7 dan 8.



Gambar 7. Scatter Plot Ruangan Tanpa AC



**Gambar 8.** Grafik Pengujian Ruangn Tanpa AC

Setelah nilai tersebut diunggah ke *thingspeak* maka selanjutnya diukur sebagai percobaan. Hal ini sebagai bentuk percobaan untuk menilai bahwa perhitungan yang dimasukkan ke algoritma dapat diimplementasikan dengan lingkungan sebenarnya. Hasil yang didapat adalah ada pada tabel 2.

**Tabel 2.** Hasil klasifikasi ruangan tanpa AC

Nilai Temperatur	Nilai Kelembaban (rata-rata)	Hasil Klasifikasi
28	78%	Tidak Normal
29	75%	Tidak Normal
30	70%	Tidak Normal

Berdasarkan hasil perhitungan pada tabel 2, dapat dilihat bahwa hasil pada ruangan tanpa AC adalah tidak normal. Hal ini tentu saja didapat karena ruangan tersebut lembab dikarenakan tidak ada AC yang mengeluarkan udara diruangan tersebut. Pengujian ini digunakan untuk mencoba hasil dari alat apabila diletakkan pada ruangan yang tidak terdapat AC didalamnya.

#### 4.2.4 Analisis Hasil Pengujian Data

Pengujian merupakan kegiatan untuk mengukur tingkat akurasi pada sistem yang dibangun. Bertujuan untuk mengetahui apakah sistem bekerja dengan benar apa tidak. Data yang diambil adalah data terakhir yang dikirim ke *thingspeak*. Dalam pengujian diatas, AC A didapat hasil sebagai AC “Normal” dan AC B sebaliknya, “Tidak Normal”.

Hasil dari pengujian kemudian disesuaikan dengan kondisi dilapangan, dimana memang benar AC B tidak memberikan keluaran suhu yang baik atau secara singkat tidak dingin. Hal ini berarti hasil dari algoritma yang diterapkan pada sistem benar dalam mempelajari pola dan mengklasifikasikan hasil keluaran AC.

Pengujian pada ruangan tanpa AC memberikan hasil “Tidak Normal”. Hal ini terjadi karena ruangan tanpa AC tidak ada suatu udara yang masuk secara terus menerus. Bila pada ruangan yang dipasang AC, tentunya ada udara yang dihembuskan ke dalam ruangan berasal dari AC tersebut. Inilah yang memberi pengaruh nyata pada nilai kelembaban yang ada pada ruangan. Dengan perhitungan yang memberikan hasil Tidak Normal pada pengujian tersebut, maka hasil ini adalah benar.

## 5. Kesimpulan

### 5.1 Kesimpulan

Algoritma K-Nearest Neighbor dapat diimplementasikan pada perangkat IoT untuk mengklasifikasikan suhu sesuai kinerja AC. Hasil yang diberikan juga memiliki tingkat akurasi yang bagus. Algoritma menghitung perbandingan jarak antara temperatur dan kelembaban yang ada. Semakin tinggi kelembaban maka semakin kering ruangan, dimana hal tersebut menjadi salah satu indikasi bahwa AC bekerja mengeluarkan udara dengan baik

Alat SmartAC yang dibangun dengan mikrokontroler dan sensor bekerja dengan baik sesuai yang diharapkan, alat dapat menangkap udara keluaran dari AC dan mengklasifikasikan bahwa keluaran tersebut normal atau tidak normal. Alat ini diintegrasikan dengan API sehingga dapat melakukan klasifikasi menggunakan algoritma KNN dan kemudian memberikan pemberitahuan kepada user.

## 5.2 Saran

Berdasarkan pengujian yang telah dilakukan, masih banyak kekurangan yang dapat diperbaiki salah satunya adalah dengan memanfaatkan tidak hanya satu sensor saja, dengan begitu bisa didapat parameter lainnya sehingga perhitungan dapat lebih akurat. Algoritma KNN dalam penelitian ini memang memberikan nilai akurasi yang baik, tapi ada kemungkinan bahwa algoritma lain dapat lebih direkomendasikan untuk dipakai. Selain itu sebaiknya menggunakan *platform* monitoring yang lebih lengkap. Penelitian selanjutnya juga diharapkan dapat mengembangkan baik dari sisi algoritma maupun fungsionalitas alat yang dibangun misalnya dengan menambahkan *auto adjustment* pada alat sehingga tidak perlu ada pengaturan suhu sendiri oleh user.

## Daftar Pustaka

- [1] J. Dejvices and N. Tanthanuch, "A Simplified Air-conditioning Systems Model with Energy Management," 2016.
- [2] B. Yu, Z. Hu, M. Liu, H. Yang, Q. Kong and Y. Liu, "Review of Research on Air-conditioning Systems and Indoor Air Quality Control for Human Health," 2009.
- [3] Mustakim and G. O. F, "Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa," 2016.
- [4] Y. D. Aryandhi and M. W. Talakua, "Penerapan Inferensi Fuzzy untuk Pengendali Suhu Ruangan Secara Otomatis Pada Air Conditioner (AC)," 2013.
- [5] L. Chettri, S. Pradhan and R. Chettri, "Internet of Things Comparative Study on Classification Algorithms k\_NN Naive Bayes and Case based Reasoning," 2015.
- [6] H. Yuliansyah, "Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture," 2016.
- [7] H. M. Hussain, K. Benkrid and H. Seker, "An Adaptive Implementation of a Dynamically Reconfigurable K-Nearest Neighbour Classifier On FPGA," 2012.
- [8] T. Srivastava, "Analytics Vidhya," 10 October 2014. [Online]. Available: <https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/>. [Accessed 19 October 2017].
- [9] M. A. G. Maureira, D. Oldenhof and L. Teernstra, "ThingSpeak – an API and Web Service for the Internet of Things," 2015.
- [10] S. Pasha, "Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis," 2016.