

# Pengamatan Cuaca Lokal secara Multi Node dengan Internet of Things dan Django Framework

Muh Aris Saputra <sup>a,1,\*</sup>, Wahyu Cahyo Utomo <sup>a,2</sup>, Ahmad Bagus Setiawan <sup>a,3</sup>, Ilham KhEFI Ramadhanu <sup>a,4</sup>

<sup>a</sup>Teknik Informatika, Universitas Nusantara PGRI Kediri, 64112, Indonesia

<sup>1</sup>[aris.saputra@unpkdr.ac.id](mailto:aris.saputra@unpkdr.ac.id) \*; <sup>2</sup>[wahyu.utomo@unpkdr.ac.id](mailto:wahyu.utomo@unpkdr.ac.id); <sup>3</sup>[bagus.este@gmail.com](mailto:bagus.este@gmail.com); <sup>4</sup>[ilhamramadhanu25@gmail.com](mailto:ilhamramadhanu25@gmail.com)

\* Korespondensi penulis

Submission 15/4/2024, Revision 22/5/2024, Accepted 24/5/2024

## Abstract

Weather is one of the challenges that humans must experience in their activities. Like the MSMEs of sand crackers in Kediri Regency. This MSME uses sunlight as a means of drying products. What often becomes a problem is unpredictable weather conditions which reduce productivity and quality. Therefore, a real-time local weather observation system is needed to anticipate sudden weather changes. In this research, an IoT-based local weather condition measurement tool will be connected to a system built using the Django framework. This system and tools were tested for eight days. So it was concluded that the system was successfully built with data collection accuracy of 96.31%. Measurements are carried out every 5 minutes or a time frame of 5 minutes. In addition, this system supports observations in several places at once. This multiple node concept is used to detect local weather changes in the surrounding area. So it is not concentrated in the MSME area.

Keywords: Internet of Things, Django Framework, local weather, MSME, Convection Clouds

## Abstrak

Cuaca merupakan salah satu tantangan yang harus dialami manusia dalam beraktivitas. Seperti halnya UMKM kerupuk pasir di Kabupaten Kediri. UMKM ini memanfaatkan sinar matahari sebagai alat pengeringan produk. Yang sering menjadi permasalahan adalah kondisi cuaca yang tidak dapat diprediksi sehingga menurunkan produktivitas dan kualitas. Oleh karena itu, diperlukan sistem pengamatan cuaca lokal secara real-time untuk mengantisipasi perubahan cuaca yang tiba-tiba. Dalam penelitian ini, alat pengukuran kondisi cuaca lokal berbasis IoT akan dihubungkan pada sistem yang dibangun menggunakan framework Django. Sistem dan alat ini diuji selama delapan hari. Sehingga disimpulkan sistem berhasil dibangun dengan akurasi pengumpulan data sebesar 96,31%. Pengukuran dilakukan setiap 5 menit atau rentang waktu 5 menit. Selain itu, sistem ini mendukung observasi di beberapa tempat sekaligus. Konsep multiple node ini digunakan untuk mendeteksi perubahan cuaca lokal di area sekitar. Jadi tidak terkonsentrasi di bidang UMKM.

Kata kunci: Internet of Things, Django Framework, Cuaca Lokal, UMKM, Awan Konveksi

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## 1. Pendahuluan

Setiap individu dalam kesehariannya kemungkinan besar pernah mengalami hujan lokal atau biasa disebut dengan cuaca lokal. Kondisi ini terjadi ketika hujan turun didaerah tertentu saja dan tidak merata dalam satu kota bahkan kecamatan. Salah satu case-nya adalah seseorang menggunakan sepeda motor untuk berkendara. Pada saat berada dikecamatan A mengalami hujan dan menggunakan jas hujan. Kemudian pengendara melanjutkan perjalanan ternyata kecamatan B tidak hujan sama sekali. Hal ini merupakan permasalahan bagi manusia apabila melakukan kegiatan luar ruangan. Lebih khusus jika suatu UMKM melakukan proses produksi menggunakan panas matahari. Kondisi cuaca lokal ini disebabkan oleh awan konvektif. Awan ini menyebabkan penyebaran hujan yang tidak merata disebabkan adanya proses konveksi yang tidak stabil dalam

atmosfer [1]. Dalam penelitian yang sudah ada, cuaca lokal dapat diamati dengan beberapa variabel. Seperti variabel curah hujan, suhu dan kelembapan [2].

Cuaca lokal ternyata juga mempengaruhi UMKM yang menjadi produk unggulan kabupaten kediri. Terutama dalam bidang produksi krupuk pasir pada kecamatan tarokan kabupaten kediri [3]. Dalam prosesnya, UMKM ini masih menggunakan proses yang sederhana. Seperti pengeringan produk menggunakan sinar matahari secara langsung [3]. Apabila tidak ada matahari atau sering terjadi hujan mengakibatkan produksi tidak akan berjalan. Bahkan mempengaruhi kualitas produk yang dihasilkan bahkan gagal produksi karena produk tidak dapat kering dengan sempurna. Sedangkan pembeli menginginkan produk yg berkualitas dan tepat waktu dalam *supply* produk. Selain itu, apabila produk dijemur secara langsung dan terjadi cuaca lokal maka pelaku UMKM akan mengeluarkan *effort* untuk menjemur dan meneduhkan produk secara terus menerus. Sehingga dapat disimpulkan bahwa UMKM mengalami 2 permasalahan jika terjadi cuaca lokal yaitu, masalah dalam produksi dan *effort* yang lebih dalam menanggulangi hujan yang turun tidak merata.

Pemantauan yang dilakukan selama ini oleh pelaku UMKM adalah menggunakan metode visual dan manual. Dengan melihat bentuk dan warna awan. Hal ini masih sangat subjektif karena setiap individu dapat berbeda hasil pengamatannya. Dengan adanya permasalahan pengamatan cuaca maka dibutuhkan pendekatan lainnya. Dalam penelitian [4], [5], [6] dan [7], pendekatan yang digunakan adalah IOT (*Internet of Things*). Dalam penelitian [6], IOT berhasil daplikasikan kedalam kasus pemantauan rumah hijau. Sistem ini dibuat otomatis untuk memantau data kelembapan tanah, pencahayaan dan keadaan suhu. Setelah itu, sistem secara otomatis melakukan proses penyiraman. Pada penelitian lain, pendekatan IOT juga dapat diaplikasikan dalam monitoring tempat pengelolaan budidaya jamur tiram [7]. Dari dua penelitian dapat disimpulkan bahwa dengan pendekatan IOT faktor subjektifitas dapat dikurangi karena menggunakan sensor sehingga hasil pengukuran cenderung sama. Setelah sensor membaca data, sistem akan dikirimkan kedalam sebuah server sehingga terpusat kedalam satu tempat walaupun alat yang digunakan banyak (*multiple node*).

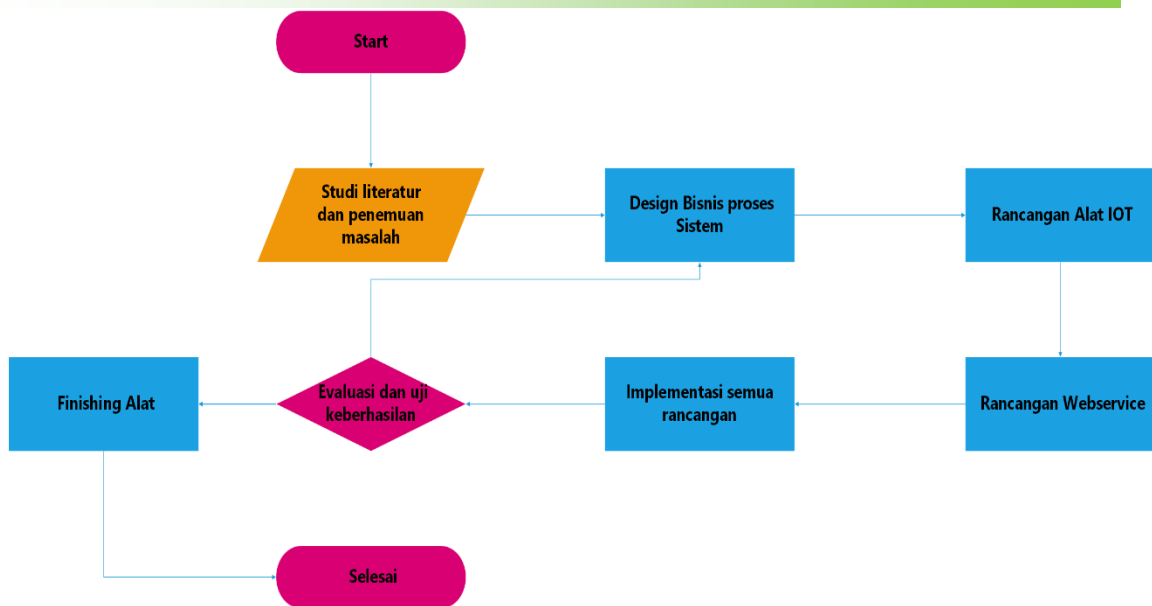
Data yang telah diambil oleh sensor IOT akan dikirimkan kedalam *web server* yang ada pada *cloud hosting*. Cara atau salah satu langkah pengiriman ini menggunakan *framework webservice API*. Pada penelitian [8], [9] dan [10], data IOT dikirimkan dengan menggunakan *web service API* yang dibangun menggunakan *flask framework*. Sedangkan pada penelitian [11], [12] dan penelitian [13], peneliti melakukan pengiriman data pembacaan sensor menggunakan *django framework*. Kedua *framework* memiliki kesamaan dalam bahasa pemrograman. Yaitu menggunakan bahasa pemograman *python*. Berdasarkan penelitian [8 - 12] dan [13], *framework* yang berpotensi dan terbukti bisa digunakan untuk mengirimkan data berbasis IOT adalah *flask framework* dan *django framework*. Selain itu, penggunaan sistem *berbasis webservice API* akan mendukung *input multiple node* dan mendukung berbagai *platform* untuk pengembangan kedepan.

Data sensor yang dikumpulkan dengan *webservice API* memiliki rangkaian waktu atau disebut dengan *time frame*. Secara sederhana *time frame* adalah rentan lama waktu atau jeda waktu dalam pengambilan data. *Time frame* dalam konsep OHLC (*open, high, low, close*) memiliki beberapa jenjang mulai dari satu menit, jam, 4 jam, harian, minggu hingga tahunan [14]. Salah satu kesimpulan dalam penelitian tersebut adalah, semakin kecil nilai *time frame* maka data akan semakin bagus untuk penelitian selanjutnya. Karena semakin kecil *time frame* tingkat *error* semakin kecil. Hal ini wajib dipertimbangkan, sebab data ini berpotensi untuk dilakukan penelitian lanjutan. Selain itu, OHLC dapat digunakan untuk menjelaskan *time frame* lebih besar berdasarkan *time frame* lebih kecil. Pernyataan tersebut sejalan dengan penelitian [15] yang membahas bahwa *time frame* yang semula bulanan bisa dirubah menjadi *time frame semester*.

Berdasarkan uraian permasalahan yang dihadapi oleh UMKM krupuk pasir kabupaten kediri. Serta studi literatur yang dilakukan sebelumnya. Maka dalam penelitian ini akan melakukan penelitian untuk membuat alat bantu berbasis IOT yang bersifat *multiple node* untuk merekam data cuaca lokal. Perekaman ini dilakukan dengan menggunakan *API webservice* dengan *django framework*. Sedangkan untuk *timeframe*, penelitian ini akan menggunakan *time frame* setiap 5 menit. Data yang terkumpul nantinya akan digunakan untuk penelitian selanjutnya.

## 2. Metode Penelitian

Setelah melakukan pengamatan masalah, penyimpulan masalah dan studi literatur penyelesaian, dibutuhkan beberapa langkah strategis yang harus dilakukan dengan beruntun. Dengan tujuan agar tujuan penelitian tercapai. Dikarenakan penelitian ini menggunakan *Internet of Things* maka dibutuhkan perencanaan dalam *hardware* maupun *software*. Selain itu alat IOT supaya dapat mengumpulkan data maka dibutuhkan *middleware* untuk menghubungkan masing-masing alat IOT atau *node* pengamatan. Berikut adalah penjabaran langkah tersebut.

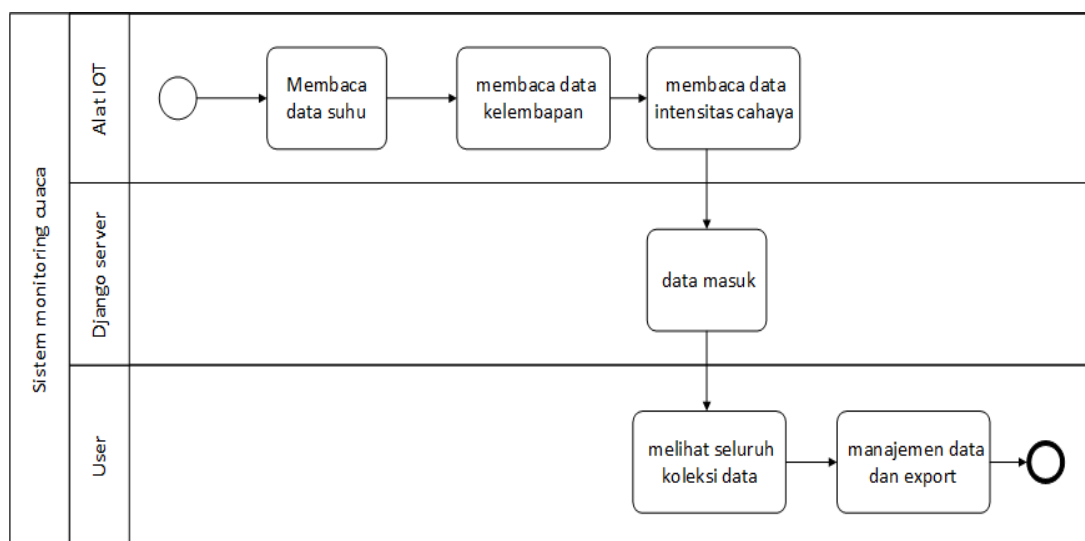


Gambar 1. Alur Rencana Pelaksanaan Penelitian

Sesuai dengan Gambar 1, penelitian ini dimulai dengan melakukan studi literatur dan menemukan potensi pendekatan yang cocok dengan permasalahan. Secara spesifik, penelitian ini akan memecahkan permasalahan pengamatan cuaca lokal. Dimana setiap *node* memiliki jarak tidak jauh. Dapat berupa jarak setiap desa satu *node* maupun setiap kecamatan satu *node*. Sehingga diharapkan data yang didapatkan merupakan interpretasi cuaca lokal bukan regional. Pengumpulan data lokal dilakukan menggunakan rangkaian sensor yang melakukan pengamatan secara *real time* 24 jam. Teknik atau metode yang digunakan dapat disebut dengan IOT. Karena setiap alat atau *node* saling terhubung ke server dengan menggunakan Internet. Sedangkan metode pengiriman atau komunikasi dalam penelitian ini adalah *REST API webservice*. Berikut adalah langkah spesifik yang dijabarkan sesuai dengan Gambar 1. (1)

2.1. Bisnis Proses Sistem

Dalam sistem yang dibangun tentu terdapat beberapa bagian yang saling terhubung. Sehingga dibutuhkan rancangan yang spesifik untuk menggambarkan sistem dan peran masing-masing elemen. Berikut adalah rancangan bisnis proses yang akan diaplikasikan.



Gambar 2. Rancangan Bisnis Proses

Pada gambar Gambar 2, sistem dimulai dari alat IOT yang membaca data suhu. Dilanjutkan alat membaca data kelembapan dan data intensitas cahaya. Data ini dikirimkan ke *django server*. Transfer ini menggunakan *webservice API* dengan *django framework*. Data yang sudah ditampung akan ditampilkan kepada user dalam

bentuk *dashboard*. Data yang dapat dilihat adalah data hasil pengamatan sensor. Selain itu sistem akan mendukung user untuk melakukan analisis sederhana. Seperti *filter*, *sorting* dan *searching*. Dan yang terpenting adalah sistem mendukung untuk melakukan *export* data. Dikarenakan, Data hasil pengamatan ini berpotensi untuk digunakan untuk melakukan penelitian maupun analisa.

## 2.2. Rancangan Alat IOT

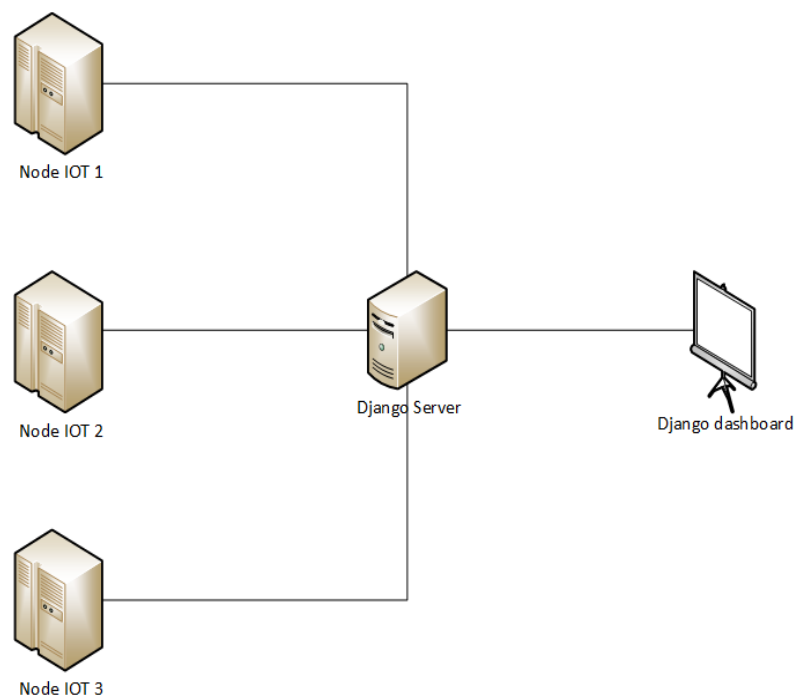
Alat IOT yang dibangun membutuhkan beberapa alat sesuai kebutuhan penelitian. Baik sensor maupun alat pendukung. Kebutuhan alat dan sensor lebih detail dituliskan pada Tabel 1.

Tabel 1. Alat dan kebutuhan sensor

| No | Nama Alat/Sensor  | Kegunaan                           |
|----|-------------------|------------------------------------|
| 1. | ESP32             | Mikrokontroler                     |
| 2. | DHT22             | Sensor pembaca suhu dan kelembapan |
| 3. | LDR Modul 4 Pin   | Sensor pembaca intensitas cahaya   |
| 4. | RTC Ds1307        | Sensor waktu realtime              |
| 5. | 4 CHANNEL IIC I2C | Konverter logic 5v ke 3v           |
| 6. | Stepdown LM2596   | Penurun tegangan                   |
| 7. | LM7805            | IC Penuruan tegangan ouput 5V 1A   |

Alat yang tercantum pada Tabel 1 merupakan alat pendukung untuk melakukan pengamatan data suhu, kelembapan dan intensitas cahaya. Selain itu, pada Tabel 1 terdapat beberapa alat pendukung seperti alat pendukung merubah tegangan maupun arus. Hal ini digunakan karena ESP32 membutuhkan *input* 5v. Sedangkan semua sensor yang digunakan merupakan sensor dengan kebutuhan *output* tegangan 5v. Sedangkan ESP32 hanya mendukung *output* sensor 3v. oleh karena itu, dibutuhkan konverter untuk dapat menggunakan sensor. Alat tersebut adalah IIC I2C.

## 2.3. Rancangan Sistem Penerima Data



Gambar 3. Multiple Node IOT

Alat dalam penelitian ini dirancang untuk *multiple node*. Dalam artian alat akan melakukan pengamatan dalam beberapa tempat sekaligus. Dalam Gambar 3, *multiple node* ini disimbolkan dan diberi nama *Node IOT*. Masing-masing alat akan mengirim data ke *server django* yang berada pada *hosting* atau *cloud*. Data tersebut akan dapat dilihat melalui *django dashboard*. Fitur ini merupakan satu paket yang disediakan oleh *django framework*.

Tabel 2. Rencana *Fields Webservice*

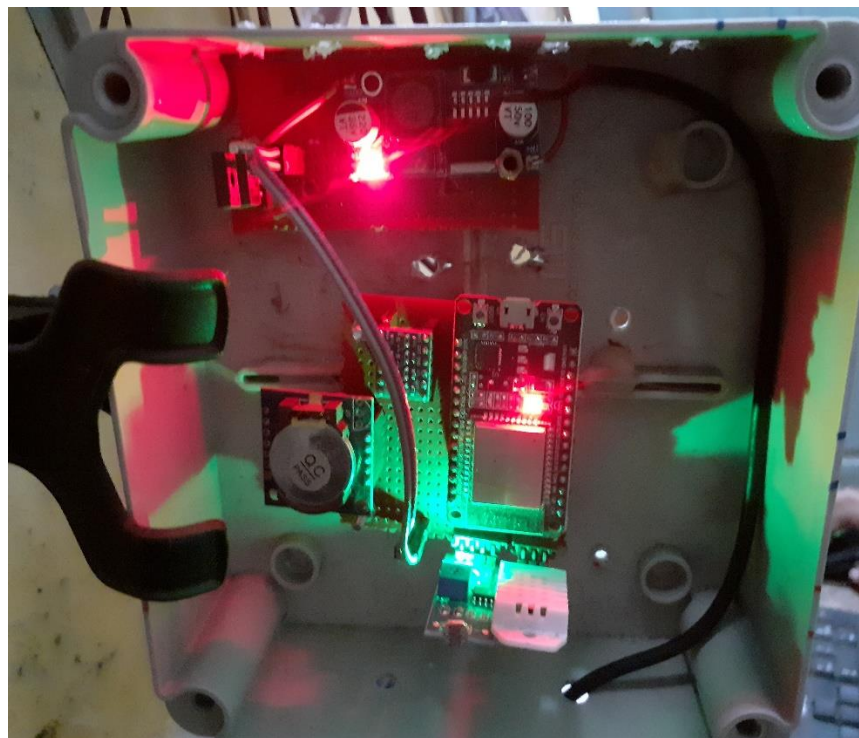
| No | Nama <i>field</i> | Keterangan                       |
|----|-------------------|----------------------------------|
| 1  | Stasiun Name      | Nama node alat IOT               |
| 2  | Suhu              | Nilai suhu dalam derajat celsius |
| 3  | Kelembapan        | Nilai kelembapan                 |
| 4  | Cahaya            | Nilai intensitas cahaya          |
| 5  | Waktu buat        | Waktu data masuk dalam sistem    |

Pada Tabel 2, terdapat beberapa *fields* data yang akan ditampung didalam *django framework*. Data tersebut dikumpulkan dengan menggunakan *web API* atau *backend framework*. *Fields* tersebut adalah data *stasiun name* yang merupakan nama *stasiun* alat pemantauan. *Fields* ini dapat berbeda jika menggunakan *multiple node*. Sehingga secara spesifik diketahui data dari mana dan dikirim oleh alat mana. Selanjutnya adalah *fields* data suhu, kelembapan dan cahaya. Ketiga *fields* ini merupakan hasil pengukuran sensor masing-masing sensor. *Fields* yang terakhir adalah waktu buat. *Fields* ini digunakan untuk memberikan keterangan waktu secara spesifik. Dalam artian kapan data diambil. Hal ini digunakan sebagai penunjang kedepan untuk diolah lebih dalam lagi.

### 3. Hasil dan Pembahasan

#### 3.1. Rancangan Alat Berhasil Diaplikasikan

Kebutuhan alat yang sebelumnya sudah direncanakan pada Tabel 1 berhasil diaplikasikan dan dapat berfungsi. Berikut adalah bentuk alat yang sudah saling terhubung dan dapat bekerja.



Gambar 4. Rangkaian Alat IOT

Keberhasilan rangkaian alat IOT pada Gambar 4 dapat dilihat dari lampu indikator yang menyala dengan baik dan terhubung. Lampu indikator merah pada bagian atas adalah rangkaian elektronik penurun tegangan yang semula 12 volt menjadi 5v. Besaran ini dapat diukur menggunakan *multi tester*. Selain itu apabila tegangan dibawah 5v bahkan terlalu besar akan berefek kepada alat yang lain. Pada penelitian ini, rangkaian penurun berjalan dengan baik sehingga sensor dan esp32 dapat beroperasi dengan baik. Indikator sensor berjalan dengan baik adalah lampu indikator yang menyala. Yaitu indikator warna hijau dan merah. Dalam tahapan awal ini maka dapat disimpulkan bahwa alat sudah siap untuk dihubungkan dengan *web API* berbasis *django framework*.

### 3.2. Rancangan Sistem Berhasil Diaplikasikan

Sistem yang dibangun terdiri dari 2 bagian utama. Pertama adalah bagian *backend* atau *webservice API*. Bagian ini bertugas sebagai penghubung antara alat IOT dengan *server cloud*. Masing-masing *node* akan mengirimkan data dengan menggunakan *HTTP GET method*. Bagian ini dibangun dengan menggunakan *framework django rest API*. Sedangkan bagian kedua adalah sistem yang dibangun dengan menggunakan *django administrator*. Keduanya merupakan bagian dari *django* tetapi memiliki bagian yang berbeda.

*Django administrator* digunakan sebagai *dashboard* penghubung *user* dengan *sistem*. *Framework* ini cukup *powerfull* karena mendukung berbagai DBMS (*database management system*). Pada *case* penelitian ini DBMS yang digunakan adalah *sqlite*. Selain itu, *framework* juga mendukung konsep *models*. Yaitu merubah *relation database* kedalam bentuk *class object oriented*. Sehingga transformasi antar DBMS sangat dinamis tanpa memikirkan *query* DBMS yang berbeda-beda. Berikut adalah kedua bagian sistem yang telah dibangun.

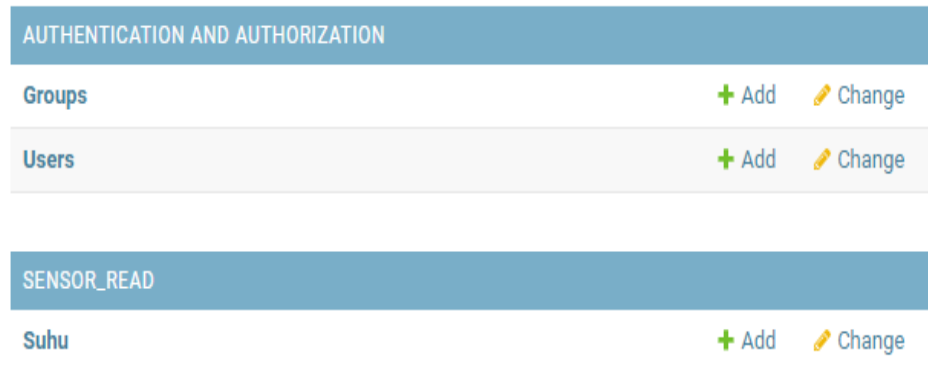


Gambar 5. Webservice Penghubung Alat IOT

Pada Gambar 5, *webservice* akan mengirimkan data berupa 4 bagian. Yaitu suhu, kelembapan, intensitas cahaya dan waktu pengamatan. Yang menarik dari sistem ini adalah waktu pengambilan bukan diambil dari sistem *django*. Tetapi dari *RTC time module* alat IOT. Sehingga waktu benar-benar *realtime* dan tidak terpengaruh *latency* koneksi internet. Sistem API ini akan mengirimkan ke sistem yang kedua. Yaitu sistem *django administrator* seperti gambar berikut.

# Django administration

## Site administration

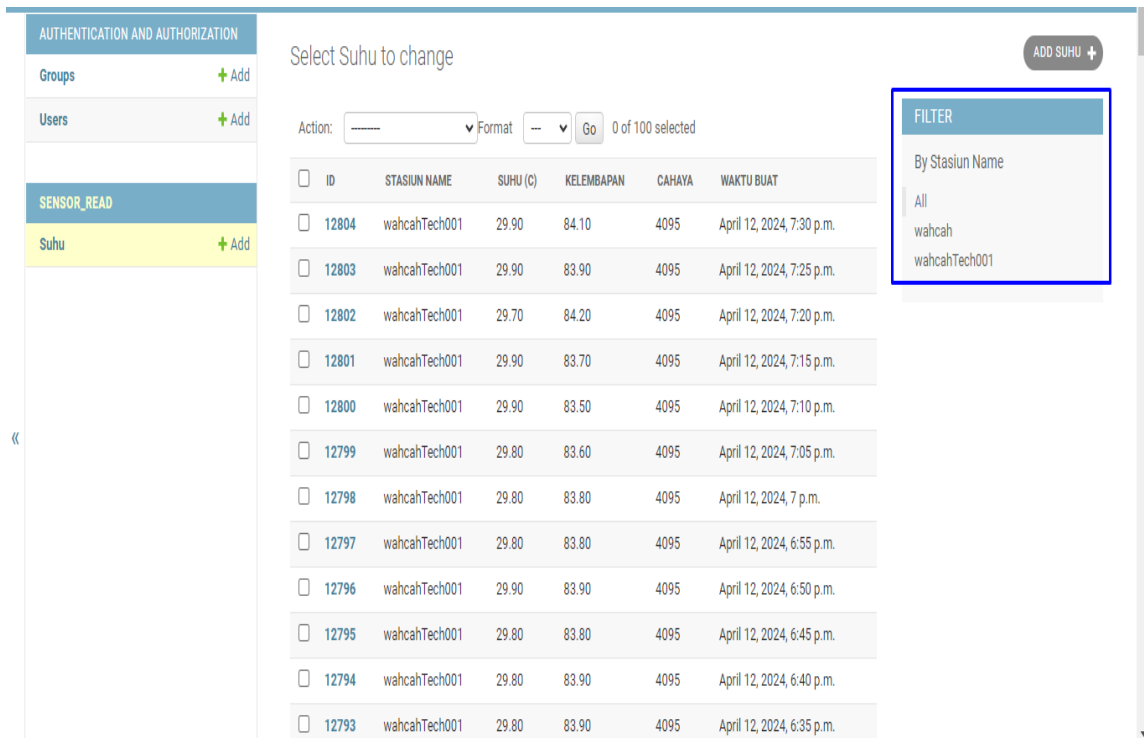


Gambar 6. Dashboard Django Administration

Django administrator memiliki fitur yang cukup lengkap. Seperti pada Gambar 6, django administrator mendukung multiple user dan groups. Sehingga user dapat diberikan hak akses khusus. Langkah ini dilakukan untuk memberikan setiap user level admin yang berbeda. Selain itu sistem ini bersifat sangat dinamis dan powerfull. Karena setiap perubahan database, sistem akan otomatis menyesuaikan. Sebagai case pada Gambar 6, IOT data akan ditampung kedalam tabel suhu.

### 3.3. Data Logging Multi Node

Data logging ini ditampilkan kedalam bentuk dashboard. Sistem sudah support multiple node sehingga data masing-masing node dapat dibedakan. Berikut adalah tampilan sistem serta data yang sudah masuk ke dalam sistem.



Gambar 7. Data View dalam Django Dashboard

Data view pada Gambar 7, secara spesifik memiliki data *fields* stasiun *name*. *fields* ini digunakan untuk membedakan masing-masing *node*. Dengan penambahan *fields* ini, *django* dapat memberikan fitur tambahan berupa *filter*. Fungsi ini digunakan untuk mempermudah *user* menemukan data. Jika *user* melakukan aksi klik pada kotak biru atau salah satu stasiun *name* maka data yang tampil adalah data stasiun *name* itu saja. Selain itu, data view ini juga mendukung data *export* berupa csv, excel dan format lainnya. Tujuan dari fitur ini adalah mempermudah pengguna untuk melakukan pengolahan data dalam penelitian lebih lanjut.

### 3.4. Penggunaan Gambar

Alat IOT dan Sistem yang dibangun menggunakan *django framework* diujicoba selama delapan hari. Berikut adalah rincian hasil *collect* data yang masuk ke dalam sistem.

Tabel 3. Presentase Keberhasilan Pengumpulan Data Harian

| Tanggal        | Data terkumpul | Data Gagal    | Presentase Keberhasilan |
|----------------|----------------|---------------|-------------------------|
| 12-04-2024     | 287            | 1             | 99,65%                  |
| 11-04-2024     | 288            | 0             | 100,00%                 |
| 10-04-2024     | 280            | 8             | 97,22%                  |
| 09-04-2024     | 273            | 15            | 94,79%                  |
| 08-04-2024     | 271            | 17            | 94,10%                  |
| 07-04-2024     | 269            | 19            | 93,40%                  |
| 06-04-2024     | 277            | 11            | 96,18%                  |
| 05-04-2024     | 274            | 14            | 95,14%                  |
| <b>Total</b>   | <b>2219</b>    | <b>85</b>     | <b>-</b>                |
| <b>Average</b> | <b>277,375</b> | <b>10,625</b> | <b>96,31%</b>           |

Alat yang dibangun sudah dilakukan uji coba selama delapan hari, yaitu tanggal 05 april 2024 sampai tanggal 12 april 2024. Pengujian ini menggunakan *time frame* atau *range* pengambilan data setiap lima menit. Dalam Tabel 3, terdapat ringkasan data yang didapatkan dari hasil ujicoba.

Data yang seharusnya terkumpul adalah 288. Dikarenakan setiap jam terdiri dari 12 kali pengambilan data. Sehingga satu hari adalah 12 dikali 24 jam. Namun dalam hasil ujicoba diperoleh data yang gagal ditransfer ke sistem *cloud*. Seperti pada tanggal 12 april 2024, diperoleh data gagal ditransformasikan sebanyak 1. Berdasarkan hasil analisis dan pengamatan, kegagalan ini disebabkan karena adanya gangguan *provider* internet, pemadaman listrik maupun *cloud* yang sedang perbaikan rutin.

Selain itu, dalam Tabel 3 juga terdapat informasi menarik. Sistem berhasil tereleasisasikan dengan akurasi pengumpulan data sebesar 96.31 %. Dengan artian bahwa dari total data yang terkumpul sebesar 2219 terjadi kegagalan pengumpulan data hanya sebesar 85. Hal ini cenderung sedikit bahkan jika tidak terjadi kesalahan faktor eksternal yang diuraikan sebelumnya. Sistem dapat mencapai akurasi 100% pengambilan data. Berikut adalah analisis lebih dalam waktu terjadinya kesalahan.

Tabel 4. Waktu Kesalahan dalam Interval *Time Frame*

| <i>Time frame</i> (5 Menit) | Data terkumpul | Data Gagal  | Presentase Keberhasilan |
|-----------------------------|----------------|-------------|-------------------------|
| 0                           | 187            | 5           | 97,40%                  |
| 5                           | 184            | 8           | 95,83%                  |
| 10                          | 185            | 7           | 96,35%                  |
| 15                          | 186            | 6           | 96,88%                  |
| 20                          | 181            | 11          | 94,27%                  |
| 25                          | 184            | 8           | 95,83%                  |
| 30                          | 185            | 7           | 96,35%                  |
| 35                          | 187            | 5           | 97,40%                  |
| 40                          | 187            | 5           | 97,40%                  |
| 45                          | 179            | 13          | 93,23%                  |
| 50                          | 185            | 7           | 96,35%                  |
| 55                          | 189            | 3           | 98,44%                  |
| <b>Total</b>                | <b>2219</b>    | <b>85</b>   | <b>-</b>                |
| <b>Average</b>              | <b>184,92</b>  | <b>7,09</b> | <b>96,31%</b>           |

Dari hasil pengamatan Tabel 4, kegagalan pengiriman data tidak terpusat pada salah satu interval saja tetapi menyebar kedalam seluruh interval. Sehingga terlihat bahwa keagal bukan karena ada faktor tertentu dalam sistem.

Tetapi terdapat faktor eksternal yang mempengaruhi. Seperti pemadaman listrik, gangguan *provider* dan *maintenance* rutin *provider cloud*. Namun perlu digaris bawahi, bahwa analisis ini dilakukan pada salah satu



*node*. Apabila salah satu *node* gagal mengirimkan data tidak akan mempengaruhi *node* yang lain karena berbeda lokasi dan *provider* yang digunakan.

#### 4. Kesimpulan

Berdasarkan uraian hasil penelitian, dapat disimpulkan bahwa penelitian ini berhasil diaplikasikan untuk membantu melakukan pemantauan kondisi cuaca lokal dalam berbagai tempat (multiple node) dengan satu sistem tersentral pada cloud. Hal ini Sesuai dengan konsep IOT (Internet of Things). Selain itu, Sistem ini berhasil dibangun menggunakan django framework dan menghasilkan akurasi keberhasilan pengumpulan data sebesar 96.31 % pada penggunaan time frame interval 5 menit.

Oleh karena itu, sistem dalam penelitian ini memiliki 2 kebermanfaatan. Yang Pertama, sistem dapat digunakan oleh UMKM krupuk pasir kabupaten kediri untuk pengamatan cuaca secara realtime dan mengurangi subjektifitas pengamatan. Dan yang Kedua, penelitian ini dapat menjadi dasar penelitian selanjutnya untuk penggunaan data maupun penyempurnaan alat IOT.

#### 5. Daftar Pustaka

- [1] Ninggar, R. D., Siregar, D. C., & Perdana, I. F. P. P. (2023). ANALISIS POLA DISTRIBUSI AWAN KONVEKTIF DI WILAYAH PROVINSI BANTEN BERBASIS RADAR CUACA. *Buletin Meteorologi, Klimatologi dan Geofisika*, 3(6), 35-43.
- [2] Arrashid, H., Sucahyono, D., Haryanto, Y. D., & Qomariyatuzzamzami, L. N. (2023). KONDISI DINAMIKA ATMOSFER SAAT HUJAN LEBAT DI KALIMANTAN SELATAN (PERIODE 12–17 JANUARI 2021). *Buletin Meteorologi, Klimatologi dan Geofisika*, 3(4), 32-44.
- [3] Appita, N. (2020). Produksi Krecek Kerupuk ditinjau dari Produksi Islam (Studi pada Produsen Krecek Kerupuk Desa Bulusari Kecamatan Tarokan Kabupaten Kediri) (Undergraduate (S1) thesis, IAIN Kediri).
- [4] Boboy, J. R., & Bata, E. S. (2023). Prototipe Sistem Monitoring Pelanggaran Zebra Cross di Lampu Merah Berbasis Internet of Things. *JITU: Journal Informatic Technology And Communication*, 7(2), 82-90.
- [5] Syelly, R., & Pranata, M. (2024). Teknologi Cerdas Untuk Pengendali Polusi Udara Pada Peternakan Ayam. *Technologica*, 3(1), 32-41.
- [6] Ismail, I., & Amin, R. M. (2023). Sistem Pemantauan Otomatis (IOT-Internet of Thing) Bagi Rumah Hijau. *Journal on Technical and Vocational Education*, 8(1), 156-169.
- [7] Abi Hudhoifah, M., & Mulyana, D. I. (2024). Implementasi Monitoring Suhu dan Kelembapan Kumbung jamur pada Budidaya Jamur Tiram dengan NodeMCU-ESP8266 di Desa Wirasana Purbalingga: Implementation of Temperature and Humidity Monitoring in Mushroom House for Oyster Mushroom Cultivation Using NodeMCU-ESP8266 in Wirasana Village Purbalingga. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(2), 472-480.
- [8] Kornienko, D. V., Mishina, S. V., Shcherbatykh, S. V., & Melnikov, M. O. (2021, November). Principles of securing RESTful API web services developed with python frameworks. In *Journal of Physics: Conference Series* (Vol. 2094, No. 3, p. 032016). IOP Publishing.
- [9] Avrylya, T. P., & Susetyo, Y. A. (2024). Perbandingan Response Time Pencarian Menggunakan Text Indexing Pada MongoDB dan ArangoDB Berbasis Web. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(3), 777-785.
- [10] Rahma, I. A., & Suadaa, L. H. (2023, December). Automated Indonesian Text Augmentation with Web-Based Application Using Flask Framework. In *Proceedings of The International Conference on Data Science and Official Statistics* (Vol. 2023, No. 1, pp. 96-108).
- [11] Wijaya, F., Jacobus, A., & Sambul, A. (2021). Implementation Of Web Services On University Library Information Systems. *Jurnal Teknik Informatika*, 16(4), 421-428.
- [12] Chang, Y., Ou, S., Jia, Q., Mo, M., & Wang, S. (2023, December). Design and Implementation of Django-based Quality Traceability System for Medicinal Dendrobium. In *3rd International Conference on Digital Economy and Computer Application (DECA 2023)* (pp. 615-627). Atlantis Press.
- [13] Ganesh, C., Saran, A., Flora, G. D., & Saravanan, M. (2022, December). A cloud-based router management system for enterprise network management. In *2022 International Conference on Computer, Power and Communications (ICCPC)* (pp. 153-157). IEEE.
- [14] Utomo, W. C., Wulanningrum, R., & Farida, I. N. (2023). Pengaruh Pemilihan Banyak Data dan Time Frame dalam Finance Forecasting dengan Linear Regression. *Joutica: Journal of Informatic Unisla*, 8(2), 18-23.

- 
- [15] Utomo, W. C. (2023). Prediksi Pergerakan Saham BBRI ditengah Issue Ancaman Resesi 2023 dengan Pendekatan Machine Learning. *Jurnal Teknologi dan Manajemen Informatika*, 9(1), 20-27.