

DESAIN SISTEM MONITORING BUILDING AUTOMATION BERBASIS IoT

Kukuh Prio Pambudi, Muhammad Dimas Arofah, A. Damar Aji, Muchlishah

¹²³Jurusan Teknik Elektro, Program Studi Teknik Otomasi Listrik Industri, Politeknik Negeri Jakarta
Jl. Prof. DR. G.A. Siwabessy, Kampus UI, Depok 16425, Telp/Fax (021) 7270036

ABSTRACT

IoT-Based Monitoring Building System Design is made to view and monitor sensors that work in the building so that it can make it easier to see the system of a building in one place. This system uses several sensors such as Flame sensor, DHT 11, MQ 135 and SW-420. Then for the microcontroller used is Arduino Mega 2560 then to send data so that it goes online using NodeMCU ESP8266. After that, the platform used to monitor the system uses the GOIOT platform. The way this system works is by detecting the sensor, so in monitoring it can be immediately known which sensor detects and is where the sensor is located. There are several tests such as the test between Arduino Mega 2560 and NodeMCU which has a speed of 1 second. ESP8266, further testing from NodeMCU ESP8266 to GOIOT also has a data transmission speed of 1 second. Then for sending data from the sensor to GOIOT, there are many time differences in each sensor, such as the fastest data from the sensor to GOIOT is the Vibration 2 or SW-420 Sensor which is 2.8 seconds while the longest is the DHT11-2 sensor with a time of 32.35 seconds.

Keywords: *BAS, Arduino Mega 2560 ;NodeMCU ESP8266, GOIOT*

ABSTRAK

Desain Sistem Monitoring Building Automation Berbasis IoT dibuat untuk melihat dan memantau sensor yang bekerja di dalam gedung sehingga dapat memudahkan dalam melihat sistem suatu gedung dalam 1 tempat. Pada sistem ini menggunakan beberapa sensor seperti *Flame sensor, DHT 11, MQ 135 dan SW-420*. Kemudian untuk *microcontroller* yang digunakan adalah Arduino Mega 2560 kemudian untuk mengirim data sehingga *online* menggunakan NodeMCU ESP8266. Setelah itu platform yang digunakan untuk *me-monitoring* sistem tersebut menggunakan platform GOIOT. Cara kerja sistem ini adalah dengan mendeteksi sensor maka dalam sistem *monitoring* dapat langsung mengetahui sensor yang mendeteksi dan berada dimana sensor tersebut berada. Terdapat beberapa pengujian seperti pengujian antara Arduino Mega 2560 dengan NodeMCU memiliki kecepatan sebesar 1 detik. ESP8266, selanjutnya pengujian dari NodeMCU ESP8266 ke GOIOT juga memiliki kecepatan pengiriman data sebesar 1 detik. Kemudian untuk Pengiriman data dari sensor ke GOIOT memiliki banyak perbedaan waktu di masing masing sensor, seperti data yang tercepat dari sensor keGOIOT adalah Sensor *Vibration 2* atau SW-420 yang waktunya adalah 2,8 detik sedangkan yang terlama dalah sensor DHT11-2 dengan waktu 32,35 detik.

Kata kunci: *BAS, Arduino Mega 2560, NodeMCU ESP8266, GOIOT*

1. PENDAHULUAN

Keamanan dan keselamatan manusia merupakan faktor utama yang harus dipertimbangkan dalam mendirikan sebuah gedung, karena kualitas dari sebuah gedung tidak hanya dinilai dari desain atau strukturnya saja. Banyak sekali permasalahan timbul sebagai akibat buruknya kualitas keamanan dan keselamatan pada suatu gedung.

Berdasarkan permasalahan tersebut dibutuhkan penerapan sistem otomasi pada gedung (*Building Automation System*). *Building Automation System*

adalah sebuah sistem jaringan komunikasi yang mengatur bermacam macam layanan pada gedung terdapat *monitoring*, kontrol penerangan, atau *temperature* yang bertujuan untuk penghematan energi didalan gedung [1].

Khususnya untuk sistem keamanan dan keselamatan pengguna gedung. Sehingga timbullah ide untuk membuat sebuah sistem yang berjudul, " *Desain Sistem Monitoring Building Automation Berbasis IoT* ". Di mana dalam sistem tersebut pengguna dapat mengetahui kondisi keamanan dan keselamatan sistem dalam gedung secara *real time*.

Desain Sistem Monitoring Building Automation Berbasis IoT

Adapun kedua sistem tersebut dapat dimonitor dari jarak jauh karena terintegrasi *Internet of Things*. Sehingga dengan adanya sistem tersebut dapat meminimalisir atau bahkan meniadakan risiko bahaya di dalam gedung.

Didalam BAS terdapat *input dan output* yang bekerja untuk membantu BAS ini bekerja didalam gedung. Pada sistem ini terdapat beberapa sensor yang digunakan seperti *Flame sensor*, sensor DHT 11, sensor M1 135 dan SW 420. *Flame sensor* merupakan sensor yang dapat mendeteksi api kemudian mengubahnya menjadi sinyal *analog*. Sensor api ini berbeda dengan sensor panas. Kalau sensor panas parameter yang diukur adalah suhunya, sedangkan sensor api ini yang dideteksi adalah nyala apinya. Modul sensor api ini memiliki 3 kaki/*pinout* yang terdiri dari Vcc 5V, Gnd, AO. Ada juga modul sensor api dengan 4 kaki, yakni dengan tambahan kaki atau *pin* untuk *Digital Output (DO)* [2].

DHT 11 adalah Sensor Suhu–Kelembaban dilambangkan dengan DHT 11, berfungsi mengukur suhu dan kelembaban di model khas individu. Suhu (T) dan Kelembaban (H), Fitur sensor (T), dan (H) sensor kompleks dengan *output* sinyal *digital* yang dikalibrasi. Modul tersebut memastikan keandalan tinggi dan jangka relatif yang sangat baik karena akuisisi sinyal *digital* eksklusif berkaitan dengan teknologi penginderaan. Sensor ini merupakan sensor resistif menggunakan suhu NTC yang terhubung kepada mikrokontroler. DHT sensor mengukur baik (T) dan (H) yang menyerahkan pembacaan ke Arduino dan mengirim ke NodeMCU untuk mengirim data ke *server IoT* [3].

MQ 135 adalah untuk mengontrol dan mendeteksi gas atau kualitas udara pada gedung. Gas gas yang di deteksi yaitu mendeteksi NH₃, Nikotin, Benzena, Asap, dan CO₂. Modul sensor MQ 135 hadir dengan *pin digital* yang memungkinkan sensor ini bekerja bahkan tanpa mikrokontroler dan bermanfaat untuk mendeteksi spesifikasi gas. Gas dalam PPM dihitung menggunakan *analog pin*. *Pin analog* ditenagai oleh TTL dan bekerja pada 5V, dan karenanya dapat digunakan dengan sebagian besar mikrokontroler modern [4].

SW 420 adalah Sensor yang mengukur getaran pada benda kemudian data tersebut akan dirubah untuk percobaan ataupun digunakan untuk sebuah sistem keamanan sehingga dapat mengantisipasi kemungkinan mara bahaya yang dapat terjadi. Jenis sensor yang mendeteksi getaran dan mengukur percepatan sebuah benda adalah *accelerometer* [5].

2. METODE PENELITIAN

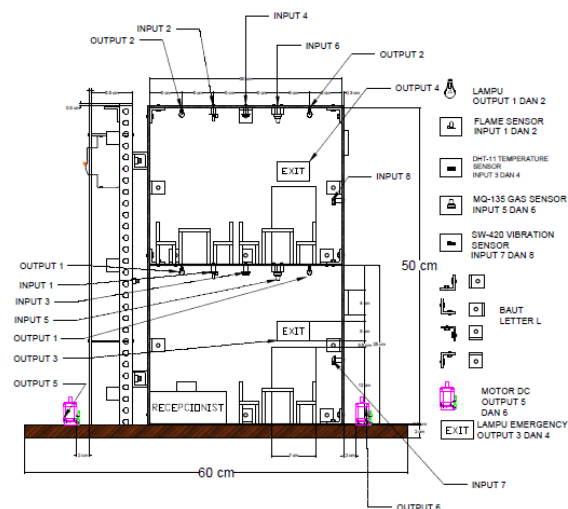
Metode penelitian yang digunakan terdiri dari beberapa tahap sebagai berikut.

1. Perancangan sistem *monitoring* pada BAS.

2. Pemrograman pada Arduino Mega 2560 dan NodeMCU ESP8266.
3. Pembuatan tampilan platform GOIOT
4. Mengatur sistem GOIOT supaya terhubung dan sesuai dengan mikrokontroler.
5. Pengujian sistem *monitoring* dimulai dari mikrokontroler, sensor, dan data *logger*.
6. Analisa kinerja sistem *monitoring*.

Sistem *monitoring* dapat dilihat pada Gambar 1 tampak depan dari alat ini. Sistem ini dibuat sebagai *safety system* dari sebuah gedung berbasis IoT. Pada sistem ini terdapat sistem *fire fighting*, untuk menjalankan sistem *fire fighting* menggunakan sensor seperti *Flame sensor* untuk mendeteksi adanya api di dalam ruangan, sensor MQ 135 untuk mendeteksi kualitas udara dalam ruangan, dan sensor DHT 11 untuk mengukur suhu dan kelembapan dalam ruangan.

Selain itu, terdapat sistem *secure* di dalam gedung dengan menggunakan Sensor SW 420. Sensor ini berfungsi untuk mendeteksi getaran yang terjadi pada kaca, di saat kaca bergetar akibat adanya indikasi tindakan pencurian maka sensor SW 420 akan mendeteksi getaran pada kaca lalu mengirimkan *output* sinyal menuju Arduino Mega 2560 untuk mengaktifkan *buzzer* sebagai peringatan dan mengirim juga menuju NodeMCU ESP8266 untuk dikirim menuju *server GOIOT* untuk diproses dan dikirim *notifikasi* menuju *email*.



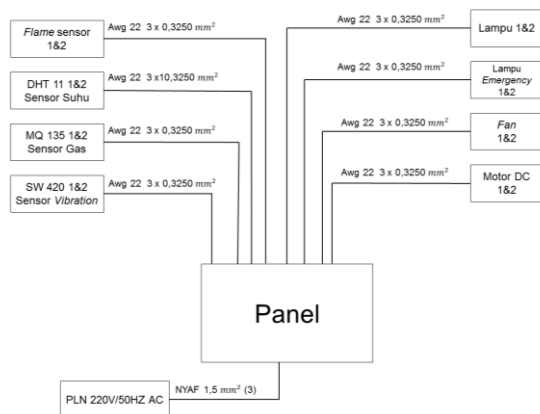
Gambar 1. Tampak Depan Bangunan

Kemudian, sistem ini dibuat dapat *dimonitoring* melalui GOIOT secara *Real-time* yang dikirim oleh NodeMCU ESP8266 melalui *protocol* komunikasi MQTT. GOIOT juga dapat memberikan pemberitahuan kepada pengguna dari jarak jauh dengan menggunakan *email*. Jadi pada sistem GOIOT menyalakan *Event Rules* untuk

Desain Sistem Monitoring Building Automation Berbasis IoT

mengirimkan peringatan atau pesan lainnya melalui *email*.

Terdapat 2 *email* yang dikirimkan dari GOIOT, yang pertama dikirimkan untuk pemilik gedung untuk lebih mudah mengetahui jika terjadi sesuatu yang tidak diinginkan. Untuk yang kedua GOIOT mengirimkan *email* ke pemadam kebakaran terdekat dan didalamnya terdapat alamat gedung sehingga mempermudah mengetahui posisi gedung berada.

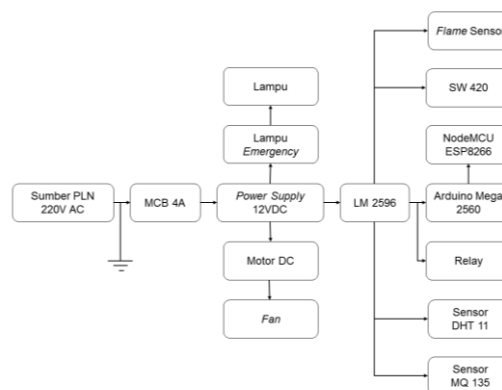


Gambar 2. Diagram Blok Elektrikal

Pada Gambar 2 dapat dilihat Diagram Blok Elektrikal di mana terdapat komponen *input* yang terdiri dari *Flame* sensor yang bekerja sebagai pendeteksi sinar inframerah yang dikeluarkan oleh api. Sensor kedua adalah sensor DHT 11, sensor ini mendeteksi *temperature* dan *humidity* di sekitar. Sensor ketiga adalah sensor MQ 135, sensor ini mendeteksi kualitas udara yang berada di sekitar. Kemudian sensor terakhir adalah sensor SW 420 atau sensor *vibration* di mana sensor ini mendeteksi suatu getaran.

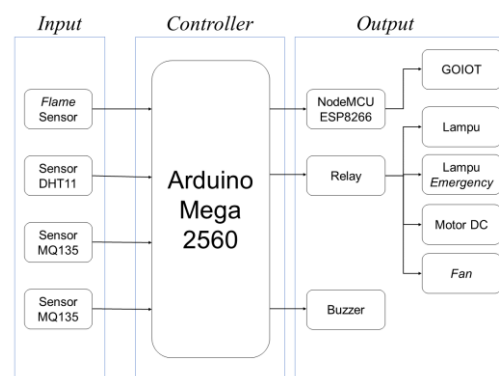
Setelah sensor menerima titik *point* tertentu sesuai dengan spesifikasi dan program maka kemudian sensor akan mengirim data ke Arduino Mega 2560 yang berada di dalam panel, kemudian setelah Arduino Mega 2560 menerima data yang dikirim oleh sensor kemudian sensor akan diproses apakah sudah sesuai dengan program. Jika sudah maka Arduino Mega 2560 akan mengirim data menuju NodeMCU ESP8266 dan *relay*.

Setelah NodeMCU ESP 8266 menerima data dari Arduino Mega 2560 maka NodeMCU ESP8266 akan mengirim data menuju GOIOT, sehingga mudah dilihat dan *dimonitoring* bagaimana sistem ini berjalan, dan data yang ditampilkan di GOIOT adalah mulai dari sensor hingga *actuator*. Lalu *relay* yang juga menerima *input* dari Arduino Mega 2560 maka akan menyalakan beban seperti lampu, lampu *emergency*, *fan*, dan motor DC. Untuk *buzzer* adalah beban yang langsung dinyalakan dari Arduino Mega 2560 tidak melalui *relay* terlebih dahulu.



Gambar 3. Diagram Blok Power

Gambar 3. diagram blok adalah diagram blok power dimulai dari sumber PLN AC 220V menuju MCB 4A, kemudian setelah MCB 4A menuju *power supply* untuk diubah dari yang sebelumnya AC menjadi DC. Kemudian setelah diubah menjadi DC maka akan memberikan *supply* pada beban seperti lampu, lampu *emergency*, motor DC, *fan*, dan LM 2596. Pada LM 2596, tegangan diturunkan lagi dari yang sebelumnya 12VDC menjadi 5VDC untuk memberikan sumber pada Arduino Mega 2560, *Relay*, dan sensor seperti *Flame* sensor, SW-420, DHT 11, MQ 135. Kemudian untuk NodeMCU mendapat sumber melalui Arduino Mega 2560.



Gambar 4. Diagram Blok Monitoring

Pada Gambar 4 dapat dilihat gambar proses Desain Sistem *Monitoring Building Automation* Berbasis IoT dimulai dari sensor yang memberikan data dari kondisi yang sudah diprogram lalu mengirim data ke Arduino Mega 2560 untuk diproses kemudian dikirim kembali menuju beban yang sesuai, proses ini dapat *monitoring* pada GOIOT yang dikirim oleh NodeMCU ESP8266 menuju GOIOT.

3. HASIL DAN PEMBAHASAN

A. Pengujian pengiriman data dari Arduino Mega 2560 menuju NodeMCU ESP8266.

Aplikasi Arduino.IDE adalah software yang digunakan untuk memprogram mikrokontroller Arduino Mega 2560 dan NodeMCU ESP8266 untuk bekerja sesuai dengan deskripsi yang sudah ditentukan. Dalam software terdapat serial monitoring yang didalamnya dapat dilihat hasil dari program tersebut.

Selisih waktu mulai dari pengirim data hingga penerima data adalah 1s, ini karena dalam program menggunakan delay 1s. Dalam program ini dibutuhkan delay karena jika tidak menggunakan delay pada program maka akan menyebabkan Arduino cukup kesulitan dalam mengirim data karena data yang dibaca dan dikirim cukup banyak. Untuk sistem pengirimannya antara Arduino Mega 2560 dan NodeMCU ESP8266 sudah cukup baik karena dalam 10 percobaan dan waktu yang berbeda tingkat keberhasilannya adalah 100% tanpa ada kegagalan.

Untuk sistem ini sudah termasuk real-time supaya dalam monitoring dan sensor tidak jauh dalam pembacaannya. Menurut [6] bahwa suatu sistem dikatakan real-time jika sebuah sistem tidak hanya mementingkan perintah atau program yang sudah diatur tetapi juga menyelesaikan tugas tersebut dengan waktu yang sudah ditentukan. Sedangkan kalau suatu sistem yang tidak real-time adalah suatu sistem yang menyelesaikan pekerjaan tanpa menggunakan deadline atau batas waktu yang sudah diatur sebelumnya. Oleh karena itu sistem inipun sudah termasuk kedalam sistem real-time karena proses yang dilakukan oleh Arduino dan NodeMCU dapat bekerja dengan selesai sesuai dengan waktu yang sudah ditentukan.

Pada Flame sensor lantai 1 dan 2, yang dibaca oleh Arduino Mega 2560 adalah langsung 1 atau on karena dalam sensornya berada terbalik namun dalam monitoring dan alarm sudah diputar juga sehingga bisa menyesuaikan dengan Flame sensor tersebut yang jika dia mendeteksi api maka keluarannya 0 dan buzzer akan menyala.

Untuk pembacaan sensor Digital lain seperti sensor SW 420 atau vibration adalah 0 untuk off dan 1 untuk on ini sudah cukup baik dan berfungsi normal. Kemudian untuk sensor DHT 11 dan MQ 135 data yang dihasilkan adalah analog sehingga data yang dikeluarkan adalah berupa angka sesuai dengan cara kerjanya masing masing.

B. Pengujian menghubungkan dari NodeMCU ESP8266 ke Wi-Fi dan GOIOT.

Pada Tabel 1 terlihat bahwa pengujian dilakukan dimulai dari NodeMCU ESP8266 ke Wi-Fi, dan data yang dibutuhkan untuk mengkoneksikannya adalah 5s, ini tergantung dari

sinyal Wi-Fi yang digunakan pada komputer. Namun pada pengujian kali ini terlihat stabil dengan rata-rata mengkoneksikannya adalah 5s.

Tabel 1 Pengujian NodeMCU ESP8266 ke Wi-Fi dan GOIOT

NO	Pengujian	Dibutuhkan waktu
1	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
2	Connect NodeMCU ESP8266 ke Wi-Fi	5s
	Connect NodeMCU ESP8266 ke GOIOT	1s
3	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
4	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
5	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
6	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
7	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
8	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
9	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s
10	Connect NodeMCU ESP8266 ke Wi-Fi	5 s
	Connect NodeMCU ESP8266 ke GOIOT	1s

Kemudian untuk pengujian yang kedua adalah menghubungkan dari NodeMCU ESP8266 dengan platform GOIOT. Sistem ini juga bisa dilihat pada serial monitor setelah NodeMCU ESP8266 telah terkoneksi oleh Wi-Fi. Ini juga tergantung pada internet yang digunakan dan untuk pengujian kali ini kecepatan koneksi antara NodeMCU ESP8266 setelah terkoneksi pada Wi-Fi adalah 1s.

C. Pengujian Flame 1 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam respond time, untuk waktu respon tercepat adalah 3,12 detik dan yang terlama 7 detik. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan delay. Kemudian akibat disaat pengujian api dideteksi oleh Flame sensor 1 terdapat angin yang menyebabkan api tidak stabil yang mengakibatkan sensor tidak dapat membaca api yang terus bergerak dan berada di luar jangkauan Flame sensor 1 tersebut seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Waktu Pembacaan Flame 1 pada GOIOT

No	Flame 1	Waktu Pembacaan GOIOT
1	ON	5.3 s
2	ON	7 s
3	ON	4.76 s
4	ON	6 s
5	ON	5.59 s

Desain Sistem Monitoring Building Automation Berbasis IoT

6	ON	6.17 s
7	ON	4.49 s
8	ON	5.11 s
9	ON	3.12 s
10	ON	6.64 s

D. Pengujian Flame 2 menuju GOIOT

Dari pengujian tersebut yang ditunjukkan pada Tabel 4 masing masing memiliki perbedaan dalam respond time, untuk waktu respon tercepat adalah 3,7 detik dan yang terlama 6,22 detik, seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Waktu Pembacaan Flame sensor 2 pada GOIOT

No	Flame 2	Waktu Pembacaan GOIOT
1	ON	5.97
2	ON	6.22
3	ON	4.22
4	ON	3.88
5	ON	6.21
6	ON	3.07
7	ON	3.95
8	ON	4.07
9	ON	4.8
10	ON	5.14

Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*. Kemudian akibat disaat pengujian api dideteksi oleh Flame sensor 2 terdapat angin yang menyebabkan api tidak stabil yang mengakibatkan sensor tidak dapat membaca api yang terus bergerak dan berada di luar jangkauan Flame sensor 2 tersebut.

E. Pengujian Sensor Vibration 1 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam respon time, untuk waktu respon tercepat adalah 3,55 detik dan yang terlama 5,62 detik. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*. Kemudian disaat pengujian getaran dideteksi oleh sensor Vibration 2 getaran yang diberikan kurang besar sehingga sensor tidak membaca dengan jelas, seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Waktu Pembacaan Vibration 1 pada GOIOT

No	Vibration 1	Waktu Pembacaan GOIOT
1	ON	4.45
2	ON	4.35
3	ON	4.31
4	ON	3.77
5	ON	5.17
6	ON	5.62
7	ON	5.49
8	ON	0
9	ON	3.55
10	ON	3.7

F. Pengujian Sensor Vibration 2 menuju GOIOT

Dari pengujian yang ditunjukkan pada Tabel 5, masing masing memiliki perbedaan dalam *respond time*, untuk waktu respon tercepat adalah 2,8 detik dan yang terlama 4,93 detik. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*. Kemudian disaat pengujian getaran dideteksi oleh sensor Vibration 2 getaran yang diberikan kurang besar sehingga sensor tidak membaca dengan jelas.

Tabel 5. Waktu Pembacaan Vibration 2 pada GOIOT

No	Vibration 2	Waktu Pembacaan GOIOT
1	ON	4.16
2	ON	3.72
3	ON	4.93
4	ON	3.8
5	ON	2.8
6	ON	4.16
7	ON	4.56
8	ON	3.45
9	ON	3.84
10	ON	3.41

G. Pengujian Sensor MQ 135-1 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam *respond time*, untuk waktu respon tercepat adalah 3,59 detik dan yang terlama 8,27 detik. Untuk pengujian kali ini terdapat

pebedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*. Terdapat juga masalah dari luar seperti angin yang menyebabkan asap atau gas yang akan dibaca oleh sensor MQ 135 terganggu oleh angin yang membuat asap tersebut keluar dan menjauh dari sensor sehingga sensor sulit membaca, seperti yang ditunjukkan pada Tabel 6

Tabel 6. Hasil Waktu Pembacaan Sensor MQ 135-1 pada GOIOT

No	MQ 135-1	Waktu Pembacaan GOIOT
1	ON	7.83
2	ON	6.75
3	ON	6.57
4	ON	5.59
5	ON	6.08
6	ON	8.04
7	ON	8.27
8	ON	3.59
9	ON	4.54
10	ON	8.04

H. Pengujian Sensor MQ 135-2 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam *respond time*, untuk waktu respon tercepat adalah 3,7 detik dan yang terlama 8,05 detik, seperti yang ditunjukkan pada Tabel 7. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*. Terdapat juga masalah dari luar seperti angin yang menyebabkan asap atau gas yang akan dibaca oleh sensor MQ 135 terganggu oleh angin yang membuat asap tersebut keluar dan menjauh dari sensor sehingga sensor sulit membaca.

Tabel 7. Hasil Waktu Pembacaan Sensor MQ 135-2 pada GOIOT

No	MQ 135-2	Waktu Pembacaan GOIOT
1	ON	7.86
2	ON	3.7
3	ON	2.11
4	ON	7.97
5	ON	6.66
6	ON	7.47
7	ON	8.3
8	ON	3.37
9	ON	8.05

10	ON	6.48
----	----	------

I. Pengujian Sensor DHT 11-1 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam *respond time*, untuk waktu respon tercepat adalah 3,2 detik dan yang terlama 17,54 detik. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*, seperti yang ditunjukkan pada Tabel 8.

Tabel 8. Hasil Waktu Pembacaan Sensor DHT 11-1 pada GOIOT

No	DHT11-1	Waktu Pembacaan GOIOT
1	ON	19.25
2	ON	15.07
3	ON	17.54
4	ON	13.5
5	ON	17.15
6	ON	14.75
7	ON	16.63
8	ON	13.6
9	ON	17.7
10	ON	13.2

J. Pengujian Sensor DHT 11-2 menuju GOIOT

Dari pengujian tersebut masing masing memiliki perbedaan dalam *respond time*, untuk waktu respon tercepat adalah 12,95 detik dan yang terlama 32,35 detik. Untuk pengujian kali ini terdapat perbedaan pengiriman data dan terdapat juga pengiriman data ini diakibatkan oleh sinyal yang diterima NodeMCU ESP8266 dan akan dikirim menuju GOIOT sehingga data yang diterima GOIOT mengalami perbedaan waktu dan *delay*, seperti yang ditunjukkan pada Tabel 9.

Tabel 9. Hasil Waktu Pembacaan Sensor DHT 11-2 pada GOIOT

NO	DHT11-2	Waktu Pembacaan GOIOT
1	ON	32.35
2	ON	19.49
3	ON	12.95
4	ON	15.33
5	ON	16.77
6	ON	18.57

NO	DHT11-2	Waktu Pembacaan GOIOT
7	ON	13.72
8	ON	18.45
9	ON	15.44
10	ON	16.72

7. SIMPULAN DAN SARAN

Kesimpulan yang didapatkan pada penelitian ini adalah:

1. Faktor jaringan menentukan pengiriman data dari sensor ke GOIOT.
2. Monitoring sistem keamanan dan keselamatan dapat bekerja dengan menggunakan GOIOT dan menggunakan mikrokontroler Arduino Mega 2560 untuk membaca sensor dan mengirimnya dengan menggunakan NodeMCU ESP8266.
3. Dalam GOIOT terdapat eror atau bug ketika menggunakan tag lebih dari 20.
4. Dalam pengambilan data di GOIOT jam tidak sesuai dengan data diinginkan.
5. Sistem *Monitoring* dari sistem keamanan dan keselamatan bekerja secara *real-time* dengan kecepatan respon dari sensor hingga monitoring pada GOIOT hanya membutuhkan 1 detik.
6. Pada tampilan GOIOT terdapat sensor pada setiap lantai dan saat mendeteksi maka Arduino Mega 2566 membaca dan mengirim ke NodeMCU ESP8266, kemudian NodeMCU ESP8266 mengirim kembali data yang diterima dari Arduino MEGA 2560 ke GOIOT.
7. Menggunakan jaringan yang lebih stabil untuk pengiriman data lebih cepat.
8. Menggunakan tag pada GOIOT tidak sampai 20 untuk menghindari *bug*.
Pada saat mengambil data dari GOIOT lebih baik *mendownload* setiap 1 jam sehingga lebih mudah dalam pembacaan

DAFTAR PUSTAKA

- [1] N. Fadilla, "BUILDING AUTOMATION SYSTEM BERBASIS MIKROKONTROLER UNTUK MONITORING DAN KONTROL ENERGI," *Skripsi Tek. Elektro*, 2015.
- [2] S. Kasus, D. Sepanjang, J. Slamet, and R. Surakarta, "Laporan penelitian," vol. 22, no. 2, pp. 184–206, 2009.
- [3] D. A. Aziz, "Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module," no. July, 2018.
- [4] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of Smart Healthcare Monitoring System in IoT Environment," *SN Comput. Sci.*, vol. 1, no. 3, pp. 1–11, 2020.

- [5] L. D. Mustafa, P. Studi, J. Telekomunikasi, J. T. Elektro, P. N. Malang, and A. L. Belakang, "IMPLEMENTASI WIRELESS SENSOR NETWORK PADA SIMULASI PERINGATAN GEMPA BUMI MENGGUNAKAN," pp. 38–44, 2020.
- [6] A. A. A. Hidayat and M. Uliyah, "Rancang Bangun Prototype Perangkat Lunak Gaya Berjalan Atlet Secara Real Time Sebagai Perangkat Dukung Peningkatan Performa Atlet Jalan Cepat," no. 0718126802, 2013.