

SISTEM PEMANTAUAN DAN PENGENDALI SUHU RUANGAN SERVER BERBASIS IOT DENGAN NOTIFIKASI ALARM DAN TELEGRAM

Raihan Naufal Ichsan

Fakultas Teknik, Program Studi Teknik Elektro

Universitas Muria Kudus

Email: raihannaufalichsan@gmail.com

ABSTRAK

Ruang server adalah ruangan yang biasa digunakan untuk menyimpan peralatan jaringan seperti router, server komputer pribadi, dan barang-barang penting lainnya. Ruangan ini membutuhkan perhatian khusus dan pemeliharaan yang baik, serta harus dipantau secara efektif untuk melindungi peralatan yang berada di sana. Dampak dari tidak memberikan perhatian khusus pada ruang server adalah kenaikan suhu aktivitas jaringan di atas normal dapat menyebabkan kinerja perangkat keras yang kurang optimal karena perangkat terlalu panas. Pada penelitian ini akan dirancang suatu sistem kontrol dan monitoring ruang server berbasis Internet of Things dengan alarm dan notifikasi telegram untuk mengukur kondisi suhu, kelembaban, getaran dan energi listrik pada ruang server. Metodologi yang digunakan dalam penelitian ini adalah sistem Research and Development. Sistem dirancang untuk dapat mencapai hasil yaitu memantau ruang server, mengontrol suhu komponen di ruang server, dan sistem dapat mengeluarkan peringatan ketika kondisi ruangan tidak normal. Sistem dapat mengirimkan data sensor dari slave ke master melalui Modbus RTU yang terhubung ke RS485. Sistem kemudian dapat mengirimkan data pada LCD 16x2 ke perangkat dan mengeluarkan alarm seperti alarm rendah, tinggi, dan kritis serta alarm ketika kondisi ruangan yang tidak normal terdeteksi. Saat sistem mendeteksi kondisi ruangan yang tidak normal, kipas mendinginkan komponen yang panas di ruang server.

Kata kunci: Ruang Server, Suhu, Kelembaban, Getaran, Arus listrik, Modbus Rtu

ABSTRACT

Server room is a room that is usually used to store network equipment such as routers, personal computer servers, and other important items. These spaces require special attention and proper maintenance, and must be monitored effectively to protect the equipment located there. The impact of not paying special attention to the server room is that the temperature rise of network activity above normal can cause less than optimal hardware performance due to device overheating. In this research, a server room control and monitoring system based on the Internet of Things will be designed with alarms and telegram notifications to measure temperature, humidity, vibration and electrical energy conditions in the server room. The methodology used in this study is the Research and Development system. The system is designed to achieve results, namely monitoring the server room, controlling the temperature of components in the server room, and the system can issue warnings when room conditions are abnormal. The system can send sensor data from slave to master via Modbus RTU connected to RS485. The system can then send data on the 16x2 LCD to the device and issue alarms such as low, high and critical alarms as well as alarms when abnormal room conditions are detected. When the system detects abnormal room conditions, fans cool hot components in the server room.

Keywords: Server Room, Temperature, Humidity, Vibration, Electric current, Modbus Rtu.

1. PENDAHULUAN

Ruang *server* adalah ruangan yang biasa digunakan buat menyimpan perangkat jaringan seperti *outer*, personal komputer *server* dan perangkat penting lainnya. Ruang *server* ini membutuhkan perhatian khusus dan perawatan yang baik serta harus diawasi secara aktual. Ruangan ini memerlukan standar keamanan buat melindungi perangkat didalamnya seperti kondisi suhu udara, kelembapan, kebakaran, dan bencana lainnya [1]. Adapun dampak dari tidak adanya perhatian spesifik pada ruang *server* yaitu adanya kenaikan suhu diatas normal pada aktivitas jaringan personal komputer yang bisa menyebabkan kurang optimalnya kinerja perangkat keras ataupun perangkat lainnya menjadi kerjanya kurang maksimal dikarenakan perangkat terlalu panas (*overheat*) [2],

Ruang *server* harus memiliki suhu dan kelembapan pada kondisi stabil dan sesuai menggunakan standarisasi yang telah dipengaruhi. standar suhu ruang *server* pada Indonesia memiliki standar 21-27°C (70-74°F) buat suhu serta 45%-70% buat kelembapan dimana rendahnya suhu bisa membuat kinerja melambat bahkan terhenti sedangkan tingginya suhu menyebabkan personal komputer serta jaringan terlalu panas sebagai akibatnya akhirnya mati [3].

Maka dari itu, ruang *server* wajib dipantau secara berkala, agar Jika suatu waktu terjadi sesuatu yang bisa membahayakan dapat terdeteksi dan ditindaklanjuti buat menghindari kerusakan pada *server*. Salah satu cara yang bisa dipergunakan buat mengawasi kondisi lingkungan pada ruang *server* agar tetap terjaga ialah menggunakan cara sistem monitoring.

Berdasarkan penelitian tersebut, pada penelitian ini akan dibangun suatu sistem pemantau dan pengendali suhu ruangan server berbasis IoT notifikasi alarm dan telegram untuk mengukur kondisi suhu, kelembapan, getaran, dan arus listrik pada rack perangkat keras diruang *server*. Sistem menggunakan 2 slave yang akan diletakan didalam ruang *server* dan 1 *master* yang berfungsi untuk menampilkan kondisi ruangan pada layar LCD. *Master* dan *Slave* terhubung melalui *module* RS485 pada pengiriman data menggunakan Modbus RTU. Sistem juga mengirimkan peringatan berupa *alarm* yang memiliki *level Minor, Major, dan critical* dan Notifikasi melalui telegram ketika mendeteksi kondisi ruangan tidak normal. Perancangan alat ini menggunakan Modbus RTU (*Remote Terminal Unit*) sebagai protokol komunikasi antara *master* dan *slave*. Nantinya, komunikasi dilakukan melalui telegram berupa kabel *one twisted-pair* sepanjang 4 meter antara *slave 1* dan *master* serta sepanjang 5 meter antara *slave 1* dan *slave 2*. Apabila sistem mendeteksi kondisi suhu pada rack perangkat keras tidak normal maka fan/kipas akan mendinginkan komponen perangkat keras yang panas terdapat di ruangan server. Keuntungan dari protokol komunikasi ini yaitu memiliki *throughput* data yang lebih baik dibandingkan dengan ASCII saat menggunakan nilai yang sama pada *baudrate* sehingga protokol ini cocok digunakan untuk sistem *monitoring*.

2. METODOLOGI PENELITIAN

Metode penelitian adalah sistem *Research And Development*. Metode penelitian ini adalah metode penelitian dimana dari produk yang sudah ada kemudian diteliti dan dipelajari, kemudian dikembangkan menjadi produk baru dan diuji keefektifannya. Penelitian ini akan membuat sebuah sistem pemantau dan pengendali suhu, kelembapan, getaran, dan arus listrik pada perangkat keras di ruang *server* dengan protokol komunikasi Modbus RTU memakai modul RS485 dalam merancang sistem ini.

2.1 Waktu dan Tempat

Proses perancangan dan pembuatan alat berlangsung pada bulan Juli 2023 sampai selesai di dua lokasi yakni dirumah dan lokasi penelitian sistem pemantauan dan pengendalian suhu ruangan server berbasis *internet of things* yaitu usaha Wi-Fi rumahan / UMKM skala kecil. Dari usaha yang bernama DF.NET yang berlokasi di daerah Samirejo, Gebog, Kudus. Perencanaan lokasi penelitian komponen pada ruangan server.

2.2 Tahapan Alur Penelitian

Pengembangan sistem ini ada beberapa langkah yang harus diselesaikan untuk memudahkan penelitian. Langkah yang harus diselesaikan dalam penelitian meliputi identifikasi masalah, perancangan *hardware* dan *software*, pembuatan alat dan pengujian alat dimana alat yang sudah dibuat kemudian diuji apabila dirasa belum bekerja dan berfungsi dengan baik maka akan dievaluasi dari tahapan sebelumnya. Kemudian apabila alat sudah bekerja dan berfungsi dengan baik maka berlanjut ke pengambilan data dan analisis data.



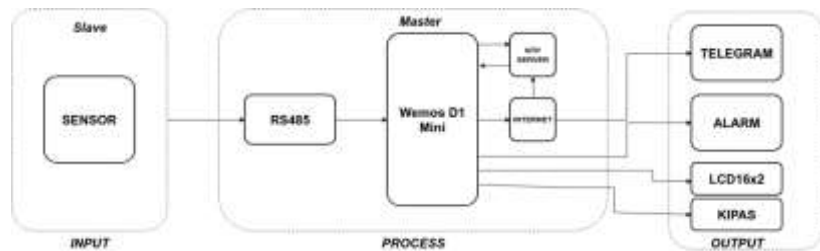
Gambar 1. Flowchart Penelitian

2.3 Perancangan Hardware

Perancangan *Hardware* ini guna dapat mengetahui komponen yang akan digunakan dalam pembuatan sistem pemantauan dan pengendali suhu diruangan server sehingga dalam pelaksanaanya lebih efektif, efisien, dan terstruktur dengan baik.

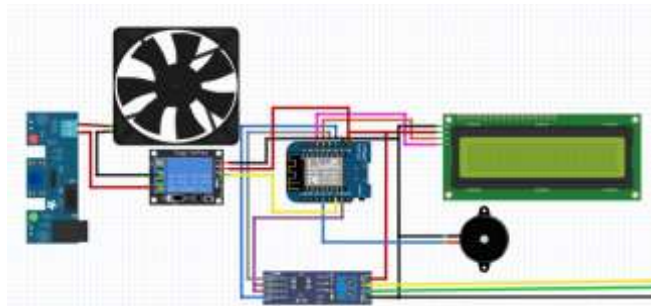
2.3.1 Diagram Blok Sistem

Pada blok diagram sistem, untuk slave dan untuk master menggunakan sumber listrik dari charger yang terhubung langsung ke terminal listrik. input dari masing-masing slave yaitu sensor DHT11 untuk slave 1 serta sensor SW-420 dan SCT-013-000 untuk slave 2 akan diproses didalam arduino nano yang akan dikirimkan ke master melalui modul MAX485 TTL to RS485 menggunakan Modbus RTU. Sedangkan unntuk output berada di master berupa buzzer 3-24V DC, Notifikasi telegram dan LCD 16x2 untuk menampilkan bebrapa informasi terkait kondisi suhu, kelembapan, getaran, dan arus listrik pada ruangan.



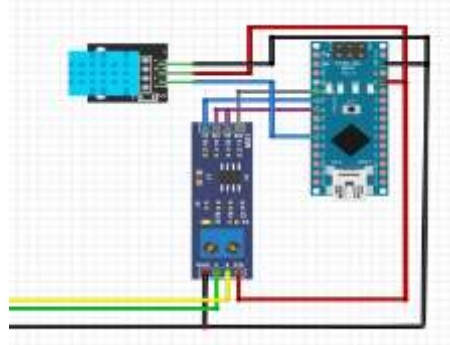
Gambar 2. Diagram Blok Sistem

Tujuan dari perancangan perangkat keras adalah untuk membuat sistem sesuai dengan tujuan. Dari blok diagram diatas, akan dijabarkan secara skematik antar pin pada komponen perangkat keras yang akan digunakan. Dalam perancangan untuk sistem yang akan dibuat dibagi menjadi *master*, *slave 1* yang terdapat sensor DHT11, dan *slave 2* yang terdapat sensor SCT-013-000 dan SW420. Pada *master* komponen perangkat keras yang digunakan yaitu Wemos D1 Mini sebagai mikrokontroller, MAX485 TTL to RS485, *Buzzer* 3-24V DC sebagai *alarm* dan LCD 16x2 untuk menampilkan data dari *slave*. *Master* berfungsi untuk menerima semua data dari *slave* yang dimana data tersebut akan diproses untuk mengetahui kondisi ruangan apakah normal atau tidak. Apabila kondisi ruangan tidak normal, maka *master* akan mengirimkan notifikasi ke telegram dan menjalankan kipas DC dan *alarm* dengan kategori yang telah ditentukan yaitu *minor*, *major*, dan *critical*. Selain itu, *master* juga akan mengirimkan data yang diterima *slave* sesuai jadwal yang ditentukan.



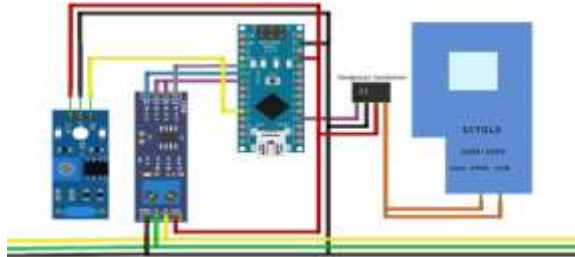
Gambar 2. Rangkaian Perangkat Keras Master

Slave 1 menggunakan sensor DHT11 untuk mengambil data kelembapan dan suhu yang terhubung ke Arduino Nano sebagai mikrokontroller yang nantinya data tersebut akan dikirim ke *master* melalui modul MAX485 TTL to RS485. Apabila hasil pembacaan sensor DHT11 melebihi standart yang telah disetting akan mengaktifkan kipas pendingin komponen ruang server yang panas.



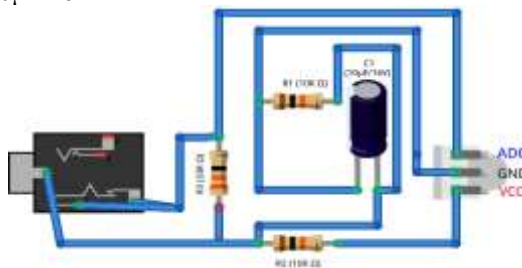
Gambar 3. Rangkain Perangkat Keras Slave 1

Pada *Slave 2* komponen perangkat keras yang digunakan yaitu Arduino Nano, sensor SW420 untuk mendeteksi getaran dan sensor SCT-013-000 100A/50mA untuk mendeteksi arus. Data pada *slave 2* akan dikirim ke *master* melalui modul MAX485 TTL to RS485.



Gambar 4. Rangkaian Perangkat Keras Slave2

Untuk menghubungkan sensor SCT-013-000 100A/50mA memerlukan rangkaian tambahan yang berfungsi sebagai amplifier. Komponen yang dibutuhkan untuk rangkaian tambahan ini adalah Stereo Jack 3.5mm Female 5 pin, 2 Resistor bernilai 10K Ω , 1 Resistor bernilai 33R Ω dan Capacitor bernilai 10 μ F/16V.

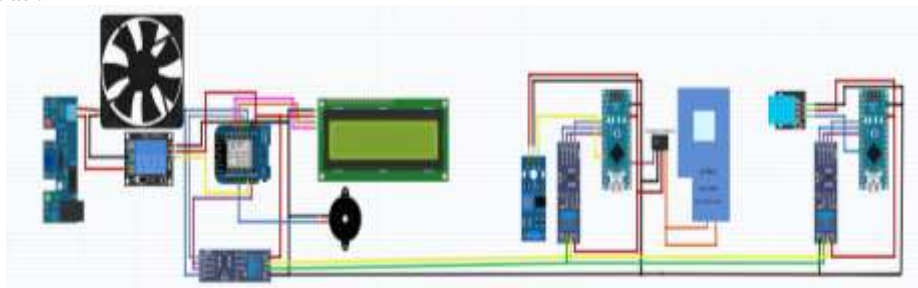


Gambar 2. 5 Rangkaian Tambahan Sensor SCT-013-000

Sistem saling terhubung melalui modul RS485 pada masing-masing mikrokontroler dan menggunakan kabel *one twisted pair* sebagai telegram penghubung antar *master-slave*. Kabel yang digunakan memiliki panjang 4 meter untuk *slave 1* ke *master* dan 5 meter untuk *slave 1* ke *slave 2*.

2.3.1. Rangkaian Skematik

Berikut rangkaian skematik alat yang menggambarkan wiring antar alat dan komponen menjadi kesatuan. Perancang pada rangkaian ini menggunakan *software fritzing*. Berikut rangkaian skematik alat :

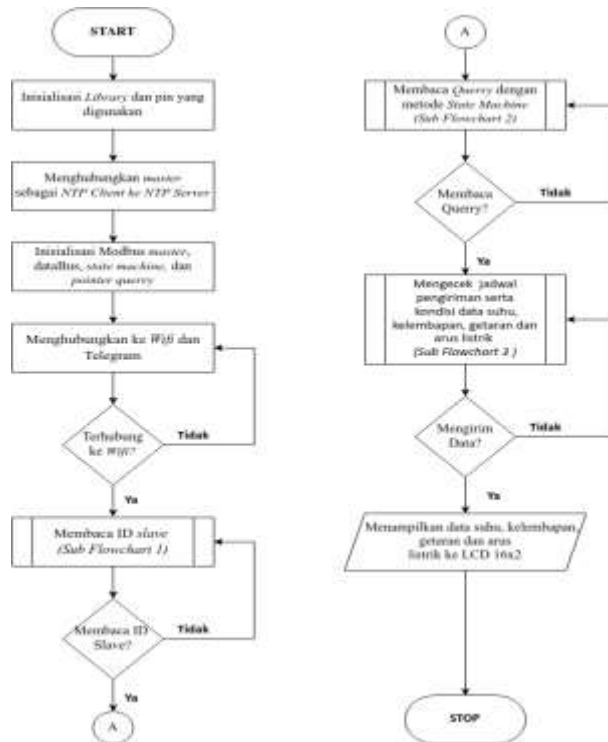


Gambar 6. Rangkaian Skematik

2.4. Perancangan Software

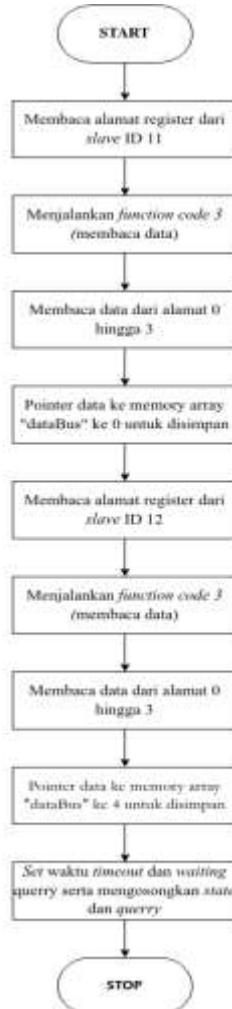
Saat mendesain perangkat lunak pada sistem, bagan alir menjelaskan bagaimana program dimulai pada master dan slave dan sistem bekerja dari awal. Sistem kemudian mendeteksi input berupa suhu, kelembaban, getaran dan arus dan mengirimkan data dari slave ke master, yang memproses data tersebut mengirimkan alarm notifikasi telegram dan notifikasi alarm dan menjalankan kipas DC jika kondisi ruangan tidak normal.

2.4.1. Flowchart Master

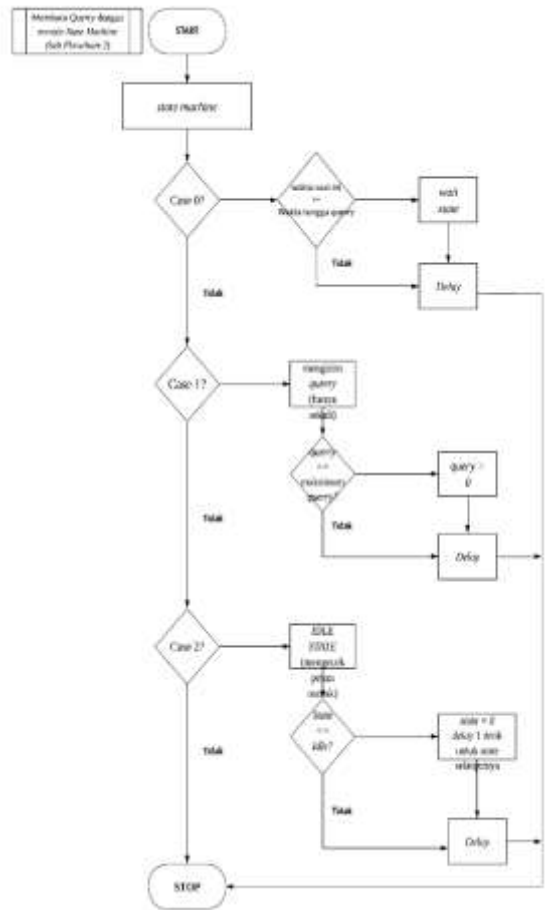


Gambar 7. Flowchart Utama Master

Membaca ID slave
(Sub Flowchart 1)



Gambar 9. Sub Flowchart 1 Master



Gambar 10. Sub Flowchart 2 Master



Gambar 11. Sub Flowchart 3 Master

Pada Gambar 8 merupakan gambaran umum atau *flowchart* utama dari *master*. Pertama sistem menginisialisasi *library* yang dibutuhkan dan pin yang digunakan. Kemudian, menghubungkan Wemos D1 Mini (*master*) sebagai *NTP Client* ke *NTP Server* untuk mendapatkan jam, hari dan tanggal secara *real-time*. Selanjutnya, *master* akan melakukan inisialisasi sebagai *master* dan membuat array untuk Data dari Modbus, inisialisasi *state machine* dan melakukan pointer query untuk data Modbus. Setelah melakukan inisialisasi yang dibutuhkan, *master* akan menghubungkan ke *wifi* yang ingin disambungkan dan ke telegram. Apabila sistem tidak terhubung ke *wifi*, maka sistem akan direset dan kembali menghubungkan ke *wifi* hingga berhasil terhubung. Kemudian, sistem akan menjalankan proses inisialisasi Modbus RTU, Modbus RTU *State* mengecek data berdasarkan kondisi yang telah ditentukan, mengirimkan data ke telegram sesuai jadwal dan menampilkan data ke LCD 16x2.

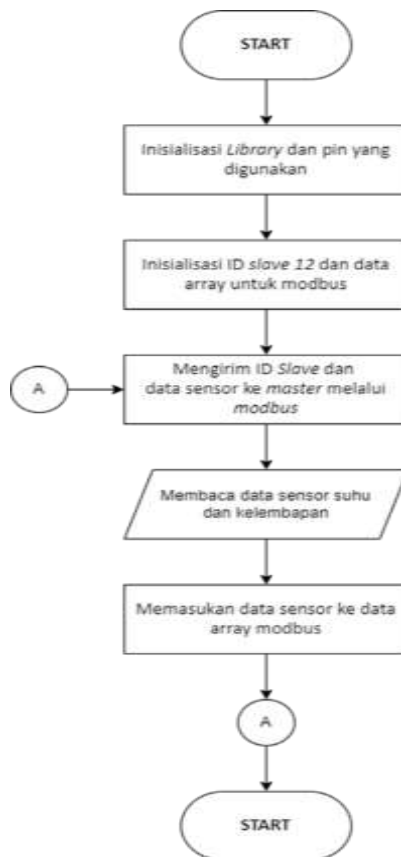
Pada Gambar 9 merupakan *flowchart* dari sub-proses pada *master* untuk proses Inisialisasi Modbus RTU. Sebelum memulai menerima data, sistem akan melakukan inisialisasi Modbus RTU terlebih dahulu. Pertama, sistem akan melakukan inisialisasi *slave* dengan cara membaca alamat register dari “*slave ID 11*” dan kemudian sistem akan memilih *function code 3* yaitu membaca data untuk ID tersebut. Setelah itu, sistem akan memilih alamat data yang akan di baca pada *slave ID 11* yaitu alamat 0 hingga 3. Kemudian, data yang dibaca akan di *pointer* ke ke *memory array*

“dataSensor” ke 0 untuk disimpan. Setelah membaca melakukan inisialisasi untuk “slave ID 11”, sistem akan melakukan proses seperti sebelumnya untuk “slave ID 12”. Namun, data yang dibaca akan di *pointer* ke *memory array* “dataSensor” ke 4 untuk menghindari data yang “slave ID 11” tertimpa. Setelah itu, *set waktu timeout* dan *waiting state* bernilai +1 detik dari waktu saat ini serta melakukan *set state=0* dan *query=0*.

Pada Gambar 10 merupakan *flowchart* dari sub-proses pada master untuk proses Modbus RTU State. Pada proses ini, sistem menjalankan *state machine* untuk memproses data yang masuk dari *slave* dengan melakukan pengecekan menggunakan *switch-loop* dengan 3 *case* dengan mengecek nilai “state”. Pada *case* ketika “state” bernilai 0 sistem akan mengecek apakah waktu saat ini lebih besar dari waktu tunggu *query* yang telah diatur sebelumnya. Apabila lebih besar, maka sistem mengubah nilai “state” menjadi 1 untuk menandakan telegram sedang digunakan dan kemudian melakukan *break* ke proses selanjutnya. Pada *case* ketika “state” bernilai 1 sistem akan mengirim *query* sebanyak satu kali ke *slave* dan mengubah nilai “state” menjadi 2 dan menambah 1 nilai “*query*”. Kemudian mengecek apakah nilai “*query*” lebih besar dari maksimum *query* yang telah ditentukan, apabila lebih besar maka *query* akan dikosongkan dan melakukan *break* ke proses selanjutnya. Pada *case* ketika “state” bernilai 2 sistem akan mengambil data dari *slave* dan kemudian mengecek apakah *state* dalam keadaan “*idle*”. Apabila *state* dalam keadaan *idle* maka sistem akan mengubah nilai “state” menjadi 0 dan memberikan delay 1 detik untuk *state* selanjutnya. Kemudian melakukan *break* ke proses selanjutnya.

Pada Gambar 11 merupakan *flowchart* dari sub-proses pada master untuk proses pengecekan jadwal dan kondisi. Pada proses ini sistem akan mengecek jadwal yang telah ditentukan untuk pengiriman ke telegram dan mengecek kondisi untuk data suhu, kelembapan, getaran dan arus listrik. Pertama, sistem akan melakukan update waktu, hari dan tanggal berdasarkan NTP Server kemudian mengecek apakah waktu sesuai dengan jadwal pengiriman. Apabila waktu sesuai dengan jadwal pengiriman, maka sistem akan mengirimkan data suhu, kelembapan, getaran dan arus listrik ke telegram. Kemudian, sistem akan melakukan pengecekan kondisi apabila suhu dibawah 21°C maka sistem akan menjalankan alarm dengan level major dan mengirim notifikasi ke telegram. Apabila tidak memenuhi kondisi tersebut sistem akan mengecek kondisi selanjutnya yaitu apabila suhu lebih tinggi dari 27°C maka sistem akan menjalankan alarm dengan level critical dan mengirimkan notifikasi ke telegram. Apabila masih belum memenuhi kondisi tersebut sistem akan mengecek kondisi selanjutnya yaitu apabila arus listrik bernilai 0 dan getaran bernilai lebih dari 0 maka sistem akan menjalankan alarm dengan level critical dan mengirimkan notifikasi ke telegram.

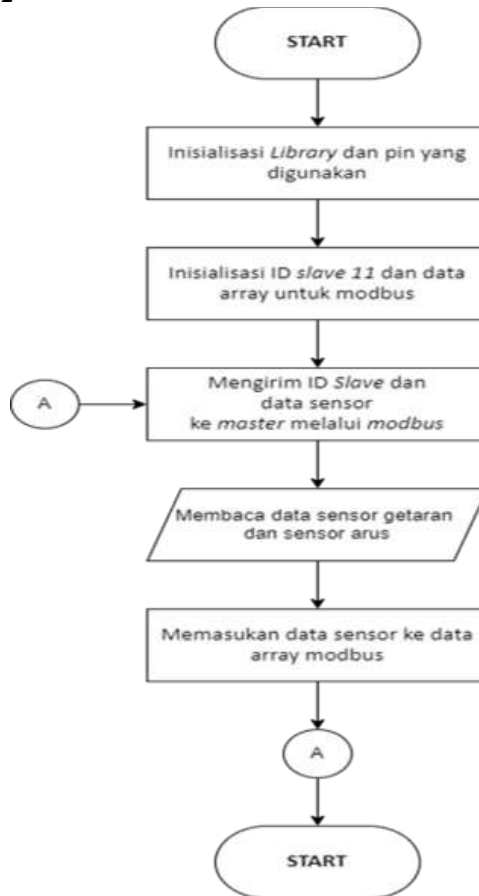
2.4.2. Flowchart Slave 1



Gambar 12. Flowchart Slave 1

Pada Gambar 12 merupakan *flowchart* dari *slave 1* mulai dari inisialisasi hingga pengiriman data ke *master*. Pertama, *slave* akan melakukan inisilasi *library* dan pin yang dibutuhkan terlebih dahulu untuk memulai sistem. Kemudian, sistem melakukan inisialisasi *slave* menggunakan ID 11 yang menandakan *slave 1* dan melakukan inisialisasi data array untuk Modbus bernama “dataregister” dengan format uint16_t. Setelah melakukan seluruh inisialisasi maka *slave* akan mengecek telegram komunikasi terlebih dahulu apakah sedang dipakai atau tidak, apabila tidak maka *slave* akan mengirimkan data. Kemudian, *slave* membaca data sensor suhu dan kelembapan kemudian memasukan data tersebut ke data array dari Modbus untuk dikirimkan ke *master* nantinya. Proses ini dilakukan berulang hingga sumber daya *slave* terputus mulai dari mengirim data, membaca data sensor, dan memasukan data sensor ke data ke array modbus.

2.4.3. Flowchart Slave 2



Gambar 13. Flowchart Slave 2

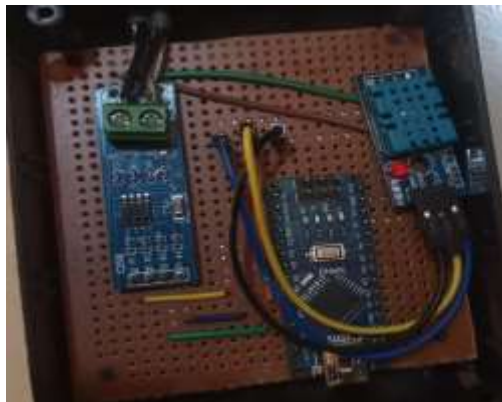
Pada Gambar 13 merupakan *flowchart* dari *slave 2* yang dimana prosesnya hampir sama seperti proses pada *slave 1*. Pertama, *slave* akan melakukan inisilasi *library* dan pin yang dibutuhkan terlebih dahulu untuk memulai sistem. Kemudian, sistem melakukan inisialisasi *slave* menggunakan ID 12 yang menandakan *slave 1* dan melakukan inisialisasi data array untuk Modbus bernama “*dataregister*” dengan format `uint16_t`. Setelah melakukan seluruh inisialisasi maka *slave* akan mengecek telegram komunikasi terlebih dahulu apakah sedang dipakai atau tidak, apabila tidak maka *slave* akan mengirimkan data. Kemudian, *slave* membaca data sensor getaran dan sensor arus listrik kemudian memasukan data tersebut ke data array dari Modbus untuk dikirimkan ke *master* nantinya. Proses ini dilakukan berulang hingga sumber daya *slave* terputus mulai dari mengirim data, membaca data sensor, dan memasukan data sensor ke data ke array modbus.

3. HASIL DAN PEMBAHASAN

Hasil perancangan alat sistem pemantauan dan pengendali suhu ruangan server berbasis *Internet Of Things* dengan notifikasi alarm dan telegram yang mencakup perancangan diagram blok sistem, perancangan *wiring*, perancangan *software*, perancangan *box* alat, pada gambar berikut ini.



Gambar 14. Rangkaian Perangkat Keras Master



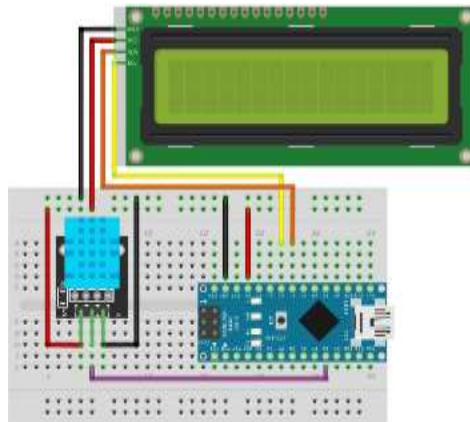
Gambar 15. Rangkaian Perangkat Keras Slave 1



Gambar 16. Rangkaian Perangkat Keras Slave 2

3.1. Pengujian Sensor DHT 11

Pengujian sensor getar dilakukan untuk mengetahui apakah sensor ini dapat diaplikasikan sebagai pendeteksi suhu dan kelembapan dapat bekerja dengan baik atau tidak, dengan cara yang dilakukan adalah merakit sensor DHT11, Arduino Nano, LCD16x2 dengan baterai pada gambar berikut.



Gambar 17. Pengujian Sensor DHT 11

Setelah merakitnya kemudian diberikan program agar dapat menjalankan pengambilan data suhu dan kelembapan, dan sebagai acuan untuk memastikan bahwa pengukuran sensor akurat atau tidaknya perlu adanya perbandingan suhu dan kelembapan ruangnya yaitu memakai alat HTC 2, untuk hasilnya dapat dilihat pada daftar tabel dibawah.

Tabel 1. Hasil Pengujian Sensor DHT 11

<i>NO</i>	<i>Termometer HTC (°C)</i>	<i>DHT 11 (°C)</i>	<i>Selisih</i>	<i>Error (%)</i>	<i>Keakuratan (%)</i>
1	25°C	26°C	1°C	4%	96%
2	25°C	27°C	2°C	8%	92%
3	26°C	27°C	1°C	4%	96%
4	25°C	27°C	2°C	8%	92%
5	25°C	26°C	1°C	4%	96%
<i>Rata –Rata</i>				5,6%	94,4%

Pada tabel diatas pengujian sensor DHT11 dilakukan dengan 5 kali percobaan dan melakukan perbandingan antara nilai pembacaan sensor dan alat yang terkalibrasi. Pada hasil pengujian diatas dapat disimpulkan bahwa sensor DHT 11 dapat bekerja dengan baik dengan nilai rata-rata *error* 5,6% dan nilai rata-rata keakurasiannya adalah 94,4%, maka dapat disimpulkan bahwa sensor ini layak untuk digunakan dalam pengukuran penelitian ini.

3.2 Pengujian pemantauan dan pengendali suhu, kelembapan, getaran dan arus listrik

Tujuan pengujian ini adalah untuk menjelaskan hasil pengujian sesuai dari rumusan masalah serta mengetahui apakah sistem dapat apabila suhu diatas ambang batas maksimal 27°C akan mengaktifkan kipas dan apakah sistem dapat apabila kelembapan diatas ambang batas maksimal 70% akan mengaktifkan kipas, serta menjawab rumusan masalah yang telah dibuat. Pengujian ini dimaksudkan untuk mengetahui apakah sensor pada slave dapat bekerja dengan baik atau tidak, serta LCD dan Telegram yang terdapat pada master dapat menampilkan data sensor dan menjalankan kipas DC sebagai pengendalian suhunya

Tabel 2. Hasil pengujian sistem pemantauan dan pengendali suhu, kelembapan, getaran dan arus listrik

NO	Data Pembacaan Master				Kondisi Kipas DC
	Suhu	Kelembapan	Getaran	Arus Listrik	
1	24°C	47%	1	0.12 A	OFF
2	26°C	70%	1	0.13 A	ON
3	31°C	57%	1	0.65 A	ON
4	21°C	45%	1	0.15 A	OFF
5	28°C	49%	1	0.55 A	ON

Pada pengujian kali ini dilakukan percobaan sebanyak 5 kali ini untuk mengetahui apakah sistem dapat bekerja dengan baik atau tidak apabila suhu sudah melampaui batas maksimal yang telah ditentukan dan batas kelembapan yang telah ditentukan. Pada pengujian ini didapat bahwa sistem *slave* mampu bekerja dengan baik melakukan pengiriman data ke *master*, karena kipas akan hidup dan alarm aktif ketika suhu atau kelembapan melebihi batas maksimal yang telah ditentukan.

4. KESIMPULAN

Setelah selesai melakukan pengujian dan menganalisis hasil yang diperoleh maka dapat ditarik kesimpulan untuk menjawab rumusan masalah. Berikut merupakan kesimpulan telah didapatkan :

1. Sistem dapat melakukan pemantauan kondisi suhu, kelembapan, getaran dan arus listrik. Berdasarkan pengujian *persentase error* sensor dengan membandingkan data hasil uji dengan perhitungan manual yang menghasilkan *persentase error* untuk suhu 5,6%, untuk sensor SW-420 dapat mendeteksi getaran, dan untuk sensor SCT-013-000 dapat membaca dengan rata rata error 1,2%. Selain itu, kondisi ruang dapat ditampilkan pada layar LCD 16x2 yang terdapat pada *master* dan kondisi ruangan dapat dikirim ke telegram sesuai jadwal yang telah ditentukan.
2. Sistem dapat mengirimkan data dari *slave* ke *master* menggunakan modbus RTU yang terhubung melalui RS485 dengan nilai *timeout* yang digunakan dari hasil analisis sebesar 3574ms. Rata- rata untuk waktu interval pengiriman data dari *slave* 1 yaitu 195,4 dan rata- rata untuk waktu interval pengiriman data dari *slave* 2 yaitu 693,5ms dengan data yang dibaca oleh *master* memiliki data yang terstruktur.
3. Sistem dapat memberikan peringatan ketika kondisi ruangan tidak normal. Alarm *buzzer* aktif dengan *tone* yang berdasarkan berdasarkan level *Minor*, *Major* dan *Critical*. Selain itu, sistem berhasil mengirimkan notifikasi peringatan ke telegram dan notifikasi dapat diterima oleh semua orang yang berada digrup telegram.

DAFTAR PUSTAKA

- [1] S. Bahri and Suhardiyanto, "Sistem Keamanan Ruang Server Menggunakan Teknologi RFID dan Password," *J. Elektum*, vol. 15, no. 1, pp. 12–18, 2018.
- [2] S. W. Hendra Budianto, "Rancang Bangun Dan Web Monitoring Pengukur Temperatur Suhu Dengan Modul Komunikasi Arduino Uno," *J. Ilmu Komput.*, vol. 1, no. 1, pp. 1–10, 2018.
- [3] ASHRAE data center, "ASHRAE data center guidelines to HP ProLiant-based facilities," no. December 2012, pp. 1–5, 2018.