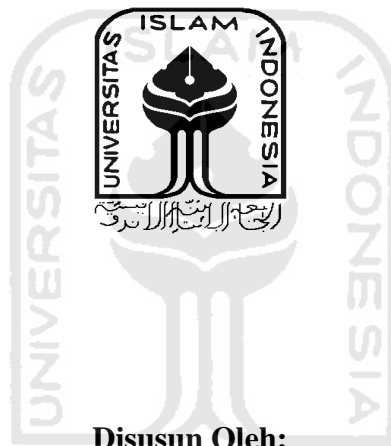


**Perancangan Prototipe Alat Monitoring Sistem Pengkondisi
Udara Berbasis *IoT* (*Internet Of things*)**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh:

Nama : Riefky Iqbal Mahendra

No. Mahasiswa : 16525106

NIRM : 2016080697

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2021

LEMBAR PENGESAHAN DOSEN PEMBIMBING

Perancangan Prototipe Alat Monitoring Sistem Pengkondisi Udara Berbasis *IoT* (*Internet Of things*)

TUGAS AKHIR



Disusun Oleh :

Nama : Riefky Iqbal Mahendra

No. Mahasiswa : 16525106

NIRM : 2016080697

Yogyakarta, 12-07-2021

Pembimbing I,

A handwritten signature in black ink, appearing to read 'Agung Nugroho Adi'.

Agung Nugroho Adi, S.T., M.T.

PERNYATAAN KEASLIAN

Demi Allah, semua yang saya tulis pada Tugas Akhir ini adalah hasil karya saya sendiri kecuali kutipan atau ringkasan yang saya ambil sebagai referensi dan telah saya cantumkan sumber-sumbernya. Apabila dikemudian hari, pengakuan saya terbukti tidak benar, maka saya bersedia mengikuti hukuman ataupun sanksi yang diberikan, sesuai sanksi hokum yang berlaku di Universitas Islam Indonesia.

Yogyakarta, 02 Agustus 2021



Riefky Iqbal Mahendra

16525106

LEMBAR PENGESAHAN DOSEN PENGUJI

Perancangan Prototipe Alat Monitoring Sistem Pengkondisi Udara Berbasis *IoT (Internet Of things)*

TUGAS AKHIR

Disusun Oleh :

Nama : Riefky Iqbal Mahendra

No. Mahasiswa : 16525106

NIRM : 2016080697

Tim Penguji

Agung Nugroho Adi, S.T., M.T

Ketua



Tanggal: 02-08-2021

Purtojo, S.T., M.Sc.

Anggota I



Tanggal: 02-08-2021

Donny Suryawan, S.T., M.Eng.

Anggota II



Tanggal: 02-08-2021

Mengetahui

Ketua Jurusan Teknik Mesin



Dr. Eng. Risdhyono, S.T., M.Eng.

HALAMAN PERSEMBAHAN

Dengan segala puji syukur kehadirat Allah SWT berkat rahmat dan karuniaNYA serta do'a dari orang-orang tercinta, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan waktu yang telah ditentukan. Tugas Akhir ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana Teknik Strata Satu (S1) di Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia, maka dengan itu saya sampaikan rasa bahagia, bangga dan ucapan terima kasih yang sebesar-besarnya kepada:

Bapak dan ibu saya yang selalu memberi dukungan, do'a, dan kasih sayangnya tanpa batas untuk penulis hingga saat ini, sehingga penulis dapat menyelesaikan tugas akhir ini sebagai salah satu bakti penulis kepada kedua orang tua.

Dosen pembimbing, dosen dan staff Teknik Mesin Universitas Islam Indonesia yang telah memberikan ilmu, arahan, dan pelajaran berharga bagi penulis selama masa perkuliahan.

Keluarga, saudara, dan teman-teman yang selalu memberi dukungan, nasehat dan semangat sehingga laporan Tugas Akhir ini selesai.

HALAMAN MOTTO

Natas, nitis, netes

(Anonim)

Hasil tidak akan mengkhianati prosesnya

(Anonim)



KATA PENGANTAR ATAU UCAPAN TERIMA KASIH

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat melaksanakan Tugas Akhir Program Studi Teknik mesin Fakultas Teknologi Industri Universitas Islam Indonesia tanpa halangan suatu apapun.

Tugas Akhir yang kami jalani Teknik Mesin Universitas Islam Indonesia merupakan salah satu tahapan yang harus ditempuh untuk menyelesaikan program Strata-1 di Jurusan Teknik Mesin Fakultas Teknologi Industri Universitas Islam Indonesia, Yogyakarta.

Tersusunnya laporan kerja praktek ini dapat kami selesaikan tidak lepas dari dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini kami ucapkan terima kasih kepada :

1. Allah SWT karena atas segala berkah dan rahmatnya kami masih diberikan kemampuan untuk dapat menyelesaikan laporan kerja praktek ini.
2. Orang tua atas kasih sayang, dukungan, dan doanya sehingga dapat melaksanakan kerja praktik dengan baik.
3. Bapak Dr. Eng. Risdiyono, S.T., M.Eng., selaku Ketua Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia
4. Bapak Agung Nugroho Adi, S.T., M.T. selaku Dosen Pembimbing Tugas Akhir.
5. Serta semua pihak yang telah membantu dalam menyusun laporan ini tidak dapat dituliskan satu persatu.

Penulis menyadari bahwa dalam pelaksanaan dan penyusunan laporan Tugas Akhir ini masih terdapat banyak kekurangan dan kesalahan. Oleh sebab itu penulis mengharap kritik dan saran yang membangun sehingga penulis dapat menyusun laporan dengan lebih baik lagi.

Wassalamu'alaikum Wr. Wb.

ABSTRAK

Pada zaman modern ini hampir setiap rumah memiliki pengkondisi udara AC (*Air Conditioning*). Perawatan AC menjadi kegiatan yang perlu dilakukan oleh masyarakat agar kondisi AC mereka tetap prima. Apabila terdapat keterlambatan dalam melakukan perawatan AC, maka kerusakan parah dan pemborosan listrik dapat terjadi. Keterlambatan ini biasanya ditandai dengan kotornya filter udara evaporator. Kondisi temperatur dan kelembaban yang keluar maupun masuk ke evaporator dapat menjadi indikator kondisi filter udara evaporator. Oleh sebab itu perancangan alat pemantau kondisi filter udara evaporator ini dapat menjadi solusi. Alat pemantau ini memanfaatkan perangkat keras mikrokontroler ESP32 Wemos Lolin32 Lite dan sensor temperatur dan kelembaban BME280 yang berfungsi untuk memantau dan mengolah data. Selain itu alat ini memanfaatkan teknologi IoT (*Internet of things*), teknologi IoT digunakan untuk mengirimkan data ke antarmuka digital dan menuju database. Hasil perancangan ini telah mampu mengirimkan kondisi filter udara dan data yang diperoleh menggunakan protokol MQTT (*Message Queuing Telemetry Transport*) menuju antarmuka berupa aplikasi *website* dan pangkalan data. Pengiriman data tersebut dibantu oleh aplikasi Arduino IDE, Node-Red dan XAMPP. Berdasarkan analisis Wilcoxon Rank Sum-Test, dapat diketahui terdapat perbedaan antara filter udara kotor dan filter udara bersih. Kemudian digunakan metode Logistik Regresi untuk mendapatkan model matematika dari kondisi filter udara pada evaporator. Setelah didapatkan model matematika, model tersebut kemudian divalidasi pada kondisi filter udara kotor dan bersih. Validasi tersebut memiliki hasil pembacaan peluang filter udara dalam keadaan bersih pada kondisi kotor = $0,68 \times 10^{-13} \%$ kemudian pada kondisi bersih = 99%.

Kata kunci: IoT, MQTT, server, evaporator, filter udara, pengkondisi udara

ABSTRACT

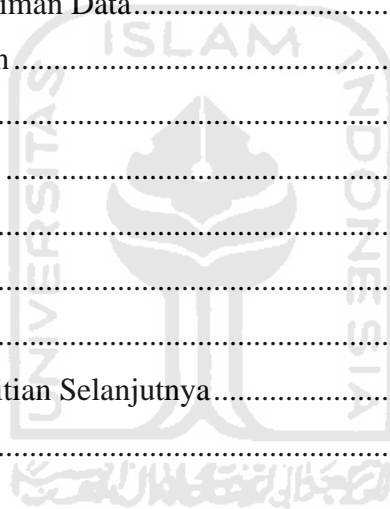
In this modern era, almost every house has air conditioning. AC maintenance is an activity that needs to be carried out by the community so that their AC conditions remain prime. If there is a delay in carrying out AC maintenance, severe damage and waste of electricity can occur. This delay is usually indicated by a dirty evaporator air filter. Temperature and humidity conditions that go out or enter the evaporator can be an indicator of the condition of the evaporator air filter. Therefore, the design of this evaporator air filter condition monitoring device can be a solution. This monitoring tool utilizes the ESP32 Wemos Lolin32 Lite microcontroller hardware and the BME280 temperature and humidity sensor which functions to monitor and process data. In addition, this tool utilizes IoT (Internet of things) technology, IoT technology is used to transmit data to digital interfaces and to databases. The results of this design have been able to send air filter conditions and data obtained using the MQTT (Message Queuing Telemetry Transport) protocol to an interface in the form of a website application and database. The data transmission is assisted by the Arduino IDE, Node-Red and XAMPP applications. Based on the Wilcoxon Rank Sum-Test analysis, it can be seen that there is a difference between a dirty air filter and a clean air filter. Then used the Logistics Regression method to get a mathematical model of the condition of the air filter on the evaporator. After obtaining the mathematical model, the model is then validated on the condition of the dirty and clean air filter. The validation has the result of reading the probability of the air filter being clean in a dirty condition = $0.68 \times 10^{(-13)}$ % then in a clean condition = 99%.

Keywords: IoT, MQTT, server, evaporator, air filter, air conditioner

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	ii
Lembar Pengesahan Dosen Penguji	iv
Halaman Persembahan	v
Halaman Motto	vi
Kata Pengantar atau Ucapan Terima Kasih	vii
Abstrak	viii
<i>ABSTRACT</i>	ix
Daftar Isi	x
Daftar Tabel	xii
Daftar Gambar	xiii
Daftar Notasi	xv
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian atau Perancangan	4
1.5 Manfaat Penelitian atau Perancangan	4
1.6 Sistematika Penulisan	5
Bab 2 Tinjauan Pustaka	6
2.1 Kajian Pustaka	6
2.2 Sensor	8
2.2.1 12C dan SPI	10
2.3 Internet of Things (IoT)	11
2.3.1 MQTT	11
2.4 Sistem Air-Conditioning (AC)	12
2.5 Grafik Psikrometrik	13
2.6 Wilcoxon Rank Sum-Test	15
2.7 Logistik Regresi	18
Bab 3 Metode Penelitian	20

3.1	Alur Penelitian	20
3.1.1	Kriteria Desain.....	21
3.1.2	Pemilihan Komponen	22
3.2	Perancangan	26
3.2.1	Perancangan Komunikasi	26
3.2.2	Perancangan Perangkat Keras	27
3.2.3	Perancangan Perangkat Lunak.....	29
3.3	Pengujian	35
Bab 4	Hasil dan Pembahasan	36
4.1	Hasil Perancangan.....	36
4.1.1	Hasil Pembacaan Sensor.....	38
4.1.2	Hasil Pengiriman Data.....	39
4.2	Hasil Pengukuran.....	40
4.3	Analisis	42
4.4	Pengujian Model	49
4.5	Pembahasan	51
Bab 5	Penutup.....	54
5.1	Kesimpulan	54
5.2	Saran atau Penelitian Selanjutnya.....	54
Daftar Pustaka	56



DAFTAR TABEL

Tabel 4-1	Sampel Selisih Temperatur Evaporator.....	44
Tabel 4-2	Sampel Selisih Kelembaban Evaporator.	44
Tabel 4-3	Tabel Ranking Temperatur.....	45
Tabel 4-4	Tabel Ranking Kelembaban.	46
Tabel 4-5	Hasil Analisis Perbedaan Selisih Temperatur Evaporator.....	46
Tabel 4-6	Hasil Analisis Perbedaan Selisih Kelembaban Evaporator.	47



DAFTAR GAMBAR

Gambar 2-1	Rangkaian Perangkat Keras Alat <i>Monitoring</i> AC Arus Listrik. ...	7
Gambar 2-2	Datasheet BME280.	9
Gambar 2-3	Perbedaan Jumlah Kabel Pada 12C dan SPI.	10
Gambar 2-4	Model TCP/IP.	12
Gambar 2-5	Sistem Kerja AC atau Kulkas.	13
Gambar 2-6	Pembacaan Grafik Psikrometrik.	15
Gambar 2-7	Dua Populasi yang Berdampingan.	16
Gambar 2-8	Pemberian Ranking Pada Sampel.	16
Gambar 2-9	Nilai <i>t_j</i> Untuk Nilai Ranking yang Sama.	17
Gambar 2-10	Tabel <i>Corresponding</i> z-Value.	18
Gambar 3-1	Diagram Alir Metode Penelitian.	20
Gambar 3-2	Antarmuka Lolin32.	23
Gambar 3-3	BME-280.	24
Gambar 3-4	Modul BME-280.	25
Gambar 3-5	Ringkasan Perancangan Komunikasi.	27
Gambar 3-6	Skema Rangkaian Alat Pemantau.	28
Gambar 3-7	Posisi Alat.	28
Gambar 3-8	Posisi Sensor Pada Alat.	29
Gambar 3-9	Alur Perancangan Perangkat Lunak.	30
Gambar 3-10	Alur Program Node-Red.	32
Gambar 3-11	Program Node-Red.	32
Gambar 3-12	Hasil Pembuatan Pangkalan Data.	33
Gambar 3-13	Perancangan Antarmuka.	34
Gambar 3-14	Fitur XAMPP.	34
Gambar 4-1	Hasil Perancangan Perangkat Keras.	36
Gambar 4-2	Peletakan Alat Pemantau.	37
Gambar 4-3	Sensor Masukan Evaporator.	37
Gambar 4-4	Sensor Keluaran Evaporator.	38
Gambar 4-5	Hasil Pembacaan Sensor.	38
Gambar 4-6	Pangkalan data Pengujian.	39

Gambar 4-7	Hasil Pengiriman Data Menuju Antarmuka.....	40
Gambar 4-8	Kondisi Filter Udara Kotor.	41
Gambar 4-9	Kondisi Filter Udara Bersih.	41
Gambar 4-10	Grafik Psikrometri.	43
Gambar 4-11	Grafik Hubungan Probabilitas dengan Data Sampel.	48
Gambar 4-12	Kondisi Filter Udara Bersih.....	49
Gambar 4-13	Hasil Pengujian Model Matematika.	49
Gambar 4-14	Kondisi Filter Udara Kotor.....	50
Gambar 4-15	Hasil Pengujian Kondisi Filter Kotor.	51
Gambar 4-16	Grafik Data Selisih Temperatur.....	53
Gambar 4-17	Grafik Data Selisih Kelembaban.	53



DAFTAR NOTASI

AC	=	<i>Air Conditioner</i>
ADC	=	<i>Analog Digital Converter</i>
API	=	<i>Application Programming Interface</i>
BTU	=	<i>British Thermal Unit</i>
DAC	=	<i>Digital Analog Converter</i>
HTTP	=	<i>Hyper Text Transfer Protocol</i>
HTTPS	=	<i>Hyper Text Transfer Protocol Secure</i>
I2C	=	<i>Inter Integrated Circuit</i>
IoT	=	<i>Internet of things.</i>
IP	=	<i>Internet Protocol</i>
LCD	=	<i>Liquid Crystal Display</i>
MQTT	=	<i>Message Queue Telemetry Transport</i>
Pub	=	<i>Publish</i>
SCL	=	<i>Serial Clock</i>
SDA	=	<i>Serial Data</i>
SPI	=	<i>Serial Peripheral Interface</i>
SQL	=	<i>Structured Query Language</i>
Sub	=	<i>Subscribe</i>
TCP	=	<i>Transmission Control Protocol</i>
USB	=	<i>Universal Serial Bus</i>

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perangkat pengkondisi udara seperti AC dan refrigerator (kulkas) merupakan alat pendingin yang memiliki kesamaan cara kerja, perbedaan perangkat AC dan refrigerator terletak pada tujuan pengkondisian udara. AC digunakan untuk mendinginkan ruangan sedangkan kulkas untuk mendinginkan makanan (Cengel & Boles, 2006). Alat-alat tersebut menggunakan fluida cair sebagai media untuk memindahkan panas. Fluida cair yang digunakan biasanya berupa refrigeran R22, R410A, dan R32.

Komponen utama dari sistem pengkondisi udara adalah evaporator, kompresor, kondensor dan piranti ekspansi. Evaporator merupakan komponen sistem pengkondisi udara yang berfungsi untuk memindahkan panas dari udara ke fluida cair freon, sedangkan kondensor sebaliknya (Hundy dkk., 2016). Pada evaporator dan kondensor, terdapat komponen filter udara yang berfungsi untuk memisahkan kotoran dengan udara yang masuk ke dalam ke evaporator maupun kondensor.

Walaupun demikian, filter udara yang memisahkan kotoran dengan udara pada evaporator serta kondensor dapat menjadi kotor. Akibatnya kecepatan aliran udara yang masuk ke evaporator dan kondensor akan berkurang serta membuat performa pendinginan menurun dan membuat konsumsi energi meningkat (Rodriguez dkk., 1996).

Kondisi dari filter udara yang bersih akan membantu filter refrigeran untuk membersihkan refrigeran dan pipa refrigeran, menurut Chao dkk (2012) kondisi pipa yang kotor akan membuat temperatur keluaran katup ekspansi menurun. Efek panjang dari penurunan temperatur ini dapat mengakibatkan rusaknya komponen seperti pompa dan pipa refrigeran.

Alat pemantau atau *monitoring* menjadi solusi untuk mengetahui waktu perawatan AC yang tepat. Mayoritas alat yang telah dibuat menggunakan *mikrokontroler*, sebelum perancangan ini dilakukan terdapat beberapa alat

pemantau sistem pengkondisi udara. Ilham dan Ramschie (2013) membuat alat pemantau sistem pengkondisi udara hanya menggunakan *mikrokontroler* AVR ATmega 8535, dan sensor arus ACS 712. Selain itu kelebihan alat ini telah mengetahui pemborosan pemakaian energi listrik pada AC kotor di settingan suhu 24-25 °C, sedangkan kekurangannya antarmuka yang digunakan tidak bisa dilihat dimanapun.

Kekurangan alat tersebut dijawab oleh Diori (2019) dengan merancang pemantau AC dengan antarmuka digital dan penyimpanan data berbasis IoT (*Internet of Things*), alat tersebut dapat mengetahui kerusakan pada AC, kerusakan pada tendon air. Penggunaan banyak komponen menjadi kelemahan alat ini. Selain itu untuk menghubungkan Node-MCU dengan antarmuka web membutuhkan *script* API (*Application Programming Interface*) yang panjang (Nasution dkk., 2019).

Mahasiswa prodi Teknik Mesin UII pernah melakukan perancangan dengan memanfaatkan sistem IoT (Mukti, 2020). Sistem IoT dimanfaatkan untuk pembuatan *remote* AC dan pemantau sistem pengkondisi udara yang memiliki antarmuka aplikasi web yang mudah digunakan. Walaupun demikian alat pemantau sistem pengkondisi udara tersebut belum mampu mengirimkan data hasil pembacaan sensor ke dalam sebuah pangkalan data. Sehingga alat tersebut belum dilengkapi model kendali untuk membedakan kondisi filter udara bersih maupun filter udara kotor.

Temperatur dan kelembaban dapat menjadi indikator kebersihan dari filter udara evaporator maupun kondensor, selain itu kemudahan dalam mengaplikasikan dan memahami perangkat keras maupun lunak untuk membuat sistem IoT telah tersedia saat ini, Oleh sebab itu pembuatan alat pemantau sistem pengkondisi udara yang sedikit komponen, transmisi data ke antarmuka yang mudah, serta memperhatikan kondisi temperatur dan kelembaban untuk memodelkan kondisi filter udara bersih maupun kotor, dapat menjadi solusi dari permasalahan tersebut.

Pertimbangan tersebut mengarahkan penulis untuk merancang sebuah alat pemantau sistem pengkondisi udara, yang dapat memberikan peringatan dini apabila filter udara evaporator maupun kondensor sudah mengalami kekotoran

berdasarkan kondisi temperatur dan kelembaban dari masukan dan keluaran evaporator atau kondensor. Alat pemantau sistem pengkondisi udara (AC) tersebut harus memiliki transmisi data yang mudah untuk dirancang, dengan kemudahan dalam mengaplikasikan dan memahami perangkat keras atau lunak untuk membuat sistem IoT telah tersedia saat ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, maka dapat dirumuskan Rumusan masalah sebagai berikut :

1. Bagaimana membuat alat pemantau sistem pengkondisi udara yang dapat memberikan peringatan dini terhadap kekotoran filter udara evaporator maupun kondensor?
2. Apa perbedaan parameter temperatur dan kelembaban pada evaporator dan kondensor dalam kondisi filter udara kotor dan bersih?
3. Bagaimana menyusun model kendali yang dapat mendeteksi perbedaan kondisi filter udara evaporator dan kondensor yang bersih dengan yang bersih?

1.3 Batasan Masalah

Batasan Masalah yang digunakan dalam perancangan dan penelitian ini adalah :

1. Penelitian hanya dilakukan pada satu AC split merk Daikin dengan kapasitas pendinginan 5000 BTU/h, dan daya listrik 380 W untuk mendinginkan ruangan 4 x 3 x 3 m.
2. Perancangan alat hanya sampai pada tahap prototipe, sehingga memungkinkan dilakukan pengembangan selanjutnya.
3. Penelitian yang dilakukan hanya mencari karakteristik filter udara kotor dan bersih pada evaporator, tidak sampai pada kerusakan pada komponen lain.
4. Perancangan alat ini digunakan untuk mendeteksi, memonitoring dan memberikan notifikasi terkait kondisi filter udara evaporator pengkondisi udara.

5. Perancangan dilakukan untuk mendeteksi kondisi temperatur pengkondisi udara yang dikonfigurasi pada temperatur 22 °C dengan penggunaan pada siang hari.
6. Hasil perancangan alat akan memonitoring kondisi filter udara kotor dari evaporator dan kondensor, namun hanya filter udara evaporator saja yang akan dimodelkan.

1.4 Tujuan Penelitian atau Perancangan

Tujuan dari perancangan dan penelitian ini adalah:

1. Membuat alat pemantau sistem pengkondisi udara yang dapat memberikan peringatan dini terhadap kekotoran filter udara evaporator.
2. Mengetahui perbedaan parameter temperatur dan kelembaban pada evaporator dalam kondisi filter udara kotor dan bersih.
3. Menyusun model kendali yang dapat mendeteksi perbedaan kondisi filter udara evaporator yang kotor dengan yang bersih.

1.5 Manfaat Penelitian atau Perancangan

Manfaat perancangan dan penelitian ini bagi peneliti, masyarakat, industri, dan kampus adalah:

1. Peneliti, masyarakat, industri, dan kampus dapat memanfaatkan kegunaan AC secara maksimal.
2. Mengurangi pemborosan energi listrik dalam kehidupan peneliti, masyarakat, industri, dan kampus.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang, waktu pelaksanaan, tujuan, manfaat, dan sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang penelitian atau perancangan yang telah ada serta penggunaan teori pada penelitian atau perancangan yang dilakukan.

BAB III METODE PENELITIAN

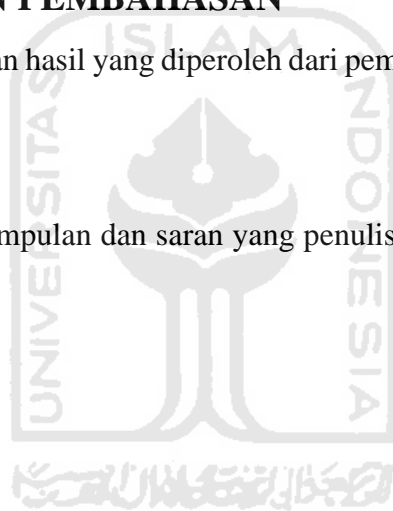
Bab ini menguraikan secara singkat tentang proses penelitian atau perancangan yang telah dilakukan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil yang diperoleh dari pembahasan maupun analisis yang telah dilakukan.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran yang penulis paparkan dalam laporan Kerja Praktik ini.



BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penggunaan AC secara terus-menerus dapat membuat filter evaporator maupun kondensor menjadi kotor. Kotornya filter AC dapat menimbulkan kerugian bagi pengguna AC, contohnya adalah performa pendinginan AC akan menurun, pemborosan listrik hingga kerusakan komponen kompresor. Alat pemantau AC berfungsi untuk mencegah timbulnya kerugian bagi pengguna.

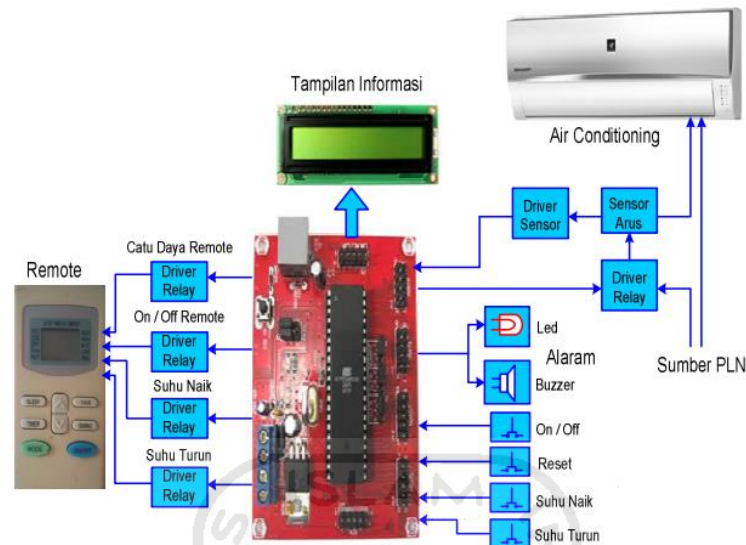
Beberapa alat pemantau AC telah dibuat, salah satunya adalah alat pemantau AC buatan Ilham dan Ramschie (2013). Alat tersebut menggunakan sensor arus ACS 712 yang dipasang pada arus listrik yang diperlukan oleh kompresor AC. Pemantau AC ini digunakan untuk mendeteksi adanya pemborosan arus listrik yang digunakan oleh kompresor AC.

Hasil pembacaan sensor ACS 712 kemudian akan dikirimkan ke *microcontroller* AVR ATmega 8535. AVR ATmega 8535 berfungsi untuk mengubah besar arus listrik menjadi energi listrik per jam serta membandingkan kondisi energi listrik pada saat kondisi AC bersih dan kondisi AC kotor. Suhu AC yang digunakan pada penelitian tersebut berkisar 24-25 °C.

Microcontroller AVR ATmega 8535 juga dihubungkan dengan LCD yang berfungsi untuk menampilkan besar konsumsi energi listrik saat pengambilan data. Selain itu LCD juga berfungsi untuk menampilkan kondisi dari AC baik maupun sedang dalam keadaan kotor.

Alat pemantau AC ini telah mampu membedakan kondisi filter AC kotor maupun filter AC bersih, selain itu alat ini dapat memberikan peringatan apabila AC dalam keadaan kotor. Walaupun demikian, alat ini akan mendeteksi kekotoran AC apabila telah terjadi pemborosan energi listrik terlebih dahulu, sehingga alat ini belum mampu memberikan peringatan dini adanya kekotoran AC yang menimbulkan kerugian pemborosan energi.

Hal lain yang menjadi kelemahan alat tersebut adalah penggunaan antarmuka LCD yang terlihat pada Gambar 2-1. Antarmuka LCD hanya mampu digunakan apabila pengguna berada dekat dengan AC atau alat pemantau tersebut.



Gambar 2-1 Rangkaian Perangkat Keras Alat Monitoring AC Arus Listrik.

Sumber: Ilham & Ramschie (2013)

Alat pemantau AC lainnya yang dirancang oleh Diori (2019) telah menggunakan antarmuka berbasis web. Antarmuka ini berfungsi untuk memberikan peringatan dini mengenai kondisi air pada tandon AC dan kondisi motor DC 5V pada AC. Sensor yang digunakan pada alat ini adalah sensor ketinggian air untuk memeriksa kondisi dari air pada tandon AC. Sedangkan pada kondisi motor DC 5V digunakan sensor *hall effect* untuk mengetahui kondisi putaran motor. Hasil pembacaan sensor kemudian akan dikirimkan menuju Arduino menggunakan transmisi kabel.

Papan Arduino yang digunakan telah terhubung dengan papan modul Node-MCU. Modul Node-MCU tersebut berfungsi untuk mengirimkan data dari Arduino menuju server web Thingspeak. Komunikasi yang digunakan oleh Node-MCU dan Thingspeak adalah API (*Application Programming Interface*) (Nasution dkk., 2019). API yang digunakan berjalan dengan menggunakan protokol komunikasi HTTP (*HyperText Transfer Protocol*), dengan HTTP, proses

komunikasi memiliki beberapa perintah seperti GET, PUT, POST dan DELETE (Saputra & Lukito, 2017).

Secara keseluruhan alat pemantau sistem pengkondisi udara ini telah mampu membedakan kondisi refrigeran dan motor DC bersih. Selain itu alat ini telah dilengkapi dengan antarmuka berbasis web, sehingga dapat dilakukan pemantauan kondisi AC dari tempat yang jauh. Meskipun demikian, alat ini belum mampu memberikan peringatan dini kerusakan komponen AC, serta komunikasi antara perangkat keras dengan perangkat lunak masih dinilai terlalu sulit.

Alat pemantau sistem pengkondisi udara dan pengendali sistem pengkondisi udara yang dibuat oleh Mukti (2020) menggunakan protokol MQTT dalam pengiriman data dari Wemos Lolin-Lite ke antarmuka. Pembuatan antarmuka dilakukan menggunakan aplikasi web Node-Red, sedangkan server MQTT yang digunakan beralamat moussquitto.org. Wemos Lolin-Lite dan Node-Red sangat mudah digunakan untuk membangun sistem IoT.

Meskipun begitu alat pemantau sistem pengkondisian udara tersebut belum mampu membedakan kondisi filter udara kotor maupun filter udara bersih. Selain itu alat ini belum mampu mengirimkan data pembacaan sensor ke pangkalan data. Sehingga pengolahan data untuk menemukan model filter udara bersih maupun filter udara kotor belum bisa dilakukan.

Berdasarkan tinjauan pustaka yang telah dipaparkan perancangan ini akan mencoba untuk menghilangkan kekurangan perancangan yang telah dilakukan oleh Fauzi. Namun dengan sedikit perbedaan dalam menggunakan sensor, penambahan program Node-Red ke pangkalan data dan pemodelan kondisi filter udara bersih dan kotor. Pemodelan yang dilakukan bertujuan untuk mengetahui komponen sistem pengkondisi udara dalam keadaan kotor atau tidak, seperti yang dilakukan oleh Diori (2019) dan Ilham dan Ramschie (2013), namun pemodelan yang dilakukan hanya untuk kondisi filter udara evaporator.

2.2 Sensor

Untuk mengetahui kondisi filter AC melalui temperatur dan kelembaban udara dari masukan serta keluaran evaporator maupun kondensor, dibutuhkan sensor kelembaban dan temperatur. Terdapat beberapa *low-energi* sensor

kelembaban dan temperatur, seperti DHT11, DHT22, LM35, BMP180, dan BME280.

Terdapat perbedaan spesifikasi seperti, akurasi, presisi, rentang pengukuran dan protokol komunikasi dari beberapa sensor tersebut. Sebagai contoh BME280 yang memiliki resolusi pembacaan temperatur sebesar 0,01 °C dan akurasi sebesar 1% seperti yang ditunjukkan pada Gambar 2-2, kemudian rentang pengukuran -40 °C sampai 85 °C, selain itu BME280 juga mampu membaca tekanan (Sensortec, 2014). BME 280 memiliki komunikasi I2C dan SPI, hal ini tergantung pada modul yang digunakan. Spesifikasi tersebut dapat dijadikan acuan dalam memilih sensor untuk sebuah penelitian atau perancangan.

Absolute accuracy temperature ⁷	A _{T,full}	0...65 °C		±1.0		°C
	A _{T,ext} ⁸	-20 ... 0 °C		±1.25		°C
	A _{T,ext} ⁹	-40 ... -20 °C		±1.5		°C
Output resolution	R _T	API output resolution		0.01		°C
RMS noise	N _T	Lowest oversampling		0.005		°C

Gambar 2-2 Datasheet BME280.

Sumber :(Sensortec, 2014).

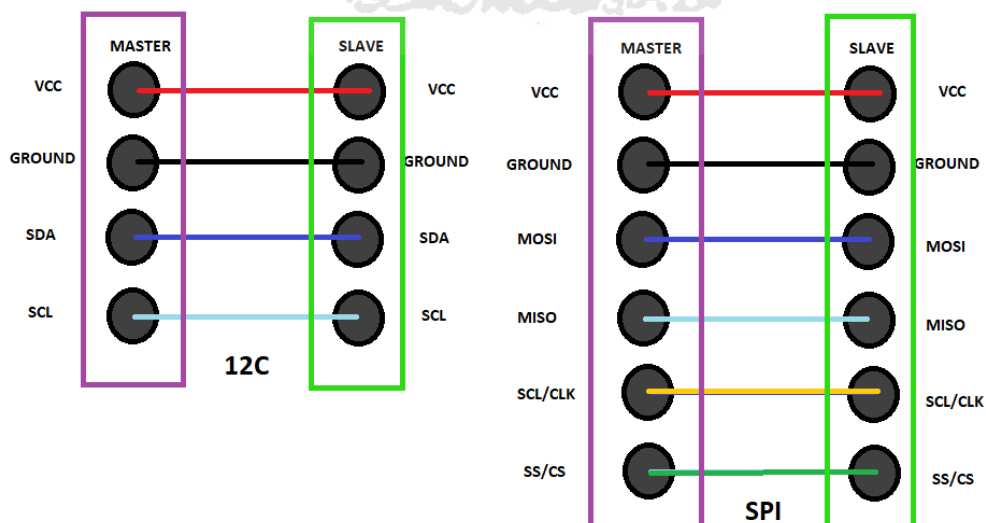
Contoh spesifikasi sensor lainnya adalah DHT11 dengan spesifikasi resolusi 1 °C dengan rentang pengukuran temperatur 0 °C sampai 50 °C dan kelembaban 20% sampai 90% (Gay, 2018). Kemudian untuk DHT22 memiliki resolusi 0,5°C dan 2% dan rentang pengukuran temperatur -40 °C sampai 80 °C dan pengukuran kelembaban 0% sampai 100% (Liu, 2013).

Berbeda dengan BME280 maupun BMP180 yang memiliki komunikasi SPI atau I2C (Sensortec, 2013). DHT11 dan DHT22 memiliki komunikasi serial *single bus* atau *1-Wire*. Komunikasi tersebut hanya menggunakan satu kabel untuk mengirimkan data. Apabila I2C atau SPI terdapat lebih dari satu kabel untuk mengirimkan data. Pada komunikasi *1-Wire* tidak ada line untuk mengirimkan data *clock*.

2.2.1 I2C dan SPI

I2C maupun SPI merupakan protokol komunikasi serial, kedua protokol ini biasanya digunakan untuk mengirimkan data dari sensor ke *mikrokontroler*. Baik I2C maupun SPI mengenal istilah Master dan *Slave*. Master merupakan komponen yang mengirimkan data *clock*, sedangkan *slave* merupakan komponen yang menerima data *clock*.

Komunikasi serial merupakan komunikasi yang mengirimkan data bit hanya melalui satu port dan setiap pengiriman data bit akan dihitung menggunakan data *clock*. Hal ini membuat komunikasi serial lebih lambat dibandingkan dengan komunikasi paralel, yang mengirimkan data melalui banyak port. Perbedaan yang mendasar antara I2C dengan SPI adalah, I2C bekerja secara *half-duplex* sedangkan SPI bekerja secara *full-duplex* (Leens, 2009). Artinya SPI dapat mengirimkan data dari master ke slave dan sebaliknya, sedangkan I2C hanya mampu mengirimkan data dari master ke slave atau slave ke master. Gambar 2-3 menunjukkan bagaimana sambungan kabel antara I2C dan SPI digunakan. Gambar 2-3 menginformasikan bahwa SPI membutuhkan lebih banyak kabel dari pada I2C. Mengkomunikasikan 2 sensor atau lebih ke satu mikrokontroler dengan menggunakan protokol I2C diperlukan rangkaian *pull-up* pada SDA dengan pin dan SCL dengan pin.



Gambar 2-3 Perbedaan Jumlah Kabel Pada I2C dan SPI.

Diadaptasi Dari: Leens (2009)

2.3 Internet of Things (IoT)

Internet of Things atau IoT merupakan sebuah tujuan untuk membuat perangkat yang biasanya terdiri dari sensor dan aktuator, agar mampu berkomunikasi melalui jaringan IP (*Internet Protocol*) (Cirani dkk., 2019). IoT dapat digunakan dalam berbagai sektor, contohnya sektor perumahan, industri, pertanian, perkantoran sampai perkotaan (*Smart Cities*).

Penggunaan IP (*Internet Protocol*) di masa sekarang, lebih banyak untuk mengkomunikasikan antara gawai (komputer, laptop, telepon pintar) dengan server. Contoh IP (*Internet Protocol*) yang sering kita gunakan adalah HTTP (*Hypertext Transfer Protocol*) dan HTTPS (*Hypertext Transfer Protocol Secure*).

HTTP dapat digunakan dalam pembuatan IoT, biasanya HTTP akan berdampingan dengan REST (*Representational State Transfer*). Salah satu fungsi dari REST adalah *scalability* (mengatur interaksi antara pengguna atau objek dengan server) (Cirani dkk., 2019). Selain HTTP terdapat protokol lain yang disebut dengan MQTT (*Message Queue Telemetry Transport*).

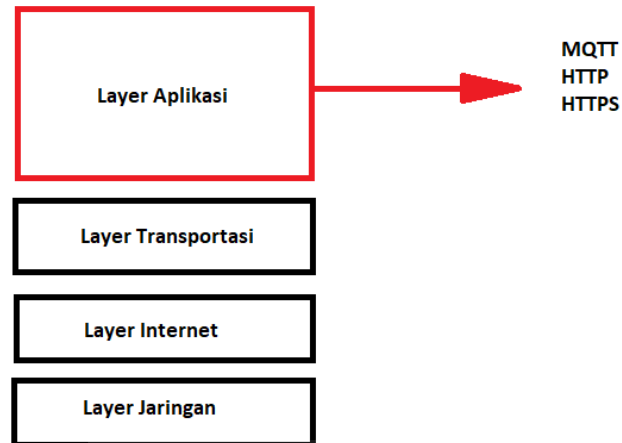
2.3.1 MQTT

MQTT (*Message Queue Telemetry Transport*) merupakan protokol komunikasi yang ringan serta menggunakan metode Pub/Sub (*publish/subscribe*) dalam mengirimkan data. Pesan yang dikirimkan/di *publish* menuju broker MQTT dilengkapi dengan topik tertentu. Pesan dapat diakses/*subscribe* oleh pengguna, apabila pengguna mengetahui topik tersebut.

Kelebihan menggunakan MQTT dan Pub/Sub adalah pesan yang dikirimkan lebih ringan dan lebih mudah dalam merancang komunikasinya daripada menggunakan protokol HTTP dan REST. Sebenarnya metode Pub/Sub dapat dibuat dengan menggunakan protokol HTTP, namun MQTT merupakan protokol yang lebih ringan dari HTTP (Hillar, 2017).

Protokol MQTT bekerja pada layer *Application* dalam model protokol tradisional TCP/IP (*Transmission Control Protocol / Internet Protocol*) sama seperti HTTP maupun HTTPS. Posisi dari layer *Application* dapat dilihat pada Gambar 2-4. Gambar 2-4 memperlihatkan beberapa layer pada model protokol

tradisional TCP/IP. Layer-layer tersebut menunjukkan urutan mengirimkan sebuah data melalui jaringan internet.



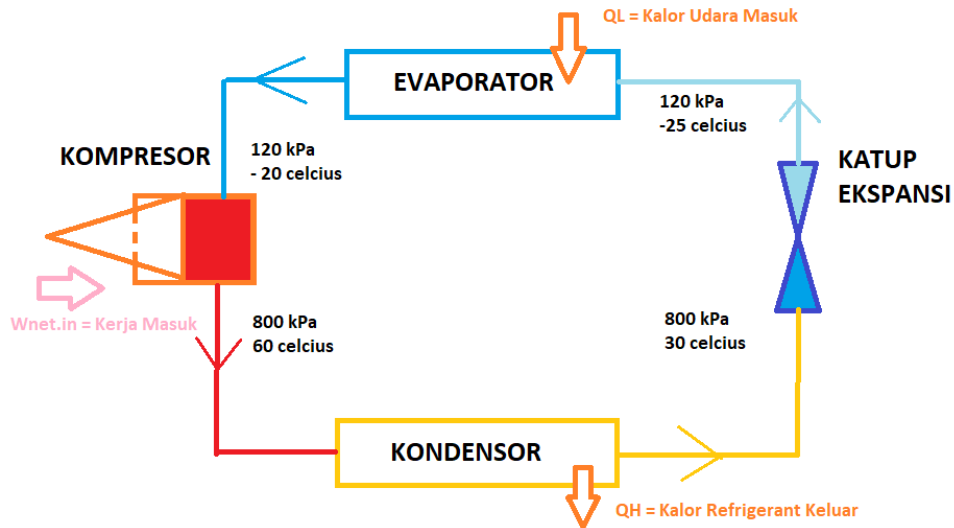
Gambar 2-4 Model TCP/IP.

Diadaptasi Dari: Hunt (2002)

Aplikasi layer akan menyusun data yang ingin dikirimkan sesuai protokol yang digunakan. Penyusunan data tersebut biasanya dilakukan oleh aplikasi browser seperti, Google Chrome, Opera Mini dan lain-lainnya.

2.4 Sistem Air-Conditioning (AC)

Air-Conditioning (AC) merupakan sebuah sistem pemindah kalor yang memiliki sistem kerja seperti *refrigerator* (kulkas). Sistem kerja AC maupun kulkas memiliki komponen utama seperti kompresor, kondensor, evaporator dan katup ekspansi atau pompa (Cengel & Boles, 2006).



Gambar 2-5 Sistem Kerja AC atau Kulkas.

Diadaptasi Dari: Cengel & Boles (2006)

Gambar 2-5 merupakan sistem kerja AC atau kulkas dengan empat komponen utama. Kondisi air yang melewati keempat komponen utama (kulkas) juga dijelaskan pada Gambar 2-5. Berdasarkan Gambar 2-5 juga diketahui bahwa pada komponen kompresor terdapat kerja yang masuk ($W_{net.in}$) sedangkan pada evaporator terdapat kalor yang masuk (Q_L) dan pada kondensator terdapat kalor yang keluar (Q_H). Nilai dari Q_H maupun Q_L dirumuskan dalam persamaan (1. 1):

$$Q_{L/H} = m \times c \times (T_{in} - T_{out}) \quad (1. 1)$$

m merupakan massa dari benda atau zat yang melepas atau menerima kalor dalam (Kg) sedangkan c kalor jenis zat dalam (J/kg °C) serta $(T_{in} - T_{out})$ merupakan selisih dari temperatur udara masuk dan temperatur udara keluar.

2.5 Grafik Psikrometrik

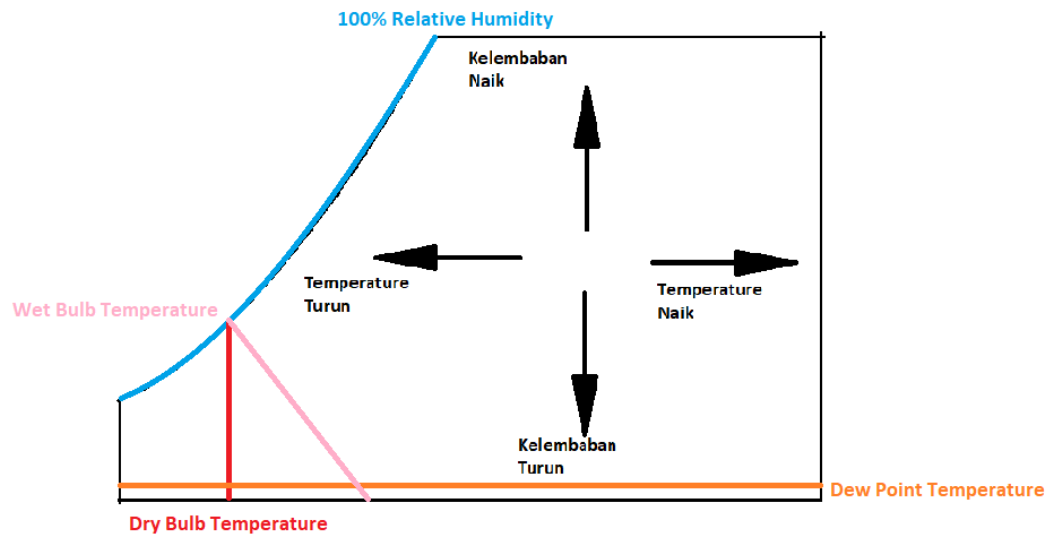
Grafik Psikrometrik merupakan grafik yang berfungsi untuk menghubungkan beberapa indikator udara dari kondisi udara. Indikator-indikator tersebut berupa *dry-bulb temperature*, *wet-bulb temperature*, *relative humidity*, *specific humidity*, dan *enthalpy*. *Specific humidity* merupakan massa uap air yang terkandung dalam setiap 1 kg udara kering.

Sedangkan untuk *relative humidity* merupakan kelembaban yang biasanya berada pada kondisi saturasi udara dan merupakan nilai perbandingan antara nilai kelembaban spesifik pada suatu waktu dengan kelembaban spesifik yang dapat ditampung pada temperatur dan tekanan yang sama (Cengel & Boles, 2006).

Indikator lain berupa *dry-bulb temperature* merupakan nilai asli hasil pengukuran temperatur pada udara atmosfer. Untuk *dew-point temperature* merupakan nilai temperatur ketika kondensasi mulai terjadi pada udara yang didinginkan pada tekanan konstan atau *dew-point temperature* merupakan nilai temperatur yang ada pada kondisi saturasi pada grafik T-s (Cengel & Boles, 2006).

Wet-bulb temperature merupakan nilai temperatur yang diukur menggunakan termometer dengan ujung dari termometer diberikan sebuah kapas basah untuk mensimulasikan kondisi *Wet-bulb temperature* tanpa harus membuat sebuah pipa yang panjang. Pipa panjang tersebut nantinya akan menghasilkan proses adiabatic (Cengel & Boles, 2006). Kemudian untuk membuat Grafik Psikrometrik ini biasanya digunakan untuk menggambarkan kondisi udara pada siklus AC melalui indikator tersebut.

Sebelum hadirnya grafik psikrometri untuk melihat kondisi udara melalui indikator udara, dapat dilakukan pengukuran menggunakan alat ukur konvensional serta persamaan matematika. Untuk mempermudah pembacaan kondisi udara maka dibuatlah sebuah grafik, grafik tersebut dikenal dengan grafik psikrometrik. Cara pembacaan dari grafik psikrometrik ditunjukkan pada Gambar 2-6, arah panah yang ditunjukkan oleh Gambar 2-6 menunjukkan kondisi udara pada siklus AC.

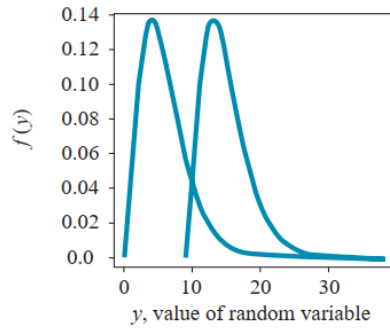


Gambar 2-6 Pembacaan Grafik Psikrometrik.

Diadaptasi Dari: Cengel & Boles (2006)

2.6 Wilcoxon Rank Sum-Test

Terdapat banyak untuk mengetahui perbedaan dari dua populasi data. Metode yang paling umum digunakan adalah uji hipotesis dua sampel *Z-test* dan *T-test*. Untuk menggunakan metode *T-test* diperlukan dua populasi yang berdistribusi normal. Kasus perancangan ini memiliki populasi data yang berdistribusi tidak normal. Oleh sebab itu untuk mengetahui perbedaan dari dua populasi data digunakan metode Wilcoxon Rank Sum-Test. Metode Wilcoxon Rank Sum-Test dapat digunakan pada *independent random samples* yang diperoleh dari dua populasi yang memiliki distribusi yang sama atau saling berdampingan seperti yang ditunjukkan pada Gambar 2-7 (Ott & Longnecker, 2001). Metode ini tidak mengharuskan distribusi dari kedua populasi adalah normal.



Gambar 2-7 Dua Populasi yang Berdampingan.

Sumber :Otto dan Longnecker (2001)

Metode ini menggabungkan 2 sampel dari 2 populasi kemudian memberikan nilai atau ranking seperti yang ditunjukkan pada Gambar 2-8. Gambar 2-8 menunjukkan terdapat 24 ranking. Ranking-ranking tersebut didapatkan dengan mengurutkan berdasarkan nilai terkecil sampai terbesar pada 12 sampel sebelum pembersihan (n_1) dan (n_2) 12 sampel sesudah pembersihan. Masing-masing sampel terdiri dari $n_1 = 12$ data dan $n_2 = 12$. Apabila nilai n (jumlah data pada setiap sampel) >10 , maka dapat digunakan perhitungan Z -test karena distribusi dari ranking sampel dianggap normal (Ott & Longnecker, 2001).

Before Cleanup		After Cleanup	
11.0	(10)	10.2	(1)
11.2	(14)	10.3	(2)
11.2	(14)	10.4	(3)
11.2	(14)	10.6	(4,5)
11.4	(17)	10.6	(4,5)
11.5	(18)	10.7	(6)
11.6	(19)	10.8	(7,5)
11.7	(20)	10.8	(7,5)
11.8	(21)	10.9	(9)
11.9	(22,5)	11.1	(11,5)
11.9	(22,5)	11.1	(11,5)
12.1	(24)	11.3	(16)
$T = 216$			

Gambar 2-8 Pemberian Ranking Pada Sampel.

Sumber :Otto dan Longnecker (2001)

Nilai dari z dalam Wilcoxon Rank Sum Test sendiri untuk n (jumlah data pada setiap sampel) >10 dirumuskan pada persamaan (1. 2).

$$z = \frac{(T - \mu_T)}{\sigma_T} \quad (1.2)$$

T menunjukkan jumlah dari ranking sampel yang akan diobservasi, contoh dari nilai T dapat dilihat pada Gambar 2-8, dengan nilai $T = 216$. Nilai tersebut merupakan jumlah ranking pada sampel pertama (n_1) "Before Clean-up". Sedangkan nilai μ_T dirumuskan pada persamaan (1.3):

$$\mu_T = \frac{n_1 \times (n_1 + n_2 + 1)}{2} \quad (1.3)$$

Kemudian untuk σ_T dirumuskan dalam persamaan (1.4) :

$$\sigma_T = \frac{n_1 \times n_2}{12} \left[(n_1 + n_2 + 1) - \frac{\sum t_j(t_j^2 - 1)}{(n_1 + n_2)(n_1 + n_2 - 1)} \right] \quad (1.4)$$

t_j merupakan nilai dari jumlah data dengan nilai ranking yang sama seperti yang ditunjukkan pada Gambar 2-9 untuk data seperti yang diperoleh pada Gambar 2-8.

Rank(s)	Group	t_j	Rank(s)	Group	t_j
1	1	1	14, 14, 14	10	3
2	2	1	16	11	1
3	3	1	17	12	1
4.5, 4.5	4	2	18	13	1
6	5	1	19	14	1
7.5, 7.5	6	2	20	15	1
9	7	1	21	16	1
10	8	1	22.5, 22.5	17	2
11.5, 11.5	9	2	24	18	1

Gambar 2-9 Nilai t_j Untuk Nilai Ranking yang Sama.

Sumber : Otto dan Longnecker (2001)

Kemudian setelah nilai z dapat ditentukan maka nilai z akan dibandingkan dengan nilai $z_{\alpha/2} = 1,645$. Nilai $z_{\alpha/2}$ diperoleh dari Gambar 2-10.

Confidence Coefficient (1 - α)	Value of $\alpha/2$	Area in Table 1 1 - $\alpha/2$	Corresponding z-Value, $z_{\alpha/2}$
.90	.05	.95	1.645
.95	.025	.975	1.96
.98	.01	.99	2.33
.99	.005	.995	2.58

Gambar 2-10 Tabel *Corresponding z-Value*.

Sumber :Otto dan Longnecker (2001)

2.7 Logistik Regresi

Regression logistic atau logistik regresi merupakan metode dalam statistika yang memiliki kesamaan dengan metode regresi biasa. Kedua metode tersebut berguna untuk meramalkan nilai suatu variabel terikat dengan menggunakan nilai dari variabel bebas (Ott & Longnecker, 2001). Variabel terikat pada diagram kartesius biasanya berada pada sumbu Y sedangkan variabel bebas terletak pada sumbu X .

Variabel terikat pada logistik regresi biasanya terdiri dari dua kondisi yang saling berlawanan contohnya, besar dan kecil, panjang dan pendek, baik dan buruk, kasar dan halus dan lain-lainnya. Kondisi tersebut kemudian digambarkan dalam probabilitas bilangan numerik 0 sampai 1. Probabilitas tersebut dirumuskan sebagai persamaan (1. 5),

$$p(x) = \frac{e^x}{(1 + e^x)} \quad (1. 5)$$

Nilai $p(x)$ merupakan nilai probabilitas dari x , dimana x merupakan nilai dari skor yang dirumuskan dengan persamaan (1. 6)

$$x = (A) + (B \times x_1) + (C \times x_2) \dots (Z \times x_n) \quad (1. 6)$$

Untuk nilai y dan $A, B, \dots Z$ diperoleh dari persamaan regresi normal atau regresi logistik yang dirumuskan, sebagai persamaan (1. 7) dan (1. 8)

$$y = A + Bx_1 \dots \dots \dots \text{(Regresi Normal)} \quad (1. 7)$$

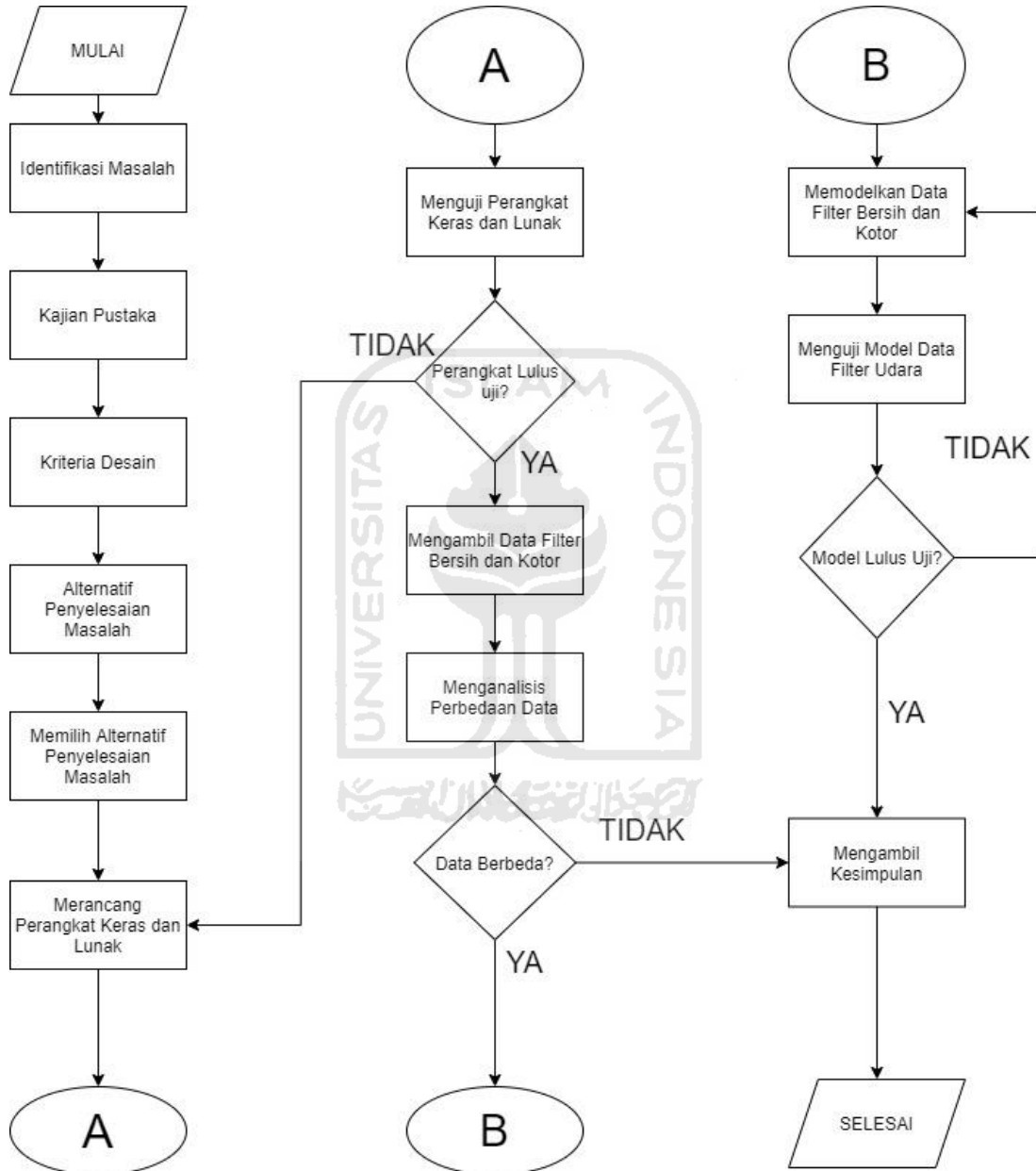
$$p(x) = (A + Bx_1 + Cx_2 \dots Zx_n) \dots (\text{Regresi Logistik}) \quad (1.8)$$

Menurut Ott & Longnecker (2001) perbedaan dari kedua metode tersebut terletak pada jumlah jenis variabel bebas. Regresi normal dapat digunakan apabila data memiliki satu variabel bebas dan satu variabel terikat. Sedangkan logistik regresi dapat digunakan apabila data memiliki lebih dari satu variabel bebas dan satu variabel terikat.



BAB 3 METODE PENELITIAN

3.1 Alur Penelitian



Gambar 3-1 Diagram Alir Metode Penelitian.

Penelitian ini menggunakan metode penelitian yang tertera pada Gambar 3-1. Berdasarkan diagram alir tersebut diketahui terdapat beberapa tahapan dalam Perancangan Prototipe Alat

Monitoring AC (*Air Conditioning*) Berbasis IoT. Tahapan pertama adalah mengidentifikasi masalah dalam perawatan pengkondisi udara AC. Secara singkat, perawatan pengkondisi udara yang terlambat dapat menyebabkan kerusakan permanen pada komponen AC. Hal ini harus dihindari, karena proses perbaikan komponen pengkondisi udara AC memerlukan biaya yang tidak sedikit. Selain itu keterlambatan perawatan dapat menyebabkan menurunnya kenyamanan penggunaan. Setelah permasalahan didapatkan maka langkah selanjutnya adalah melakukan kajian pustaka.

Kajian pustaka ini bertujuan untuk mencari alternatif penyelesaian masalah dan menentukan kriteria desain. Alternatif penyelesaian ini bisa berupa sebuah perangkat keras maupun perangkat lunak. Beberapa alternatif penyelesaian masalah perawatan AC telah didapatkan seperti yang dilakukan oleh Ilham dan Ramschie (2013), Diori (2019) dan Mukti (2020) telah merancang alat yang mampu mengetahui waktu perawatan dilakukan melalui temperatur dan kelembaban.

3.1.1 Kriteria Desain

Hal lain yang didapatkan dalam melakukan kajian pustaka adalah kriteria desain. Kriteria desain yang akan digunakan dalam perancangan ini adalah mampu membaca temperatur dari 16°C- 30 °C dan kelembaban dari 0% - 100% pada filter udara evaporator. Kriteria ini digunakan karena melihat alat yang dirancang oleh Mukti (2020) yang dirasa efektif untuk melihat kondisi filter udara.

Kriteria desain yang kedua adalah mampu mengirimkan hasil pembacaan temperatur dan kelembaban ke database dan antarmuka digital berupa aplikasi *website* menggunakan jaringan internet selama kurang lebih 3 jam. Pemilihan kriteria ini didasarkan oleh perancangan yang dilakukan Mukti (2020) dan Diori (2019) yang menggunakan antarmuka digital dalam perancangannya. Antarmuka berbasis aplikasi *website* ini memiliki kelebihan dalam hal mobilitas dibandingkan dengan antarmuka layar menggunakan LCD.

Untuk kriteria desain yang terakhir adalah kemampuan alat ini dalam membedakan data filter udara evaporator yang kotor dan bersih menggunakan model matematika pada rentang temperatur 16 °C - 30°C dan kelembaban dari 0% - 100% . Berdasarkan kriteria desain tersebut maka dipilih alternatif penyelesaian

dengan menggunakan komponen-komponen yang akan dijelaskan pada sub bab-bab berikutnya.

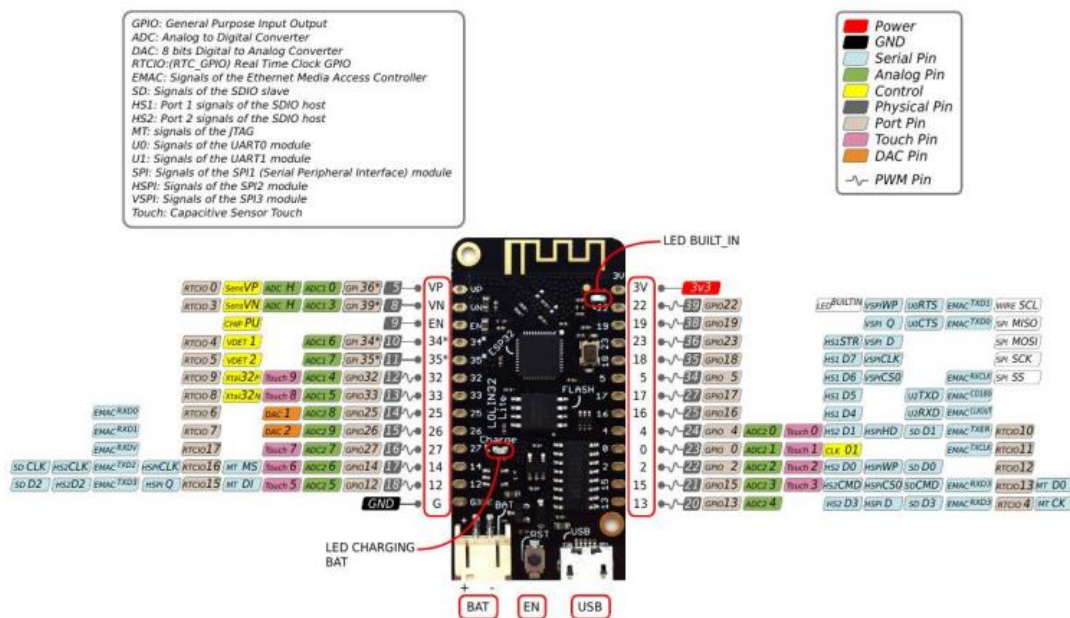
3.1.2 Pemilihan Komponen

Komponen yang digunakan dalam perancangan ini terdiri dari:

3.1.2.1 Pemilihan Mikrokontroler

Terdapat banyak pilihan mikrokontroler yang tersedia di pasaran. Sebelumnya modul komunikasi melalui internet terpisah dengan mikrokontrolernya. Namun saat ini telah hadir modul ESP32 maupun ESP8266 yang terhubung dengan mikrokontroler. ESP32 merupakan penerus dari ESP8266 dengan tambahan fitur Bluetooth. Banyak jenis-jenis mikrokontroler yang menggunakan modul ES32 dan ESP8266, contohnya ESP32 Wemos Lolin32 Lite, Node-Mcu, DOIT DEVKIT V1, ESP32 Thing dan lain-lainnya. Kelebihan dari mikrokontroler tersebut adalah bentuknya yang kecil dan ramping daripada mikrokontroler dengan modul ESP yang terpisah.

Untuk perancangan kali ini dipilih ESP32 Wemos Lolin Lite merupakan modul ESP32 yang dikeluarkan perusahaan Wemos. ESP32 sendiri merupakan *chip microcontroller* yang terintegrasi dengan Wi-Fi dan Bluetooth. ESP32 Wemos Lolin32 Lite memiliki tegangan kerja sebesar 3.3 V serta memiliki 19 antarmuka *input/output*.



Gambar 3-2 Antarmuka Lolin32.

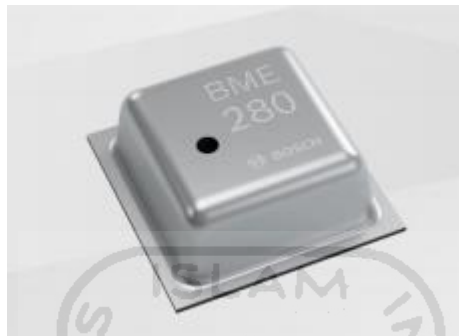
Sumber : de Lallana (2020)

Gambar 3-2 menunjukkan semua antarmuka pada ESP32 Wemos Lolin32 Lite, dari Gambar 3-2 terlihat antarmuka *micro USB*, JST 2.0, dan beberapa pin catu daya. Berdasarkan Gambar 3-2 diketahui pada setiap pin memiliki fungsi-fungsi lainnya seperti ADC (*Analog Digital Converter*), DAC (*Digital Analog Converter*), *Serial* dan fungsi-fungsi lainnya.

3.1.2.2 Pemilihan Sensor

Sensor yang akan dipilih harus mampu memenuhi kriteria desain “mampu membaca temperatur dari 16°C sampai 30 °C dan kelembaban dari 0% sampai 100%”. Sensor yang mampu membaca temperatur dan kelembaban sangatlah banyak seperti DHT11, DHT22, LM35, BMP180, dan BME280. Sensor – sensor tersebut memiliki kelebihan dan kekurangan masing-masing. BME280 menjadi pilihan dalam perancangan alat kali ini, karena hasil pembacaan sensor BME280 paling mendekati hasil pembacaan alat ukur yang telah terstandar. Hal ini dibuktikan oleh (Rocha dkk., 2019) dalam penelitiannya yang membandingkan pembacaan sensor DHT11, DHT22, LM35, BMP180, dan BME280 dengan alat ukur HMP45C.

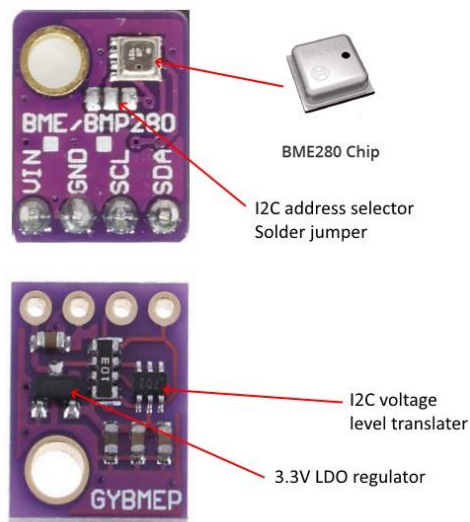
Pertimbangan lain dari pemilihan BME280 adalah daya tahan dari sensor tersebut dan BME280 memiliki resolusi paling kecil dalam pembacaan temperatur dibandingkan sensor lain seperti DHT11, DHT22, LM35, dan BMP180. BME280 mampu bertahan selama beberapa bulan dengan mempertahankan performanya (Stoev & Mutkov, 2018).



Gambar 3-3 BME-280.

Sumber : Sensortec (2014)

BME280 merupakan sensor temperatur, kelembaban dan tekanan yang dibuat oleh perusahaan Bosch. BME280 ini dapat digunakan dengan menggunakan protokol komunikasi I2C dan SPI. Gambar 3-3 menunjukkan bentuk fisik dari BME280. Sedangkan Gambar 3-4 menunjukkan bentuk fisik dari modul BME280. Penggunaan modul BME280 bertujuan untuk mempermudah perancangan perangkat keras, dikarenakan dalam modul BME280 telah dilengkapi dengan regulator catu daya dan antarmuka yang lebih besar daripada antarmuka BME280. Perancangan ini membutuhkan 2 sensor BME280.



Gambar 3-4 Modul BME-280.

Sumber : Adafruit Datasheet

3.1.2.3 Pemilihan Protokol Komunikasi dan Perangkat Lunak

Untuk menghubungkan BME280 dengan Wemos Lolin32 Lite digunakan protokol I2C, hal ini disebabkan karena protokol I2C membutuhkan lebih sedikit kabel dibandingkan protokol SPI. Keunggulan dari SPI adalah dapat mengirimkan data dari master ke slave dan slave ke master secara bersamaan. Perancangan ini tidak membutuhkan fitur tersebut, sehingga protokol I2C dianggap lebih cocok daripada SPI.

Selain komunikasi sensor dengan mikrokontroler terdapat komunikasi mikrokontroler dengan antarmuka dan database. Terdapat banyak sekali protokol komunikasi yang tersedia, contohnya MQTT, HTTP, HTTPS dan lain-lainnya. Untuk perancangan ini digunakan protokol komunikasi MQTT karena protokol ini dirancang untuk mengirimkan data yang sangat ringan. Data yang diambil berupa besar pembacaan temperatur dan kelembaban memiliki beban yang ringan. Untuk mempermudah perancangan komunikasi menggunakan MQTT digunakan aplikasi web Node-Red dan database MySQL. Node-Red ini dipilih karena untuk menggunakannya sangat mudah. Tidak perlu mempelajari bahasa pemrograman yang rumit dan antarmuka yang mudah dipahami menjadi kelebihan dari Node-Red. Selain itu fitur-fitur untuk mengaplikasikan IoT sangatlah mudah dan banyak.

Selain komponen-komponen yang telah disebutkan terdapat beberapa komponen pendukung lainnya, seperti:

1. Terminal Blok.
2. Pin Header.
3. Solder.
4. Timah Solder.
5. Karton Bekas.
6. Selotip Bermata Dua.
7. Laptop.
8. Kabel Mikro USB.
9. PCB Lubang.
10. Kabel Jumper.

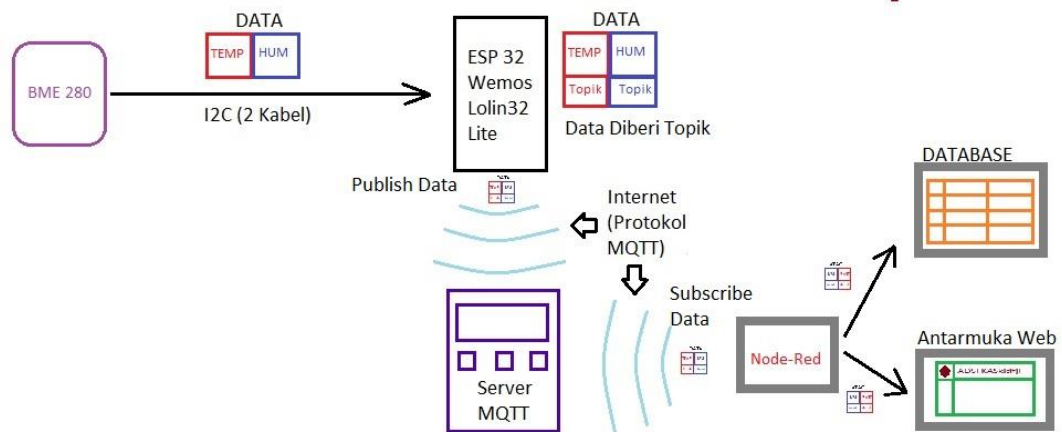
3.2 Perancangan

Proses perancangan dilakukan menjadi dua jenis yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras dilakukan untuk menghubungkan BME280 dengan ESP32 Wemos Lolin Lite serta menghubungkan ESP32 Wemos Lolin Lite dengan catu daya. Perancangan selanjutnya adalah perancangan perangkat lunak yang bertujuan untuk membaca temperatur dan kelembaban udara yang masuk dan keluar dari evaporator dan kondensor AC rumah. Perancangan perangkat lunak juga bertujuan untuk mengirim data dari ESP32 Wemos Lolin Lite menuju pangkalan data dan menampilkan kondisi AC pada antarmuka.

3.2.1 Perancangan Komunikasi

Berdasarkan Gambar 3-5 terdapat dua perancangan komunikasi yang dilakukan dalam perancangan ini. Pertama adalah perancangan komunikasi I2C dan yang kedua adalah perancangan komunikasi MQTT. Perancangan komunikasi I2C ini digunakan untuk menghubungkan mikrokontroler dengan sensor. Pada mikrokontroler digunakan pin 23 dan 19 untuk menghubungkan dengan sensor masukan evaporator. Pin 23 dihubungkan dengan pin SDA pada sensor dan pin 19 dihubungkan dengan pin SCL pada sensor. Untuk sensor keluaran evaporator

digunakan pin 33 yang terhubung dengan SDA dan pin 32 yang terhubung dengan SCL.

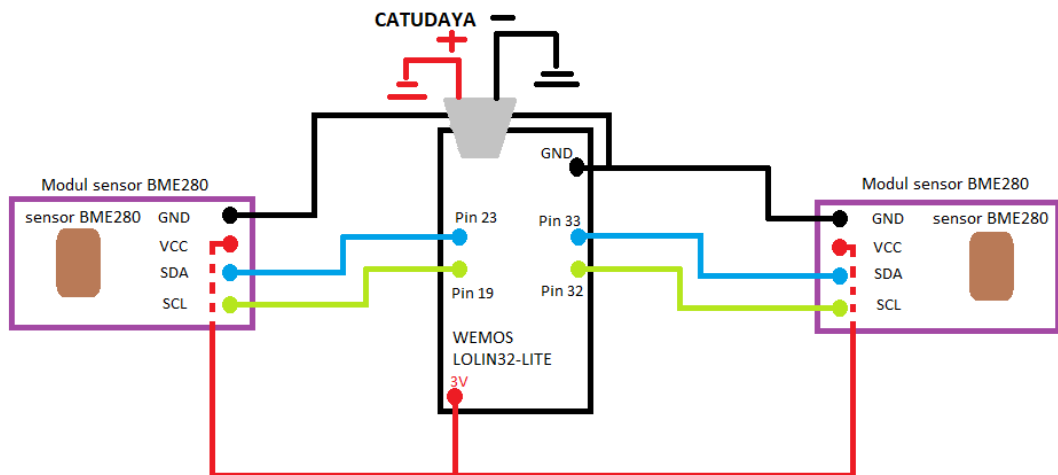


Gambar 3-5 Ringkasan Perancangan Komunikasi.

Perancangan komunikasi yang kedua adalah perancangan komunikasi MQTT. MQTT digunakan untuk mengirimkan data ke server sehingga data sensor dapat diambil oleh Node-Red untuk ditampilkan pada antarmuka dan disimpan dalam pangkalan data. Server yang digunakan memiliki alamat internet mousquito.org. Data sensor tersebut akan di **publish** menggunakan topik yang berbeda-beda. Topik yang digunakan pada data masukan evaporator adalah **inEva/temp** dan **inEva/hum** sedangkan untuk topik keluaran evaporator digunakan **outEva/temp** dan **outEva/hum** seperti yang ditunjukkan pada Lampiran 1- 5-4 dan Lampiran 1- 5-5. Sedangkan untuk mengirimkan kondisi filter udara AC digunakan topik **Eva/AC**.

3.2.2 Perancangan Perangkat Keras

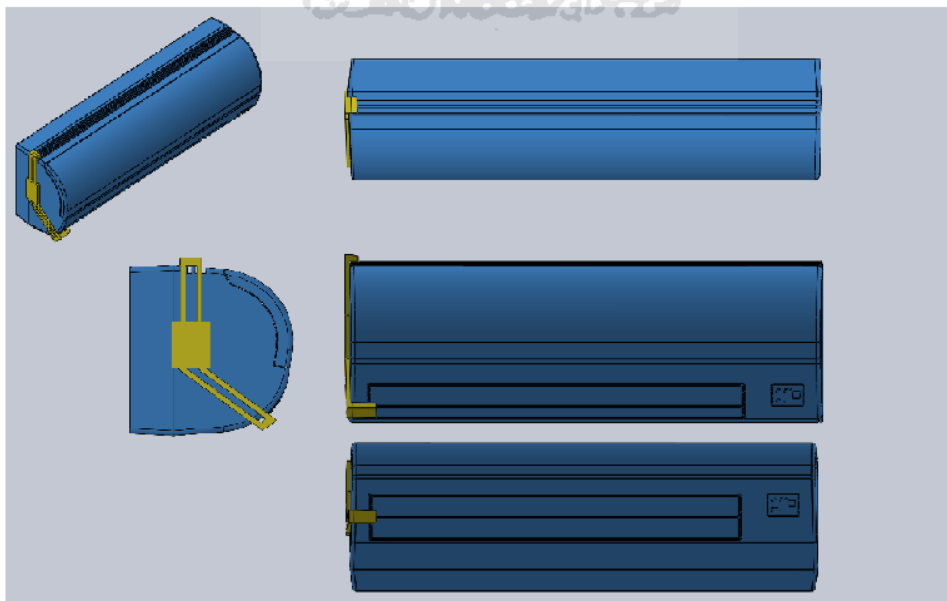
Langkah kedua adalah menghubungkan ESP32 dengan Modul BME280. Modul BME280 yang digunakan, memiliki protokol komunikasi 12C dan SPI. Sedangkan pada perancangan ini digunakan protokol 12C, menggunakan lebih sedikit kabel daripada SPI, untuk jenis pin yang digunakan terlihat pada Gambar 3-6. Langkah kedua adalah menghubungkan ESP32 dengan catu daya. Gambar 3-6 juga memperlihatkan skema rangkaian yang digunakan untuk menghubungkan BME280 dan catu daya dengan ESP32/Wemos Lolin32-Lite



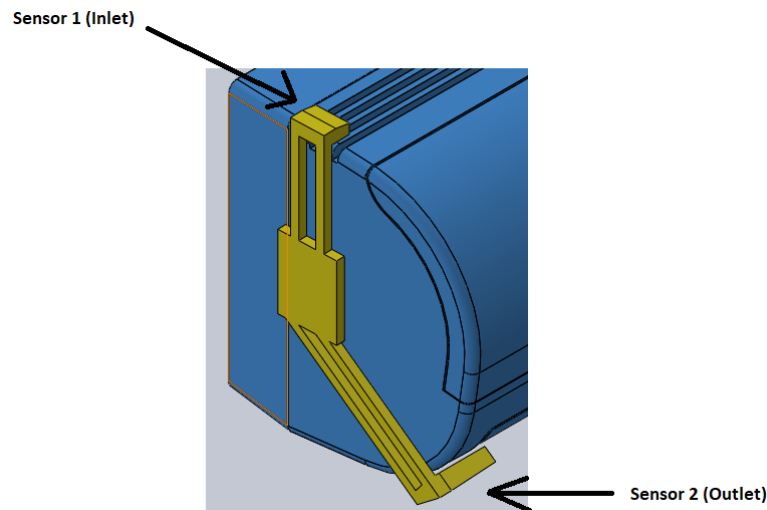
Gambar 3-6 Skema Rangkaian Alat Pemantau.

Selain itu untuk menghubungkan sensor BME280 dengan mikrokontroler dibutuhkan rangkaian *pull-up* untuk menghubungkan SDA dan SCL, rangkaian *pull-up* tersebut sudah dirangkai menjadi satu modul BME280. Sehingga dalam Gambar 3-6 tidak digambarkan.

Langkah kedua adalah menentukan posisi alat akan dipasang pada evaporator AC. Posisi yang dipilih berada di samping evaporator AC. Posisi tersebut dipilih berdasarkan kondisi keluaran evaporator AC yang akan diamati memiliki aliran udara yang lebih besar pada bagian samping evaporator. Bagian samping yang dipilih dapat dilihat pada Gambar 3-8 . Untuk Gambar 3-9 memperlihatkan posisi sensor 1 dan sensor 2 pada alat.



Gambar 3-7 Posisi Alat.

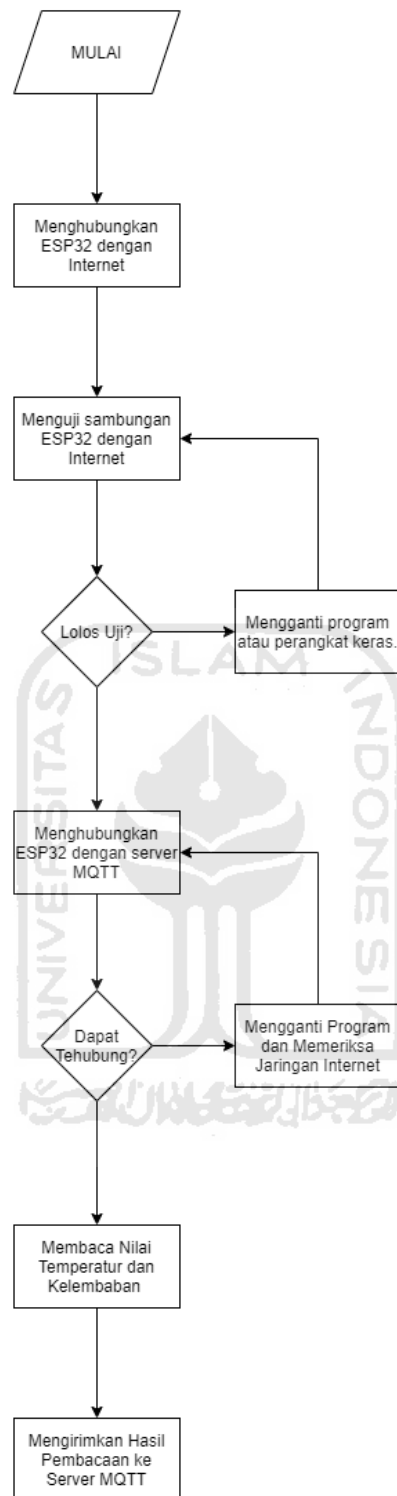


Gambar 3-8 Posisi Sensor Pada Alat.

3.2.3 Perancangan Perangkat Lunak

Secara garis besar terdapat tiga perangkat lunak yang akan digunakan dalam perancangan ini. Ketiga perangkat lunak tersebut adalah Arduino IDE, Node-Red, dan XAMPP. Hubungan dari ketiga perangkat lunak tersebut dalam perancangan ini dapat dilihat pada Gambar 3-9.

Penulisan naskah program pertama dilakukan pada Arduino IDE. menunjukkan bahwa Arduino IDE berfungsi untuk menerjemahkan naskah program yang telah ditulis menjadi bahasa yang dikenali oleh komputer. Alur keseluruhan perancangan perangkat lunak terlihat pada Gambar 3-9.

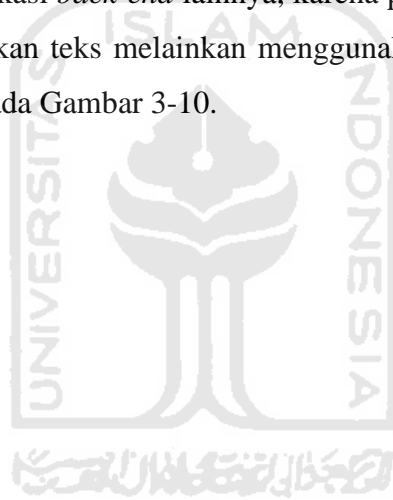


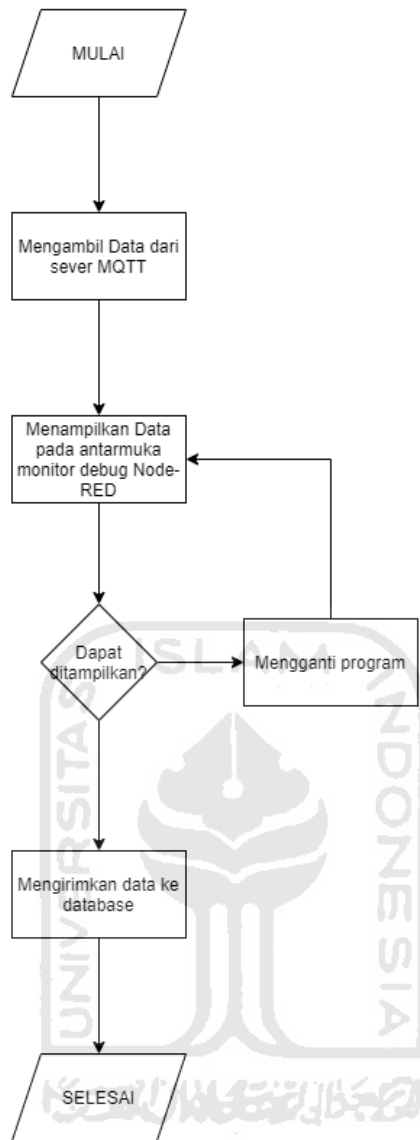
Gambar 3-9 Alur Perancangan Perangkat Lunak.

Untuk pembuatan protokol MQTT digunakan server MQTT mousquito dengan alamat server test.mosquitto.org dengan nomor port 1883. Penggunaan

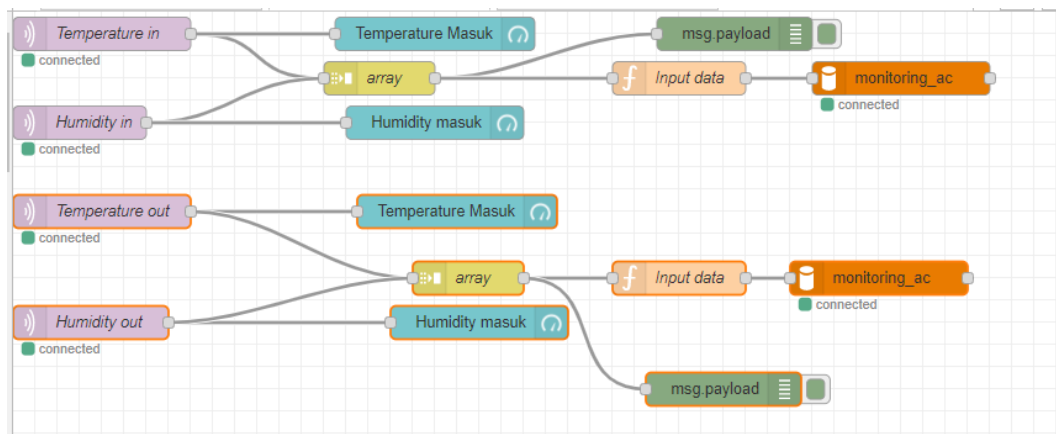
port 1883 memiliki arti data tidak terenkripsi dan tidak memiliki *authentication* (tanpa *password* dan *username*) pada proses pengiriman data. Terdapat beberapa port dengan fitur enkripsi dan *authentication*, seperti port 1884, 8883, 8884 dan port lain-lainya.

Penulisan naskah program kedua dilakukan pada perangkat lunak Node-red. Node-Red merupakan aplikasi web yang dibuat menggunakan framework Node.js yang dirancang dengan bahasa pemrograman Javascript. Node-Red dirancang untuk mempermudah pembuatan *back-end* maupun *front-end* dalam sebuah *website*. Contoh pekerjaan *back-end* adalah menghubungkan antarmuka *website* dengan pangkalan data, sedangkan contoh pekerjaan *front-end* adalah membuat tampilan antarmuka. Penggunaan Node-Red sendiri dirasa sangat mudah dibandingkan dengan aplikasi *back-end* lainnya, karena proses pembuatan naskah program tidak menggunakan teks melainkan menggunakan blok. Alur program Node-Red dapat dilihat pada Gambar 3-10.





Gambar 3-10 Alur Program Node-Red.

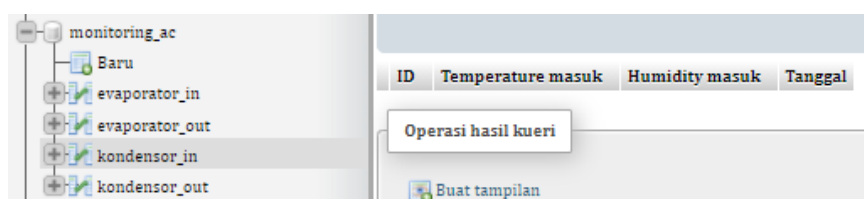


Gambar 3-11 Program Node-Red.

Berdasarkan Gambar 3-11 diketahui beberapa blok dengan warna yang berbeda-beda. Blok dengan warna ungu merupakan blok yang berfungsi untuk mengambil data dari server MQTT, kemudian blok berwarna biru berfungsi untuk menampilkan data pada antarmuka. Selanjutnya blok berwarna coklat terang berfungsi untuk menggabungkan data dalam bentuk array yang kemudian diambil oleh blok berwarna hijau untuk ditampilkan pada debug Node-Red untuk mengetahui apakah data yang diambil dari server sudah sesuai atau belum.

Selain diambil oleh blok debug (hijau) data array juga diambil oleh blok fungsi yang berwarna krem untuk dieksekusi dalam naskah program Javascript yang kemudian naskah program tersebut dikirim ke blok berwarna oranye dan dimasukkan ke dalam pangkalan data SQL yang memiliki alamat IP tertentu. Untuk pengaturan dari setiap blok dapat dilihat pada Lampiran 2- 1 sampai Lampiran 2- 4. Setiap blok memiliki pengaturan yang berbeda, terkadang terdapat blok yang perlu menuliskan program dalam bentuk teks, seperti blok fungsi.

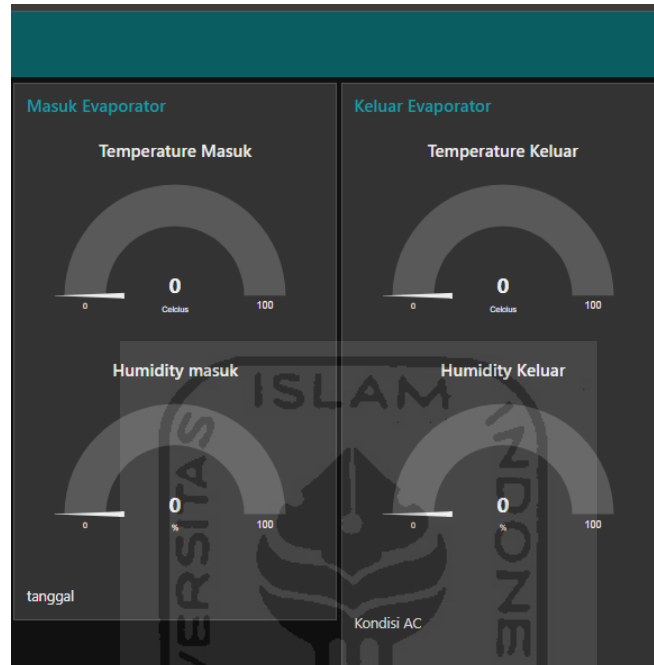
Perancangan perangkat lunak yang terakhir adalah membuat pangkalan data pada laptop. Untuk pembuatan pangkalan data digunakan perangkat lunak XAMPP. XAMPP sendiri berfungsi untuk mengubah komputer pribadi menjadi sebuah komputer server, dengan beberapa fitur layanan. Fitur dari XAMPP terlihat ditunjukkan oleh Gambar 3-12. Namun penggunaan XAMPP pada perancangan kali ini hanya membutuhkan fitur Apache dan MySQL. Apache merupakan fitur yang dapat mengubah komputer atau laptop pribadi menjadi sebuah komputer server. Fitur selanjutnya adalah fitur MySQL, fitur tersebut memungkinkan untuk membuat sebuah pangkalan data dimana dalam satu database dapat dibuat beberapa tabel dan dalam satu tabel terdapat beberapa kolom.



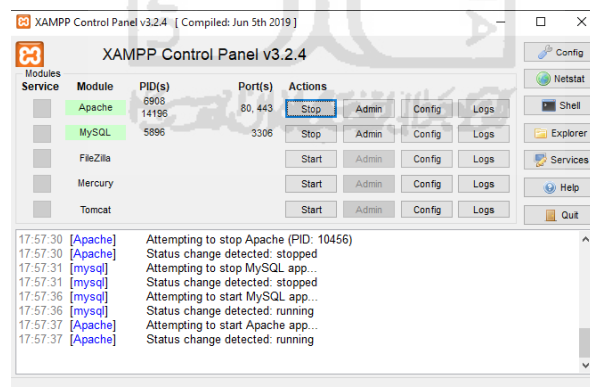
Gambar 3-12 Hasil Pembuatan Pangkalan Data.

Pangkalan data yang dibuat dalam perancangan ini diberi nama **monitoring_ac** dan tabel-tabel yang dibuat diberi nama **evaporator_in**, **evaporator_out**, **kondensor_in** dan **kondensor_out**. Gambar 3-12 menunjukkan

pembuatan pangkalan data dan tabel-tabelnya serta beberapa kolom di satu tabel yang dibuat menggunakan antarmuka *phpMyadmin*. Antarmuka tersebut dijalankan menggunakan aplikasi XAMPP dengan fitur-fitur yang ditunjukkan pada Gambar 3-14. Kemudian untuk perancangan antarmuka Node-Red ditunjukkan pada Gambar 3-13.



Gambar 3-13 Perancangan Antarmuka.



Gambar 3-14 Fitur XAMPP.

Untuk penjelasan lebih lanjut mengenai perancangan perangkat lunak telah ditunjukkan dalam BAB Lampiran. BAB Lampiran akan menjelaskan bagaimana cara kerja program Arduino baik untuk evaporator maupun kondensor, selain itu telah dijelaskan dalam lampiran pengaturan blok-blok yang digunakan pada Node-Red. Proses pembuatan pangkalan data juga dapat dilihat pada BAB Lampiran.

3.3 Pengujian

Perancangan ini bertujuan untuk membuat alat dapat bekerja sesuai dengan yang diinginkan. Pengujian ini dilakukan pada kondisi perangkat keras maupun perangkat lunak. Perangkat keras menjadi hal yang pertama kali diuji dengan melihat setiap sambungan kabel sudah terhubung atau belum.

Pengujian selanjutnya adalah pengujian perangkat lunak, dalam pengujian perangkat lunak ini akan diuji komunikasi antara sensor dan mikrokontroler, serta komunikasi mikrokontroler dengan server menggunakan protokol MQTT, selain itu juga akan dilihat bagaimana sensor mampu membaca kondisi lingkungan.

Bentuk validasi yang digunakan dalam pengujian perangkat keras adalah hasil pembacaan sensor, sedangkan bentuk validasi pengujian perangkat lunak adalah hasil pengiriman data. Apabila hasil pembacaan sensor dapat terlihat pada *serial monitor* Arduino IDE dan menunjukkan nilai yang sesuai, maka perangkat keras siap untuk digunakan. Kemudian apabila hasil pengiriman data ke antarmuka dan pangkalan data telah mampu dilakukan, maka perangkat lunak telah siap digunakan.

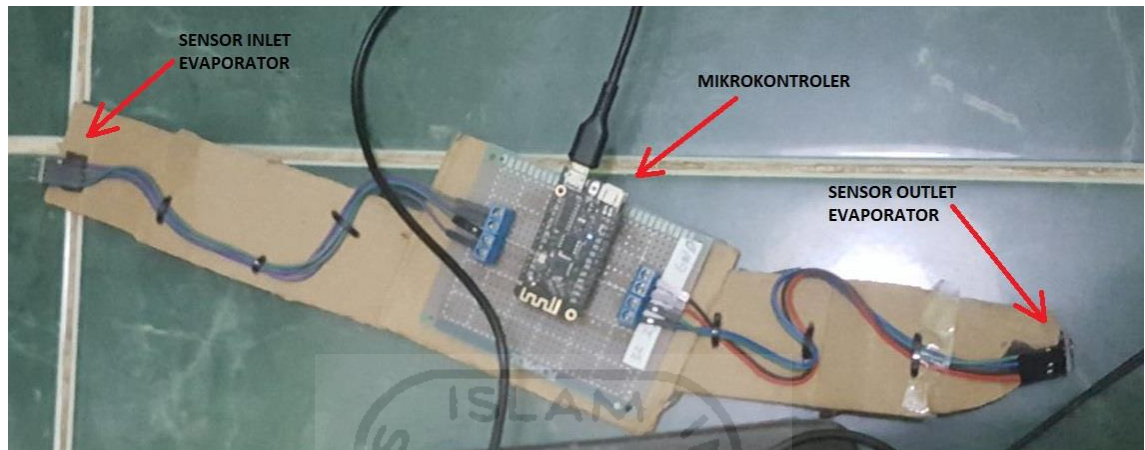
Setelah melakukan pengujian perangkat keras maupun lunak dan telah diperoleh data kondisi filter udara dengan dua kondisi maka dapat dilakukan pengujian statistika untuk mengetahui apakah terdapat perbedaan pada kondisi filter udara bersih dengan filter udara kotor. Metode pengujian statistika yang dipilih adalah Wilcoxon Rank Sum Test. Metode tersebut digunakan untuk mempermudah menentukan bentuk dan parameter validasi. Validasi yang didapatkan akan menunjukkan adanya perbedaan data atau tidak.

Pengujian yang terakhir adalah pengujian model matematika, setelah diketahui adanya perbedaan pada data bersih dan kotor dilakukan pemodelan kondisi filter udara menggunakan metode Regresi Logistik. Model matematika yang didapatkan kemudian diuji pada kondisi filter sedikit kotor dan kondisi filter sedikit bersih. Validasi dilakukan dengan melihat pesan yang dikirimkan ke antarmuka. Apabila pesan yang dikirim sesuai dengan kondisi filter udara maka dapat dikatakan model matematika telah mewakili kondisi filter udara.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan



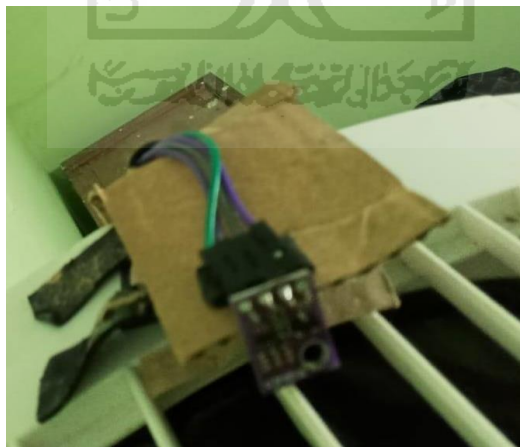
Gambar 4-1 Hasil Perancangan Perangkat Keras.

Hasil perancangan kali ini terbagi menjadi dua bagian, yaitu bagian pembacaan temperatur dan kelembaban, serta bagian hasil perancangan sistem IoT. Untuk hasil perancangan perangkat keras seperti pada Gambar 3-6 dapat dilihat pada Gambar 4-1. Berdasarkan Gambar 4-1 dapat diketahui bahwa sumber tenaga menggunakan kabel USB (*Universal Serial Bus*). Hal ini disebabkan port JST 2.0 yang tersedia pada Wemos Lolin32 -Lite tidak dapat berfungsi.

Untuk pemasangan alat pemantau pada evaporator dapat dilihat pada Gambar 4-2, Pemasangan alat diletakan di sebelah kiri evaporator, hal ini disebabkan agar alat tidak menutupi lubang keluaran evaporator apabila diletakan di tengah. Peletakan juga tidak dilakukan di sebelah kanan evaporator, karena membutuhkan lebih banyak *base* (kardus). Sehingga peletakan alat di sebelah kiri dirasa cukup efektif. Untuk posisi kedua sensor ditunjukan oleh Gambar 4-3 dan Gambar 4-4. Gambar 4-3 menunjukkan posisi sensor masukan evaporator dan Gambar 4-4 menunjukkan posisi sensor keluaran evaporator.



Gambar 4-2 Peletakan Alat Pemantau.



Gambar 4-3 Sensor Masukan Evaporator.



Gambar 4-4 Sensor Keluaran Evaporator.

4.1.1 Hasil Pembacaan Sensor

Hasil pembacaan sensor terdapat dua variabel yaitu, temperatur dan kelembaban. Untuk dapat mengkomunikasikan sensor BME280 dengan Wemos Lolin32-Lite digunakan protokol komunikasi 12C sehingga membutuhkan 8 kabel untuk 2 sensor seperti yang ditunjukkan pada Gambar 4-1. Untuk hasil pembacaan sensor dan hasil penyambungan Wemos-Lolin32 ke internet dapat dilihat pada Gambar 4-5. Sedangkan untuk naskah program pada Arduino IDE dapat dilihat pada Lampiran 1- 5-1. Berdasarkan Gambar 4-5 dapat diketahui bahwa Wemos Lolin32-Lite telah mampu berkomunikasi dengan server MQTT.

```
COM4
WiFi connected
IP address:
192.168.2.158
BME280 1 test
BME280 2 test
-- Test OKE --

Attempting MQTT connection...connected
-----SENSOR 1-----
Temperature = 30.10 *C
Humidity = 76.37 %

-----SENSOR 2-----
Temperature = 29.26 *C
Humidity = 75.23 %

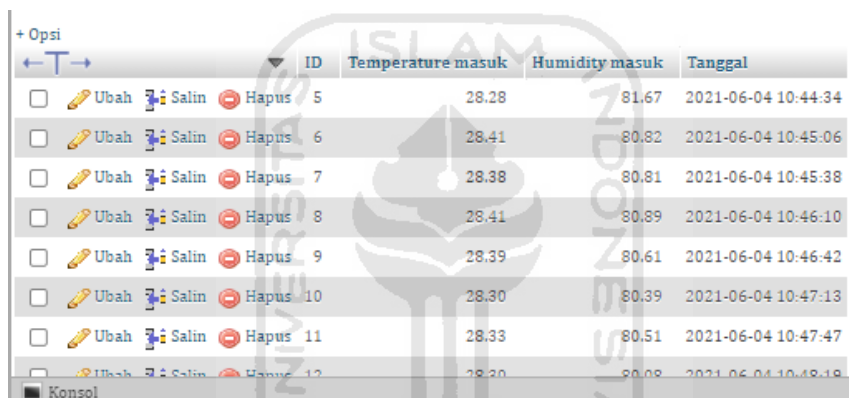
 Scroll automatic  Show timestamp
```

Gambar 4-5 Hasil Pembacaan Sensor. .

4.1.2 Hasil Pengiriman Data

Hasil pengiriman data dari Arduino IDE ke server MQTT Mousquito dibuktikan dengan masuknya data ke database dan tampil-nya data di antarmuka. Topik yang digunakan pada data masukan evaporator adalah **inEva/temp** dan **inEva/hum** sedangkan untuk topik keluaran evaporator digunakan **outEva/temp** dan **outEva/hum** seperti yang ditunjukkan pada Lampiran 1- 5-4 dan Lampiran 1-5-5.

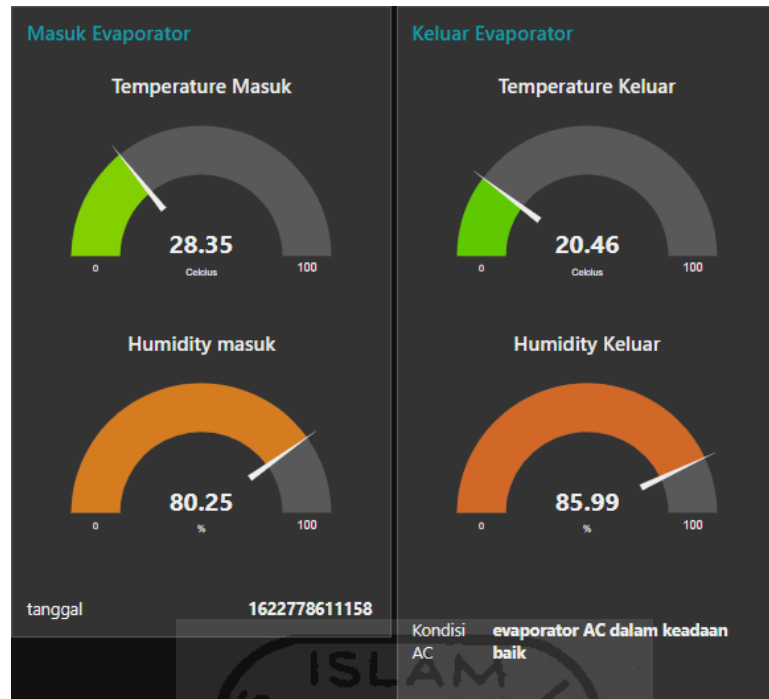
Hasil dari penulisan program pada Arduino IDE dan Node-Red adalah masuknya data ke dalam pangkalan data yang bernama **monitoring_ac** seperti yang ditunjukkan pada Gambar 4-6. Gambar 4-6 juga menunjukkan tabel-tabel yang ada di dalam pangkalan data **monitoring_ac**.



ID	Temperature masuk	Humidity masuk	Tanggal
5	28.28	81.67	2021-06-04 10:44:34
6	28.41	80.82	2021-06-04 10:45:06
7	28.38	80.81	2021-06-04 10:45:38
8	28.41	80.89	2021-06-04 10:46:10
9	28.39	80.61	2021-06-04 10:46:42
10	28.30	80.39	2021-06-04 10:47:13
11	28.33	80.51	2021-06-04 10:47:47
12	28.20	80.08	2021-06-04 10:48:18

Gambar 4-6 Pangkalan data Pengujian.

Data yang telah dikirim ke Node-Red dan pangkalan data juga dikirimkan ke antarmuka. Setiap pangkalan data terdapat beberapa tabel seperti yang ditunjukkan pada Gambar 4-6 tabel tersebut bernama evaporator_in, evaporator out dan lain-lainnya. Setiap tabel berisikan 4 kolom IDE, temperatur, kelembaban, dan tanggal. Sedangkan untuk hasil perancangan antarmuka ditunjukkan oleh Gambar 4-7.



Gambar 4-7 Hasil Pengiriman Data Menuju Antarmuka.

4.2 Hasil Pengukuran

Pengujian yang dilakukan bertujuan untuk mengetahui perbedaan dan karakteristik dari filter udara evaporator dengan kondisi bersih dan kotor. Untuk mengetahui perbedaan kondisi tersebut dilakukan pengukuran temperatur dan kelembaban pada masukan dan keluaran evaporator. Hasil pengukuran tersebut dimasukkan ke dalam pangkalan data **monitoring_ac**. Data dari pangkalan data kemudian dipindahkan ke Microsoft Excel untuk kebutuhan analisis.

Pengukuran dilakukan dengan perbedaan kondisi filter udara evaporator yang ditunjukkan oleh Gambar 4-8 dan Gambar 4-9. Gambar 4-8 merupakan kondisi filter udara kotor dan Gambar 4-9 merupakan kondisi filter udara setelah dibersihkan. Pengukuran dilakukan dengan kondisi filter udara kotor terlebih dahulu baru dilanjutkan dengan kondisi filter udara bersih. Untuk jarak pengukuran kondisi bersih dan kotor sekitar 5 hari seperti yang di informasikan pada Lampiran 3- 1 Tabel Pengukuran Filter Udara Kotor dan Lampiran 3- 2 Tabel Pengukuran Filter Udara Bersih.



Gambar 4-8 Kondisi Filter Udara Kotor. .



Gambar 4-9 Kondisi Filter Udara Bersih. .

Hasil pengukuran temperatur dan kelembaban pada kondisi filter udara kotor ditunjukkan Lampiran 3-1 Tabel Pengukuran Filter Udara Kotor dan untuk kondisi filter udara bersih ditunjukkan pada Lampiran 3-2 Tabel Pengukuran Filter Udara Bersih. Data-data tersebut diambil pada pukul 09.00 sampai 11.00. Setiap kondisi filter udara dilakukan pengambilan data selama dua jam setiap 30 detik sekali. Seharusnya hasil pengukuran yang didapatkan berjumlah 60 pada setiap kondisi filter udara, namun terjadi *delay* saat data dikirim ke server MQTT melalui internet, akibatnya hanya ada 57 data dalam satu kondisi filter udara.

4.3 Analisis

Langkah pertama adalah digunakan diagram psikrometrik untuk menggambarkan perbedaan selisih temperatur dan kelembaban saat kondisi filter udara bersih dan kotor. Untuk hasil diagram psikrometrik dapat dilihat pada Gambar 4-10. Berdasarkan Gambar 4-10 tersebut kita dapat mengetahui terjadi proses penurunan temperatur dan kenaikan kelembaban pada kedua grafik. Melihat kedua grafik tersebut dapat diketahui bahwa proses pendinginan pada kondisi filter udara bersih lebih besar, daripada kondisi filter udara kotor.

Hasil yang ditunjukkan pada diagram psikrometri perlu dilakukan pengujian secara statistika. Hal ini dilakukan untuk melihat lebih yakin terhadap perbedaan kondisi filter udara. Pengujian statistika akan digunakan metode Wilcoxon Rank Sum Test. Untuk hasil pengujian statistika akan dijelaskan pada sub-bab berikutnya.

Setelah diketahui perbedaan kondisi kotor dan kondisi bersih maka dilakukan pemodelan kondisi filter udara bersih dan kotor. Model tersebut nantinya akan dimasukkan kedalam naskah program Arduino untuk mengirimkan peringatan kondisi filter udara ke antarmuka. Pemodelan tersebut dilakukan dengan metode Logistik Regresi.

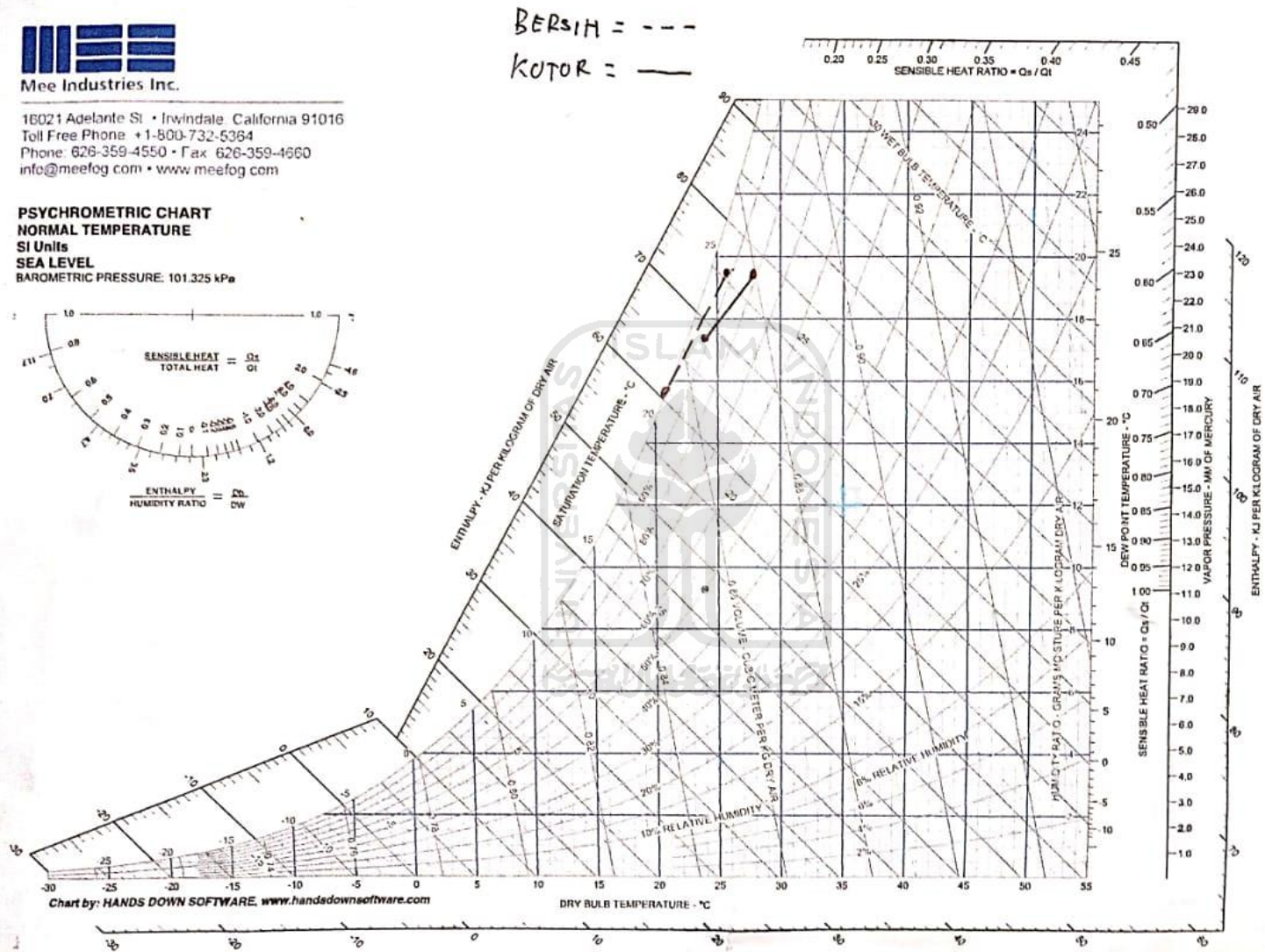
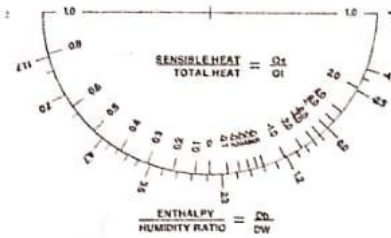
Secara ringkas analisis dan pembahasan yang dilakukan bertujuan mengetahui perbedaan variabel temperatur dan kelembaban pada kondisi filter udara bersih dan kotor. Kemudian berdasarkan perbedaan tersebut dilakukan pemodelan untuk setiap kondisi filter udara. Pemodelan dilakukan dalam bentuk matematika agar model tersebut dapat dimasukkan ke naskah program Arduino. Sehingga Wemos Lolin 32-Lite dapat mengirimkan peringatan perawatan AC ke antarmuka digital.

Untuk menyakinkan perbedaan kondisi filter udara dari setiap variabel, maka dilakukan analisis Wilcoxon Rank Sum-Test. Metode ini dipilih karena populasi dari data pada Lampiran 3- 1 dan Lampiran 3- 2 memiliki distribusi tidak normal baik pada variabel temperatur maupun kelembaban.



16021 Adelante St • Irwindale, California 91016
Toll Free Phone +1-800-732-5364
Phone 626-359-4550 • Fax 626-359-4660
info@meefog.com • www.meefog.com

PSYCHROMETRIC CHART
NORMAL TEMPERATURE
SI Units
SEA LEVEL
BAROMETRIC PRESSURE: 101.325 kPa



Gambar 4-10 Grafik Psikrometri.

Metode ini digunakan untuk membandingkan selisih temperatur ($T_{masuk} - T_{keluar}$) kondisi bersih dengan selisih temperatur ($T_{masuk} - T_{keluar}$) kondisi kotor. Untuk selisih kelembaban ($H_{masuk} - H_{keluar}$) kondisi bersih dibandingkan dengan selisih kelembaban ($H_{masuk} - H_{keluar}$) kondisi kotor.

Tabel 4-1 Sampel Selisih Temperatur Evaporator.

Selisih Sampel Temperatur Kotor	Selisih Sampel Temperatur Bersih
1,11	5,14
2,39	5,42
3,65	5,42
3,66	5,43
3,76	5,48
3,91	5,64
4,28	5,69
4,58	5,7
4,74	5,71
4,74	5,72
4,96	5,73
5,01	5,75

Tabel 4-2 Sampel Selisih Kelembaban Evaporator.

Selisih Sampel Kelembaban Kotor	Selisih Sampel Kelembaban Bersih
0,52	8,36
2,04	8,46
3,4	8,48
4,33	8,59
5,02	8,68
5,32	8,85
6,48	8,9
7,63	9,13
8,05	9,51
8,11	9,65
8,2	9,84
8,68	10,61

Metode ini menggunakan *independent random sample (n)*, untuk itu telah didapatkan nilai *independent random sample (n)* untuk populasi selisih temperatur,

yang dapat dilihat pada Tabel 4-1, kemudian untuk sampel selisih kelembaban ditunjukkan pada Tabel 4-2. Untuk hipotesis dalam analisis ini H0 didefinisikan sebagai “sama” dan H1 didefinisikan sebagai “berbeda”. Data -data pada tabel Tabel 4-1 dan Tabel 4-2 diambil secara acak kemudian diurutkan dari yang terkecil sampai terbesar. Sehingga akan terdapat 24 ranking untuk setiap parameter temperatur dan kelembaban, seperti yang ditunjukkan pada

Tabel 4-3 Tabel Ranking Temperatur.

Sample	Rank
1,11	1
2,39	2
3,65	3
3,66	4
3,76	5
3,91	6
4,28	7
4,58	8
4,74	9,5
4,74	9,5
4,96	11
5,01	12
5,14	13
5,42	14,5
5,42	14,5
5,43	16
5,48	17
5,64	18
5,69	19
5,7	20
5,71	21
5,72	22
5,73	23
5,75	24

Tabel 4-4 Tabel Ranking Kelembaban.

Sample	Rank
2,04	1
4,33	2
5,02	3
5,32	4
6,48	5
7,63	6
8,05	7
8,11	8
8,36	9
8,46	10
8,48	11
8,59	12
8,68	13,5
8,68	13,5
8,85	15
8,9	16
9,13	17
9,49	18
9,51	19
9,65	20
9,84	21
10,61	22
12,42	23
14,28	24

Ranking-ranking tersebut kemudian akan digunakan untuk melakukan analisis menggunakan metode Wilcoxon Rank Sum-Test. Hasil analisis ini dirangkum pada Tabel 4-5 dan Tabel 4-6.

Tabel 4-5 Hasil Analisis Perbedaan Selisih Temperatur Evaporator.

n_1	12
n_2	12
T	267
μT	150
$\sum t_j x(t_j^2 - 1)$	12
$\sigma^2 t$	300
σt	17,32
z	6,75

Tabel 4-6 Hasil Analisis Perbedaan Selisih Kelembaban Evaporator.

n_1	12
n_2	12
T	185,5
μT	150
$\sum t_j x(t_j^2 - 1)$	6
$\sigma^2 t$	300
σt	17,32
z	2,05

Kedua tabel Tabel 4-5 dan Tabel 4-6 menunjukkan nilai Z yang menjadi acuan untuk menentukan hipotesis mana yang akan diterima. Nilai Z akan dibandingkan dengan nilai $z_{\alpha/2}$ yang terlihat pada Gambar 2-10 Tabel *Corresponding z-Value*. Karena nilai $\frac{\alpha}{2}$ yang digunakan adalah 0,05, maka nilai $z_{\alpha/2}$ yang dipilih adalah 1,645. Apabila nilai z pada Tabel 4-5 dan Tabel 4-6 lebih dari $z_{\alpha/2}$ maka bisa dikatakan kedua data 95% berbeda (H_0 ditolak), namun apabila nilai z kurang dari $z_{\alpha/2}$ bisa dikatakan kedua data 95% sama (H_0 diterima). Berdasarkan perbandingan antara hasil pembahasan dengan hipotesis maka dapat disimpulkan bahwa H_0 ditolak atau tidak dapat diterima sehingga data dapat dikatakan 95% berbeda. Untuk langkah-langkah dalam perhitungan analisis ini dapat dilihat pada Lampiran 4- 1.

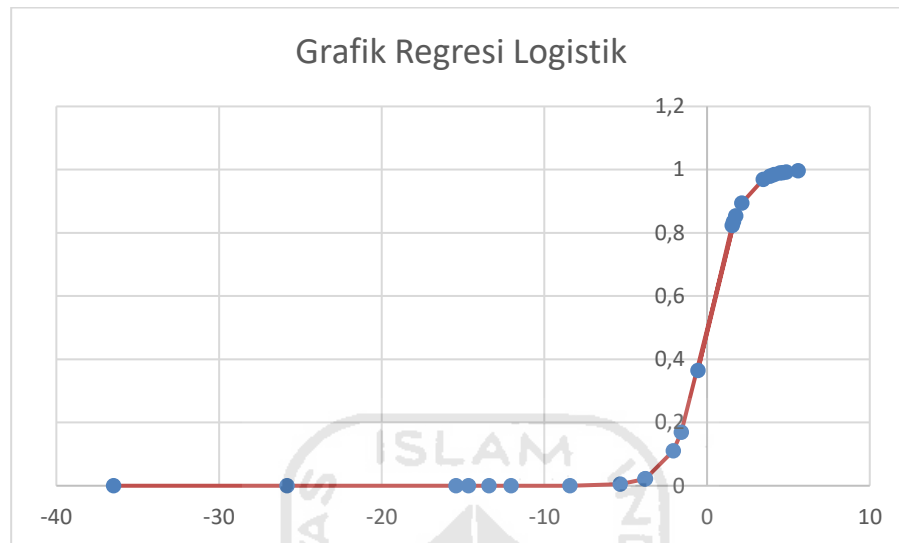
Setelah kedua kondisi dari filter udara bersih dan kotor dipastikan berbeda maka pemodelan persamaan matematika dapat dilakukan. Pemodelan ini menggunakan metode logistik regresi dengan data sampel yang digunakan sama seperti data yang ditunjukkan pada Tabel 4-5 dan Tabel 4-6. Hasil dari pemodelan logistik regresi ditunjukkan oleh persamaan (1. 9):

$$y = p(x) = \frac{e^x}{(1+e^x)} \text{ dengan nilai ,} \quad (1. 9)$$

$$x = -45,12 + 7,43x_1 + 0,75x_2$$

Nilai -45.12, 7.43 dan 0.75 didapatkan dari perhitungan menggunakan fitur *Solver* pada aplikasi excel, dengan menggunakan parameter, nilai variabel-variabel regresi normal berganda. Variabel x_1 untuk perancangan ini adalah nilai dari selisih

temperatur masuk dan keluar. Sedangkan x_2 merupakan nilai selisih kelembaban masuk dan keluar. Hasil hubungan antara persamaan tersebut dengan data-data yang ditunjukkan oleh Tabel 4-1 dan Tabel 4-2 ditunjukkan oleh Gambar 4-11. Untuk cara perhitungan detailnya dapat dilihat pada Lampiran 4- 2.



Gambar 4-11 Grafik Hubungan Probabilitas dengan Data Sampel.

Berdasarkan Gambar 4-11 terdapat kemungkinan yang diberi nilai 0 dan 1. Nilai 0 menunjukkan kemungkinan kondisi filter udara evaporator kotor, sedangkan nilai 1 menunjukkan kemungkinan kondisi filter udara evaporator bersih. Pada Gambar 4-11 nilai 0 dan 1 terletak pada sumbu Y. Sedangkan nilai yang terletak pada sumbu X merupakan nilai skor (x) yang berasal dari persamaan (1. 10) :

$$x = -45,12 + 7,43x_1 + 0,75x_2 \quad (1. 10)$$

Sebelum persamaan di masukan ke dalam program, model matematika tersebut diuji terlebih dahulu. Pengujian dilakukan dengan dua tahap pertama adalah memasukan nilai variabel x_1 dan x_2 dengan data-data pada Lampiran 3- 1 dan Lampiran 3- 2 . Apabila nilai kemungkinan yang muncul melebihi nilai 1 atau kurang dari 0 maka persamaan tersebut perlu dilakukan revisi. Pengujian ini juga berfungsi untuk mengetahui nilai tengah antara kondisi bersih dengan kondisi kotor.

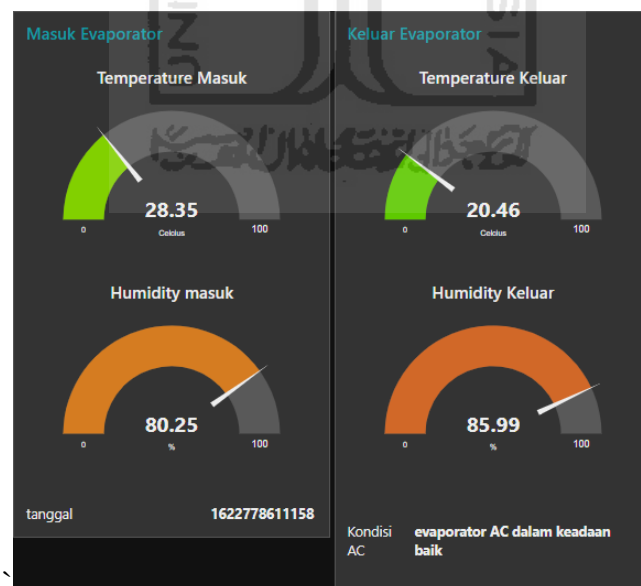
4.4 Pengujian Model

Tahap kedua yang dilakukan adalah memasukan persamaan dan nilai batas ke alat, kemudian melakukan pengukuran terhadap filter udara yang tidak dibersihkan selama 1 bulan. Apabila alat mampu membaca kondisi filter udara maka persamaan dianggap sudah mewakili kondisi filter udara evaporator.



Gambar 4-12 Kondisi Filter Udara Bersih.

Gambar 4-12 menunjukkan kondisi filter udara evaporator ketika dilakukan pengujian model matematika pada alat pemantau. Hasil pemantauan ditunjukkan pada Gambar 4-13. Berdasarkan Gambar 4-13 diketahui bahwa model persamaan telah mewakili kondisi filter udara.



Gambar 4-13 Hasil Pengujian Model Matematika.

Selain itu dilakukan pengujian model matematika terhadap kondisi filter udara yang sengaja diberi kotoran. Kondisi filter udara yang sengaja diberi kotoran dapat dilihat pada Gambar 4-14. Meskipun karakteristik dari kotoran yang

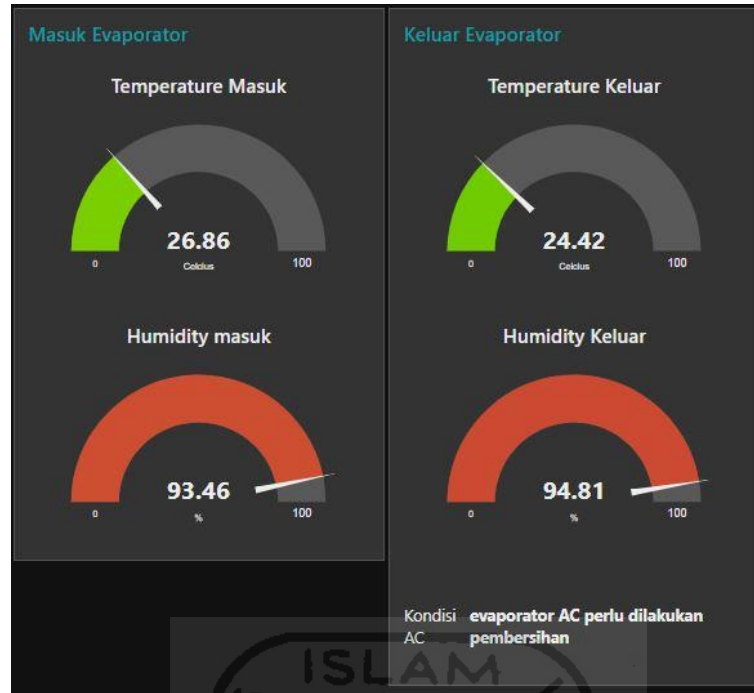
diberikan berbeda dengan kotoran yang timbul akibat proses AC, namun hal tersebut dianggap cukup mewakili kondisi filter udara kotor. Hal tersebut juga lebih efisien dibandingkan menunggu kondisi filter udara menjadi kotor akibat proses alami AC yang membutuhkan waktu sangat panjang.

Berdasarkan data yang ditunjukkan pada Gambar 4-13 yang kemudian dilakukan perhitungan dengan model matematika. Setelah dilakukan perhitungan didapatkan nilai $p(x) = 0,99$, artinya kemungkinan bahwa filter udara dalam keadaan bersih adalah 0,99 atau 99% filter dalam keadaan bersih. Berdasarkan nilai tersebut maka filter dalam keadaan 1 bulan setelah pemakaian dapat dikatakan dalam keadaan bersih.



Gambar 4-14 Kondisi Filter Udara Kotor.

Hasil dari pengujian dengan kondisi yang ditunjukkan pada Gambar 4-14 dapat dilihat pada Gambar 4-15. Melihat Gambar 4-15 diketahui bahwa antarmuka telah menunjukkan peringatan kondisi filter udara kotor dalam bentuk tulisan. Pada pengujian ini kondisi kelembaban lingkungan memang sedikit lebih tinggi dan kondisi temperatur lebih rendah daripada kondisi lingkungan pada pengujian pertama.



Gambar 4-15 Hasil Pengujian Kondisi Filter Kotor.

Berdasarkan data yang ditunjukkan pada Gambar 4-15, data tersebut kemudian dimasukkan kedalam model matematika untuk memperoleh nilai kemungkinan filter udara dalam kondisi bersih. Setelah nilai data dimasukkan kedalam model matematika dan telah dilakukan perhitungan maka diperoleh nilai $p(x) = 0,68 \times 10^{-13}$, yang artinya didapatkan kemungkinan filter udara dalam keadaan bersih sebesar $0,68 \times 10^{-13}$ atau $0,68 \times 10^{-13}$ %. Berdasarkan hasil perhitungan tersebut maka kondisi filter yang dibuat kotor dalam kondisi kotor.

4.5 Pembahasan

Proses perancangan dan penelitian ini menghasilkan beberapa hal. Salah satunya adalah perbedaan kondisi filter udara evaporator dalam kondisi kotor dan bersih. Pada awalnya pengambilan data AC tidak dilakukan pada satu jenis AC saja namun dilakukan pada berbagai jenis AC yang terdapat pada Laboratorium Teknik Mesin UII, akibat pandemi yang hadir diawal tahun membuat pengambilan AC dilakukan di satu AC yang terdapat di salah satu rumah warga.

Penggunaan AC di salah satu rumah warga sebagai objek penelitian membuat proses pengambilan data tidak bisa dilakukan dimalam hari. Karena beban biaya listrik ditanggung oleh pemilik rumah yang bukan sebuah organisasi

besar seperti universitas. Akibat hadirnya pandemi ini juga membuat proses pembuatan prototipe tidak maksimal. Penggunaan material yang kurang memadai menjadi salah satu kekurangan dalam proses perancangan ini. Terlebih jumlah objek pengamatan yang sedikit membuat proses pengujian model menjadi lebih sulit

Adanya pengaruh internal juga membuat proses perancangan ini belum begitu sempurna. Pengaruh internal tersebut berakibat pada proses pemodelan kondisi filter udara kondensor yang tidak bisa dilakukan dan penggunaan alat dan bahan yang tidak efisien seperti, penggunaan mikrokontroler yang menyisakan banyak pin, dan penggunaan PCB yang terlalu besar perlu diefisiensikan.

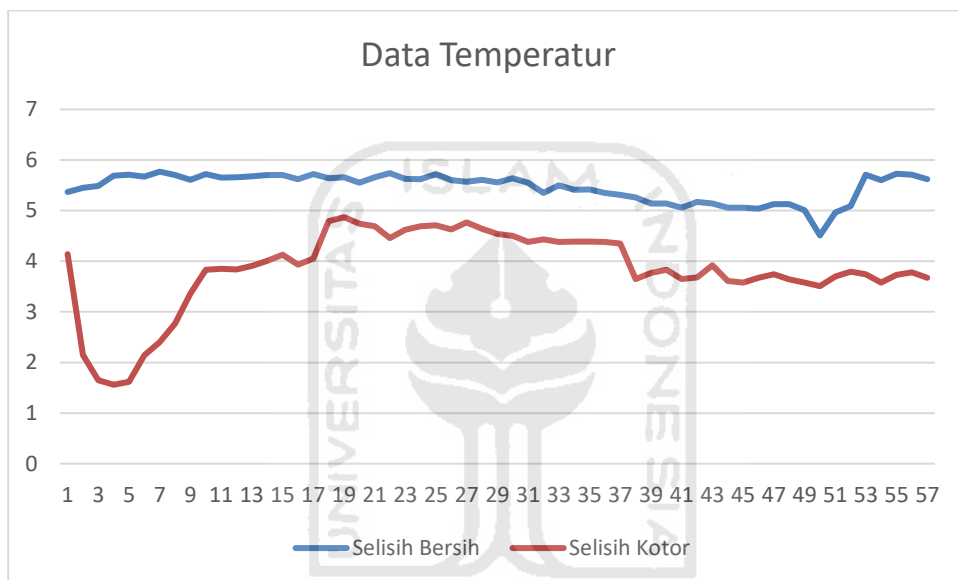
Walaupun demikian perancangan alat monitoring AC berbasis IoT ini telah mampu membaca data temperatur dari rentang 16-30 °C dan kelembaban dari rentang 0-100 % yang digunakan dalam proses siklus sistem pengkondisi udara AC. Selain itu berdasarkan analisis yang dilakukan perancangan ini mampu membedakan kondisi filter udara bersih dengan filter udara kotor. Proses pemodelan filter udara sistem pengkondisi udara ini telah mendapatkan hasil yang memuaskan.

Secara teknis perancangan ini mampu membuat mikrokontroler berkomunikasi dengan server dan saling bertukar data. Perancangan ini juga telah menghasilkan sebuah antarmuka yang mudah untuk dipahami oleh pengguna. Selain itu perancangan ini mampu mengirimkan data menuju pangkalan data melalui jaringan internet dengan sangat mudah dan cepat. Pemilihan perangkat lunak yang mudah digunakan juga merupakan hasil perancangan ini.

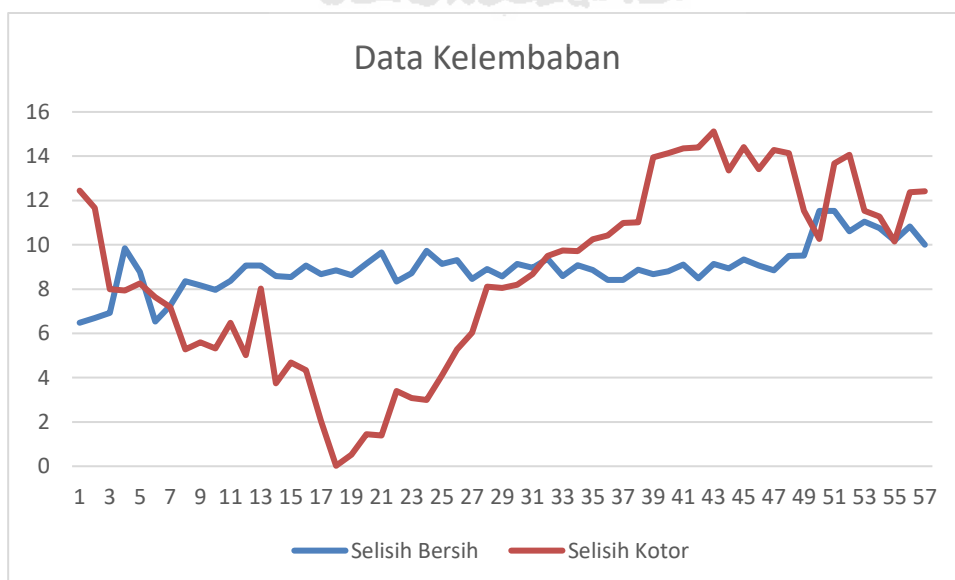
Terdapat beberapa hal penting yang belum mampu dilakukan pada perancangan kali ini. Hal tersebut berupa analisis data kondisi evaporator belum mampu dilakukan, penggunaan material dalam perancangan ini masih dengan material seadanya, proses desain alat ini belum dilakukan secara sempurna dan masih terdapat permasalahan dalam pengambilan data.

Pengambilan data yang tidak dilakukan dengan menggunakan 2 AC menjadi kekurangan lainnya dalam perancangan ini. Kekurangan lainnya adalah pengaruh external karena selama proses pengambilan data ruangan belum mampu dikondisikan kosong, sehingga mempengaruhi data yang telah diperoleh. Hal

tersebut ditunjukkan pada Gambar 4-16 dan Gambar 4-17, keduanya menunjukkan bahwa pada pengambilan data dengan kondisi kotor masih terdapat fluktuasi yang sangat besar. Terdapat banyak kemungkinan yang terjadi sehingga membuat data pada kondisi kotor memiliki fluktuasi yang tinggi, contohnya adalah kondisi ruangan tidak kosong atau terdapat orang, kondisi dari filter kotor berpengaruh pada program kendali AC, dan kemungkinan lainnya adalah kondisi cuaca yang berubah saat pengambilan data. Oleh sebab itu dalam perancangan ini masih perlu dilakukan pengembangan lebih lanjut untuk membuat alat ini menjadi lebih sempurna.



Gambar 4-16 Grafik Data Selisih Temperatur.



Gambar 4-17 Grafik Data Selisih Kelembaban.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan dan penelitian yang telah dilakukan maka dapat ditarik beberapa kesimpulan. Kesimpulan pertama berkaitan dengan telah dirancangnya sebuah alat yang mampu memberikan peringatan dini terhadap kekotoran yang terjadi pada filter udara evaporator. Untuk membuat alat yang mampu memberikan peringatan dini terhadap kondisi filter udara evaporator, maka didapatkan kesimpulan kedua mengenai diketahuinya perbedaan yang signifikan dengan nilai kepercayaan hingga 95% dari parameter temperatur dan kelembaban pada masukan dan keluaran evaporator.

Perbedaan tersebut perlu dipahami oleh manusia sebagai penerima peringatan dini pengkondisi udara dan komputer sebagai pemberi peringatan dini pengkondisi udara. Oleh sebab itu perbedaan tersebut perlu dimodelkan kedalam bahasa yang dimengerti oleh manusia dan komputer. Kesimpulan ketiga berkaitan dengan telah dilakukannya pemodelan perbedaan temperatur dan kelembaban dalam bentuk persamaan matematika membuat manusia dan komputer dapat memahami perbedaan yang ada. Untuk persamaan model filter udara evaporator dituliskan sebagai persamaan (1. 11):

$$y = p(x) = \frac{e^x}{(1+e^x)} \text{ dengan nilai ,} \quad (1. 11)$$
$$x = -45,12 + 7,43x_1 + 0,75x_2$$

5.2 Saran atau Penelitian Selanjutnya

Proses perancangan dan penelitian yang dilakukan memiliki banyak kekurangan. Sehingga harapan untuk penelitian selanjutnya mengenai permasalahan yang sama adalah:

1. Pembuatan Prototipe menggunakan material yang lebih baik.

2. Pengamatan parameter temperatur dan kelembaban pada kondensor sebaiknya dilakukan dan dijadikan pertimbangan untuk menyelesaikan permasalahan ini.
3. Pemilihan alat dan bahan seperti protokol dan perangkat keras maupun lunak diharapkan lebih efisien dan efektif.
4. Pengambilan data yang digunakan diharapkan memiliki lebih banyak variasi kondisi lingkungan seperti kondisi malam atau badai.



DAFTAR PUSTAKA

- Cengel, Y. A., & Boles, M. A. (2006). *Thermodynamics: An engineering approach*. McGraw Hill.
- Chao, S., Yiqiang, J., Yang, Y., & Shiming, D. (2012). Experimental performance evaluation of a novel dry-expansion evaporator with defouling function in a wastewater source heat pump. *Applied Energy*, 95, 202–209. <https://doi.org/10.1016/j.apenergy.2012.02.030>
- Cirani, S., Ferrari, G., Picone, M., & Veltri, L. (2019). *Internet of things: Architectures, protocols and standards* (First edition). Wiley.
- de Lallana, R. M. P. (2020). *Proyecto Fin de Grado*.
- Diori, G., Rianjani, D. A., Maulana, G., Tamzil, T. Z., Manawan, M., & Sukandi, A. (2019). *Sistem Otomatisasi dan Monitoring Perawatan Berkala AC (Air Conditioner) Berbasis Arduino yang Terintegrasi IoT (Internet of Things)*. 9(1), 184–193.
- Gay, W. (2018). DHT11 sensor. Dalam *Advanced Raspberry Pi* (hlm. 399–418). Springer.
- Hillar, G. (2017). *MQTT Essentials—A Lightweight IoT Protocol*.
- Hundy, G. F., Trott, A. R., & Welch, T. (2016). *Refrigeration, air conditioning and heat pumps* (Fifth edition). Butterworth-Heinemann is an imprint of Elsevier.
- Hunt, C. (2002). *TCP/IP network administration* (Vol. 2). O'Reilly Media, Inc.
- Ilham, A. A., & Ramschie, A. (2013). Sistem Monitoring dan Kendali Kerja Air Conditioning Berbasis Mikrokontroler ATmega 8535. *Jurnal Ristek*, 2(1).

- Larson, R., & Farber, B. (2019). *Elementary statistics*. Pearson Education Canada.
- Leens, F. (2009). An introduction to I²C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1), 8–13.
- Liu, T. (2013). Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302). *Aosong Electronics*. [En línea].
Disponível: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> [Último acesso: 09/2018].
- Mukti, F. (2020). *Remote dan Monitoring AC Dengan Memanfaatkan Internet of things (IoT)*.
- Nasution, T., Muchtar, M., Seniman, S., & Siregar, I. (2019). *Monitoring temperature and humidity of server room using Lattepanada and ThingSpeak*. 1235(1), 012068.
- Ott, L., & Longnecker, M. (2001). *An introduction to statistical methods and data analysis* (5th ed). Duxbury.
- Rocha, L. A. G., da Silva, B. A. R., & Costa, D. I. (2019). Comparação de desempenho de sensores de baixo custo com sensores comerciais para a coleta de dados ambientais. *Periódico Eletrônico Fórum Ambiental da Alta Paulista*, 15(4).
- Rodriguez, A. G., O'Neal, D., Davis, M., & Kondepudi, S. (1996). Effect of reduced evaporator airflow on the high temperature performance of air conditioners. *Energy and Buildings*, 24(3), 195–201.
[https://doi.org/10.1016/S0378-7788\(96\)00976-0](https://doi.org/10.1016/S0378-7788(96)00976-0)

- Saputra, L. K. P., & Lukito, Y. (2017). *Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266*. 126–130.
- Sensortec, B. (2013). Data sheet BMP180 Digital pressure sensor. *Np: Bosch Sensortec*, 1–29.
- Sensortec, B. (2014). Datasheet: BME280 Combined humidity and pressure sensor. *Alemanha, publicação eletrônica*, 55p.
- Stoev, I., & Mutkov, V. (2018). *Microclimatic data collection multisensor system for design of energy model in residential buildings*. 1–3.



Lampiran 1- 5-1 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <WiFi.h>
#include <PubSubClient.h>

/* Inisialisasi pin yang digunakan pada sensor input evaporator*/
#define I2C_SDA_1 23
#define I2C_SCL_1 19
/* inisialisasi pin yang digunakan pada sensor output evaporator*/
#define I2C_SDA_2 33
#define I2C_SCL_2 32

//#define ledPin 22

/* Membuat objek untuk membuat pin dapat berkomunikasi dengan I2C */
TwoWire I2CBME1 = TwoWire(0);
TwoWire I2CBME2 = TwoWire(1);

/* Membuat objek untuk dapat memanggil fungsi pada library */
Adafruit_BME280 bme1;
Adafruit_BME280 bme2;

/* Menginisialisasikan username dan password wifi/hotspot untuk
menyambungkan ESP32 dengan Internet */

const char* ssid = "BALABALTENG";
const char* password = "BALABALTENGHADIR";

unsigned long delayTime;

/* Menginisialisasikan alamat dari broker MQTT untuk
menyambungkan ESP32 dengan broker MQTT */

const char* mqtt_server = "test.mosquitto.org";
```

Lampiran 1- 5-2 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
// Membuat array untuk menampung nilai pembacaan sensor 1/in_Eva
char temp1_inEva [8];
char hum1_inEva [8];
// Membuat array untuk menampung nilai pembacaan sensor 2/out_Eva
char temp2_outEva[8];
char hum2_outEva [8];

// Mendefinisikan konstanta untuk perhitungan model matematika
float temp1,hum1,temp2,hum2;
float x1,x2,z;
float e =2.72;
double p;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  setup_wifi(); // Memanggil fungsi setup_wifi

  /* Memulai komunikasi I2C pada pin yang digunakan */
  Serial.println(F("BME280 1 test"));
  I2CBME1.begin(I2C_SDA_1, I2C_SCL_1, 100000); //Komunikasi pada pin sensor 1/sensor
in_Eva
  Serial.println(F("BME280 2 test"));
  I2CBME2.begin(I2C_SDA_2, I2C_SCL_2, 100000);//Komunikasi pada pin sensor 2/sensor
out_Eva
```

Lampiran 1- 5-3 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
if (!status1) {
  Serial.println("Could not find a valid BME280 1 sensor, check wiring!");
  while (1);
}

status2 = bme2.begin(0x76, &I2CBME2); //Pengecekan sensor2/out_Eva
if (!status2) {
  Serial.println("Could not find a valid BME280 2 sensor, check wiring!");
  while (1);
}

Serial.println("-- Test OKE --");
/* Memulai menghubungkan ESP32 dengan broker MQTT */
client.setServer(mqtt_server, 1883);
delayTime = 1000;
Serial.println();
}

/* Fungsi untuk menyambungkan ESP32 dengan internet */
void setup_wifi() {
  delay(10);
  /* Menyambungkan ESP32 dengan Internet */
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Lampiran 1- 5-4 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
bool status1;// Objek untuk mengecek komunikasi pada sensor in_Eva
bool status2;// Objek untuk mengecek komunikasi pada sensor out_Eva

/* Mengecek komunikasi I2C pada sensor dengan ESP32 */
status1 = bme1.begin(0x76, &I2CBME1); // Pengecekan sensor 1/in_Eva
if (!status1) {
  Serial.println("Could not find a valid BME280 1 sensor, check wiring!");
  while (1);
}
status2 = bme2.begin(0x76, &I2CBME2); //Pengecekan sensor2/out_Eva
if (!status2) {
  Serial.println("Could not find a valid BME280 2 sensor, check wiring!");
  while (1);
}
Serial.println("-- Test OKE --");
/* Memulai menghubungkan ESP32 dengan broker MQTT */
client.setServer(mqtt_server, 1883);
delayTime = 1000;
Serial.println();
}
/* Fungsi untuk menyambungkan ESP32 dengan internet */
void setup_wifi() {
  delay(10);
  /* Menyambungkan ESP32 dengan Internet */
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Lampiran 1- 5-5 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
/* Fungsi untuk membaca humidity dan temperature dari sensor
 * dan mengirimkan kepada server MQTT */
void printValues() {
    // Membaca nilai temperature dan humidity pada sensor 1/in_Eva
    temp1 = bme1.readTemperature();
    hum1 = bme1.readHumidity();

    // Membaca nilai temperature dan humidity pada sensor 2/out_Eva
    temp2 = bme2.readTemperature();
    hum2 = bme2.readHumidity();

    // Memasukan nilai ke dalam array
    dtostrf(temp1,1,2,temp1_inEva);
    dtostrf(hum1,1,2,hum1_inEva);

    dtostrf(temp2,1,2,temp2_outEva);
    dtostrf(hum2,1,2,hum2_outEva);

    // Perhitungan untuk mengetahui kondisi filter udara AC
    x1 = temp1-temp2;
    x2 = hum2-hum1;
    z = -45.12 + (x1*7.43)+(x2 *0.75);
    p = pow(e,z)/(1 + pow(e,z));

    Serial.println ("-----SENSOR 1-----");
    client.publish("inEva/temp", temp1_inEva); // Mengirimkan array hasil pembacaan sensor ke
    sever MQTT dengan topic "inEva/temp" untuk pembacaan temperature sensor 1/in_Eva

    client.publish("inEva/hum", hum1_inEva); // Mengirimkan array hasil pembacaan sensor ke sever
    MQTT dengan topic "inEva/hum" untuk pembacaan humidity sensor 1/in_Eva

    Serial.print("Temperature = ");
    Serial.print(temp1);
    Serial.println(" *C");
    Serial.print("Humidity = ");
    Serial.print(hum1);
    Serial.println(" %");
    Serial.println();
}
```

Lampiran 1- 5-6 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
Serial.println ("-----SENSOR 2-----");
  client.publish("outEva/temp",temp2_outEva);// Mengirimkan array hasil pembacaan sensor ke
  sever MQTT dengan topic "outEva/temp" untuk pembacaan temperature sensor 2/out_Eva

  client.publish("outEva/hum",hum2_outEva); // Mengirimkan array hasil pembacaan sensor ke
  sever MQTT dengan topic "outEva/hum" untuk pembacaan humidity sensor 2/out_Eva

  Serial.print("Temperature = ");
  Serial.print(temp2);
  Serial.println(" *C");
  Serial.print("Humidity = ");
  Serial.print(hum2);
  Serial.println(" %");

  /* Peringatan evaporator AC rusak*/
  char OKE [50]= "evaporator AC dalam keadaan baik"; // Pesan peringatan evaporator AC baik-
  baik saja
  char EROR[50] = "evaporator AC perlu dilakukan pembersihan";// Pesan peringatan evaporator
  AC sudah tidak baik-baik saja
  char SALAH [50] = "pengukuran salah";
  if(p>=0.51 && p<=1){
    Serial.println(OKE);
    Serial.println (p);
    client.publish("Eva/AC",OKE);//Mengirimkan pesan Evaisi evaporator AC baik-baik saja
  }
  else if (p < 0.51 && p >= 0) {
    Serial.println(EROR);
    Serial.println (p);
    client.publish("Eva/AC",EROR); // Mengirimkan pesan Evaisi evaporator AC sudah rusak.
  }
  else {
    client.publish("Eva/AC",SALAH);
    Serial.println (SALAH);
  }
  Serial.println();
  delay(30000);
}
```

Lampiran 1- 5-7 PROGRAM PEMBACAAN SENSOR PADA ARDUINO IDE

```
/* Fungsi reconnect apabila ESP32 tidak terhubung dengan broker MQTT */
void reconnect() {
  // Berulang sampai terhubung
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);

    //Program untuk menghubungkan MQTT
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
    }

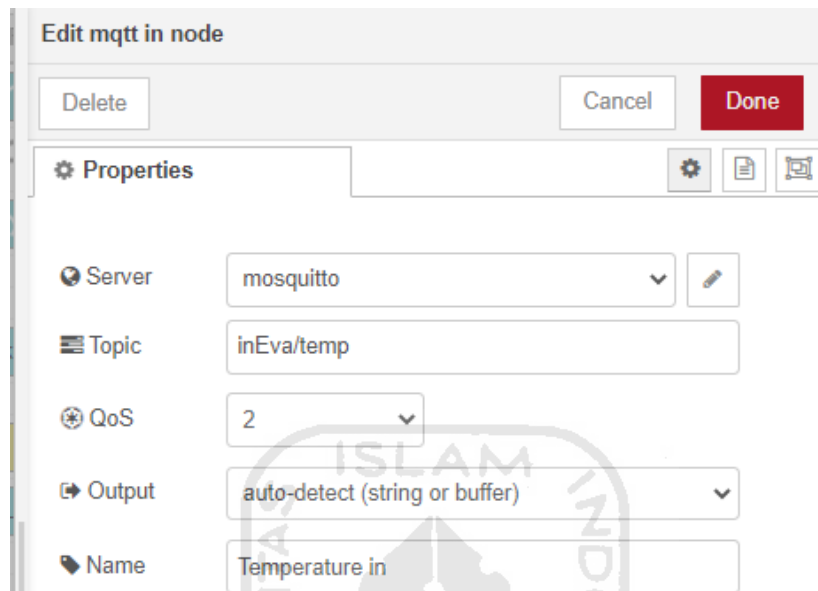
    else {
      Serial.print("failed, reconnect=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Delay lima detik sebelum berulang
      delay(5000);
    }
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  printValues();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000) {
    lastMsg = now;
  }

  delay(delayTime);
}
```


Lampiran 2- 1 PENGATURAN BLOK NODE-RED

Pengaturan Blok MQTT (ungu)



Edit mqtt in node

Delete Cancel Done

⚙️ Properties 📄 🔄

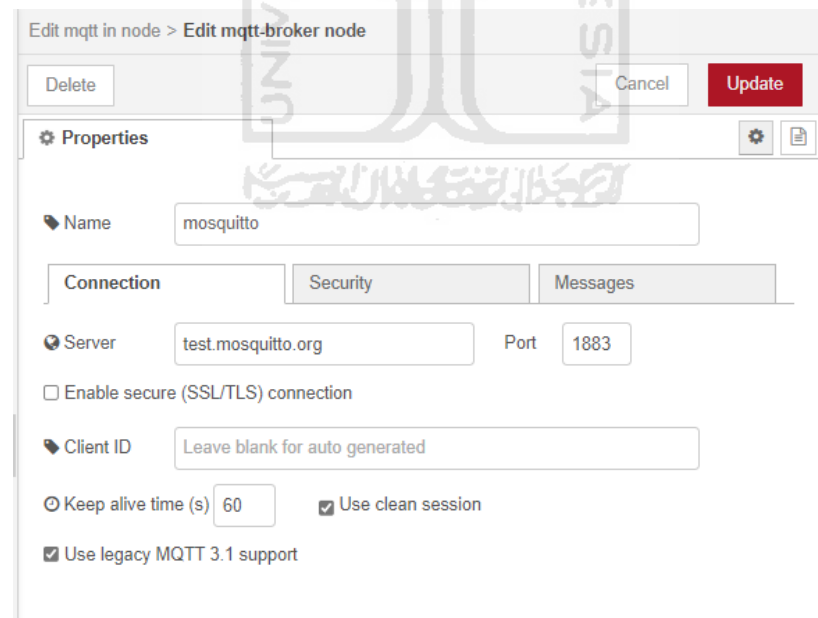
🌐 Server mosquitto

📄 Topic inEva/temp

🌐 QoS 2

➡️ Output auto-detect (string or buffer)

📄 Name Temperature in



Edit mqtt in node > Edit mqtt-broker node

Delete Cancel Update

⚙️ Properties 📄

📄 Name mosquitto

Connection Security Messages

🌐 Server test.mosquitto.org Port 1883

Enable secure (SSL/TLS) connection

📄 Client ID Leave blank for auto generated

🕒 Keep alive time (s) 60 Use clean session

Use legacy MQTT 3.1 support

Pengaturan Blok Array (Coklat terang)

Edit join node

Delete Cancel Done

⚙ Properties

Mode

Combine each

to create

Send the message:

- After a number of message parts
- After a timeout following the first message
- After a message with the `msg.complete` property set

Name

Lampiran 2- 2 PENGATURAN BLOK NODE-RED

Pengaturan Blok Antrmuka (Biru)

Edit gauge node

Delete Cancel Done

Properties

Group [Monitoring Evaporator] Masuk Evapora

Size auto

Type Gauge

Label Temperature Masuk

Value format {{value}}

Units Celcius

Range min 0 max 100

Colour gradient

Sectors 0 optional optional 100

Name

Pengaturan Blok Debug (Hijau)

Edit debug node

Delete Cancel Done

Properties

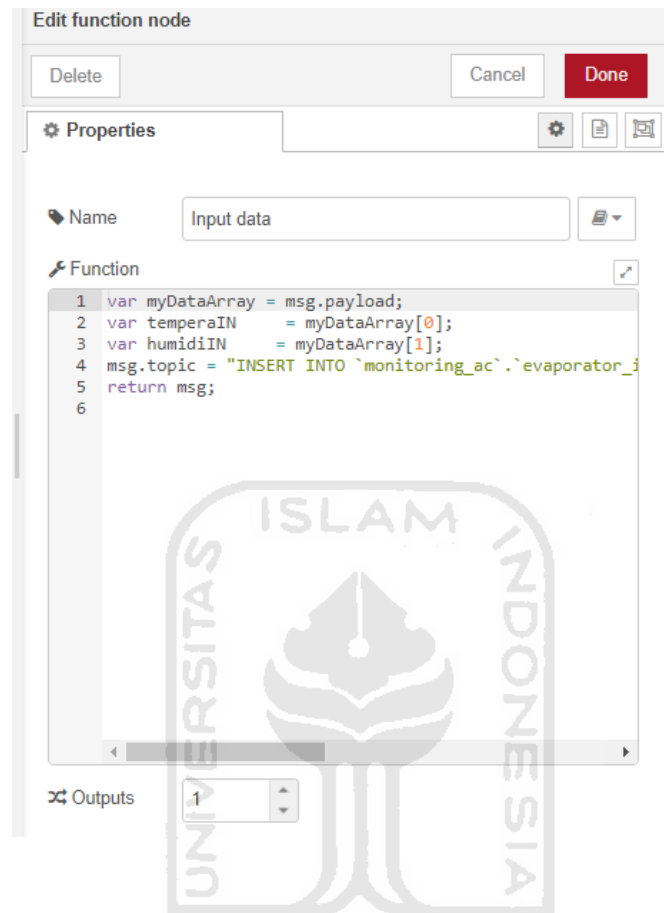
Output msg. payload

To debug window
 system console
 node status (32 characters)

Name Name

Lampiran 2- 3 PENGATURAN BLOK NODE-RED

Pengaturan Blok Fungsi (Krem)



Naskah Program Blok Fungsi :

- (1) `var mydataArray = msg.payload;`
- (2) `var temperaIN = mydataArray[0];`
- (3) `var humidiIN = mydataArray[1];`
- (4) `msg.topic = "INSERT INTO `monitoring_ac`.`evaporator_in`
(`temperature masuk`,`humidity masuk`)VALUES
(`+temperaIN`,`+humidiIN`)"`;
- (5) `return msg;`

Lampiran 2- 4 PENGATURAN BLOK NODE-RED

Pengaturan Blok SQL

Edit mysql node > Edit MySQLdatabase node

Delete Cancel Update

Properties

Host 127.0.0.1

Port 3306


User root

Password

Database monitoring_ac

Timezone

Name Name

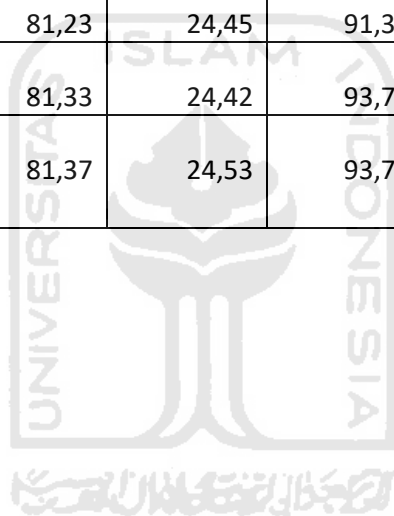


Lampiran 3- 1 Tabel Pengukuran Filter Udara Kotor

ID	Tem Masuk	Hum Masuk	Tem Keluar	Hum Keluar	Waktu
1	28,16	87,55	24,02	100	2021-03-25 10:29:45
2	28,43	86,12	26,28	97,78	2021-03-25 10:30:17
3	28,57	85,62	26,92	93,61	2021-03-25 10:30:49
4	28,65	85,46	27,09	93,4	2021-03-25 10:31:21
5	28,61	85,16	26,99	93,42	2021-03-25 10:31:53
6	28,63	85,33	26,49	92,96	2021-03-25 10:32:25
7	28,59	85,55	26,19	92,74	2021-03-25 10:32:57
8	28,58	85,22	25,81	90,49	2021-03-25 10:33:29
9	28,6	85,13	25,24	90,73	2021-03-25 10:34:01
10	28,57	84,5	24,74	89,82	2021-03-25 10:34:33
11	28,53	84,26	24,68	90,74	2021-03-25 10:35:05
12	28,48	83,94	24,64	88,96	2021-03-25 10:35:37
13	28,46	84,06	24,55	92,08	2021-03-25 10:36:09
14	28,41	83,56	24,4	87,3	2021-03-25 10:36:41
15	28,4	83,66	24,27	88,35	2021-03-25 10:37:13
16	28,37	83,43	24,44	87,76	2021-03-25 10:37:45
17	28,35	83,29	24,3	85,33	2021-03-25 10:38:17
18	28,32	82,99	23,53	83,01	2021-03-25 10:38:54
19	28,34	82,83	23,47	83,35	2021-03-25 10:39:26
20	28,31	82,32	23,57	83,76	2021-03-25 10:39:59
21	28,3	82,1	23,61	83,49	2021-03-25 10:40:31
22	28,25	82,31	23,79	85,71	2021-03-25 10:41:03

23	28,22	82,64	23,6	85,72	2021-03-25 10:41:34
24	28,21	82,18	23,52	85,17	2021-03-25 10:42:06
25	28,2	82,18	23,49	86,29	2021-03-25 10:42:38
26	28,18	82,33	23,55	87,61	2021-03-25 10:43:10
27	28,19	82,09	23,42	88,12	2021-03-25 10:43:42
28	28,18	82,11	23,54	90,22	2021-03-25 10:44:14
29	28,16	82,07	23,62	90,12	2021-03-25 10:44:46
30	28,13	81,84	23,63	90,04	2021-03-25 10:45:18
31	28,15	81,89	23,77	90,57	2021-03-25 10:45:50
32	28,15	81,82	23,72	91,31	2021-03-25 10:46:22
33	28,17	81,99	23,79	91,74	2021-03-25 10:46:55
34	28,16	81,88	23,77	91,6	2021-03-25 10:47:26
35	28,18	82,04	23,79	92,29	2021-03-25 10:47:58
36	28,17	81,66	23,79	92,08	2021-03-25 10:48:30
37	28,17	81,65	23,82	92,64	2021-03-25 10:49:02
38	28,19	81,87	24,54	92,89	2021-03-25 10:49:44
39	28,21	81,96	24,44	95,91	2021-03-25 10:50:16
40	28,2	81,77	24,37	95,91	2021-03-25 10:50:48
41	28,19	81,76	24,54	96,12	2021-03-25 10:51:20
42	28,17	81,71	24,49	96,11	2021-03-25 10:51:52
43	28,17	81,66	24,25	96,78	2021-03-25 10:52:24
44	28,18	81,65	24,57	95	2021-03-25 10:52:59
45	28,18	81,71	24,6	96,12	2021-03-25 10:53:27
46	28,18	81,85	24,51	95,26	2021-03-25 10:53:59

47	28,17	81,39	24,43	95,67	2021-03-25 10:54:31
48	28,18	82,04	24,54	96,18	2021-03-25 10:55:03
49	28,18	81,5	24,6	93,03	2021-03-25 10:55:35
50	28,19	81,76	24,68	92,02	2021-03-25 10:56:07
51	28,2	81,63	24,5	95,3	2021-03-25 10:56:39
52	28,2	81,83	24,41	95,9	2021-03-25 10:57:11
53	28,21	81,58	24,47	93,11	2021-03-25 10:57:43
54	28,19	80,93	24,61	92,21	2021-03-25 10:58:15
55	28,18	81,23	24,45	91,38	2021-03-25 10:58:47
56	28,2	81,33	24,42	93,71	2021-03-25 10:59:19
57	28,2	81,37	24,53	93,79	2021-03-25 10:59:51

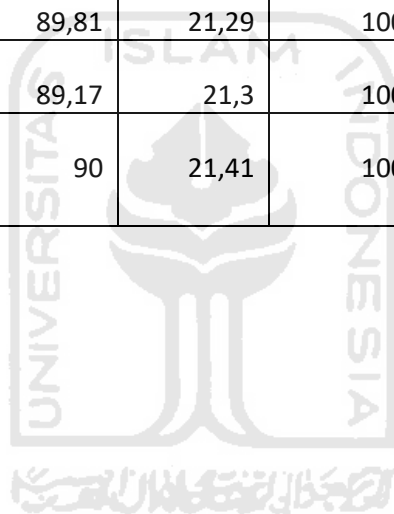


Lampiran 3- 2 Tabel Pengukuran Filter Udara Bersih

ID	Tem Masuk	Hum Masuk	Tem Keluar	Hum Keluar	Waktu
1	26,76	93,52	21,39	100	2021-03-31 09:49:15
2	26,79	93,31	21,34	100	2021-03-31 09:49:47
3	26,8	93,08	21,31	100	2021-03-31 09:50:19
4	26,72	90,16	21,03	100	2021-03-31 09:50:51
5	26,68	91,22	20,97	100	2021-03-31 09:51:23
6	26,78	93,47	21,11	100	2021-03-31 09:51:54
7	26,84	92,75	21,07	100	2021-03-31 09:52:26
8	26,81	91,64	21,11	100	2021-03-31 09:53:33
9	26,81	91,83	21,2	100	2021-03-31 09:54:06
10	26,84	92,03	21,12	100	2021-03-31 09:54:37
11	26,83	91,63	21,18	100	2021-03-31 09:55:09
12	26,81	90,94	21,15	100	2021-03-31 09:55:41
13	26,78	90,93	21,1	100	2021-03-31 09:56:13
14	26,81	91,41	21,11	100	2021-03-31 09:56:45
15	26,82	91,45	21,12	100	2021-03-31 09:57:17
16	26,8	90,94	21,18	100	2021-03-31 09:57:49
17	26,81	91,32	21,09	100	2021-03-31 09:58:21
18	26,81	91,15	21,17	100	2021-03-31 09:58:53
19	26,83	91,37	21,17	100	2021-03-31 09:59:25
20	26,8	90,85	21,25	100	2021-03-31 09:59:57
21	26,74	90,35	21,08	100	2021-03-31 10:00:28
22	26,8	91,66	21,06	100	2021-03-31 10:01:00

23	26,8	91,28	21,17	100	2021-03-31 10:01:32
24	26,78	90,27	21,16	100	2021-03-31 10:02:04
25	26,79	90,87	21,07	100	2021-03-31 10:02:36
26	26,77	90,69	21,17	100	2021-03-31 10:03:08
27	26,8	91,54	21,23	100	2021-03-31 10:03:40
28	26,83	91,1	21,22	100	2021-03-31 10:04:12
29	26,79	91,43	21,23	100	2021-03-31 10:04:44
30	26,77	90,87	21,13	100	2021-03-31 10:05:16
31	26,79	91,04	21,24	100	2021-03-31 10:05:49
32	26,74	90,63	21,39	100	2021-03-31 10:06:21
33	26,79	91,41	21,29	100	2021-03-31 10:06:53
34	26,78	90,92	21,37	100	2021-03-31 10:07:24
35	26,77	91,14	21,35	100	2021-03-31 10:07:57
36	26,79	91,59	21,44	100	2021-03-31 10:09:03
37	26,81	91,59	21,5	100	2021-03-31 10:10:07
38	26,8	91,12	21,54	100	2021-03-31 10:10:39
39	26,8	91,33	21,66	100	2021-03-31 10:11:11
40	26,84	91,19	21,7	100	2021-03-31 10:11:43
41	26,85	90,89	21,79	100	2021-03-31 10:12:15
42	26,85	91,52	21,68	100	2021-03-31 10:12:47
43	26,83	90,87	21,69	100	2021-03-31 10:13:19
44	26,84	91,06	21,78	100	2021-03-31 10:13:51
45	26,83	90,66	21,77	100	2021-03-31 10:14:24
46	26,85	90,93	21,81	100	2021-03-31 10:14:56

47	26,9	91,15	21,77	100	2021-03-31 10:15:28
48	26,89	90,5	21,76	100	2021-03-31 10:16:00
49	26,86	90,49	21,85	100	2021-03-31 10:16:32
50	26,96	88,47	22,45	100	2021-03-31 10:26:56
51	26,94	88,47	21,98	100	2021-03-31 10:27:26
52	26,97	89,39	21,88	100	2021-03-31 10:27:58
53	26,99	88,96	21,28	100	2021-03-31 10:28:30
54	26,99	89,25	21,39	100	2021-03-31 10:29:05
55	27,02	89,81	21,29	100	2021-03-31 10:29:37
56	27,01	89,17	21,3	100	2021-03-31 10:30:09
57	27,03	90	21,41	100	2021-03-31 10:30:42



Lampiran 4- 1 Langkah-Langkah Analisis Wilcoxon Rank Sum Test
Studi Kasus

Salah satu tujuan dari perancangan ini adalah mampu mengetahui perbedaan dari kondisi filter udara bersih dengan kondisi filter udara kotor pada evaporator AC. Informasi yang diketahui adalah beberapa data temperatur dan kelembaban pada masukan dan keluaran evaporator AC dengan dua kondisi filter udara evaporator. Selisih nilai temperatur masuk dan . Rata-rata dari selisih nilai temperatur pada kondisi bersih = 5,46 dan untuk kondisi kotor = 3,83. Kemudian untuk rata-rata selisih nilai kelembaban pada kondisi bersih = 8,97 dan untuk kondisi kotor = 8,69

1. **Pendefinisian Masalah** = Permasalahan dapat dilihat pada kalimat pertama

Salah satu tujuan dari perancangan ini adalah mampu mengetahui perbedaan dari kondisi filter udara bersih dengan kondisi filter udara kotor pada

Berdasarkan kalimat pertama diketahui terdapat permasalahan untuk mengetahui **perbedaan** pada kondisi filter udara bersih dan kotor. Untuk mengetahui perbedaan dari 2 populasi data dapat digunakan metode yaitu, metode T-test dan Wilcoxon Rank Sum-Test.

2. **Uji Distribusi** = Untuk mengetahui apakah data berdistribusi normal.

Pengujian distribusi ini menggunakan nilai kurtosis dan skewness sebuah populasi data nilai kurtosis dan nilai skewness yang mendekati nilai 0 menunjukkan distribusi populasi data normal. Data yang digunakan terletak pada paragraf kedua dalam studi kasus.

Informasi yang diketahui adalah beberapa data temperatur dan kelembaban pada masukan dan keluaran evaporator AC dengan dua kondisi filter udara evaporator.

Hasil dari perhitungan nilai skewnes dan kurtosis adalah :

<i>Temperatur Bersih</i>		<i>Temperatur Kotor</i>	
Kurtosis	0,913266	Kurtosis	1,749174588
Skewness	-1,1368	Skewness	1,353147774

<i>Kelembaban Bersih</i>		<i>Kelembaban Kotor</i>	
Kurtosis	1,213829	Kurtosis	-0,97333837
Skewness	0,075056	Skewness	0,281394867

Berdasarkan nilai kurtosis dan skewness populasi data temperatur dan kelembaban dalam kondisi filter udara kotor atau bersih dapat diketahui bahwa seluruh populasi **tidak berdistribusi normal**.

3. **Pemilihan Metode** = Pemilihan metode didasarkan pada uji distribusi

Berdasarkan nilai kurtosis dan skewness populasi data temperatur dan kelembaban dalam kondisi filter udara kotor atau bersih dapat diketahui bahwa seluruh populasi **tidak berdistribusi normal**.

Berdasarkan pernyataan diatas maka dipilih metode **Wilcoxon Rank Sum-Test** untuk melakukan analisis. Hal ini disebabkan metode Wilcoxon Rank Sum-Test tersebut tidak memiliki syarat distribusi normal untuk data populasi sedangkan metode T-test memiliki syarat distribusi normal.

4. **Pemilihan Sampel** = Pemilihan sampel didasarkan pada metode yang dipilih dan nilai rata-rata.

Berdasarkan pernyataan diatas maka dipilih metode **Wilcoxon Rank Sum-Test** untuk melakukan analisis

Karena metode yang dipilih adalah **Wilcoxon Rank Sum-Test** maka akan dipilih sampel acak dari variabel bebas.

Rata-rata dari selisih nilai temperatur pada kondisi bersih = 5,46 dan untuk kondisi kotor = 3,83. Kemudian untuk rata-rata selisih nilai kelembaban pada kondisi bersih = 8,97 dan untuk kondisi kotor = 8,69

rata-rata kelembaban kotor < rata-rata kelembaban bersih, maka sampel dari kondisi kotor harus lebih kecil dari kondisi bersih jumlah dari sampel yang didapatkan sebesar $n = 12$ atau sekitar 21.05 % dari total data sampel tersebut dapat dilihat pada Tabel 4-1 dan Tabel 4-2.

5. **Pendefinisian Hipotesis** = Pendefinisian Hipotesis didasarkan pada metode yang digunakan dan jumlah sampel.

Metode yang dipilih adalah **Wilcoxon Rank Sum-Test** dan sampel yang didapatkan sebesar $n = 12$

Berdasarkan metode analisis yang dipilih dan jumlah sampel (n) > 10 maka dapat dirumuskan hipotesis berdasarkan perhitungan nilai z . Hipotesis tersebut dirumuskan,

$$H_0 = \text{Kedua data memiliki kondisi yang sama}$$

$$H_1 = \text{Kedua data memiliki kondisi yang tidak sama}$$

Hipotesis yang digunakan akan memiliki tingkat keyakinan sebesar 95% dengan nilai $\alpha = 0,05$ dan nilai $z = 1,645$. Apabila nilai z perhitungan > dari 1,645 maka H_0 ditolak.

6. **Perhitungan** = Perhitungan dilakukan berdasarkan metode dan sampel.

Sampel pada Tabel 4-1 dan Tabel 4-2

Berdasarkan nilai -nilai pada sampel maka dapat dihitung nilai:

$$T = \text{Data 1} + \text{Data 2} + \dots + \text{Data } n$$

$$\mu T = \frac{n_1 (n_1 + n_2 + 1)}{2}$$

$$\sigma^2 t = \frac{n_1 n_2}{12} \left[n_1 (n_1 + n_2 + 1) - \frac{\sum t_j (t_j^2 - 1)}{(n_1 + n_2)(n_1 + n_2 - 1)} \right]$$

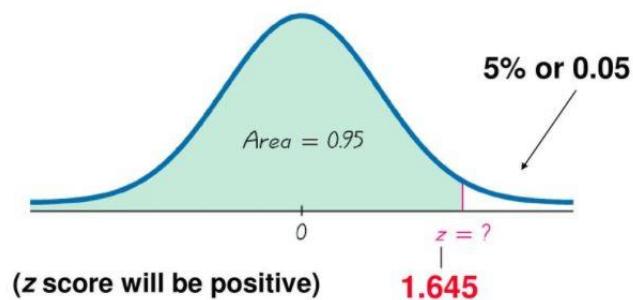
$$z = \frac{T - \mu T}{\sigma t}$$

Hasil dari perhitungan dapat dilihat pada Tabel 4-5 dan Tabel 4-6

7. **Pembahasan** = Pembahasan analisis ini berdasarkan dari hasil perhitungan dan hipotesis yang digunakan.

Hasil perhitungan menunjukkan nilai z pada temperatur = 6,75 dan pada data kelembaban $z = 2,04$

Berdasarkan hasil perhitungan dapat diketahui bahwa dalam grafik distribusi, nilai $z = 6,75$ dan $z = 2,04$ berada di sebelah kanan nilai $z = 1,645$. Karena nilai z merupakan nilai dari garis rata-rata menuju garis $\alpha = 0,05$.



Sumber : (Larson & Farber, 2019)

8. **Kesimpulan** = Kesimpulan ini didapatkan setelah melakukan perhitungan dan pembahasan kemudian dibandingkan dengan hipotesis yang telah didefinisikan.

Hasil dari pembahasan menunjukkan nilai $z = 6,75$ dan $z = 2,04$ berada di sebelah kanan nilai $z = 1,645$ atau lebih besar dari $z = 1,645$

Hipotesis yang digunakan dalam analisis ini menunjukkan apabila nilai z perhitungan lebih dari $z = 1,645$ maka H_0 ditolak atau tidak dapat diterima. Berdasarkan perbandingan antara hasil pembahasan dengan hipotesis maka dapat disimpulkan bahwa H_0 ditolak atau tidak dapat diterima sehingga data dapat dikatakan 95% berbeda.

Lampiran 4- 2 Langkah-langkah Pemodelan Regresi Logistik.

Langkah-langkah yang digunakan dalam mencari model Regresi Logistik adalah sebagai berikut:

- 1) Menentukan nilai *draft* = Nilai ini merupakan nilai dalam bentuk bilangan biner 0 dan 1 untuk mewakili dua kemungkinan kondisi filter. Kondisi filter bersih akan diwakilkan dengan nilai biner = 1 dan kondisi filter kotor akan diwakilkan dengan bilangan biner = 0. Hasil pemberian nilai *draft* ditunjukkan pada tabel dibawah ini.

Draft	Temp	Hum
1	5,75	10,61
1	5,73	9,84
1	5,72	9,65
1	5,71	9,51
1	5,7	9,13
1	5,69	8,9
1	5,64	8,85
1	5,48	8,68
1	5,42	8,59
1	5,42	8,48
1	5,14	8,46
0	5,43	8,68
0	5,01	8,36
0	4,96	8,2
0	4,74	8,11
0	4,74	8,05
0	4,58	7,63
0	4,28	6,48
0	3,91	5,32
0	3,76	5,02
0	3,66	4,33
0	3,65	3,4
0	2,39	2,04
0	1,11	0,52

- 2) Mencari Persamaan Regresi Normal Berganda = Untuk mendapatkan persamaan Regresi Normal Berganda maka digunakan fitur/*add-in Data Analyst* pada Microsoft Excel. Hasilnya diperoleh persamaan

$$x = A + Bx_1 + Cx_2$$

Dengan nilai

A	-1,07727
B	0,403519
C	-0,05086
x_1	Selisih Temperature
x_2	Selisih Kelembaban

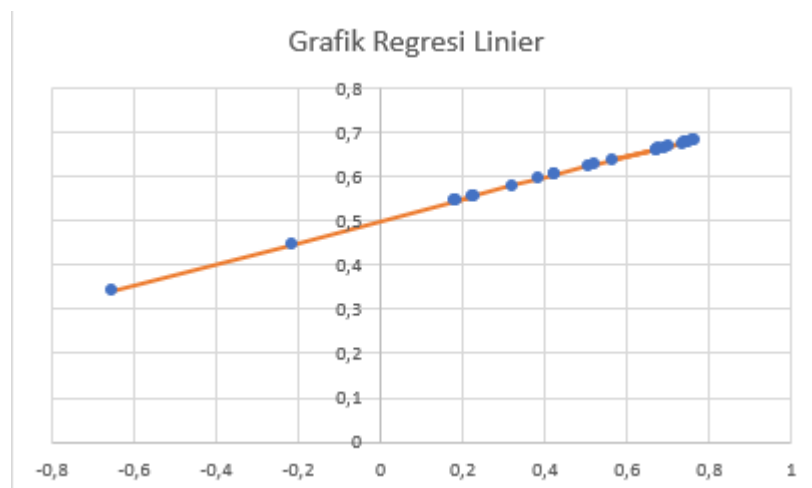
- 3) Mencari nilai X = Dengan menggunakan persamaan $x = A + Bx_1 + Cx_2$, maka didapatkan nilai X untuk setiap data pada setiap baris. Hasilnya ditunjukkan pada tabel dibawah ini. Nilai tersebut nantinya akan digunakan untuk mencari nilai probabilitas $p(x)$

Temp	Hum	X
5,75	10,61	0,703384
5,73	9,84	0,734472
5,72	9,65	0,7401
5,71	9,51	0,743184
5,7	9,13	0,758474
5,69	8,9	0,766136
5,64	8,85	0,748503
5,48	8,68	0,692585
5,42	8,59	0,672951
5,42	8,48	0,678545
5,14	8,46	0,566577
5,43	8,68	0,672409
5,01	8,36	0,519205
4,96	8,2	0,507166
4,74	8,11	0,422969
4,74	8,05	0,42602
4,58	7,63	0,382817
4,28	6,48	0,320245
3,91	5,32	0,229936
3,76	5,02	0,184665
3,66	4,33	0,179404
3,65	3,4	0,222665
2,39	2,04	-0,21661
1,11	0,52	-0,65581

- 4) Mencari Nilai $p(x)$ = Nilai ini merupakan kemungkinan yang dirumuskan dengan $p(x) = \frac{e^x}{(1+e^x)}$. Hasilnya ditunjukkan pada tabel dibawah ini

Temp	Hum	X	p(x)
5,75	10,61	0,703384	0,668938
5,73	9,84	0,734472	0,675786
5,72	9,65	0,7401	0,677018
5,71	9,51	0,743184	0,677692
5,7	9,13	0,758474	0,681022
5,69	8,9	0,766136	0,682684
5,64	8,85	0,748503	0,678852
5,48	8,68	0,692585	0,666542
5,42	8,59	0,672951	0,662164
5,42	8,48	0,678545	0,663414
5,14	8,46	0,566577	0,637973
5,43	8,68	0,672409	0,662042
5,01	8,36	0,519205	0,626962
4,96	8,2	0,507166	0,624142
4,74	8,11	0,422969	0,604193
4,74	8,05	0,42602	0,604923
4,58	7,63	0,382817	0,594552
4,28	6,48	0,320245	0,579384
3,91	5,32	0,229936	0,557232
3,76	5,02	0,184665	0,546036
3,66	4,33	0,179404	0,544731
3,65	3,4	0,222665	0,555437
2,39	2,04	-0,21661	0,446059
1,11	0,52	-0,65581	0,341682

Berdasarkan nilai X dan $p(x)$ diperoleh grafik regresi normal berganda.



- 5) Mencari LH, LogLH dan SumLogLH= Kedua nilai ini nantinya akan digunakan untuk memperoleh nilai sumLogLH yang akan digunakan dalam *add-in* Solver di Microsoft Excel. *Add-in* Solver berfungsi untuk mencari persamaan Regresi Logistik berdasarkan nilai-nilai persamaan Regresi Normal yang telah didapatkan diawal. Untuk mendapatkan nilai LH digunakan formula yang ditunjukkan pada Gambar dibawah ini

	A	B	C	D	E	F
3	draft	Temp	Hum	X	p(x)	LH
4	1	5,75	10,61	0,703384	0,668938	=IF(A4;E4;1-E4)

Sedangkan nilai LogLH digunakan formula yang diperlihatkan pada Gambar dibawah ini.

	E	F	G
	p(x)	LH	LogLH
	0,668938	0,668938	=LN(F4)

Kemudian yang terakhir dicari nilai penjumlahan dari seluruh nilai LogLH yang ada dengan formula =SUM(G4:G27).

- 6) Mencari persamaan Regresi Logistik = Untuk mendapatkan persamaan Regresi Logistik digunakan *Add-in* Solver dengan beberapa pengaturan. Pada tampilan *Add-in* Solver terdapat dua kolom yang perlu diisi. Kolom pertama diberi nama *Set Objective*, kolom ini diisi dengan nilai SumLogLH. Kemudian kolom kedua memiliki nama *By Changing Variable Cells*, kolom ini akan dimasukan beberapa kotak nilai variabel dari persamaan Regresi Normal berganda yang terlihat pada Gambar dibawah ini

Intercept	Temp	Hum
-1,07727	0,40352	-0,05086

Kemudian akan didapatkan nilai variabel untuk persamaan Regresi Logistik dengan nilai :

`-45,1185 7,429005 0,754034`

Yang artinya persamaan untuk mencari nilai X berubah menjadi

$$x = -45,118 + 7,429x_1 + 0,754x_2$$

Berdasarkan perubahan ini maka nilai X dan $p(x)$ akan berubah dan bentuk grafik akan berubah menjadi seperti

