

BAB 3 PERANCANGAN DAN IMPLEMENTASI

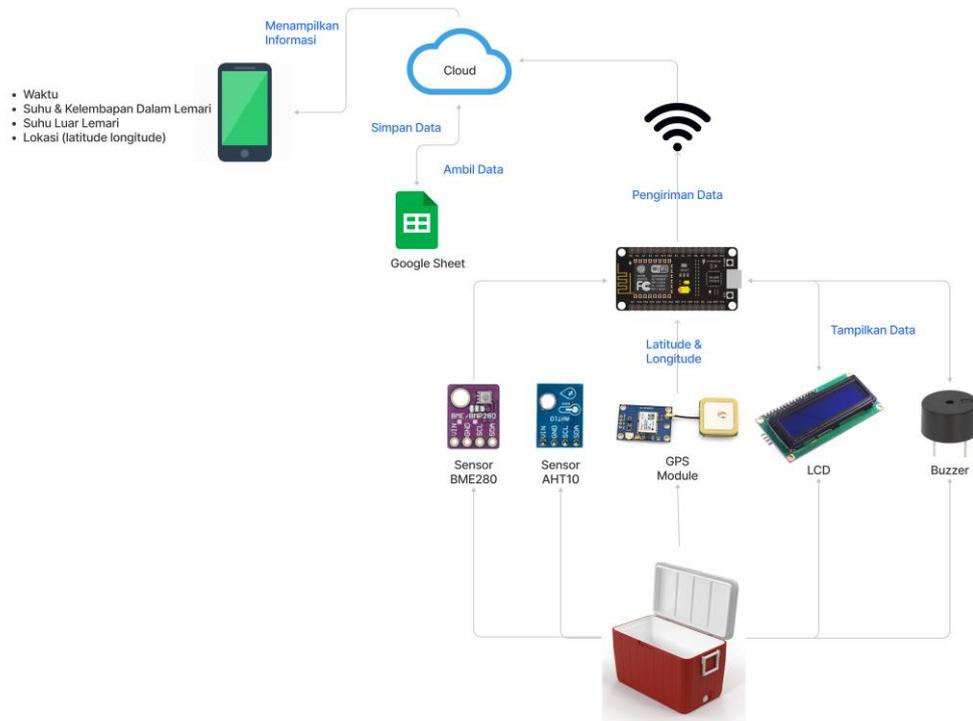
3.1 Arsitektur Sistem

Seperti yang telah dijelaskan pada bab sebelumnya, sistem *monitoring* yang dirancang pada penelitian ini akan diletakkan pada lemari pendingin / *cool box* yang biasa digunakan sebagai media penyimpanan vaksin Covid-19 selama distribusi vaksin. Diharapkan sistem ini dapat memudahkan pemantauan lemari pendingin khususnya pada pemantauan suhu di dalam lemari pendingin. Sistem monitoring lemari pendingin ini juga mengukur kelembapan di dalam lemari pendingin, suhu di luar lemari pendingin, dan posisi lemari pendingin.

Pada sistem *monitoring* lemari pendingin ini, digunakan beberapa komponen, seperti mikrokontroler, sensor suhu, *GPS module*, LCD, *buzzer*. Sensor adalah komponen atau perangkat yang berfungsi untuk mendeteksi perubahan, atau besaran fisika menjadi besaran listrik yang ada di lingkungan sekitarnya dan menghasilkan output sesuai dengan fungsinya [14].

Untuk mengukur suhu dan kelembapan di dalam lemari pendingin digunakan sensor BME280. Untuk mengukur suhu di luar lemari pendingin digunakan sensor AHT10. Untuk mengetahui posisi lemari pendingin digunakan *module* GPS. Terdapat juga LCD untuk menampilkan suhu dan kelembapan di dalam lemari pendingin dan juga *buzzer* yang digunakan sebagai alarm / *alert* yang akan berbunyi beberapa kali saat suhu di dalam lemari pendingin melewati suhu yang aman vaksin yaitu di bawah 2 °C atau di atas 8 °C. Dengan adanya bunyi dari *buzzer* ini diharapkan dapat membantu petugas untuk segera mengetahui lemari pendingin atau *cool box* mana yang sedang bermasalah.

Seluruh komponen tersebut akan diintegrasikan dengan mikrokontroler NodeMCU. Setiap pin dari masing-masing sensor dihubungkan ke pin NodeMCU. Lalu NodeMCU disambungkan ke *cloud* yaitu Google Sheet menggunakan jaringan WiFi agar data yang telah diterima NodeMCU dapat dikirim dan disimpan di *cloud*. *Dashboard* pada sistem *monitoring* lemari pendingin ini akan mengambil data dari *cloud* untuk ditampilkan sebagai parameter *monitoring* seperti pada Gambar 3.1. NodeMCU membutuhkan jaringan WiFi, untuk itu, dalam penelitian ini diasumsikan setiap pesawat/mobil memiliki *router* untuk memancarkan jaringan WiFi agar NodeMCU dapat berjalan. Untuk itu, keterbatasan sistem *monitoring* ini adalah membutuhkan jaringan WiFi untuk dapat berjalan sesuai dengan fungsinya. Berikutnya juga akan dijelaskan mengenai perangkat keras yang digunakan pada penelitian ini.



Gambar 3.1 Arsitektur Sistem

3.2 Komponen Alat Monitoring

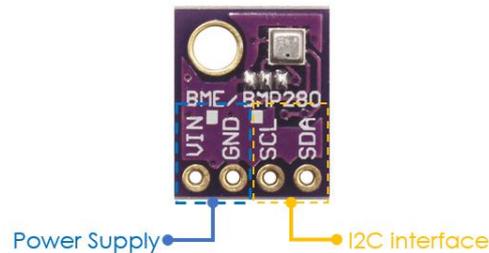
Seperti yang telah dijelaskan, dalam penelitian ini yang menjadi fokus utama adalah *hardware* atau alat *monitoring* itu sendiri. Alat *monitoring* tersebut tentu saja terdiri dari beberapa komponen. Pada bagian ini akan dijelaskan masing-masing komponen yang digunakan pada alat *monitoring* beserta dengan alasan pemilihan komponen.

3.2.1 Sensor BME280

Sensor BME280 pada Gambar 3.2 adalah sensor digital untuk mengukur suhu, kelembapan, dan tekanan. Sensor BME280 mendukung antarmuka I2C dengan *address* I2C 0x76. Sensor BME280 pada penelitian ini akan mengukur suhu dan kelembapan di dalam lemari pendingin sebagai parameter utama dalam sistem monitoring lemari pendingin ini.

Ada beberapa sensor suhu dan kelembapan yang sering digunakan beberapa di antaranya DHT11, DS18B20, BMP180, HTU21D, dan BME280. Dan yang digunakan untuk mengukur suhu dan kelembapan di dalam lemari pendingin pada penelitian ini adalah sensor BME280. Pemilihan sensor BME280 dikarenakan sensor kelembapan yang dimiliki dapat memberikan respon dalam waktu yang cepat dan sensor suhu dioptimalkan untuk pembacaan yang akurat dan *noise* terendah [15].

Selain itu, karakteristik dari sensor BME280 yang dapat dilihat pada Tabel A.1 sesuai dengan kebutuhan dalam penelitian ini. Yang pertama sensor BME280 dapat mengukur suhu dari *range* -40°C - 85°C di mana *range* ini sudah sangat mencakup *range* suhu yang ingin diukur di dalam lemari pendingin yaitu 2°C - 8°C . Yang kedua akurasi suhu BME280 sebesar $0,5^{\circ}\text{C}$ - 1°C yang sudah memenuhi kebutuhan.



Gambar 3.2 Sensor BME280 [16]

3.2.2 Sensor AHT10

Sensor AHT10 pada Gambar 3.3 adalah sensor digital untuk mengukur suhu dan kelembapan. Sensor AHT10 mendukung antarmuka I2C dengan *address* I2C $0x38$. Sensor AHT10 akan mengukur suhu di luar lemari pendingin sebagai parameter tambahan dalam sistem monitoring lemari pendingin ini.

Spesifikasi sensor AHT10 sebenarnya kurang lebih sama dengan BME280. Pemilihan AHT10 ini karena spesifikasi serupa dengan BME280 tetapi dengan harga yang jauh lebih terjangkau. Sebelumnya, dalam penelitian ini menggunakan sensor HTU21D tetapi karena HTU21D sering bertabrakan dengan GPS maka digunakan sensor AHT10.

Karakteristik dari sensor AHT10 yang dapat dilihat pada Tabel A.2 sudah sesuai dengan kebutuhan dalam penelitian ini. Yang pertama sensor AHT10 dapat mengukur suhu dari *range* -40°C - 85°C di mana *range* ini sudah mencakup *range* suhu di luar lemari pendingin sebagai parameter *monitoring* tambahan.



Gambar 3.3 Sensor AHT10 [17]

3.2.3 GPS Module

GPS (*Global Positioning System Receiver*) pada Gambar 3.4 adalah perangkat yang dapat mendeteksi lokasi dengan cara menangkap serta memproses sinyal dari satelit navigasi. GPS *module* yang terpasang pada sistem monitoring lemari pendingin ini akan mendeteksi posisi dari lemari pendingin vaksin secara *real time*.

Penerima GPS bekerja dengan mencari tahu seberapa jauh GPS dari sejumlah satelit. Satelit mengirimkan informasi tentang posisi GPS dan waktu saat ini dalam bentuk sinyal radio menuju bumi.



Gambar 3.4 GPS Module [18]

3.2.4 LCD (Liquid Crystal Display)

LCD pada Gambar 3.5 adalah salah satu komponen elektronika yang berfungsi untuk menampilkan suatu data berupa karakter, huruf, angka, ataupun simbol. LCD yang digunakan adalah LCD dengan komunikasi I2C dengan *address* I2C 0x3F.



Gambar 3.5 LCD [19]

3.2.5 Buzzer

Buzzer yang digunakan dalam penelitian ini adalah *active buzzer* seperti Gambar 3.6 yang sudah memiliki suaranya sendiri Ketika diberi tegangan listrik. Buzzer pada sistem monitoring lemari pendingin ini digunakan sebagai *alarm/alert* ketika suhu di dalam lemari pendingin melewati suhu ketentuan yang sudah ditetapkan.

Pemilihan *buzzer* sebagai alert alat monitoring dalam penelitian ini dikarenakan *alert* berupa suara lebih baik dibandingkan *alert* berupa lampu. Selain itu, ukuran penggunaan daya *buzzer* sangat minim.

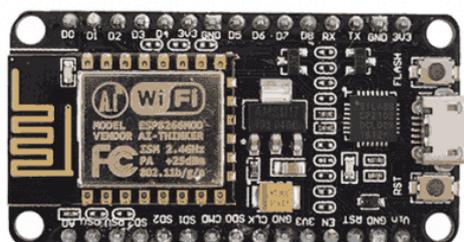


Gambar 3.6 Buzzer [20]

3.2.6 NodeMCU

NodeMCU ESP8266 pada Gambar 3.7 digunakan sebagai mikrokontroler yang di dalamnya sudah dilengkapi dengan modul WiFi sebagai komunikasi untuk pengiriman data sensor. Seluruh sensor yang digunakan dalam sistem *monitoring* lemari pendingin vaksin ini akan diintegrasikan pada NodeMCU dan di program untuk mengambil data dari seluruh sensor.

Pemilihan NodeMCU sebagai mikrokontroler dikarenakan karakteristik dari NodeMCU yang dapat membuat sistem monitoring lemari pendingin vaksin Covid-19 berjalan yaitu NodeMCU memiliki modul WiFi yang dikenal dengan sebutan ESP8266, modul tersebutlah yang dapat membuat sistem *monitoring* lemari pendingin vaksin Covid-19 berjalan dan tersambung dengan *database*.

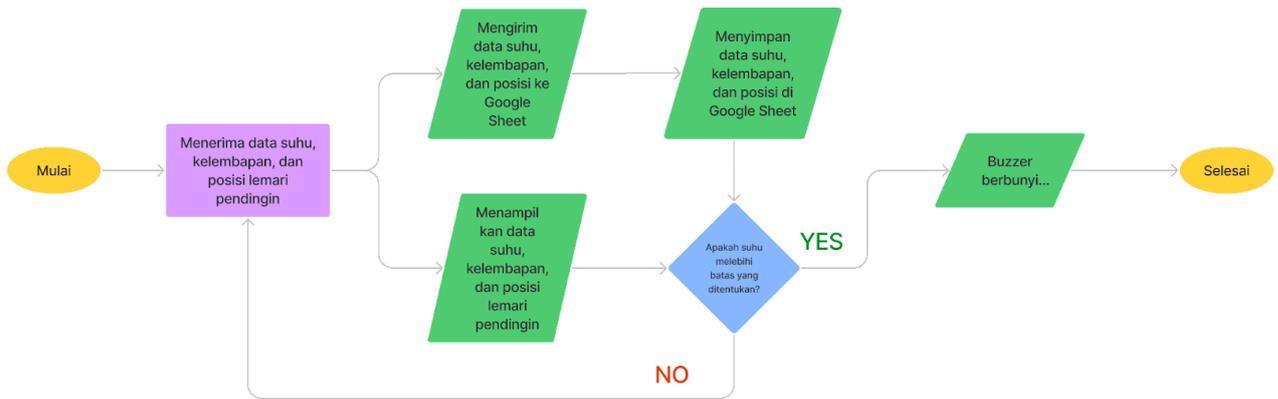


Gambar 3.7 NodeMCU [21]

3.3 Cara Kerja Sistem

Perangkat mulai bekerja dengan menerima data dari sensor. Sensor BME280 menangkap data suhu dan kelembapan di dalam lemari pendingin, sensor AHT10 akan menangkap suhu di luar lemari pendingin, dan *module* GPS akan menangkap posisi atau lokasi lemari pendingin.

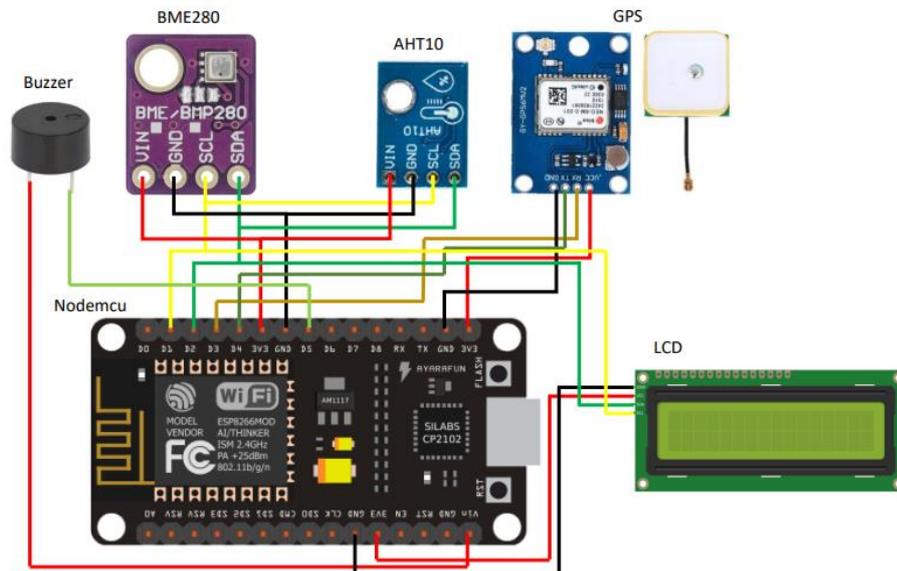
Data yang diterima akan dikirim ke *cloud* yaitu Google Sheet dengan komunikasi WiFi dan Google Apps Script sebagai penyimpanan data. Data di *cloud* akan ditampilkan di *dashboard* untuk memonitor lemari pendingin vaksin. Data suhu dan kelembapan di dalam lemari pendingin ditampilkan juga pada layar LCD yang dipasang di luar lemari pendingin. Jika suhu lemari pendingin tidak berada di antara 2 °C sampai 8 °C maka *buzzer* akan berbunyi. Untuk lebih jelasnya dapat melihat Gambar 3.8.



Gambar 3.8 Cara Kerja Sistem

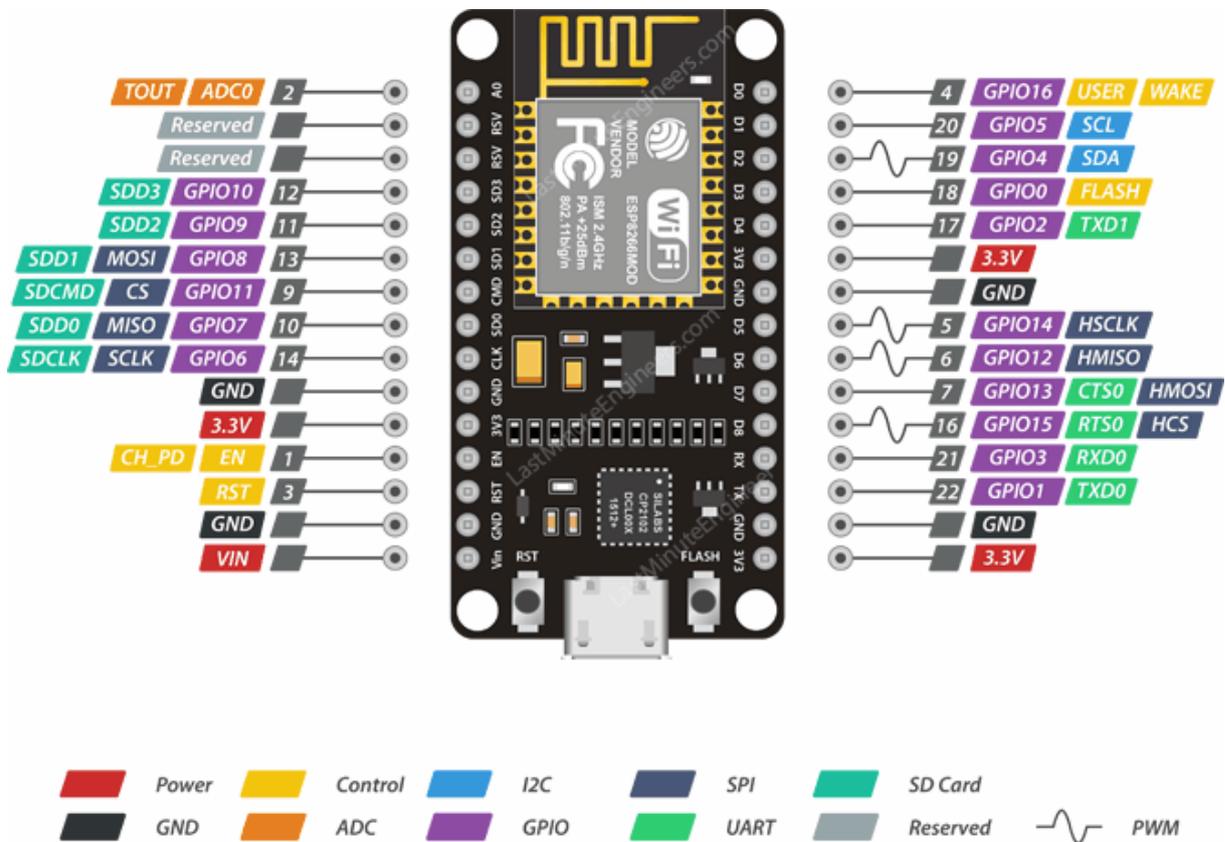
3.4 Perancangan Hardware

Perangkat *hardware* yang merupakan fokus dalam penelitian ini tentu saja akan dirancang paling pertama. Pada bagian ini akan dijelaskan mengenai *schematic* masing-masing sensor dan modul dengan NodeMCU seperti pada Gambar 3.10. Terdapat beberapa komponen utama dalam penelitian ini yaitu dua sensor suhu BME280 dan AHT10, satu GPS *module*, dan satu mikrokontroler NodeMCU ESP8266.



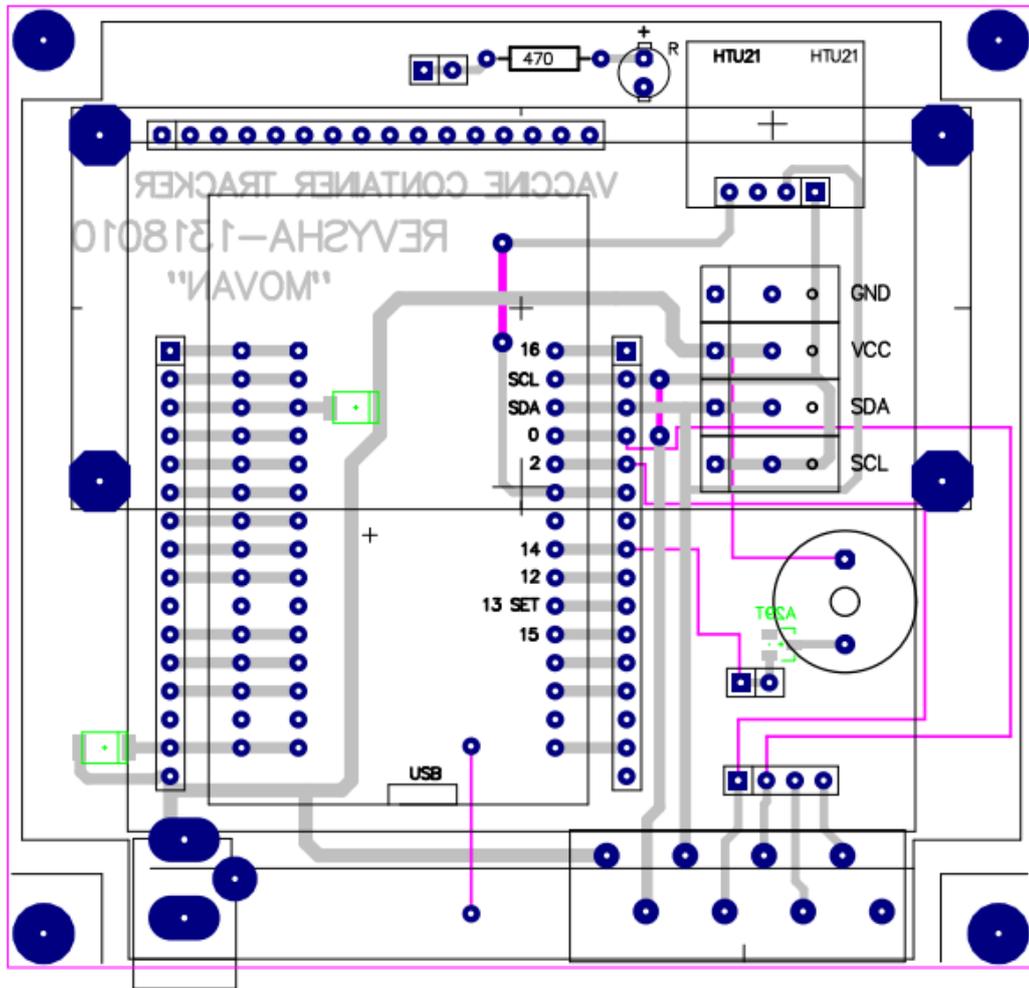
Gambar 3.9 Skematik Perangkat Keras

NodeMCU sendiri memiliki total 30 pin yang dapat dihubungkan seperti yang ada pada Gambar 3.10 dan akan dijelaskan beberapa jenis pin yang digunakan dalam penelitian ini. Terdapat empat pin *power* yang ditandai dengan warna merah sebagai catu daya. GND adalah pin *ground* yang juga berjumlah empat. Pin I2C digunakan untuk menghubungkan semua jenis sensor dan modul I2C yang digunakan. Terdapat 17 pin GPIO (*General Purpose Input-Output*) yang digunakan untuk membaca input dan mengontrol *output* berdasarkan kondisi yang sesuai dengan program yang telah dibuat. Beberapa GPIO dapat digunakan sebagai input/output digital dan sebagai *interface* untuk protokol seperti I2C, I2S, dan sebagainya.



Gambar 3.10 Interface NodeMCU [22]

Dalam penelitian ini juga akan dilakukan perancangan PCB. PCB (*Printed Circuit Board*) adalah papan dari logam yang penuh dengan sirkuit yang menghubungkan komponen elektronik tanpa kabel. Hal ini dimaksudkan dengan tujuan untuk memudahkan penggunaan prototipe dari alat *monitoring*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.11.



Gambar 3.11 Rangkaian PCB

1. NodeMCU – BME280

Ada empat pin yang perlu dihubungkan ke mikrokontroler. Ada VIN yaitu catu daya untuk modul, GND sebagai *ground*, SDA (Serial Data), dan SCL (Serial Clock). SDA dan SCL adalah penghantar khusus dari perangkat I2C. SDA berfungsi untuk mentransfer data dan SCL berfungsi untuk menghantarkan sinyal *clock*.

Tabel 3.1 *Schematic* NodeMCU – BME280

NodeMCU	BME280
3V	VIN
GND	GND
D2	SDA
D1	SCL

2. NodeMCU – AHT10

Sama seperti sensor BME280, terdapat empat pin pada sensor AHT10 yang perlu dihubungkan ke mikrokontroler NodeMCU. Ada VIN, GND, SDA, dan SCL yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 *Schematic* NodeMCU – AHT10

NodeMCU	AHT10
3V	VIN
GND	GND
D2	SDA
D1	SCL

3. NodeMCU – GPS Module

GPS *module* memiliki empat pin yang dihubungkan ke NodeMCU. Ada VCC, RX (*Receiver*), TX (*Transmitter*). TX adalah pemancar dan RX adalah penerima untuk komunikasi serial. Untuk lebih jelasnya dapat melihat Tabel 3.3.

Tabel 3.3 *Schematic* NodeMCU – GPS

NodeMCU	GPS
3V	VCC
GND	GND
D3	RX
D4	TX

4. NodeMCU – LCD

LCD yang digunakan adalah LCD dengan komunikasi I2C. Ada empat pin di LCD yang harus dihubungkan ke NodeMCU. Ada VCC, GND, SDA, dan SCL yang dapat dilihat lebih jelas pada Tabel 3.4.

Tabel 3.4 *Schematic* NodeMCU – LCD

NodeMCU	LCD
3V	VCC
GND	GND
D2	SDA
D1	SCL

5. NodeMCU – Buzzer

Buzzer memiliki dua kaki yang harus dihubungkan ke mikrokontroler NodeMCU, yaitu VCC, I/O, GND. Untuk lebih jelasnya dapat dilihat pada Tabel 3.5.

Tabel 3.5 *Schematic* NodeMCU – Buzzer

NodeMCU	Buzzer
VIN	VCC
D5	I/O
GND	GND

3.5 Perancangan Database

Seperti yang telah dijelaskan sebelumnya, *database* yang digunakan dalam penelitian ini berbasis *cloud*. *Database* digunakan untuk menyimpan data dari sensor. *Database* yang digunakan dalam penelitian ini ialah Google Sheet yang merupakan aplikasi *spreadsheet* berbasis web yang tentunya dapat digunakan secara *online* dan dilengkapi dengan formula canggih, mudah digunakan dan gratis.

Data yang diambil dari sensor BME280 yaitu suhu dan kelembapan di dalam lemari pendingin, dari sensor AHT10 berupa suhu di luar lemari pendingin, dan dari GPS *module* berupa *latitude* dan *longitude* akan langsung dikirim ke Google Sheet menggunakan Google Apps Script. Google Apps Script adalah *platform scripting* yang dikembangkan oleh Google untuk memperluas fungsionalitas Google Apps dan membangun aplikasi berbasis *cloud* yang ringan. *Database* dibagi menjadi 7 kolom, yaitu Tanggal, Jam, *Latitude*, *Longitude*, Suhu Dalam, *Humidity* Dalam, dan Suhu Luar. Untuk gambaran *database* lebih jelasnya dapat dilihat pada Gambar 3.12.

	A	B	C	D	E	F	G
1	Tanggal	Jam	Latitude	Longitude	Suhu Dalam (°C)	Humidity Dalam (%)	Suhu Luar (°C)
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

Gambar 3.12 Struktur *Database*

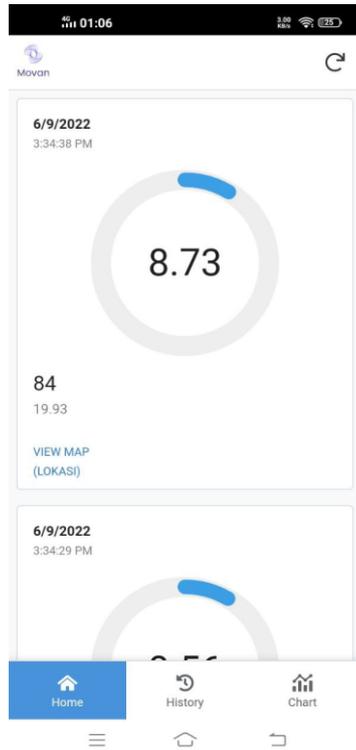
3.6 Perancangan Dashboard

Data dari *database* akan ditampilkan pada sebuah *dashboard*. *Dashboard* akan menerima input data dari *database* yang ada pada *cloud* lalu menampilkan data tersebut. Data yang ditampilkan pada *dashboard* berupa tanggal, posisi lemari pendingin berupa *longitude* dan *latitude*, suhu dan kelembapan di dalam lemari pendingin, dan suhu di luar lemari pendingin. *Dashboard* berupa aplikasi *mobile*.

Dashboard yang dirancang menggunakan *platform* AppSheet. AppSheet adalah platform pengembangan tanpa kode untuk membuat aplikasi web dan seluler. Aplikasi dari AppSheet dapat menggunakan beberapa sumber data, seperti Google Spreadsheet, Excel, Cloud SQL, Salesforce, dan konektor serupa lainnya. Desain antarmuka aplikasi menggunakan template UX untuk membuat peta, kalender, dan lainnya. Alur kerja otomatis juga dapat disertakan ke dalam aplikasi untuk melakukan berbagai hal, seperti mengirim notifikasi, membuat email, membuat laporan kostum, dan memodifikasi data.

Dashboard ini dapat menampilkan data yang ada pada *database*. *Dashboard* ini memiliki tiga halaman yaitu Home yang akan menampilkan data *current* pada *database*, dapat dilihat pada Gambar 3.13, Gambar 3.14 merupakan halaman detail dari Home di mana *latitude* dan *longitude* sudah dapat divisualisasikan dengan *maps*, dan Gambar 3.15 merupakan detail dari *maps* yang akan menampilkan lokasi saat ini dan histori lokasi sebelumnya. Halaman kedua pada Gambar 3.16 adalah History yang menampilkan tabel histori yang berisi seluruh data dalam *database*. Lalu yang terakhir ada halaman Chart pada Gambar 3.17, di mana halaman ini hanya sebagai tambahan visualisasi data saja, dengan menampilkan *chart* yang dapat

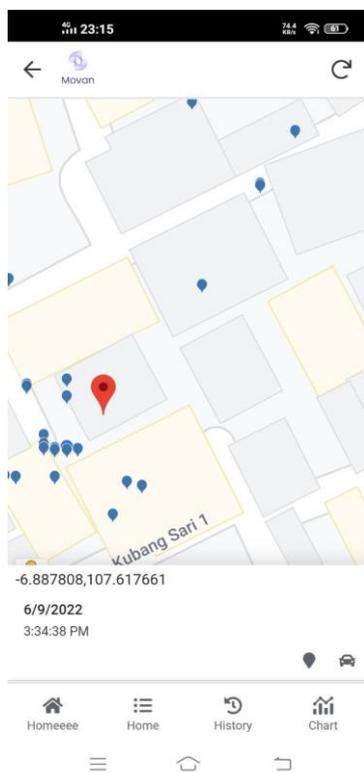
memvisualisasikan data suhu di dalam lemari pendingin dan data suhu di luar lemari pendingin. Dan fitur notifikasi juga telah berhasil dibuat dan dapat dilihat pada Gambar 3.18.



Gambar 3.13 Home Page I



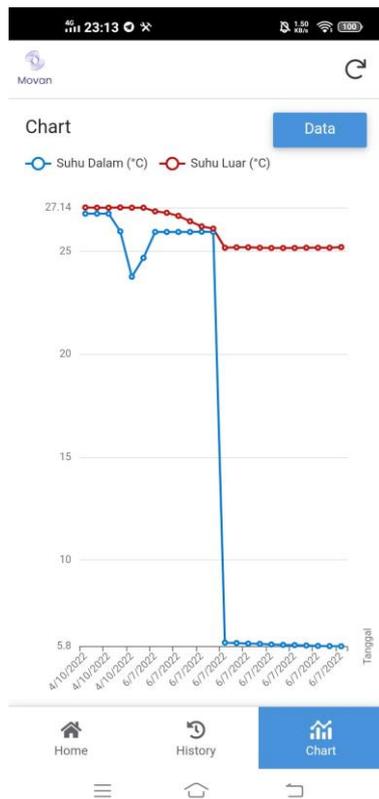
Gambar 3.14 Home Page II



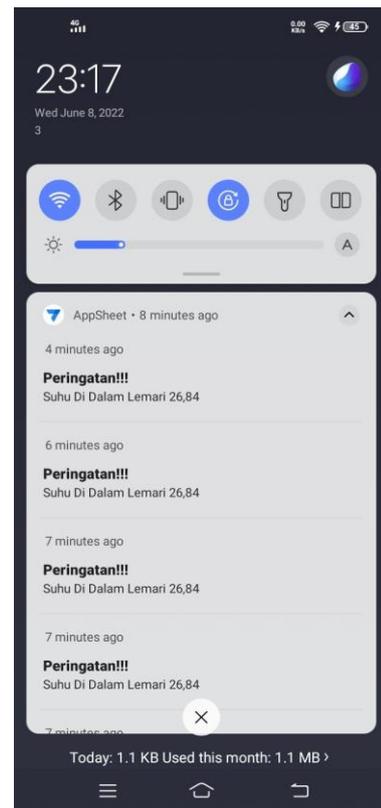
Gambar 3.15 Home Page III

Tanggal ↓	Jam	Latitude	Long
6/7/2022	10:23:15 AM	-6.887878	107.1
6/7/2022	10:23:06 AM	-6.887879	107.1
6/7/2022	10:22:58 AM	-6.887877	107.1
6/7/2022	10:22:50 AM	-6.887879	107.1
6/7/2022	10:22:42 AM	-6.887877	107.1
6/7/2022	10:22:34 AM	-6.887872	107.1
6/7/2022	10:22:26 AM	-6.887860	107.1
6/7/2022	10:22:18 AM	-6.887808	107.1
6/7/2022	10:22:11 AM	-6.887525	107.1
6/7/2022	10:22:03 AM	-6.887523	107.1
6/7/2022	10:21:55 AM	-6.887452	107.1
6/7/2022	9:31:09 AM	-6.887913	107.1
6/7/2022	9:30:58 AM	-6.887913	107.1
6/7/2022	9:30:49 AM	-6.887914	107.1
6/7/2022	9:30:30 AM	-6.887915	107.1

Gambar 3.16 History Page



Gambar 3.17 Chart Page

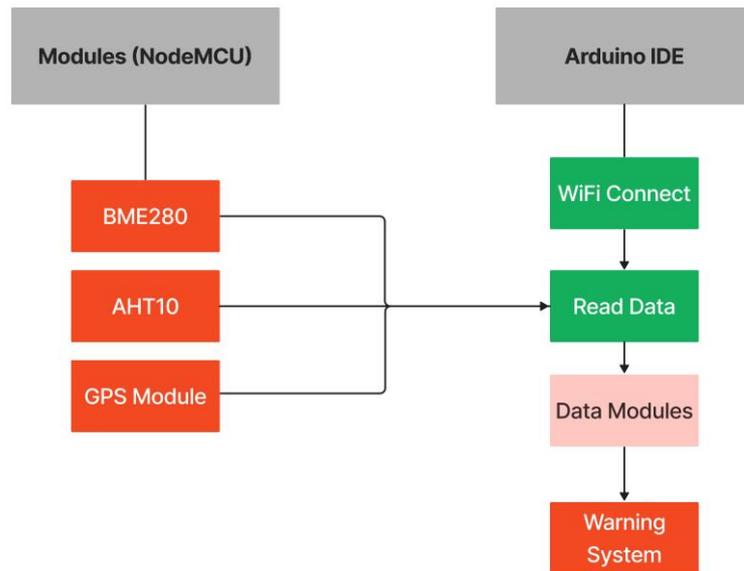


Gambar 3.18 Notification

3.7 Implementasi Sistem

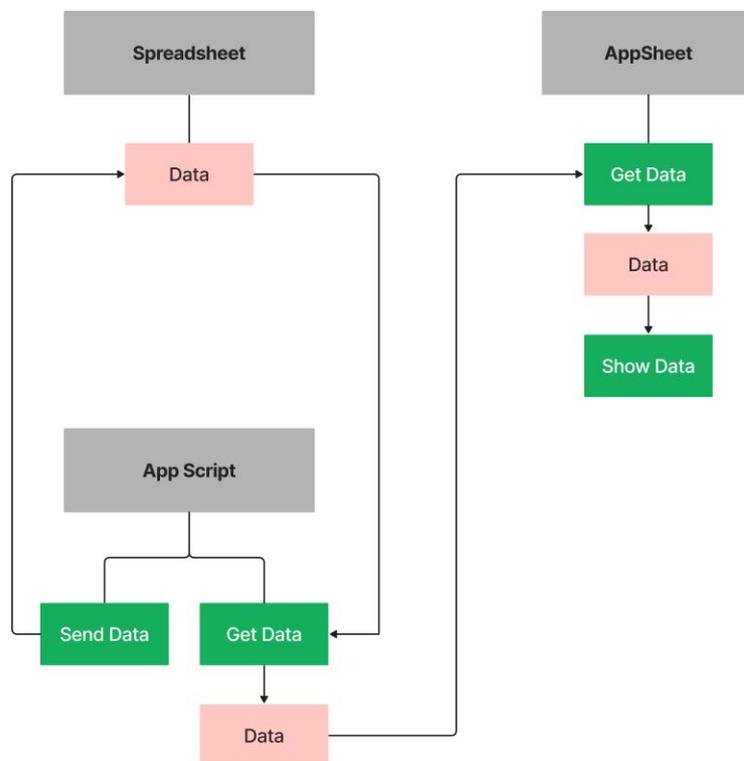
Implementasi sistem monitoring lemari pendingin vaksin Covid-19 dilakukan melalui jaringan WiFi dengan menggunakan modul NodeMCU ESP8266 berbasis mikrokontroler. Melalui modul ini data dari sensor diterima oleh mikrokontroler dan diteruskan oleh modul ESP8266 untuk dikirim ke *database* (Google Spreadsheet) melalui Apps Script. Dari *database* data tersebut akan dibaca dan ditampilkan ke *dashboard*.

Alur untuk perangkat keras dimulai dari pembacaan data suhu dan kelembapan oleh sensor BME280 dan sensor AHT10, serta pembacaan data latitude dan longitude oleh GPS module. Lalu data tersebut didapatkan oleh mikrokontroler dan data suhu di dalam lemari pendingin akan diolah di Arduino untuk dicek oleh fungsi warning system. Jika data suhu tersebut di luar batas suhu yang telah ditentukan maka, buzzer akan berbunyi. Untuk lebih jelas dapat dilihat pada Gambar 3.19.



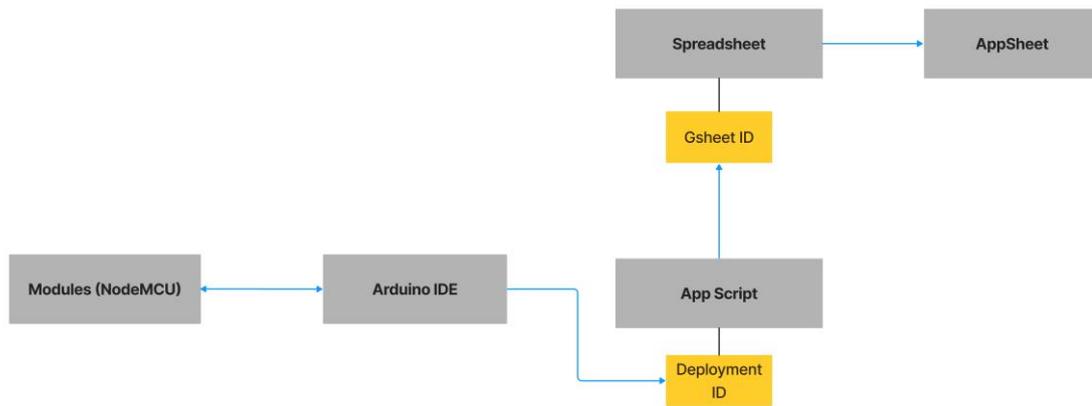
Gambar 3.19 Alur *Hardware*

Alur untuk perangkat lunak (*database* dan *dashboard*) dimulai dari pengiriman data ke database (Google Spreadsheet) melalui Apps Script. Lalu data yang ada di *database* akan diambil oleh AppSheet untuk dibaca dan ditampilkan pada *dashboard*. Untuk lebih jelas dapat dilihat pada Gambar 3.20.



Gambar 3.20 Alur *Software (Database & Dashboard)*

Alur untuk koneksi sistem dimulai dari modul sensor yang dihubungkan ke mikrokontroler NodeMCU. Lalu NodeMCU akan dihubungkan ke Arduino IDE dengan menanamkan kode di Arduino IDE ke dalam NodeMCU yang disambungkan dengan port USB. Lalu Arduino IDE dihubungkan ke *database* (Google Spreadsheet) melalui Apps Script. *Deployment ID* yang dimiliki oleh Apps Script akan dimasukkan ke kode pada Arduino IDE. Lalu *Gsheet ID* yang dimiliki oleh Google Spreadsheet akan dimasukkan ke kode pada Apps Script. Terakhir, AppSheet akan langsung terhubung pada Google Spreadsheet yang digunakan sebagai *database*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.21.



Gambar 3.21 Alur Koneksi Sistem