

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

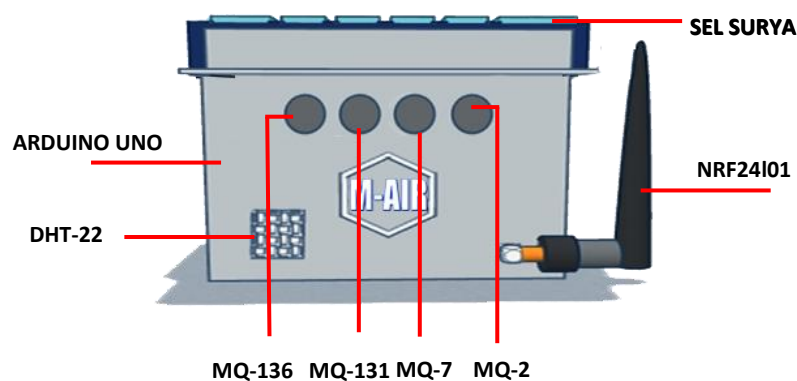
Pada bab ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak untuk menentukan kualitas udara dengan menggunakan metode *Finite State Machine*.

### 5.1 Perancangan Sistem

Perancangan sistem monitoring dan menentukan tingkat kualitas udara ini dimulai dari perancangan *prototype* alat, perancangan keras hingga perancangan perangkat lunak.

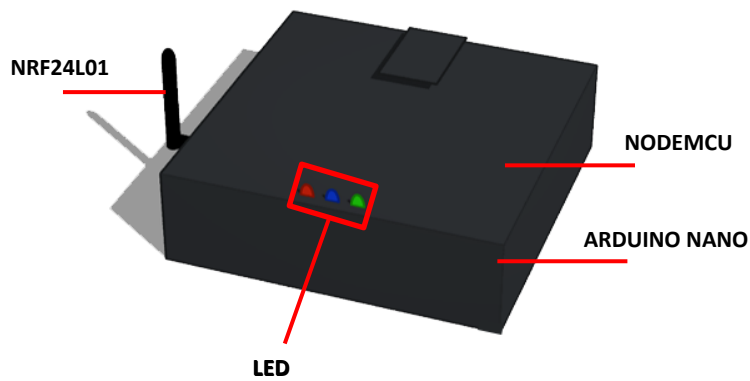
#### 5.1.1 Perancangan *Prototype* Alat

Dalam mendesain *prototype* alat harus diperhatikan letak – letak tiap komponen agar nantinya dalam implementasi dapat diterapkan sesuai dengan yang diharapkan. Pada perancangan *prototype* ini terdapat 2 desain yaitu desain di bagian *Main Device* dan *Hub Device*. Untuk desain *prototype* *Main Device* ditunjukkan pada gambar 5.1.



**Gambar 5. 1 Desain Prototype Main Device**

Pada Gambar 5.1 adalah desain *prototype* dari *Main Device*. Semua sensor diletakkan di bagian depan dimana di bagian atas terdapat sel surya yang digunakan sebagai sumber daya dari alat ini. Dan tampak bagian samping alat ini terdapat antena yang dimana itu adalah antena NRF24I01 yang digunakan untuk mengirim data ke *Hub Device*. Alat ini nanti akan ditempatkan di sudut – sudut perkotaan yang masih dapat dijangkau oleh masyarakat.

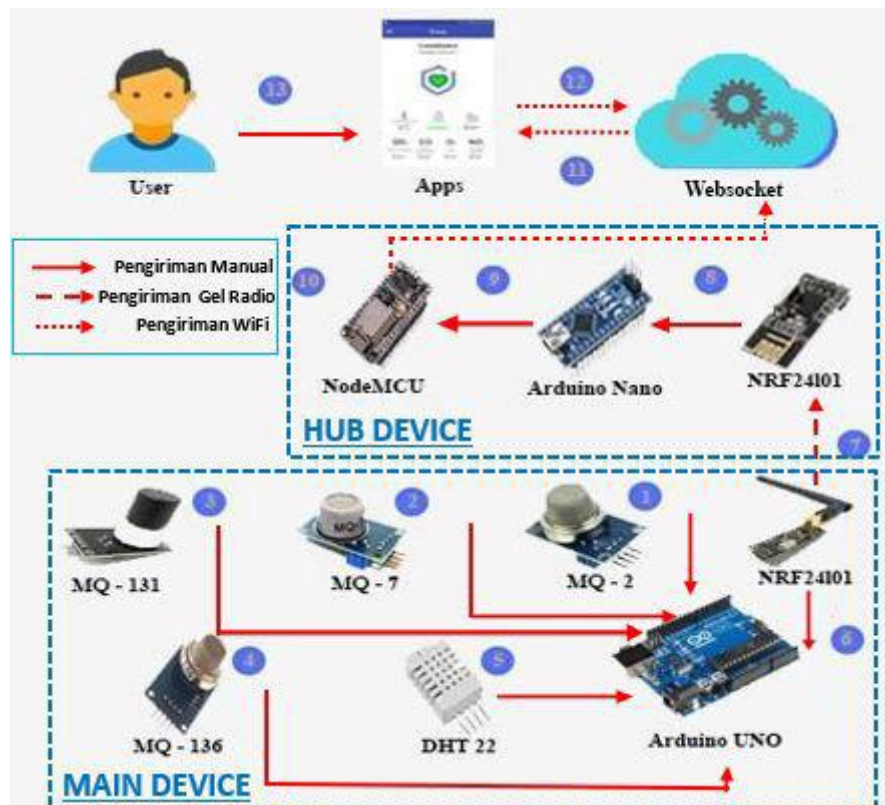


Gambar 5. 2 Desain Prototype Hub Device

Pada Gambar 5.2 adalah desain prototype dari *Hub Device*. Alat ini terbungkus dengan bentuk balok berwarna hitam. Di dalamnya terdapat 2 board yaitu Arduino Uno dan NodeMCU. Di bagian atas terdapat 3 buah led, led merah, led biru, dan led hijau. Kemudian di bagian samping tampak ada antena dari NRF24L01.

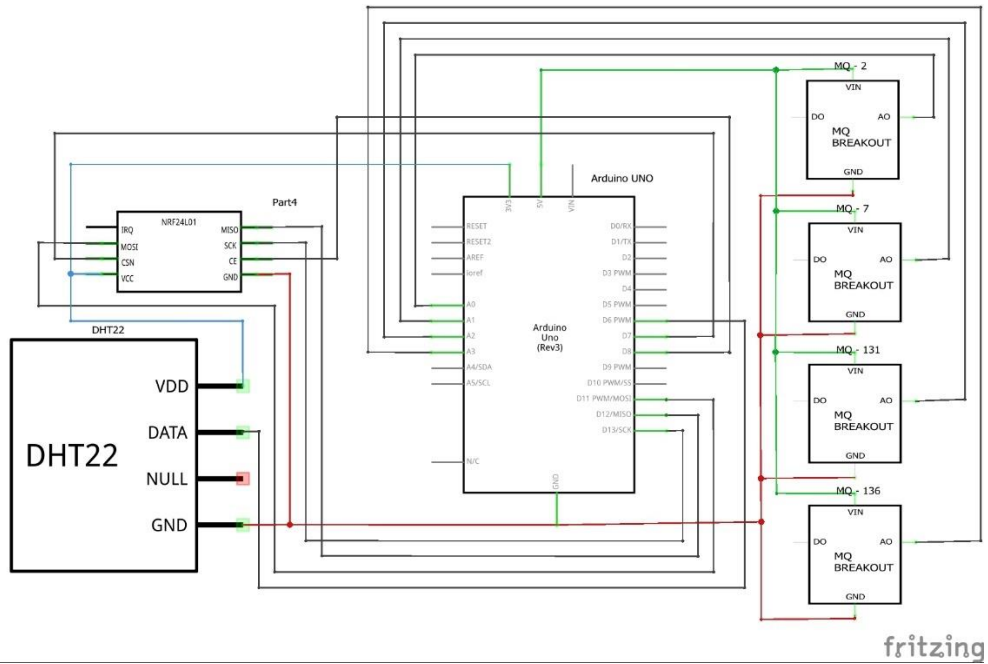
### 5.1.2 Perancangan Perangkat keras

Pada sub bab ini, menjelaskan tentang perancangan sistem dari segi perangkat keras yang meliputi penjelasan blok diagram, skematik, dan desain prototype.



Gambar 5. 3 Skema Kerja Sistem

Gambar 5.3 merupakan skema kerja dari sistem ini, terbagi menjadi 2 perangkat yang berbeda yaitu *Main Device* dan *Hub Device*. Input dari blok diagram tersebut yaitu sensor MQ -2, MQ – 7, MQ -131, MQ – 136, DHT22, sel surya. Kemudian dibagian prosesnya terdapat Arduino UNO dan NRF24I01 yang terdapat pada *Main Device*. Sedangkan dari *Hub Device* terdapat Arduino Nano, NRF24I01, dan NodeMCU. Dan untuk outputnya, adalah aplikasi pada *smartphone*.



**Gambar 5. 4 Skematik *Main Device***

Gambar 5.4 merupakan skematik dari *Main Device* yang terdapat 5 buah sensor dan 1 board NRF24I01 yang semuanya terhubung dengan board Arduino UNO. Ke empat pin analog pada Arduino UNO yaitu A0, A1, A2, A3 terhubung dengan sensor MQ.

**Tabel 5. 1 Konfigurasi pin sensor MQ - 2 dengan Arduino UNO**

Pin MQ – 2	Pin Arduino UNO
VIN	5V
GND	GND
AO	A0

Tabel 5.1 menunjukkan konfigurasi pin sensor MQ – 2 dengan Arduino UNO secara terperinci. Pada sistem ini menggunakan *module* sensor gas MQ – 2 yang mempunyai 4 buah pin diantaranya AO, DO, GND dan VIN. Namun DO tidak dipakai, sehingga yang digunakan hanya 3 pin. Sehingga konfigurasinya VIN akan dihubungkan dengan pin 5V arduino UNO, GND akan dihubungkan dengan GND, dan AO akan dihubungkan dengan A0 pada Arduino Uno.

**Tabel 5. 2 Konfigurasi pin sensor MQ - 7 dengan Arduino UNO**

Pin MQ – 7	Pin Arduino UNO
VIN	5V
GND	GND
AO	A1

Selain MQ – 2, terdapat sensor MQ – 7 yang terhubung dengan Arduino Uno. Sama seperti konfigurasi sensor MQ – 2, MQ – 7 ini terdapat 4 buah pin yaitu VIN, GND, DO dan AO. VIN akan terhubung dengan 5V pada Arduino Uno, GND akan terhubung juga ke GND Arduino Uno, dan AO akan terhubung dengan pin A1 pada Arduino Uno.

**Tabel 5. 3 Konfigurasi pin sensor MQ - 131 dengan Arduino UNO**

Pin MQ - 131	Pin Arduino UNO
VIN	5V
GND	GND
AO	A2

Tabel 5.3 merupakan konfigurasi dari pin sensor MQ – 131 dengan Arduino Uno. Untuk VIN pada MQ – 131 akan terhubung dengan 5V pada Arduino Uno. Kemudian GND sensor, akan terhubung dengan GND Arduino Uno. Dan yang terakhir AO pada sensor, akan terhubung dengan pin A2 pada Arduino Uno.

**Tabel 5. 4 Konfigurasi pin sensor MQ - 136 dengan Arduino UNO**

Pin MQ - 136	Pin Arduino UNO
VIN	5V
GND	GND
AO	A3

Sensor MQ – 136 adalah sensor yang keempat yang dipakai didalam sistem ini. Tabel 5.4 adalah konfigurasi pinnya dengan Arduino Uno. Sama seperti ketiga sensor sebelumnya, terdapat 3 pin yang dipakai. VIN terhubung dengan 5V pada Arduino Uno. GND pada sensor terhubung dengan GND pada Arduino Uno. Dan AO pada sensor terhubung dengan pin A3 pada Arduino Uno.

**Tabel 5. 5 Konfigurasi pin sensor DHT22 dengan Arduino UNO**

Pin DHT22	Pin Arduino UNO
VDD	3V3
GND	GND
DATA	D6

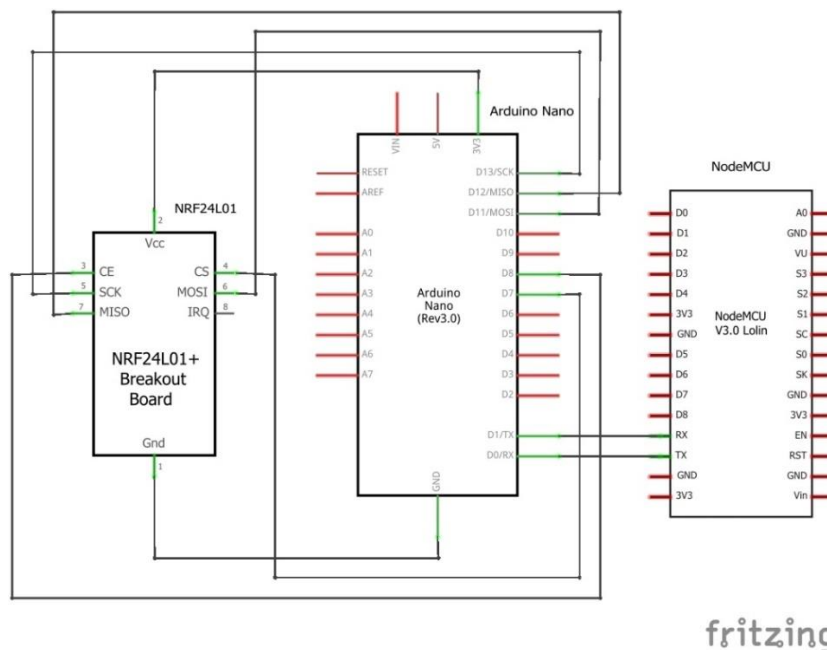
Untuk sensor DHT 22 ini merupakan sensor tambahan yang tidak termasuk kedalam menentukan kualitas udara, yaitu hanya mendeteksi suhu udara. Tabel

5.5 merupakan konfigurasi pin sensor DHT22 yang memiliki 4 buah pin yaitu VDD, DATA, GND, dan NULL. Tetapi yang dipakai hanya 3 pin saja VDD, DATA dan GND. VDD pada sensor akan terhubung dengan pin 3V3 pada Arduino Uno. Kemudian DATA pada sensor akan terhubung dengan pin D6 pada Arduino Uno.

**Tabel 5. 6 Konfigurasi pin sensor NRF24L01 dengan Arduino UNO**

Pin NRF24L01	Pin Arduino UNO
MOSI	D11
CSN	D7
VCC	3V3
MISO	D12
SCK	D13
CE	D8
GND	GND

Modul terakhir yang terhubung ke Arduino Uno adalah NRF24L01. NRF ini digunakan untuk komunikasi data ke server. NRF24L01 memiliki 8 pin, yaitu GND, VCC, CE, CSN, SCK, MOSI, MISO, dan IRQ. Namun IRQ tidak dipakai, sehingga yang terhubung ke Arduino Uno hanya 7 pin dari *module* NRF24L01. Pin yang pertama ada MOSI yang terhubung dengan pin D11 pada Arduino Uno. Kemudian CSN akan terhubung dengan D7. Yang ketiga terdapat pin VCC yang terhubung dengan 3V3 pada Arduino Uno. Selanjutnya terdapat pin MISO yang terhubung ke D12. SCK akan terhubung dengan pin D13. Kemudian pin CE akan terhubung dengan pin D8. Dan terakhir pin GND terhubung dengan pin GND pada Arduino Uno.



**Gambar 5. 5 Skematik Hub Device**

Gambar 5.5 merupakan skematik dari *Hub Device* yang merupakan node server dalam sistem ini. Terdapat 3 board dalam perangkat ini, yaitu Arduino

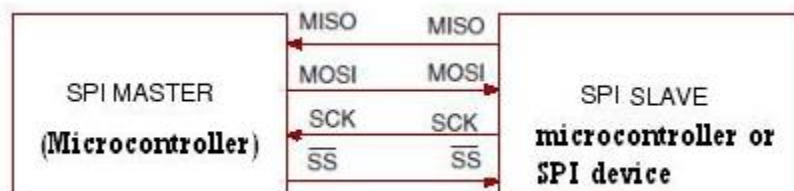
Nano, NRF24L01, dan NodeMCU. Arduino Nano sangat diperlukan dalam perangkat ini, karena digunakan untuk mengolah data dalam penerapan *Finite State Machine* dengan menerima data melalui NRF24L01. Sedangkan NodeMCU ini adalah board yang digunakan untuk local server yang nantinya akan menyediakan layanan WiFi agar pengguna dapat mengakses data. NodeMCU ini berperan penting untuk mendapatkan data dari Arduino Nano yang sudah terhubung dengan NRF24L01. Pengiriman data antara Arduino Nano dengan NodeMCU yaitu secara serial dengan menggunakan pin TX RX.

**Tabel 5. 7 Konfigurasi pin sensor NRF24L01 dengan Arduino Nano**

Pin NRF24L01	Pin Arduino Nano
MOSI	D11
CSN	D7
VCC	3V3
MISO	D12
SCK	D13
CE	D8
GND	GND

Di *Hub Device* ini juga terdapat NRF24L01 yang digunakan untuk komunikasi data ke *Main Device*. Seperti penjelasan pada *Main Device*, NRF24L01 ini memiliki 8 pin, yaitu GND, VCC, CE, CSN, SCK, MOSI, MISO, dan IRQ. Namun IRQ tidak dipakai, sehingga yang terhubung ke Arduino Nano hanya 7 pin dari *module* NRF24L01. Pin yang pertama ada MOSI yang terhubung dengan pin D11 pada Arduino Nano. Kemudian CSN akan terhubung dengan D7. Yang ketiga terdapat pin VCC yang terhubung dengan 3V3 pada Arduino Nano. Selanjutnya terdapat pin MISO yang terhubung ke D12. SCK akan terhubung dengan pin D13. Kemudian pin CE akan terhubung dengan pin D8. Dan terakhir pin GND terhubung dengan pin GND pada Arduino Nano.

Pada pin 6 dan 7 NRF24L01 terdapat MOSI dan MISO. Di penerapan NRF24L01 ke Arduino ini, menggunakan library Mirf. Sehingga untuk pin MOSI yang terdapat pada NRF24L01 terhubung ke pin 11 di Arduino. Sedangkan MISO terhubung ke pin 12 Arduino. MOSI adalah singkatan dari Master Out Slave In. MOSI ini adalah jalur data dari master dan masuk ke dalam slave. Sedangkan MISO adalah jalur data keluar dari slave dan masuk ke dalam master.



**Gambar 5. 6 SPI Diagram**

SPI (*Serial Peripheral Interface Bus*) adalah spesifikasi antarmuka komunikasi serial sinkron yang digunakan untuk komunikasi jarak pendek, terutama pada sistem embedded. Perangkat SPI berkomunikasi dalam mode full duplex menggunakan arsitektur master-slave dengan satu master. Perangkat induk berasal dari frame untuk membaca dan menulis. Beberapa perangkat slave didukung melalui seleksi dengan slave pilih individu (SS).

**Tabel 5. 8 Konfigurasi pin sensor NodeMCU dengan Arduino Nano**

Pin NodeMCU	Pin Arduino Nano
TX	RX
RX	TX

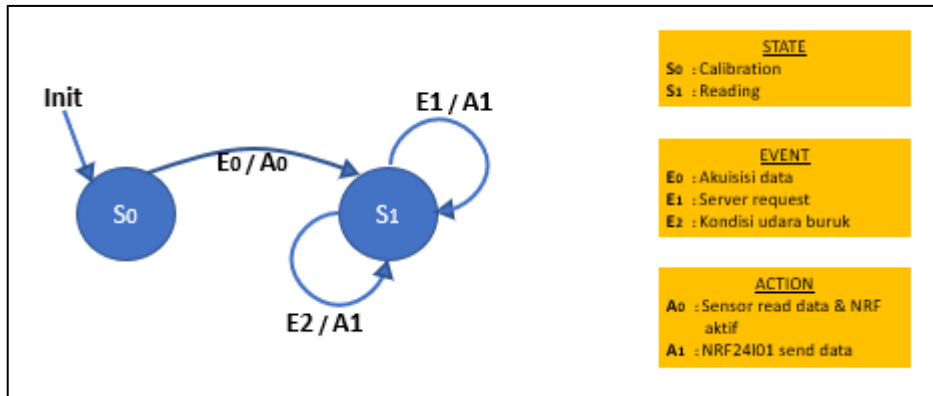
Pada board NodeMCU, pin yang digunakan hanya 2 untuk tersambung dengan Arduino Nano yaitu pin TX dan RX. Pin TX pada NodeMCU terhubung dengan pin RX pada Arduino Nano. Dan pin RX pada NodeMCU terhubung dengan pin TX pada Arduino Nano.

### 5.1.3 Perancangan Perangkat Lunak

Pada sub bab ini, menjelaskan tentang perancangan sistem dari segi perangkat lunak yang meliputi penjelasan dari diagram *state machine* dan *flowchart*.

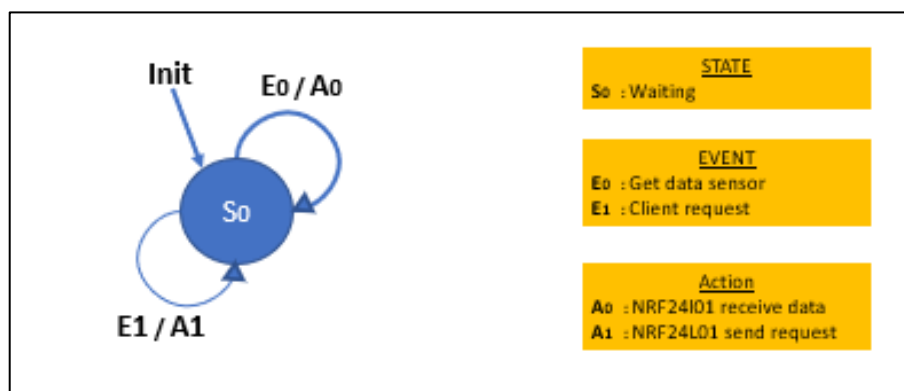
#### 5.1.3.1 Perancangan *Finite State Machine*

Di sistem ini menggunakan Finite State Machine sebagai algoritma yang di terapkan di *Main Device* dan *Hub Device* dikarenakan dengan menggunakan library *Finite State Machine* ini, microcontroller akan lebih ringan dalam memproses program dan tidak semua fungsi – fungsi yang terdapat di kode program dijalankan, melainkan hanya fokus terhadap state yang dihadapi. Di sistem ini *Finite State Machine* dapat menjalankan *State* sesuai kondisinya. Sehingga dapat memberikan keputusan dengan tepat sesuai dengan yang telah didefinisikan.



**Gambar 5. 7 Diagram State Machine Node Sensor**

Pada Gambar 5.7 merupakan diagram *State Machine* pada Node Sensor yang memiliki 2 *state* (keadaan) di dalam sistemnya. Ketika alat dinyalakan, maka sistem akan berada di S0, yaitu keadaan dimana sensor sedang melakukan kalibrasi sensor. State kalibrasi ini akan bertansisi ke state selanjutnya ketika mendapatkan trigger waktu selama 32 detik. Ketika waktu pemrosesan sudah 32 detik, maka state akan berpindah ke S1 yaitu state *reading*. Di state ini, sensor melakukan akuisisi data dan mengkonversi data yang diterima menjadi satuan ppm. Di state ini, terdapat 2 trigger yaitu ketika kondisi udara tidak sehat dan menerima request dari *user*. Sehingga ketika state mendapatkan trigger oleh kedua kondisi tersebut, maka data akan dikirim ke Hub Device melalui NRF24L01.

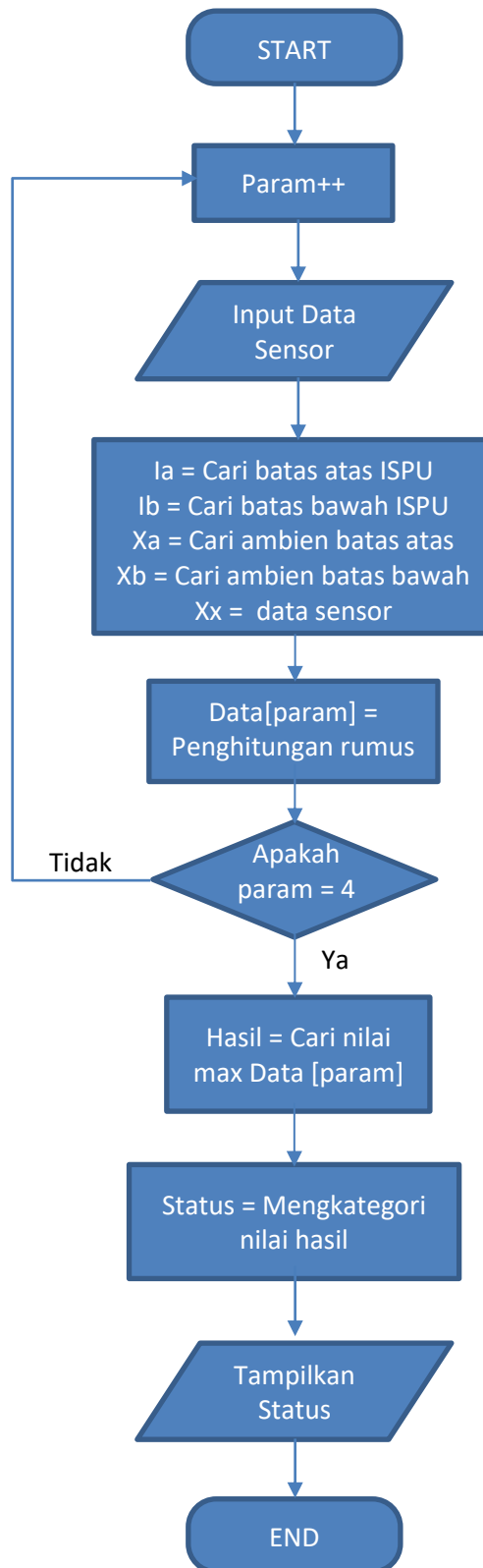


**Gambar 5. 8 Diagram State Machine Node Server**

Di sistem ini juga terdapat Node Server yang digunakan sebagai broker untuk mengakses data. Gambar 5.8 merupakan diagram state machine dari node server. Ketika alat dinyalakan, maka di node server ini menjalankan 2 kondisi secara bersamaan, yaitu S0 dan S1. S0 merupakan kondisi menunggu node sensor mengirimkan data. Dan ketika node sensor sudah menerima data maka akan looping terus ke kondisi S0 dan NRF24L01 melakukan 2 fungsi yaitu receive dan send. Kemudian S1 merupakan kondisi node server untuk menunggu request dari user. Sehingga ketika user request data maka akan meminta data ke node sensor dan diteruskan ke aplikasi melalui WiFi pada board NodeMCU.



### 5.1.3.2 Perancangan ISPU



Gambar 5. 9 Flowchart ISPU

Gambar 5.9 di atas merupakan *flowchart* ISPU yang diterapkan di aplikasi Android. Ketika data telah diterima oleh aplikasi, maka aplikasi akan melakukan perhitungan ISPU ini untuk mengkategorikan data yang masuk ke dalam kategori yang terdapat di tabel ISPU. Sehingga nanti setelah selesai melakukan perhitungan ISPU, hasil kategori tersebut akan ditampilkan di aplikasi di bagian status.

### 5.1.3.3 Perancangan GUI (Graphical User Interface) Aplikasi Android

Output dari sistem ini adalah tampilan kadar polusi beserta statusnya yang terdapat di aplikasi Android. Oleh karena itu diperlukan untuk perancangan GUI pada aplikasi Android tersebut. Perancangan GUI ini dilakukan pada software Android Studio. Tampilan di aplikasi ini terbagi menjadi beberapa komponen, yaitu:

1. **Nama Aplikasi**

Nama Aplikasi ini terdapat di bagian atas sendiri dengan nama aplikasi M-AIR

2. **Lokasi**

Maksud dari lokasi disini adalah lokasi dimana smartphone itu berada. Jadi ketika awal membuka aplikasi ini, pengguna akan diminta untuk menyalakan gps.

3. **Icon**

Di bagian tengah dari tampilan aplikasi terdapat icon love yang menggambarkan status dari setiap kondisi kualitas udara yang dideteksi dengan beragam warna. Hijau artinya kondisi normal, biru jika kondisi sedang, kuning ketika kondisi tidak sehat, merah jika kondisinya sangat tidak sehat, dan hitam ketika kondisinya berbahaya.

4. **Kadar tiap parameter**

Selanjutnya terdapat informasi dari kadar tiap parameter. Parameter tersebut yaitu Smoke, CO, SO<sub>2</sub>, NO<sub>2</sub>, O<sub>3</sub>, dan suhu. Selain itu terdapat status untuk menentukan kualitas udaranya.

5. **Button request data**

Di bagian paling bawah terdapat tombol request data yang digunakan untuk meminta data ketika kondisi udaranya normal.

## 5.2 Implementasi

Implementasi merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan perancangan yang telah dilakukan sebelumnya. Pada sub bab ini menjelaskan satu per satu secara rinci terkait implementasi *prototype*, implementasi perangkat keras serta implementasi perangkat lunak.

### 5.2.1 Implementasi Prototype Alat

Dalam mengimplementasikan *prototype* alat monitoring kualitas udara ini mengacu pada perancangan di subbab 5.1.1 dengan menggunakan 2 device, yaitu *Main Device* dan *Hub Device*. Untuk *Main Device* berbentuk balok berwarna putih dengan ukuran 13.5 x 9.5 x 8 cm<sup>3</sup>. Bahan yang digunakan dalam membuat *Main Device* ini yaitu menggunakan akrilik putih dengan ketebalan 22mm. Kemudian untuk *Hub Device* berbentuk balok juga dengan ukuran 18 x 17 x 6 cm<sup>3</sup>. Untuk bahannya menggunakan karton 2.5mm. Hasil implementasi *prototype* beserta peletakkan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.8 dan Gambar 5.10 berikut.



**Gambar 5. 10 Implementasi Prototype *Main Device***

Gambar 5.10 menunjukkan prototype dari *main device*. Tampak dari depan alat terdapat 5 buah sensor. Dan terdapat antena NRF24L01 yang terdapat pada sisi kanan alat. Dan dibagian atas terdapat panel surya yang digunakan sebagai *power supply* dari *main device*. *Main Device* ini nanti akan diletakkan di satu titik tertentu untuk mendeteksi dari kadar polusi.

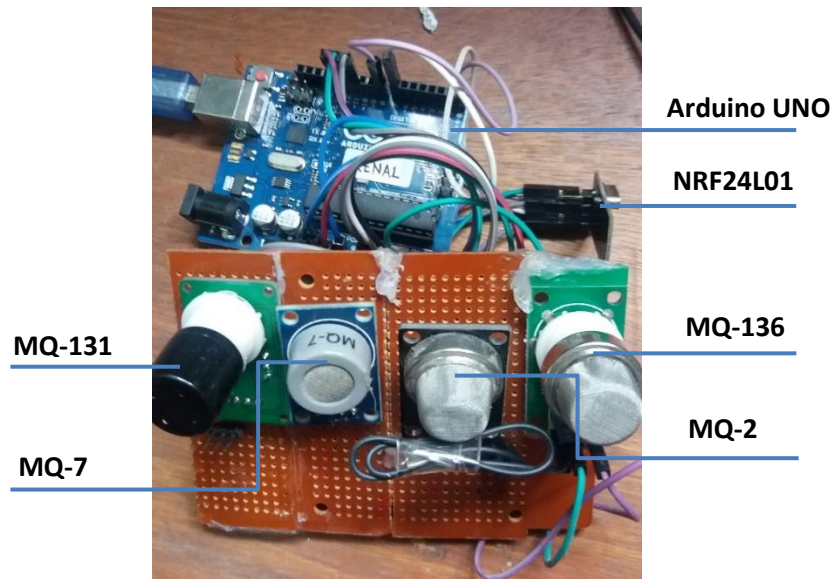


**Gambar 5. 11 Implementasi Prototype *Hub Device***

Gambar 5.11 adalah implementasi *prototype hub Device* yang memiliki desain sederhana berbentuk kotak dan disampingnya terdapat antena antena NRF24L01 yang digunakan untuk mendapatkan data dari *main device*. *Hub Device* ini akan diletakkan di dalam area yang tidak jauh dari *Main Device*. *Hub Device* ini dirancang terpisah karena membutuhkan hotspot untuk mendapatkan IP local agar dapat diakses oleh pengguna aplikasi. Selain itu *Hub Device* ini sebagai *Access Point* jika *Main Devicenya* banyak.

### **5.2.2 Implementasi Perangkat Keras**

Pada sub bab ini menjelaskan proses dari implementasi perangkat keras yang mencakup komponen elektronik. *Main Device* memiliki beberapa komponen di dalamnya, yaitu Arduino Uno, sensor MQ2, MQ7, MQ131, MQ136, DHT22, NRF24L01 dan Sel Surya. Sedangkan dari *Hub Device* terdiri dari Arduino Nano, NRF24L01, Node MCU, power bank.



**Gambar 5. 12 Komponen *Main Device***

Pada Gambar 5.12 menunjukkan hasil implementasi dari sensor MQ2, MQ7, MQ131, M136, DHT22 yang terhubung ke Arduino Uno. Dari kelima sensor tersebut terhubung dengan Arduino Uno dengan menggunakan kabel *jumper* dan komponen sensor dirangkai pada PCB.



**Gambar 5. 12 Komponen *Hub Device***

Gambar 5.13 merupakan perancangan dari *Hub Device* yang terdiri dari beberapa komponen didalamnya seperti yang dijelaskan pada subbab sebelumnya.

### 5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada subbab ini menjelaskan proses dalam menjalankan program dengan menggunakan metode FSM (*Finite State Machine*) dari Alat Monitoring Polusi Udara berdasarkan perancangan yang telah dilakukan. Dalam mengimplementasikan perangkat lunak ini, terdapat 2 software yang dipakai yaitu Arduino IDE 1.6.12 dan Android Studio 1.4.1.

#### 5.2.3.1 Implementasi *Finite State Machine*

Pada implementasi ini, sistem menggunakan Arduino Uno untuk memprogram dari mikrokontroler Arduino UNO dengan menggunakan bahasa pemrograman C.

**Tabel 5. 9 Algoritma *Main Device* Pada Mikrokontroler**

No	Algoritma
1	Mulai
2	Inisialisasi pin sensor, inisialisasi input dan output
3	Kalibrasi sensor MQ2, MQ7, MQ131, MQ136
4	while true
5	Baca data sensor MQ2,MQ7,MQ131,MQ136,DHT22
6	Waiting
7	if PM10 > 150ppm or CO > 10ppm or O3 > 253 or SO2 >
8	365 or NO2 > 100
9	kirim seluruh nilai sensor ke server
10	else if client request
11	kirim seluruh nilai sensor ke server
12	endif
	endwhile

Tabel 5.9 merupakan algoritma yang diterapkan pada Mikrokontroler. Ketika alat dinyalakan, maka mikrokontroler akan melakukan inisialisasi pin sensor. Setelah selesai inisialisasi, maka alat akan melakukan kalibrasi sensor MQ dengan tujuan mendapatkan hasil pengukuran sesuai dengan standar masing – masing sensor. Setelah mendapatkan nilai R0 masing – masing sensor, maka terjadi perulangan untuk akuisisi data dan kondisi *state* waiting.

**Tabel 5. 10 Penerapan FSM Pada *Main Device***

No	Algoritma
1	#include "Fsm.h"
2	#define TRIGGER_EVENT 0
3	State state_calibration(&start_on, NULL, NULL);

4	State state_reading(&akuisisi_data, NULL, NULL);
5	Fsm fsm(&state_calibration);
6	start_on();
7	akuisisi_data();
8	kondisi_buruk();
9	send_server();
10	terima();
11	void setup(){
12	fsm.add_transition(&state_calibration, &state_reading,
13	TRIGGER_EVENT, NULL);}
14	void loop(){
15	fsm.run_machine();
16	}

Implementasi FSM pada Mikrokontroler *Main Device* terdapat 2 state, yaitu state *calibration* dan state *reading*. Ketika sistem pertama dinyalakan maka *Main Device* akan berada di state *calibration*. Kemudian akan ke state *reading* ketika sudah 32 detik. Pada state *reading* akan mengirim data ketika kondisi udara buruk dan ketika mendapatkan *request* dari *user*.

**Tabel 5. 11 Algoritma *Hub Device* Pada Mikrokontroler**

No	Algoritma
1	Mulai
2	While true
3	Waiting get data sensor
4	Waiting request
5	if kondisi udara buruk
6	Get data sensor
7	else if client request
8	Send request
9	Get data sensor
10	Send data
11	endif
12	endwhile

Tabel 5.11 merupakan algoritma dari *Hub Device* yang diterapkan pada Mikrokontroler Arduino Nano. Di algoritma *Hub Device* ini, terdapat 1 state dalam penerapan FSM, yaitu state *waiting*. Sehingga ketika alat dinyalakan, state *waiting* akan aktif. Jika kondisi udara buruk dan client *request*, maka *Hub Device* akan menerima data dari *Main Device*. Kemudian ketika sudah mendapatkan data, maka akan dikirim secara serial ke server.

**Tabel 5. 12 Penerapan FSM Pada Hub Device**

No	Algoritma
1	#include "Fsm.h"
2	State waiting_data(&terima, NULL, NULL);
3	Fsm fsm_waiting(&waiting_data);
4	request();
5	terima();
6	check_status();
7	send_server();
8	void loop(){
9	fsm_waiting.run.machine();
10	}

### 5.2.3.2 Implementasi ISPU

Pada implementasi ini, bahasa yang digunakan adalah java android. Di program android ini akan dilakukan komputasi data yang diterima dari perangkat keras dan menentukan kategori kualitas udara dengan menggunakan rumus ISPU (Indeks Standar Pencemaran Udara).

**Tabel 5. 13 Penerapan ISPU pada Android**

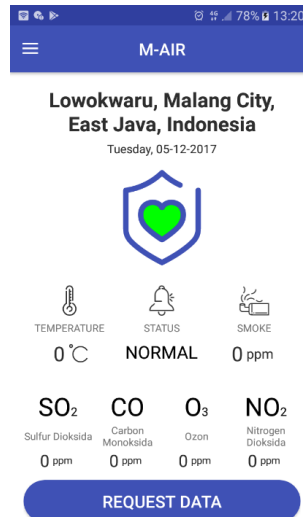
No	Algoritma
1	Mulai
2	Bagi data tiap parameter
3	Cari batas atas ISPU, batas bawah ISPU, ambien batas
4	bawah, ambien batas atas
5	Hitung dengan rumus ISPU
6	Cari nilai max ISPU tiap parameter
7	Tampilkan status ISPU

Pada tabel 5.13 menjelaskan tentang algoritma ISPU pada aplikasi Android. Di dalam algoritma ini, data tiap parameter akan dihitung nilai ISPUnya. Untuk penentuan kondisi statusnya yaitu dicari nilai ISPU maksimal dari semua parameter yang sudah dihitung.

### 5.2.3.3 Implementasi GUI (*Graphical User Interface*) Aplikasi Android

Untuk interface dari penelitian ini menggunakan platform android. Aplikasi *smartphone* android berfungsi sebagai interface yang secara langsung berhubungan dengan pengguna. Aplikasi ini dibuat dengan menggunakan software Android Studio.





**Gambar 5. 13 Tampilan GUI Aplikasi Android**

Pada gambar 5.13 adalah tampilan aplikasi pada sistem monitoring polusi udara ini. Di bagian atas terdapat nama aplikasi yaitu M-AIR dan dibawahnya adalah lokasi daerah alat dipasang. Selain itu terdapat fitur tambahan untuk tampilan hari dan tanggal. Di bagian tengah terdapat icon love dengan warna yang berbeda nantinya sesuai dengan kondisi udara. Selanjutnya terdapat info kadar dari 6 parameter yang dideteksi beserta status kondisi udaranya. Dan yang bagian bawah sendiri terdapat button untuk pengguna agar dapat *request* data.