

BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

3.1 ALAT DAN BAHAN

Dalam pembuatan alat monitoring suhu dan kelembaban runag server berbasis *internet of thing* dibutuhkan beberapa komponen seperti yang tertera pada tabel 3.1 dibawah ini:

Tabel 3.1 Alat dan Bahan

No	Alat dan Bahan	jumlah
1	Laptop	1
2	<i>Smartphone</i>	1
3	NodeMcu esp 8266	1
4	Relay	1
5	Sensor dht 11	1
6	<i>Software Arduino IDE</i>	1
7	<i>Software Whireshark</i>	1
8	Mit App Inventor	1
9	<i>Power supply</i>	1
10	<i>google firebase</i>	1
11	<i>Base board node mcu esp 8266</i>	1

3.1.1 Laptop

Pada penelitian tugas akhi ini laptop yang digunakan sebagai alat dalam mengelola seluruh data yang akan digunakan untuk terselesaikanya tugas akhir ini. Seperti pada proses pemrograman pada mikrokontroler, laptop digunakan untuk koding agar sistem yang dibangun dapat berjalan dengan baik. Spesifikasi laptop menggunakan prosesor AMD *Dual Core A9-9425 RADEON R5*, dengan kecepatan *clock* sebesar 3.7 GHz.

3.1.2 Smartphone

Smartphone yang digunakan memiliki minimum Sistem Operasi Android 2.3 (*Gingerbread*) agar aplikasi monitoring suhu dan kelembaban dapat berjalan. Pada penelitian tugas akhir ini menggunakan *smartphone* dengan sistem operasi android 7.1.2 (*Nougat*).

3.1.3 NodeMcu Esp8266

Pada *system* NodeMcu Esp8266 berfungsi sebagai *processor* yang digunakan untuk mengelola masukan dari sensor DHT 11. NodeMcu Esp8266 memiliki 13 pin GPIO, dengan standarisasi *wifi* IEEE 802.11 b/g/n.

3.1.4 Relay

Relay yaitu alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar, kemudian saklar yang digerakkan (secara mekanis) oleh daya/energi listrik [4].

3.1.5 Sensor DHT 11

DHT11 adalah salah satu sensor dapat mengukur dua parameter sekaligus yaitu suhu dan kelembaban udara. Keluaran sinyal pada DHT 11 adalah sinyal digital yang dikalibrasi dengan sensor suhu dan kelembaban, hal ini yang menyebabkan stabilitas kinerja sensor menjadi sangat baik dan responsif [5].

3.1.6 Software Arduino IDE

Pada penelitian tugas akhir, *Software Arduino IDE* digunakan untuk pengkodean pada NodeMcu Esp8266 agar sistem yang dibangun dapat berjalan dengan baik.

3.1.7 Software Wireshark

Aplikasi *Wireshark* Adalah Sebuah *Network Protocol Analyzer* untuk berbagai kebutuhan analisis unjuk kerja jaringan. *Wireshark* Difungsikan Untuk Menangkap (*Capturing*) setiap data yang lewat melalui salah *satu network interface* dari komputer di mana aplikasi tersebut terpasang. Pada penelitian tugas akhir ini *Wireshark* digunakan untuk mencari *Quality of Service*.

3.1.8 Mit App Inventor

Mit App Inventor adalah *tool* pemrograman berbasis *block* yang memungkinkan semua *user*, untuk memulai pemrograman dan membangun aplikasi yang berfungsi penuh untuk perangkat android.

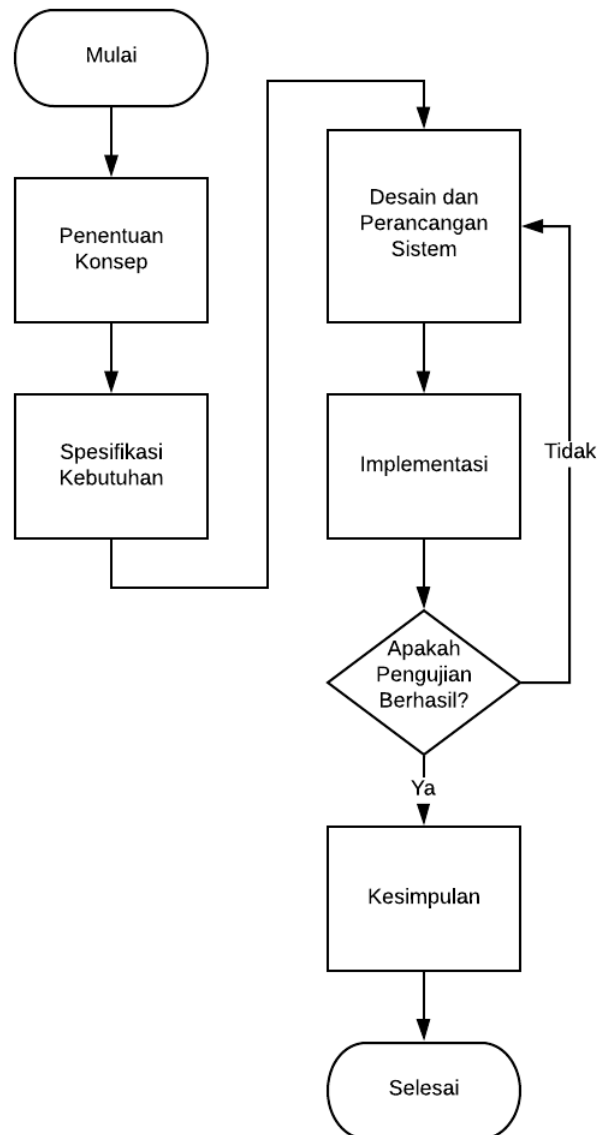
3.1.9 Power supply

Power supply digunakan sebagai sumber tegangan pada alat monitoring suhu dan kelembaban agar dapat berjalan dengan baik, pada penelitian tugas akhir ini menggunakan *Power supply* dengan tegangan *output* sebesar 12 volt.

3.1.10 Google Firebase

Google Firebase adalah salah satu layanan yang menyediakan *realtime* database yang mudah diakses dan digunakan oleh web developer, database sendiri yaitu sekumpulan informasi dan data yang tersimpan dan tersusun dalam computer secara sistematis yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program computer guna mendapatkan data atau informasi dari basis data tersebut[18].

3.2 ALUR PENELITIAN



Gambar 3.1 Flowchart Alur Penelitian

Penelitian dilakukan dalam 5 tahapan yaitu penentuan konsep, spesifikasi kebutuhan, desain dan perancangan sistem, implementasi perangkat keras dan lunak dan tahap terakhir yaitu pengujian. Pada tahap konsep perlu menentukan tujuan termasuk spesifikasi kebutuhan. Pembuatan sistem ini bertujuan untuk membuat alat monitoring suhu dan kelembaban ruang server berbasis *internet of things* dengan penggerak kipas report menggunakan aplikasi android.

Tahap spesifikasi kebutuhan merupakan tahap dimana terjadinya proses menganalisa dan memploting apa saja kebutuhan yang diperlukan dalam proses

penelitian. Mengumpulkan berbagai data yang dibutuhkan dalam pembuatan sistem seperti menganalisa dan membandingkan beberapa penelitian sebelumnya agar bisa mematangkan konsep sistem yang dibuat.

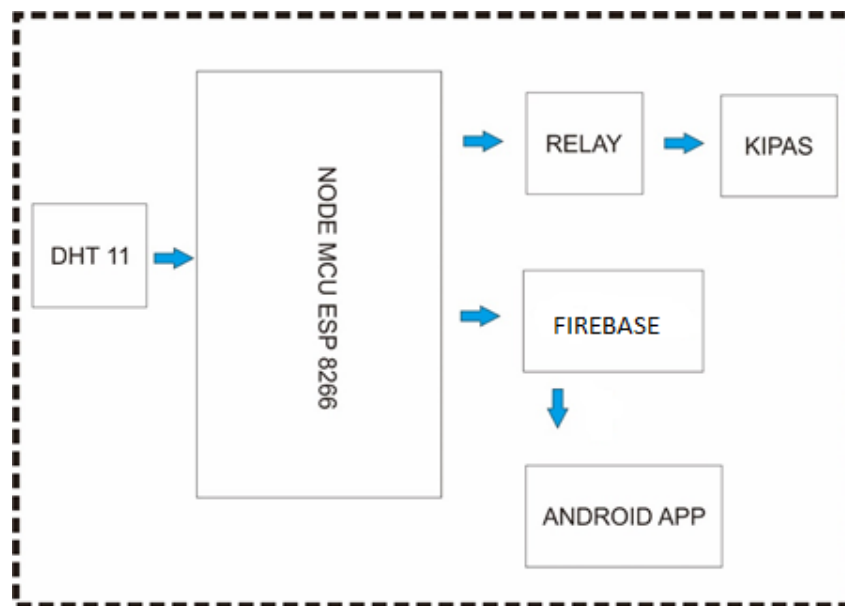
Tahap selanjutnya yaitu desain dan perancangan sistem dibuat sesuai dengan peralatan apa saja yang digunakan untuk membangun suatu sistem monitoring suhu dan kelembababn ruang server berbasis *internet of things* dengan notifikasi aplikasi android.

Pada implementasi, dari peralatan yang sudah dirancang kemudian diimplementasikan menjadi sebuah sistem yang nantinya bisa digunakan dengan baik dan diharapkan memudahkan penggunaanya, setelah perangkat keras sudah dibuat. Implementasi perangkat lunak bertujuan untuk membangun sebuah sistem dengan memberikan fungsi kontroling maupun fungsi yang kemudian menjadi perintah perintah internal dalam sebuah sistem.

Pada tahap pengujian pada tahap ini adalah kelanjutan dari tahap-tahap yang sudah dilalui dan sebagai tolak ukur bahwa suatu sistem yang sudah dirancang dan dibuat sudah pantaskah atau sudah siapkah digunakan atau belum, jika saat proses pengujian masih terjadi kesalahan kesalahan atau masih dirasa kurang maka akan kembali lagi ke tahap sebelumnya yaitu tahap desain dan perancangan sistem.

Tahapan yang terakhir adalah kesimpulan, kesimpulan bahwa sistem dapat bekerja dengan baik dan dapat diimplementasikan pada ruang *server*.

3.3 DIAGRAM *BLOCK* SISTEM



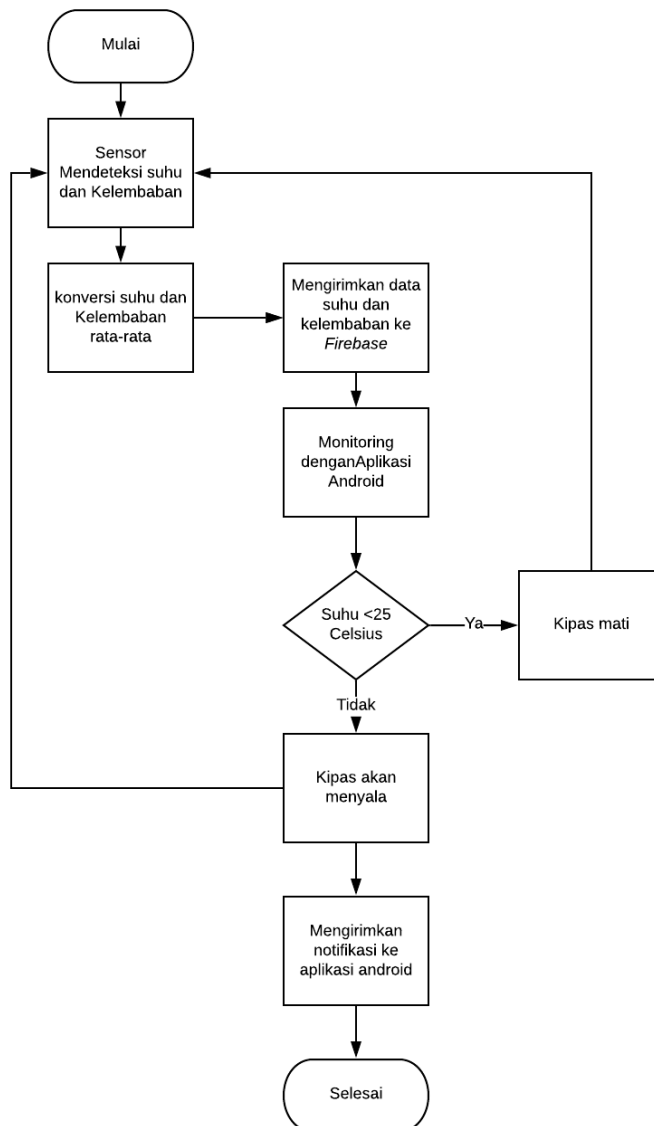
Gambar 3.2 Diagram *Block* Alat Pengendali

Berdasarkan diagram blok alat pengendali seperti pada gambar 3.2. terdapat 3 bagian pada bagian pertama yaitu input, proses dan *output*, Pada bagian input terdapat sensor DHT 11 yang berguna untuk mengukur suhu dan kelembaban baik di dalam atau diluar ruangan dengan kemampuan dapat mengukur dua parameter sekaligus yaitu suhu dan kelembaban udara. Sensor ini memiliki keluaran sinyal digital yang dikalibrasi dengan sensor suhu dan kelembaban. Hal ini membuat stabilitas kinerja sensor menjadi sangat baik dalam jangka panjang. Selain memiliki kualitas yang sangat baik, sensor ini memiliki respon cepat, kemampuan anti-gangguan dan keuntungan biaya karena dapat mengukur dua parameter sekaligus. Pada bagian pemrosesan terdapat Node MCU esp 8266, sebagai pengendali seluruh *system* di dalamnya berisi tentang program yang dijalankan dan bagian selanjutnya yaitu pada *output* ada *firebase* jadi data yang terbaca pada sensor akan di kirim ke *firebase*, *firebase* disini digunakan untuk media penyimpanan data berupa data suhu dan kelembaban yang terbaca pada sensor DHT 11 dan bisa di akses melalui aplikasi *smartphone* yang sangat memudahkan operator untuk memonitoring kondisi suhu dan kelembaban udaranya dan pada *system* ini terdapat relay dan kipas. *Relay* disini digunakan sebagai saklar otomatis yang terhubung dengan kipas, dan *relay* ini diprogram untuk mengaktifkan kipas jika suhu yang terbaca pada sensor dht 11

melebihi batas wajar yaitu 25 celsius dan kipas disini digunakan untuk membantu alat menurunkan suhu ruangan server.

3.3.1. FLOWCHART ALUR SISTEM.

Diagram alur penelitian menjelaskan mengenai alur pada *system* keseluruhan pada alat monitoring suhu dan kelembaban ruang *server* berbasis *internet of things* dengan penggerak kipas report menggunakan android app. Pada perancangan ini yaitu perancangan keseluruhan alat yang akan dibuat dengan menggunakan sensor DHT 11, *board* node mcu esp 8266, *relay* dan kipas serta beberapa kabel.

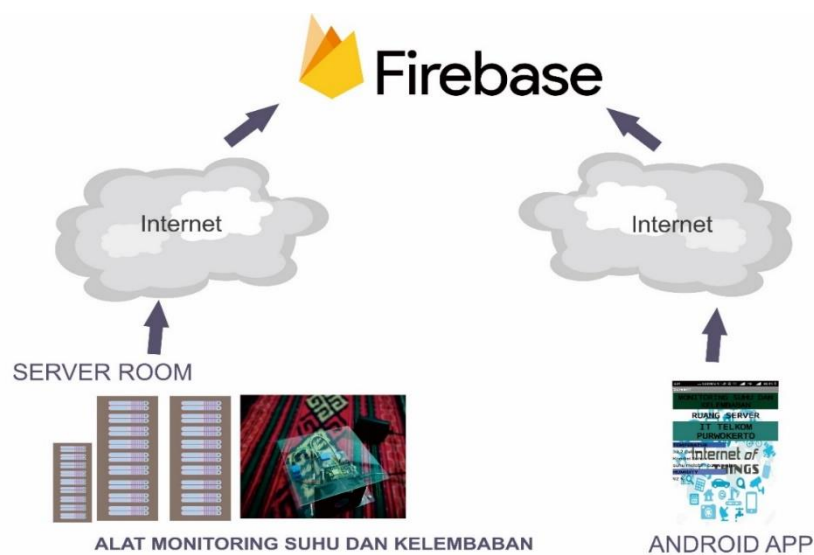


Gambar 3.3 Diagram alur sistem

Pada perancangan system monitoring ruang server berbasis *internet of things* dengan penggerak kipas report menggunakan aplikasi android. Dimulai dengan sensor dht 11 yang mendeteksi suhu diruang sekitar yaitu pada ruang server IT Telkom Purwokerto lalu data yang diperoleh oleh sensor dikonversi dan diupload ke *firebase*. Suhu yang ideal adalah 25 °C, sehingga saat sensor DHT 11 membaca suhu melebihi suhu ideal maka akan mengirimkan notifikasi suhu melewati batas wajar ke aplikasi android dan relay menyala sehingga menyebabkan kipas dalam kondisi aktif. Sedangkan jika suhu yang terdeteksi kurang dari batas ideal atau batas aman maka kondisi relay mati dan kipas dalam kondisi off.

3.3.2 PERANGKAT YANG DIGUNAKAN

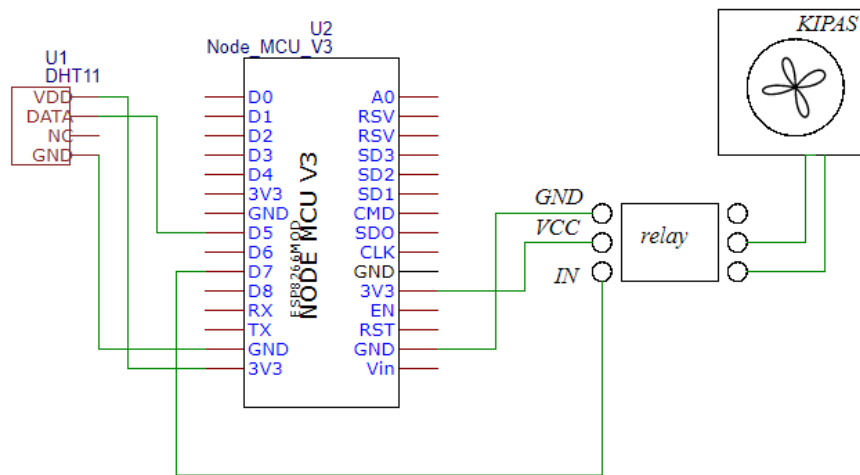
Dalam penelitian ini perangkat yang digunakan meliputi peralatan perangkat keras dan perangkat lunak. Pada gambar 3.2. Sensor DHT 11 merekam kondisi suhu dan kelembaban pada ruang *server room* yang kemudian data yang sudah direkam akan dikirimkan ke *firebase* sebagai media penyimpanan data suhu dan kelembaban kemudian aplikasi android yang sudah dibuat dengan MIT App Inventor, mengakses ke *firebase* untuk mengambil data yang sudah tersimpan dan akan ditampilkan pada aplikasi androidnya dan untuk mengakses *system* ini semua menggunakan koneksi internet dari proses *upload* data dari sensor DHT 11 ke *firebase* dan dari aplikasi android ke *firebase*.



Gambar 3.4 Perangkat yang digunakan.

3.3.3 PERANCANGAN PERANGKAT KERAS.

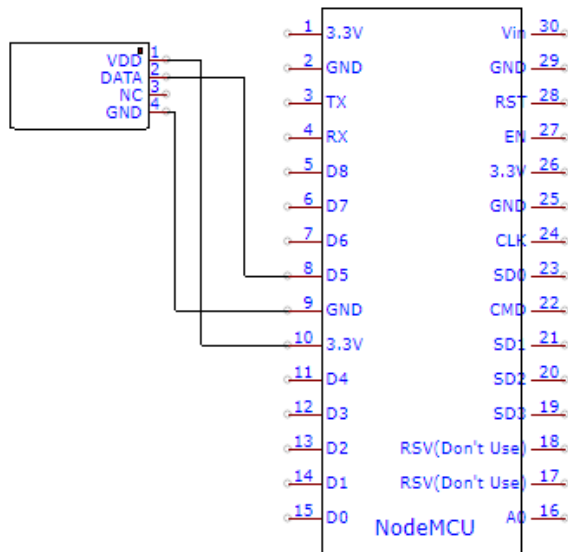
Pada perancangan monitoring suhu dan kelembaban ruang *server* berbasis *internet of things* dengan notifikasi menggunakan aplikasi android menggunakan beberapa peralatan yang membantu perancangan penelitian ini, yang akan membentuk perangkat monitoring suhu dan kelembaban ruang server.



Gambar 3.5 Perangkat Sistem Monitoring Suhu dan Kelembaban

3.3.3.1 Antarmuka NodeMcu dengan sensor DHT 11

Sensor DHT 11 sendiri pada penelitian ini digunakan untuk mengukur suhu dan kelembaban yang memiliki output sinyal digital yang telah terkalibrasi, sensor ini bisa diterapkan di suatu ruangan atau diluar ruangan. Sensor ini mencakup komponen pengukuran kelembaban resistif dan komponen pengukuran suhu NTC yang terhubung pada mikrokontroler 8 bit sehingga menawarkan kualitas yang baik, respon cepat dan kemampuan anti gangguan[5]. Pada gambar 3.6 menunjukan pembagian pin yang digunakan pada DHT 11 untuk mendeteksi suhu dan kelembaban.



Gambar 3.6 Koneksi pin NodeMcu dengan sensor dht 11

Koneksi pin antara NodeMcu dengan sensor DHT 11 dapat dilihat pada tabel 3.2 berikut:

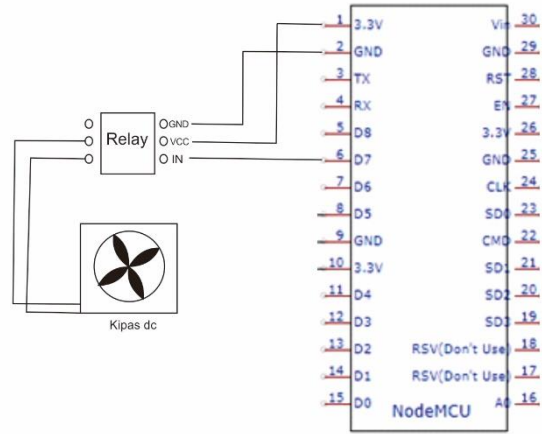
Tabel 3.2 koneksi antara NodeMcu dengan Sensor DHT 11

No	Pin	Fungsi
1	3.3v/ vcc	Catu daya sensor 3.3 V
2	GND	<i>Grounding</i>
3	D5	Pembacaan sensor di <i>port D5</i>

3.3.3.2 Antarmuka NodeMcu dengan Relay dan kipas

Pada tugas akhir ini menggunakan mikrokontroler NodeMcu esp 8266 yang digunakan untuk memenuhi kebutuhan dari monitoring suhu dan kelembaban ruang server berbasis *internet of things* dengan notifikasi aplikasi android pada *system relay* sebagai saklar otomatis yang akan aktif dan tidak sesuai kondisi yang ditetapkan pada system ini relay menghubungkan antara nodemcu dengan kipas dc dengan fungsi logika jika suhu yang terbaca oleh sensor DHT 11 melebihi 25 °C maka *relay* aktif dan akan menyalakan kipas dc, seperti pada gambar 3.6 *relay* terhubung ke sumber tegangan untuk eistem ini menggunakan sumber tegangan dari adaptor 12 volt dengan kabel negatif masuk ke kipas dc bagian negatif dan

tegangan positif masuk ke *relay*, dari *relay* dihubungkan ke positif kipas dc, pada bagian *port in* masuk ke pin D7 pada bagian NodeMcu dan GND dan Vcc masuk ke pin GND & Vcc.



Gambar 3.7 koneksi pin NodeMcu dengan *relay* dan kipas

Koneksi pin antara NodeMcu dengan *relay* dan kipas dapat dilihat pada tabel 3.3 berikut:

Tabel 3.3 koneksi antara NodeMcu dengan *relay* dan kipas

No	Pin	Fungsi
1	3.3v/ vcc	Catu daya sensor 3.3 V
2	GND	<i>Grounding</i>
3	D7	Pembacaan <i>relay</i> di <i>port</i> D7

3.3.4 PERANGKAT LUNAK UNTUK PENELITIAN.

3.3.4.1 ARDUINO IDE

Untuk memprogram board Arduino, kita butuh aplikasi IDE (*Integrated Development Environment*) bawaan dari Arduino. Aplikasi ini berguna untuk membuat, membuka, dan mengedit source code Arduino (*Sketches*), para programmer menyebut *source code* arduino dengan istilah ("*sketches*"). Selanjutnya, jika kita menyebut *source code* yang ditulis untuk Arduino, kita sebut

"*sketch*". *Sketch* merupakan *source code* yang berisi logika dan algoritma yang akan diupload ke dalam IC mikrokontroler (Arduino)[12].

```
#include <FirebaseArduino.h>
#include "DHT.h"
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
```

Gambar 3.8 Include Library pada Arduino IDE

Seperti pada gambar 3.8 menjelaskan untuk memasukan *file header* ke dalam program dibutuhkan perintah `#include` untuk memanggil *file header* tersebut, *include* seperti gambar 3.3.2.1 akan memanggil fungsi yang ada pada firebase arduino, fungsi DHT 11 dan fungsi yang lain sesuai yang dibutuhkan dalam pembentukan suatu program.

```
#define FIREBASE_HOST "percobaan-3cf9d.firebaseio.com"
#define FIREBASE_AUTH "dKCMvHV6Cb9QNPazJqEEcmVxIKhNLNgt1RIVmaA"
#define WIFI_SSID "aryaganteng"
#define WIFI_PASSWORD "12345678"

#define DHTPIN 14 // Data Pin of DHT 11 , for NodeMCU D5 GPIO no. is 14

#define DHTTYPE DHT11 // DHT 11
```

Gambar 3.9 Define pada Arduino IDE

`#define` yang tertera pada gambar 3.9 digunakan untuk mendefinisikan sebuah *variable* (pin input atau output dll) dengan nama *variable* baru. Fungsi mendefinisikan *variable* yaitu agar memudahkan dalam pembuatan program.

```
DHT dht(DHTPIN, DHTTYPE);
const int fan= 13;
```

Gambar 3.10 Pin pada Kipas

Pada program yang dibuat seperti tertera pada gambar 3.10 yaitu mendefinisikan kipas di tempatkan pada pin 13 jadi saat program berjalan, untuk pin ke 13 didefinisi untuk kipas.

```

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    pinMode (fan, OUTPUT);
    Serial.print(".");
  }
  dht.begin();
  Serial.println("");
  Serial.println ("WiFi Connected!");
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  float h = dht.readHumidity();

  float t = dht.readTemperature(); // Reading temperature as Celsius (the default)
  Firebase.setFloat ("Temp",t);
  Firebase.setFloat ("Humidity",h);
  delay(200);
}

```

Gambar 3.11 Void pada Arduino IDE

Void adalah sebuah fungsi yang ada dalam sebuah Bahasa pemrograman bisa pada c++ atau c#. Fungsi ini juga dapat disebut fungsi sebagai prosedur seperti pada gambar 3.11 menjelaskan prosedur yang berjalan pada program monitoring suhu dan kelembaban pada ruang *server* yang jika diringkas menjadi fungsi untuk membuat NodeMcu terhubung pada *wifi* menggunakan *ssid* pada gambar diatas juga menjelaskan bahwa fan digunakan sebagai *output*, saat *wifi* terhubung maka akan menghubungkan mikrokontroler ke *firebase* dan sensor suhu akan membaca suhu dilingkungan sekitar dan mengirimkan datanya ke *firebase*.

```

    if (t > 27.00){
      digitalWrite (fan,LOW);
    }

    else if (t < 27.00){
      digitalWrite (fan,HIGH);
    }
  }
}

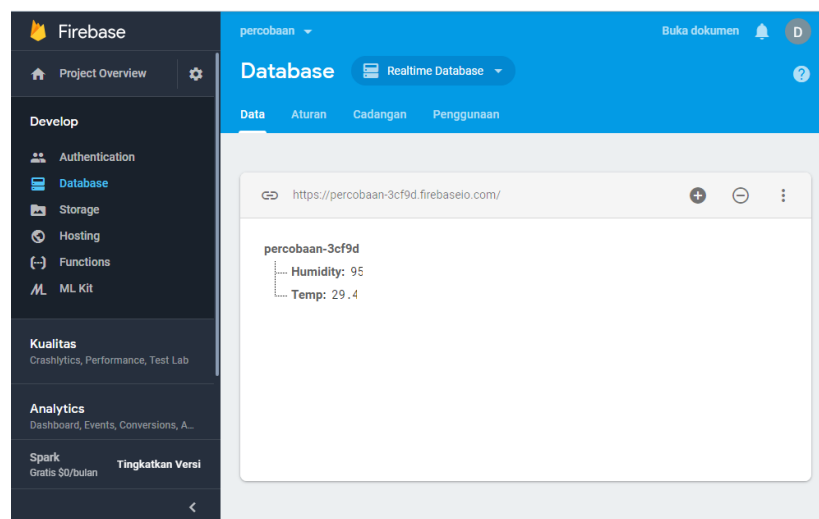
```

Gambar 3.12 Coding Relay

Pada gambar 3.12 merupakan *coding* untuk relay menggunakan fungsi *if else* sehingga menyebabkan sebab akibat pada gambar 3.12 menjelaskan bahwa jika suhu lebih dari 25 °C maka menyebabkan *relay* dalam kondisi *low* atau menyala sedangkan jika suhu kurang dari 25 °C maka akan menyebabkan *relay* dalam kondisi *high* atau mati.

3.3.4.2 FIREBASE

Firebase Realtime Database adalah database yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis.



Gambar 3.13 Tampilan Data FIREBASE

Pada gambar 3.13 setelah sensor DHT 11 membaca suhu dan kelembaban pada lingkungan sekitar dan telah diubah ke data digital maka NodeMcu mengirimkan data suhu dan kelembaban ke *firebase*.

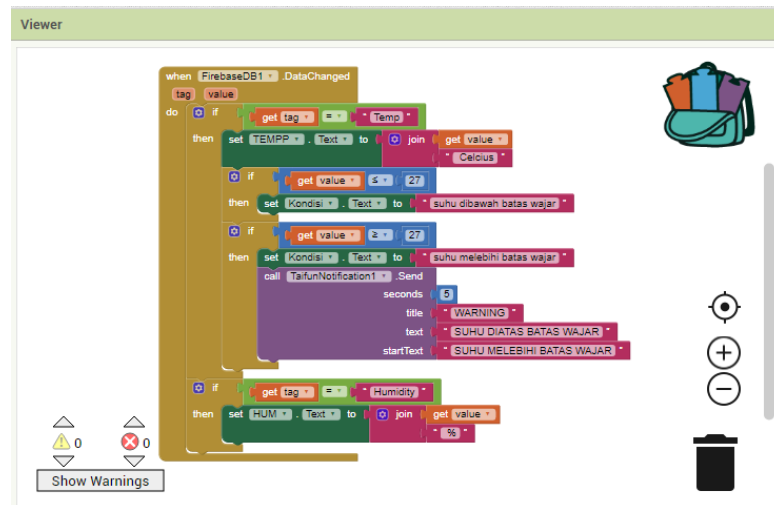


Gambar 3.14 komunikasi pada *firebase*

Pada gambar 3.14 menjelaskan bahwa *smartphone* yang sudah terhubung ke *internet* dan sudah dibekali aplikasi untuk memantau suhu dan kelembaban di ruang *server* akan dapat menampilkan data suhu dan kelembaban yang tersimpan dan terupdate di *firebase*. *Firestore* sendiri merupakan media penyimpanan data yang akan diupdate terus menerus atau *realtime* sehingga akan sangat membantu saat melakukan monitoring. NodeMcu digunakan untuk mengirim data suhu dan kelembaban ruangan ke *firebase*, sehingga *smartphone* dapat mendapatkan data suhu dan kelembaban.

3.3.4.3 MIT APP INVENTOR

MIT App Inventor adalah *tool* pemrograman berbasis blok yang memungkinkan semua *user*, untuk memulai pemrograman dan membangun aplikasi yang berfungsi penuh untuk perangkat android. Membuat aplikasi android itu bisa semudah bermain *LEGO* atau *PUZZLE*. Tidak harus mengerti bahasa pemrograman Android berbasis OOP dan Java yang pada umumnya menggunakan IDE seperti Eclipse atau netbean.[16]



Gambar 3.15 Tampilan *Block* Aplikasi di Mit App Inventor

Seperti yang dijelaskan diatas bahwa untuk menjalankan atau membuat aplikasi menggunakan Mit App Inventor, dan tidak perlu menguasai atau mengerti bahasa pemrograman android berbasis oop dan java karena Mit App Inventor adalah *tool* pemrograman berbasis *block* yang memungkinkan untuk memulai pemrograman dan membangun aplikasi yang berfungsi penuh untuk perangkat android.



Gambar 3.16 Tampilan Aplikasi Monitoring Suhu dan Kelembaban

