BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Penulis berkerja untuk IT department Universitas Multimedia Nusantara sebagai IoT engineer pada salah satu proyek yang digagaskan oleh tim IT department untuk membuat suatu perangkat Internet of Things (IoT) yang dapat memonitoring lingkungan ruang server Universitas Multimedia Nusantara. Di dalam IT department penulis berada dibagian IT infrastructure.

3.2 Tugas dan Uraian Kerja Magang

Penulis tergabung dalam proyek membangun sistem monitoring lingkungan data center UMN. Target dari proyek dibuat penulis ini adalah untuk membangun dan merancang sebuat alat Internet of Things (IoT) cerdas yang dapat mengawasi beberapa aspek, seperti suhu, kelembapan, keberadaan api, asap, dan sensor pintu yang dapat diawasi melalui website. Alat IoT yang aktif dan terkoneksi dengan internet akan mengirim data sensor yang ada secara real-time ke website yang bisa diakses oleh tim IT UMN untuk memonitoring keadaan lingkungan ruang server.

3.2.1 Tugas Kerja Magang

Tugas yang dilakukan oleh penulis adalah membangun segala yang dibutuhkan untuk membuat alat IoT monitoring ini, mulai dari mempersiapkan perangkat dan sensor, merakit, mengkoding, dan membuat kotak proyek dari awal sampai akhir. Jenis perangkat keras yang digunakan penulis adalah sistem yang cukup sederhana yaitu menggunakan ESP32 dan beberapa sensor. Berikut ini merupakan rangkuman dari kerja magang yang dilakukan oleh penulis pada saat proses kerja magang berlangsung.

Table 3. 1 Rankuman Kerja Magang

Minggu	kegiatan
1-2	1. Pengenalan awal proyek magang yang akan dibuat.

	2. Membuat flowchart magang. 3. Membuat schematic diagram project.
3-4	1. Diskusi dengan supervisor tentang rancangan schematic. 2. Mencoba melakukan drafting code. 3. Drafting code sensor. 4. Pengujian prototype code ke dalam esp32. 5. Menganalisa hasil protoype code. 6. Debugging code sensor
5-6	1. Mempersiapkan expansion board. 2. Konfigurasi sensor DHT21. 3. Konfigurasi sensor door reed switch sensor. 4. Diskusi tentang perancangan sistem backend yang akan dipakai. 5. Debugging code untuk relay
7-8	1. Melakukan testing dan menyelesaikan code sensor. 2. Melakukan riset dan analisa untuk membuat code untuk website dan web server 3. Membuat frontend website monitoring server 4. Menganalisa dan mencoba membuat backend dengan database sql 5. Menganalisa dan mencoba membuat frontend dengan express.js.
9-10	1. Melakukan riset dan analisa terhadap web server dan hosting yang mendukung proyek magang 2. Mencari problem terhadapa miskoneksi ke https 3. Membuat server hosting dengan vercel

	4. Melakukan perbaikan pengecekan performa dari proyek
11-12	1. Diskusi tentang kemajuan dari pembuatan proyek IoT dengan testing langsung ke ruang server umn. 2. Menganalisa limitasi dari vercel untuk hosting proyek 3. Melakukan sedikit pembaruan code untuk push ke vercel 4. Testing run monitoring proyek 5. Memperbaiki tampilan UI website
13-14	1. Melakukan riset untuk pembuatan kotak proyek IoT 2. Mencoba mendesain dengan aplikasi 3D seperti thinker cad 3. Mengukur ukuran proyek IoT untuk mendesain kotak IoT 4. Melakukan print 3D kotak 5. Merubah ukuran lubang baut pada kotak IoT
15-16	1. Mendesign ulang box IoT 2. Menambah beberapa penyesuaian pada kotak IoT 3. Memperbaiki ukuran sehingga kotak IoT sesuai dengan proyek
17-18	1. Menambah kabel tambahan untuk memperpanjang ukuran kabel pada beberapa sensor. 2. Mengeprint box yang baru 3. Menyesuaikan ukuran baut pada kotak IoT

	4. Merakit dan memasang ulang alat IoT ke box yang baru
19-20	1. Melakukan beberapa troubleshout pada beberapa error pada sensor 2. Mencari pemecahan terhadap error code yang tidak diketahui. 3. Meletakan proyek IoT ke ruang server Kembali. 4. Menganalisa kondisi dan performa proyek melalui website

3.2.2 Uraian Kerja Magang

Proyek pembuatan sistem monitoring lingkungan data center memiliki beberapa fitur yang dapat membantu tim IT untuk memonitoring situasi lingkungan pada ruangan server UMN. Berikut adalah rangkuman dari kegiatan utama yang dikerjakan oleh penulis semasa proses kerja magang ini berlangsung.

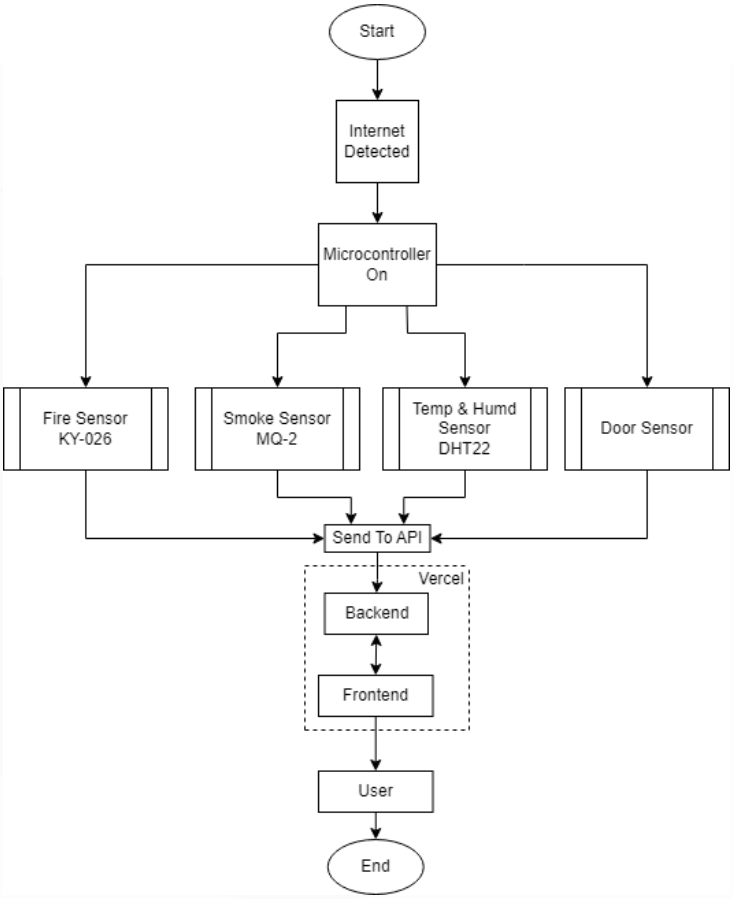
Table 3. 2 Rangkuman Kegiatan Utama

Minggu	Kegiatan
1-2	membuat flowchart, schematic mempersiapkan alat dan bahan yang dibutuhkan
3-6	Drafting code sensor, dan testing awal ESP32 dengan semua sensor yang ada
7-10	Melakukan riset tentang server hosting yang cocok, membuat frontend dan backend website
11-12	Testing, Troubleshooting, optimizing
13-18	Mendesain, mengeprint, dan menyesuaikan box IoT
19-20	Troubleshooting dan finalisasi IoT

3.2.3 Flowchart, Schematic, Alat dan bahan

Pada minggu awal kerja magang berlangsung penulis melakukan diskusi dengan supervisor dan mendiskusikan tentang bahan dan sensor awal yang dibutuhkan untuk membuat alat IoT. Pada diskusi tersebut menghasilkan persetujuan untuk menggunakan microcontroller ESP32 sebagai perangkat utama yang dipakai dalam proyek ini. Ada pula hasil lain dari diskusi yang ada yaitu tentang sensor yang dipakai pada proyek magang. Sensor yang dipakai pada proyek ini antara lain sensor suhu dan kelembapan DHT21, sensor api KY-026, sensor pintu magnetic reed switch, sensor gas MQ-2. Alat IoT ini akan di letakan di dalam rak server.

Selanjutnya penulis membuat flowchart dari proyek magang yang ada dan hasil dari flowchart tersebut akan di diskusikan kepada supervisor Kembali untuk mendapatkan perbaikan atau saran dari supervisor

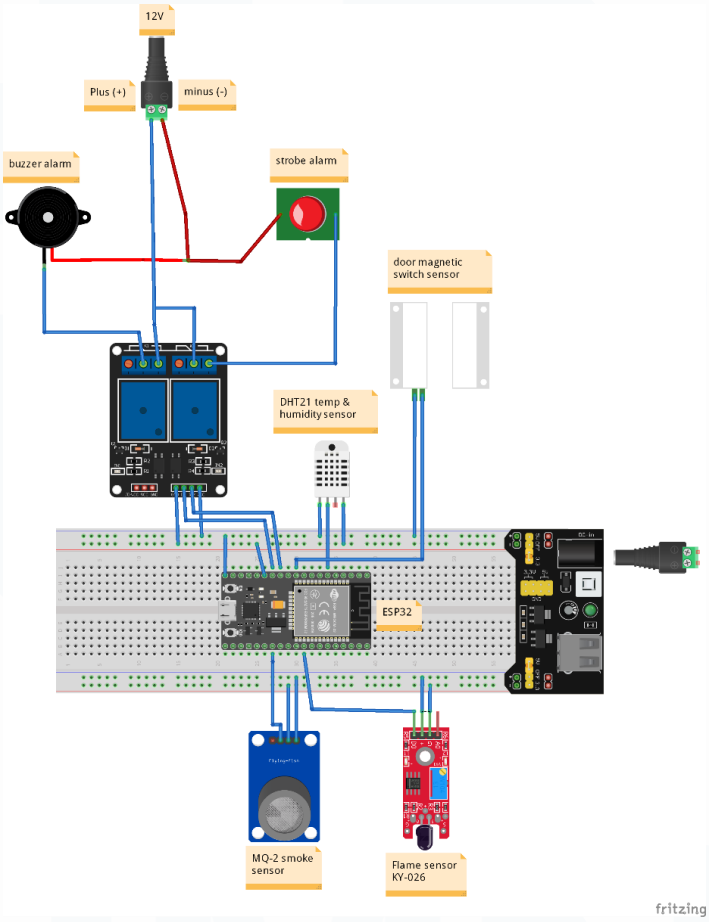


Gambar 3. 1 Konsepsional Flowchart Proyek

Dari gambar konsepsional flowchart diatas ini, dapat diperhatikan bahwa penulis menggunakan enmpat buah sistem pembacaan sensor yang nanti data node atau data sensor dari ESP32 akan dikirim ke server atau domain vercel yang sudah tersedia backend dan front yang sudah dibuat oleh penulis. Disini data node akan disimpan sementara di dalam backend pada array, nanti nya frontend akan request data yang ada di backend lalu akan ditampilkan pada frontend dan Proses pengiriman data ini berjalan secara real-time

Setelah membuat flowchart, penulis juga membuat sistem schematic dari alat IoT yang ingin dibuat. Disini penulis menggunakan aplikasi fritzing yang dimana aplikasi ini berguna untuk mendesain schematic dan topologi sistem beberapa alat IoT seperti Arduino, Raspberry Pi dan sejenis nya. Tetapi pembuatan schematic dengan aplikasi fritzing terbilang cukup berantakan dan tidak bagus, karena itu penulis hanya menggunakan aplikasi tersebut untuk

menggambarkan rangkaian yang dipakai dengan sensor yang akan dipakai penulis. Berikut gambar rangkaian atau schematic dari proyek magang.



Gambar 3. 2 Rangkaian keseluruhan Proyek

Dari gambar diatas dapat diperhatikan bahwa selain tersambung dengan sensor, alat IoT ini juga tersambung dengan relay yang tujuan penggunaannya digunakan untuk menyambungkan buzzer dan lampu strobe sebagai indicator bahwa ada kejanggalan yang dideteksi oleh sensor pada ruangan server.

Penjelsan Alat dan bahan yang digunakan penulis yaitu seperti berikut

1) ESP32

   Penulis menggunakan esp32 karena mikroprosessor ini sudah memiliki wifi module yang memungkinkan perangkat dapat terhubung ke wifi

dengan mudah, selain itu esp32 juga memiliki performa yang lebih baik dalam melakukan tugas tugas yang perlu komputasi yang lebih intensif serta ukuran[3]

2) Sensor KY-026

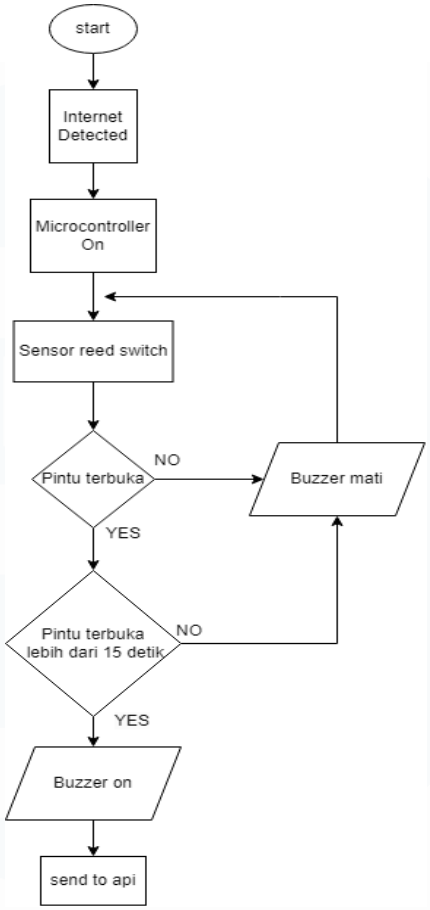
Sensor KY-026 ini sejati nya digunakan untuk mendeteksi cahaya. Tetapi tingkat pendeteksian intensitas cahaya pada sensor ini terbilang cukup tinggi sehingga dapat mendeteksi cahaya terang seperti api. Karena sensor ini yang sensitif pada cahaya maka sensor ini cocok digunakan pada lingkungan yang cukup gelap seperti ruangan server untuk mendeteksi cahaya api ketika terjadi kebakaran pada ruangan server

3) Sensor MQ-2

Sensor MQ-2 merupakan satu dari banyak sensor gas yang dapat mengukur konsentrasi berbagai jenis gas yang dapat terbakar seperti gas metana, alcohol, karbon monoksida yang membuat sensor ini dapat mendeteksi asap atau gas yang dapat muncul dari rak server pada ruangan server.

4) Sensor pintu reed switch

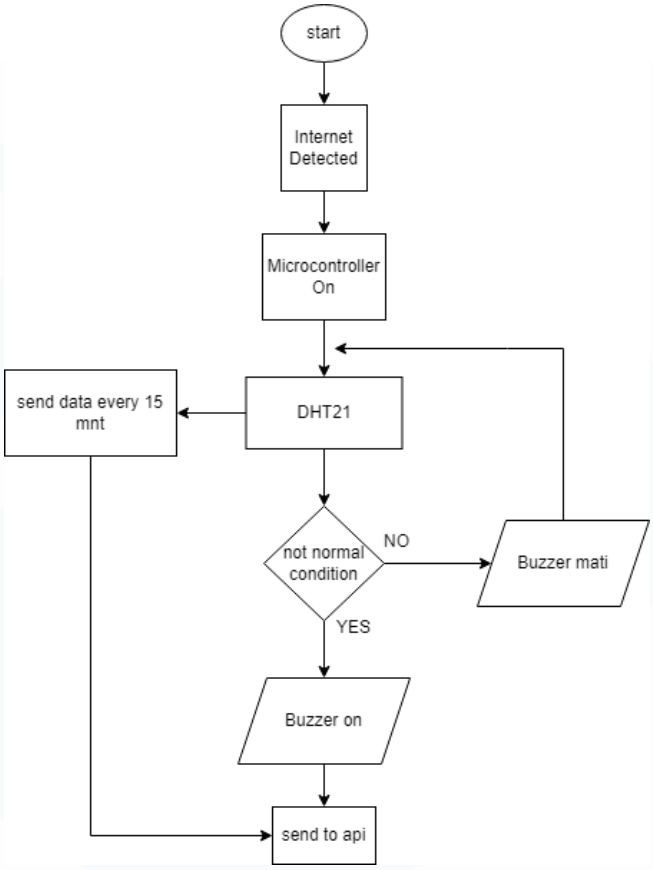
Penulis menggunakan sensor ini sebab merupakan sensor yang disarankan oleh supervisor. Penulis di beri intruksi bahwa sensor ini nanti akan digunakan untuk memberikan informasi bahwa sensor pintu di ruang server akan menyalakan alarm ketika pintu sudah terbuka lebih dari 15 detik. Gambaran flowchart dari sensor pintu reed switch seperti berikut.



Gambar 3. 3 Flowchart Sistem Sensor Reed Switch

5) Sensor DHT21

Sensor ini juga dikenal sebagai AM2301 yaitu sensor yang dapat mengukur suhu dan kelembaban relatif pada suatu lingkungan atua tempat. Pada proyek yang dibuat penulis, sensor DHT21 ini digunakan untuk mengukur suhu dan kelembapan dalam parameter suhu temperature harus di antara dari 21-23°C dan humidity harus di antara 45-55%. Penulis menggunakan sensor DHT21 karena rentang pengukuran dan akurasi sensor ini sudah cukup untuk diimplementasikan pada ruang server. Penggambaran flowchart dari sensor DHT21 ini sekiranya seperti gambar dibawah ini.



Gambar 3. 4 Flowchart Sistem Sensor DHT21

6) ESP32 expansion board

Penggunaan expansion board pada proyek kali ini di karenakan saat penulis menggunakan breadboard untuk pin esp32, hanya bisa menggunakan satu sisi breadboard saja. Karena itu penulis menggunakan expansion board untuk memaksimalkan pin yang ada pada esp32 serta expansion board yang penulis pilih terdapat relay tanam yang dapat digunakan untuk alarm.

3.2.4 Kode ESP32 dan sensor

Untuk pengkodingan alat IoT ini penulis menggunakan aplikasi Arduino IDE dengan mulai dari memasukan beberapa library pada yang digunakan untuk memperluas fungsionalitas dan intergrasi esp32 dan beberapa sensor yang lain. Hal yang pertama kali penulis koding adalah koding untuk fungsionalitas sensor dan esp32 dan menjalankan nya tanpa kode konektivitas

wifi dahulu. Contoh kodingan sensor tersebut seperti dibawah ini yang digunakan untuk sensor suhu dan kelembaban

```
void loop() {

  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print("%,  Temperature: ");
  Serial.print(t);
  Serial.println(" Celsius");

  //delay(2000); //Delay 2 sec.
  // Check if any reads failed and exit early (to try again).
  //if (isnan(h) || isnan(t) ) {
    //Serial.println(F("Failed to read from DHT sensor!"));
    //return;

    if ((h < 45) || (h > 70)){
      Serial.println("Humidity is not normal");
      //digitalWrite (Relay_1, HIGH) ; //send tone
      rHum = 1;
    } else {
      Serial.println("Humidity is Normal");
      //digitalWrite (Relay_1, LOW) ;  //no tone
      //delay(1000);
      rHum = 0;
  }
```

Gambar 3. 5 Kode Sensor DHT21

Pada gambar diatas menunjukan bahwa sensor DHT21 di setting untuk dapat menyalakan relay sesuai dengan parameter yang di inginkan. Selain itu ada pula kode untuk sensor pintu yang memerlukan penyesuaian seperti gambar berikut.

```
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
const long interval = 15000;  // door interval
```

Gambar 3. 6 Time Interval Sensor Pintu

```
// door
  unsigned long currentMillis = millis();
  doorState = digitalRead(DOOR_SENSOR_PIN); // read state
  if (doorState == HIGH) {
    Serial.println("The door is open");
    //delay (1000);
    if (currentMillis - previousMillis >= interval) {

    //digitalWrite (Relay_1, HIGH) ;
    Serial.println("Alarm BEEP");
    rDoor = 1;
    //digitalWrite (Relay_2, HIGH) ;
    //delay(1000);
    }
    unsigned long previousMillis = 0;
  }  //else
  if (doorState == LOW) {
    Serial.println("The door is closed");
    rDoor = 0;
    previousMillis = currentMillis;
    //digitalWrite (Relay_1, LOW) ;
    //digitalWrite (Relay_2, LOW) ;
    //delay(1000);
  }
```

Gambar 3. 7 Kode Sensor Pintu Reed Switch

Gambar diatas terdapat time interval untuk sensor pintu yang akan digunakan untuk menyalakan relay ketika pintu terbuka lebih dari 15 detik. Untuk selebihnya khusunya sensor yang lain menggunakan kode sederhana karena fungsi sensor yang lain digunakan hanya pendeteksian normal.

Setelah membuat kode untuk keempat sensor, penulis membuat kodingan terakhir untuk esp32 sebelum lanjut ke kodingan wifi nya, yaitu mengatur kodingan untu relay. Relay disini memiliki issue sebagai contoh ketika sensor pertama aktif dan menyalakan relay kode selanjutnya seketika mematikan relay secara sekejap, jadi relay tidak menyala dengan lama. Karena itu dengan bantuan supervisor menyarankan untuk memainkan logic codenya supaya code relay yang aktif di sensor sebelumnya tidak tertindih dengan kode sensor dibawah nya. Implementasi code nya seperti dibawah ini.

```
if ((rHum + rTemp + rDoor + rFire + rSmoke) == 0) {
    digitalWrite (Relay_1, LOW) ;
    Serial.println("Relay off");
} else {
    digitalWrite (Relay_1, HIGH) ;
    Serial.println("Relay on");
}
```

Gambar 3. 8 Kode Relay

Setelah penulis menyelesaikan semua kode sensor dan relay serta sudah melakukan test dan tidak menemukan error maka penulis langsung membuat kode untuk membuat esp32 dapat terkoneksi wifi dan kode deployment untuk mengirim data node ke web server dengan melalui API.

Di sini penulis menggunakan vercel sebagai hosting server dan domain website dari proyek magang ini. Penulis menggunakan vercel karena akses penggunaan nya yang cukup mudah dan gratis. Kode untuk dapat push ke web server seperti berikut.

```
void sendtoapi(){
  if(WiFi.status()== WL_CONNECTED){
    WiFiClientSecure *client = new WiFiClientSecure;
    if(client) {
      client->setInsecure();
      HTTPClient https;

      Serial.print("[HTTPS] begin...\n");
      if (https.begin(*client, "https://iot-server-monitoring.vercel.app/sensordata")) {
        Serial.print("[HTTPS] GET...\n");
        // start connection and send HTTP header
        https.addHeader("Content-Type", "application/json");
        json = "{\"suhu\":"+ String(suhu) +",\"humidity\":"+ String(humidity) +",\"smoke\":"+
        Serial.println(json);
        int httpCode = https.POST(json);
        if (httpCode > 0) {
         Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
          if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
            String payload = https.getString();
            Serial.println(payload);
          }
        }
```

Gambar 3. 9 Kode Deployment ke Vercel

Pada gambar diatas ditunjukan bahwa data node dari sensor akan dikirim ke server vercel melalui domain web yang sudah disiapkan untuk mengakses website proyek nya. Proses push data ini akan di deploy terus menerus dalam interval 1 menit sekali

3.2.5 Server hosting, backend, dan frontend

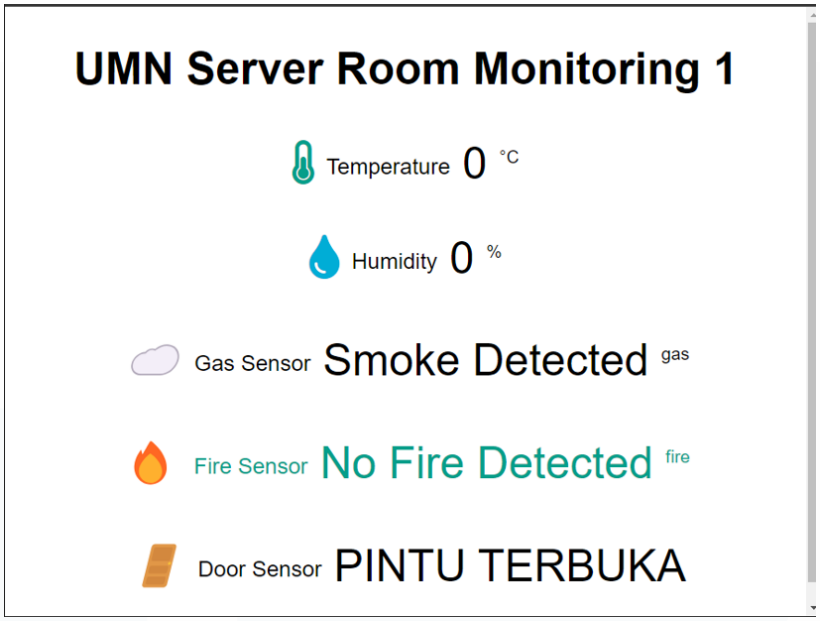
Alasan paling besar bagi penulis untuk memillih vercel sebagai hosting web adalah karena vercel merupakan platform dengan akses yang paling mudah untuk menjalankan situs web. Penulis cukup menghubungkan repository GitHub yang ada kode frontend dan backend di dalam nya serta dengan mudah melakukan deploy branch utama ke domain. Vercel juga cocok digunakan untuk mengembangkan website seperti entry level serta cocok untuk pengembangan website dengan http. [4]

Di dalam vercel, penulis memasukan kode backend dan kode frontend melalui yang di deploy melalui repository Github. Pada bagian backend, penulis menggunakan express.js. disini backend digunakan untuk menyimpan data node atau data esp32 untuk sementara pada array. Kode dari backend ini akan terlihat seperti berikut.

```
const path = require("path");
const router = express.Router();

let sensorArray = [];

let sensorDefault = {
  suhu: 0,
  humidity: 0,
  smoke: 0,
  fire: false,
  doorsensor: false,
  lokasi: "Sensor 1",
};

sensorArray.push(sensorDefault);
```

Gambar 3. 10 Kode Penyimpanan Data Sementara di Array

Disini setelah backend menerima data node melalui API, backend akan mengembalikan pesan feedback ke esp32 yang bisa dilihat di serial monitor di Arduino IDE. Data yang disimpan di array akan di request oleh frontend dan akan ditampilkan ke frontend.



Gambar 3. 11 Tampilan dummy atau tampilan awal website monitoring server

Pada gambar diatas merupakan gambaran tampilan awal frontend dari website yang dibuat. Disini terdapat lima indikator yang akan ditampilkan data nya dari sensor-sensor yang ada seperti, suhu, kelembapan, indikator gas, indikator api, dan informasi pintu jika terbuka.

3.2.6 Testing, Troubleshooting, optimizing.

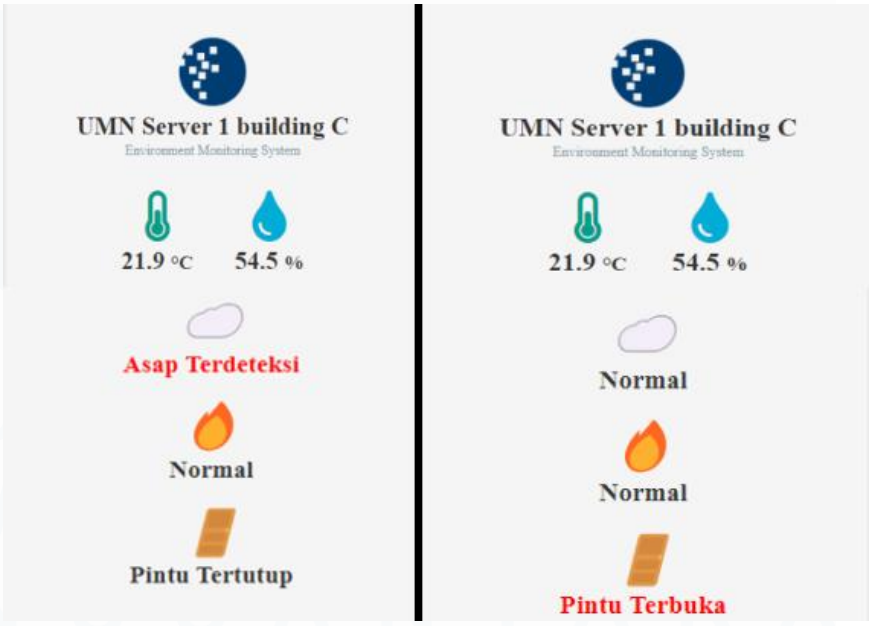
Dalam fase ini penulis lebih berfokus untuk mengatasi permasalahan dari pada esp32 ataupun dari push deployment yang sesekali terjadi error. Kasus yang paling sering jadi permasalahan adalah ketika esp32 gagal mengirim data node setelah berjalan 2-3 jam. Faktor terjadi nya error ini bisa dibilang cukup transparan dan bisa jadi memiliki banyak faktor yang menyebabkan error ini terjadi. Bisa dari koneksi internet yang kurang kuat, data yang dikirim terlalu banyak karena dikirim berulang-ulang, atau keterbatasan backend dalam menyimpan data sementara di array dan lain-lain. Walaupun permasalahan gagalnya esp32 untuk mengirim node bisa diatasi dengan mencabut ulang kabel adapter listrik nya, tapi cara tersebut sangat tidak efisien. Karena itu penulis membuat pembaruan kode yang digunakan pada pengiriman data,

dimana jika terjadi gagal pengiriman data maka esp32 akan mereset dan mengulang perangkat ke kondisi awal.

```
}
else {
  Serial.printf("DUAAAAARRRRRR[HTTPS] GET... failed, error: %s\n", https.errorToString(httpCode).c_str());
  ESP.restart();
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi connection lost. Reconnecting...");
    ESP.restart();

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Reconnecting...");
  }
  Serial.println("Reconnected to WiFi");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
 }
}
https.end();
```

Gambar 3. 12 Kode untuk Mereset ESP32

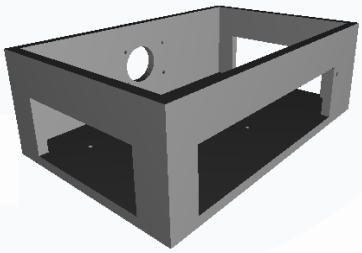
Untuk pengujian website nya sendiri penulis melakukan beberapa test pada sensor sebelum alat dapat di test di ruang server secara langsung. Pada website jika data yang ditampilkan menghasilkan value yang tidak dinginkan maka data tersebut akan ditampilkan dengan warna merah dan otomatis menyalakan relay alat. Contoh pengetasan seperti gambar dibawah ini dengan melakukan test menggunakan korek api dan mencoba menjauhkan magnetic sensor door switch.



Gambar 3. 13 Tampilan hasil testing sensor pada sensor

3.2.7 Desain Box

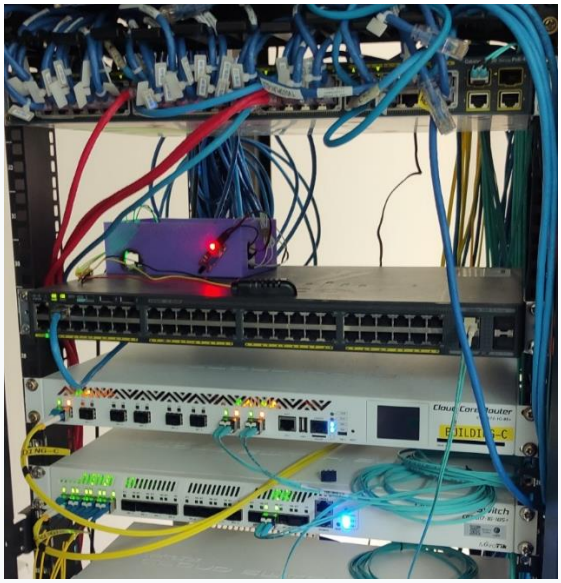
Pada tahap ini penulis melakukan beberapa research dan revisi kotak yang menghabiskan banyak waktu karena pembuatan kotak yang menggunakan printer 3D yang memerlukan waktu untuk mengeprint box nya. Desain dari revisi terakhir penulis mendesain nya dengan membuat 3 lubang persegi di 3 sisi berbeda untuk jalur koneksi kabel ke esp32. Di satu sisi dikhususkan untuk menempelkan sensor asap dan sensor api, jadi penulis perlu membuat lubang kecil di sisi tersebut.



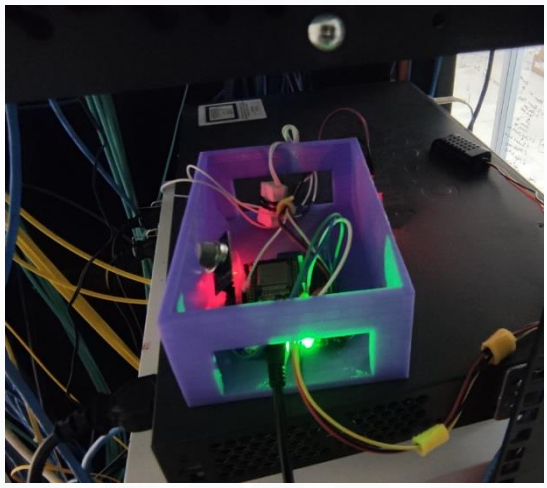
Gambar 3. 14 Desain Box IoT

3.2.8 Finishing

Pada tahap ini penulis terus memantau perkembangan dan performa dari kotak IoT di ruang server melalu website, jika diperlukan penulis perlu datang ke ruang server sesekali untuk memeriksa kabel dan koneksi internet yang tersambung pada alat IoT. Berikut dua gambar alat IoT yang diletakan pada rak server dan di test perkembangan nya secara berkala.

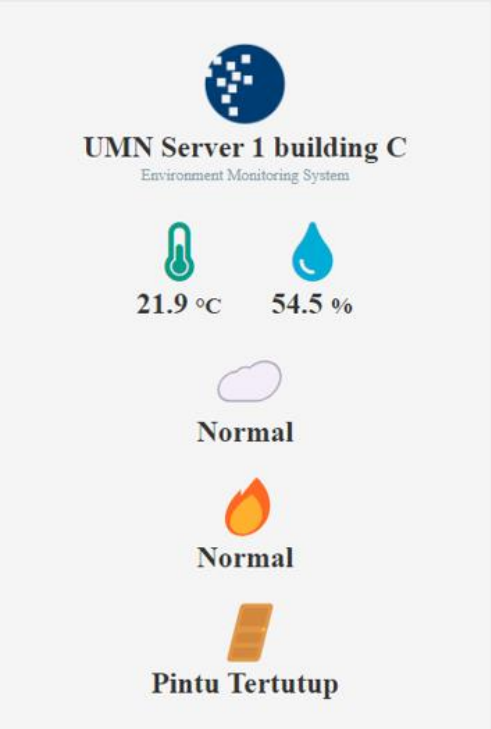


Gambar 3. 15 peletakan box IoT pada rak server



Gambar 3. 16 Box Yang Sedang di Test Pada Ruang Server

Poin penting dari perkembangan project ini adalah pihak IT department dapat melakukan monitoring suhu dan kelembapan pada ruang server serta mendapat informasi lain nya seperti indikator api, asap, dan indikator pintu. Data yang didapat oleh sensor akan di tampilkan pada website seperti gambar dibawah ini.



Gambar 3. 17 Gambar tampilan final website

3.3 Kendala yang Ditemukan

Selama pengerjaan proyek kerja magang berlangsung pernulis menenmukan beberapa kendala dalam pengerjaan nya, antara lain:

1) Pengiriman data yang kurang optimal

   Pengiriman data sensor ke server sering terjadi kendala yang dimana sering gagal mengirim yang bisa dikarenakan banyaknya data yang dikirim dan masuk ke array

2) Performa sensor yang tidak stabil

   Performa dari sensor pada proyek ini terbilang kurang optimal karena dari data yang dicatat oleh penulis khusus nya data dari sensor dht21 bahwa

suhu pembacaan suhu dapat tembus dari batas normal yang ada suhu biasanya

3) Kurang nya pengujian sensor secara real-time
   Dikarenakan ruang server yang tidak bisa dimasuki oleh sembarang orang dan memiliki waktu yang tidak banyak saat masuk ke ruang server tersebut membuat penulis tidak bisa banyak melakukan testing sensor.
4) Penggunaan vercel sebagai web hosting
   Karena penulis ditugaskan untuk membuat sistem monitoring ruang server UMN dengan pendekatan IoT serta tidak bisa memberikan akses monitoring ini ke sembarang orang menjadikan penggunaan vercel yang dimana sebagai web hosting dari server luar menjadikan proyek magang yang dibuat penulis memiliki kekurangan dari segi keamanan akses website.

3.4 Solusi atas Kendala yang Ditemukan

Dari segala kendala yang dijelaskan diatas, penulis juga mencari cara agar mengatasi masalah-masalah tersebut, seperti:

1. Menambahkan kode reset untuk esp32 jadi ketika esp32 terlalu banyak mengirim data dan terbebani lalu terjadi error maka esp32 akan mereset device agar jalan nya program bisa direstart dari awal. [5]
2. Melakukan penelitian tentang library yang cocok dengan fitur yang akan dipakai penulis dalam melakukan kerja magang agar fungsi-fungsi dari sensor dapat digunakan secara maksimal
3. Melakukan pengecekan secara berkala mulai dari konektivitas Wifi hingga perangkat sensor yang terpasang apakah masih terjadi error atau terdapat perangkat yang rusak.
4. Mempelajari batasan-batasan dari penggunaan vercel dan mencari keunggulan web hosting vercel yang bisa dimanfaatkan oleh alat IoT yang dibuat oleh penulis.