

# PERANCANGAN DAN IMPLEMENTASI APLIKASI PENDETEKSI HUJAN DI DAERAH CIGANITRI BERBASIS WEB DAN ANDROID MENGGUNAKAN ESP8266

## DESIGN AND IMPLEMENTATION OF RAIN DETECTOR APPLICATION IN CIGANITRI BASED WEB AND ANDROID USING ESP8266

Ghifary Hafidzul Ihsan<sup>1</sup>, Denny Darlis, S.Si., M.T.<sup>2</sup>, Ir. Agus Ganda Permana, M.T.<sup>3</sup>

<sup>1,2,3</sup> Prodi D3 Teknologi Telekomunikasi, Universitas Telkom

<sup>1</sup>[hafidzmost@student.telkomuniversity.ac.id](mailto:hafidzmost@student.telkomuniversity.ac.id), <sup>2</sup>[denny.darlis@tass.telkomuniversity.ac.id](mailto:denny.darlis@tass.telkomuniversity.ac.id),

<sup>3</sup>[agusgandapermana@tass.telkomuniversity.ac.id](mailto:agusgandapermana@tass.telkomuniversity.ac.id)

### Abstrak

Rata-rata wilayah di benua Asia memiliki iklim tropis, termasuk Indonesia yang pada dasarnya hanya memiliki dua musim, yaitu musim hujan dan musim kemarau. Akan tetapi, seiring berjalannya waktu, pergantian musim dari musim hujan ke musim kemarau dan sebaliknya semakin tidak menentu karena pengaruh dari pemanasan global, sehingga iklim di Indonesia pada 5 tahun terakhir sering menyebabkan hujan lokal pada daerah tertentu di wilayah Indonesia, termasuk kota Bandung dan sekitarnya yang berada di provinsi Jawa Barat. Proyek akhir yang akan dibuat berupa aplikasi dengan menggunakan sistem operasi android dan *website*. Aplikasi tersebut akan digunakan untuk memantau dua kondisi cuaca, yaitu kondisi hujan dan tidak hujan. Pemantauan dua kondisi cuaca tersebut dilakukan pada tiga titik yang berpotensi mengalami peristiwa hujan lokal di sepanjang jalan Ciganitri hingga ke Universitas Telkom. Pemantauan dua kondisi cuaca tersebut juga dilakukan secara *real-time* sehingga memungkinkan pengguna untuk memantau langsung dua kondisi cuaca pada tiga lokasi yang berbeda. Hasil simulasi dan pengujian fungsionalitas menunjukkan bahwa seluruh fungsi sistem baik *website* maupun aplikasi android dapat berjalan 100% sebagaimana mestinya. Aplikasi yang telah dirancang dan dikembangkan sudah dapat menampilkan informasi yang diperoleh di masing-masing lokasi. Artinya dapat dipastikan bahwa setiap informasi yang ditampilkan berbeda pada tiap lokasi. Performansi *website* menghasilkan nilai 55 berdasarkan penilaian dari Google Lighthouse. Artinya *website* mendapatkan nilai cukup baik jika dilihat dari rentang penilaian Google Lighthouse. Pengujian kompatibilitas aplikasi android dengan menggunakan aplikasi AppChecker mendapatkan hasil bahwa sistem operasi paling minimum yang dibutuhkan untuk melakukan instalasi aplikasi pendeteksi hujan pada perangkat android adalah pada versi 4.0 (*Ice Cream Sandwich*). Hasil pengujian alat pendeteksi hujan dinilai cukup akurat karena perbedaan hasil pengukuran alat pendeteksi hujan dengan alat ukur yang digunakan oleh acuan pengukuran tidak berbeda jauh. Diketahui rata-rata kesalahan relatif sensor DHT11 adalah 8.35% jika dibandingkan dengan pengukuran acuan. Artinya kesalahan pengukuran pada sensor DHT11 relatif rendah

**Kata Kunci:** Hujan Lokal, Website, Real-Time.

### Abstract

The average area in Asia has a tropical climate, including Indonesia, which only has two seasons, namely the rainy season and the dry season. However, over time, the change of seasons from the rainy season to the dry season and vice versa is increasingly erratic due to the influence of global warming, so that the climate in Indonesia in the last 5 years often causes local rain in certain areas in Indonesia, including the city of Bandung and its surroundings. located in West Java province. The final project will be an application using the Android operating system and a website. This application will be used to monitor two weather conditions, namely rainy and non-rainy conditions. The monitoring of the two weather conditions is carried out at three points that have the potential to experience local rain events along the Ciganitri road to Telkom University. Monitoring of the two weather conditions is also carried out in real-time, which means it would like to allow users to directly monitor two weather conditions at three different locations. The simulation results and functionality testing show that all system functions, both the website and the Android application, can run 100% as they should. Applications that have been designed and developed can display the information obtained in each location. That can ensure each information is different at each location The application Website performance scores 55 based on ratings from

Google Lighthouse. It means that the website scores quite well when viewed from the Google Lighthouse rating range. Testing the compatibility of android applications using the AppChecker application results that the minimum operating system required to install a rain detection application on an Android device is version 4.0 (Ice Cream Sandwich). The results of the rain detector test are considered pretty accurate because the difference between the measurement results of the rain detector and the measuring instrument used by the measurement reference is not much different. It noted that the average relative error of the DHT11 sensor is 8.35% when compared to the reference measurement. it means the measurement error on the DHT11 sensor is relatively low.

**Keywords:** Local Rain, Website, Real-Time.

---

## 1. Pendahuluan

Internet berasal dari kata *Interconnection Networking* yang mempunyai arti hubungan berbagai komputer dan berbagai tipe (*platform*) komputer yang membentuk sistem jaringan mencakup seluruh dunia dengan melalui jalur telekomunikasi seperti telepon, *wireless*, bahkan satelit.

Dengan menggunakan internet, informasi di tempat yang jauh dari lokasi kita saat ini dapat diakses dan secara umum dapat diketahui kondisi serta perkembangannya. Proyek Akhir ini dibuat untuk mendeteksi keadaan cuaca di tempat yang akan kita tuju, dan dibuat berdasarkan kasus yang banyak di alami mahasiswa ketika memasuki musim hujan, khususnya pada waktu akan berangkat kuliah dan saat akan pulang ke rumah atau indekos masing-masing. Masalah ini sering menjadi momok mahasiswa dikarenakan di daerah Bandung, khususnya di daerah Ciganitri yang cukup sering mengalami peristiwa hujan lokal.

Perancangan sistem pemantauan ini sebelumnya sudah diterapkan dengan satu lokasi pemantauan dan *website* sebagai media informasinya. Berbeda dari perancangan sebelumnya, perancangan sistem pemantauan ini melakukan pemantauan di tiga lokasi yang berbeda dengan menggunakan aplikasi berbasis sistem operasi android dan *website* sebagai media penyampaian informasinya.

Dengan adanya alat ini, diharapkan penggunaanya dapat mengetahui kondisi cuaca di tempat yang akan dituju oleh pengguna dan dapat melakukan tindakan pencegahan agar tidak terkena dampak hujan lokal yang sering terjadi secara tiba-tiba, terkhusus untuk mahasiswa yang menggunakan kendaraan roda dua sebagai alat transportasinya.

## 2. Dasar Teori

### 2.1 Internet of Things

*Internet of Things* merupakan sebuah contoh dari perkembangan pada dunia teknologi informasi. *Internet of Things* berasal dari komputer yang terhubung dengan perangkat *mobile*. [1] *Internet of Things* memiliki konsep dimana konektifitas internet yang memungkinkan serta dapat digunakan untuk bertukar informasi satu sama lainnya dengan benda-benda yang ada disekelilingnya.

Banyak yang memprediksi bahwa *Internet of Things* merupakan “*the next big thing*” di dunia teknologi informasi. Hal ini dikarenakan banyak sekali potensi yang bisa dikembangkan dengan teknologi *Internet of Things* tersebut. Teknologi *Internet of Things* akan memungkinkan alat-alat fisik bisa terkoneksi dengan internet. Misalnya, Kulkas, TV, Mesin Cuci dan lainnya dapat dikontrol serta dipantau menggunakan *smartphone* yang terkoneksi ke internet untuk mematikan, menghidupkan dan kegiatan lainnya. Menerapkan *Internet of Things* diyakini akan lebih mempermudah kegiatan manusia dalam melakukan berbagai aktifitas sehari-hari. Semua kegiatan dapat dilakukan dengan sangat praktis dan disatu sisi adanya sistem kontrol karena perangkat yang terhubung menyebabkan kehidupan akan lebih efektif dan efisien.

### 2.2 Website

*Website* adalah keseluruhan halaman-halaman *web* yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan. Hubungan antar satu halaman *web* dengan halaman lainnya disebut dengan *hyperlink*, sedangkan teks yang dijadikan media penghubungnya disebut dengan *hypertext*. [2]

Proyek akhir ini akan memanfaatkan *website* untuk menampilkan data masukan dari database sebelum diolah dan ditampilkan di dalam aplikasi. Data tersebut terlebih dahulu di proses oleh perangkat mikrokontroler yang digunakan untuk mengubah data analog berupa air, suhu, dan kelembapan ke dalam bentuk digital. Kemudian hasil data masukan dari perangkat tersebut dimasukkan ke *database*.

### 2.3 Arduino IDE

Arduino IDE (*Integrated Development Software*) merupakan *software* yang digunakan untuk membuat sketch yang akan disematkan ke perangkat mikrokontroler. Arduino IDE juga tergolong ke dalam bentuk *software* pengembangan program yang terintegrasi sehingga berbagai *tools* atau tampilan disediakan dan dinyatakan dalam

bentuk *interface* berbasis menu. Dengan menggunakan Arduino IDE, kesalahan dan kebenaran dari penulisan kode program dapat langsung dibuktikan. [3]

#### 2.4 Android

Android merupakan sebuah sistem operasi perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. [4] Proyek akhir ini memanfaatkan sistem operasi android untuk merancang aplikasi yang dapat menampilkan data serupa dengan data yang ditampilkan di *website*. Data masukan tersebut sebelumnya juga diperoleh dari perangkat mikrokontroler yang digunakan dengan mengubah data analog menjadi data digital.

#### 2.5 Phonegap

Phonegap merupakan sebuah *framework* untuk membuat aplikasi *mobile* terutama Android, iPhone, Blackberry dan Windows Phone 7. [5] cara kerja dari Phonegap adalah dengan membuat aplikasi menggunakan HTML5, CSS dan JavaScript. Setelah itu, mesin Phonegap dapat dikonversi ke dalam sistem operasi di *smartphone*. Proyek akhir ini memanfaatkan *framework* Phonegap untuk merancang tampilan aplikasi berbasis sistem operasi Android. Kemudian, tampilan aplikasi tersebut akan disesuaikan dengan *website* yang telah dibuat sebelumnya.

#### 2.6 JavaScript

*JavaScript* merupakan bahasa pemrograman populer yang digunakan untuk menciptakan halaman web dan kemudian dapat berinteraksi dengan pengguna dan dapat merespon event yang terjadi pada halaman web. [6] *JavaScript* bersifat *Client-Side Programming Language* yang merupakan tipe dari bahasa pemrograman dimana proses eksekusi dari kode programnya dilakukan oleh pengguna. Aplikasi pengguna yang dimaksud merujuk kepada *web browser* seperti Google Chrome, Mozilla Firefox, dsb. Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman di sisi *server* seperti PHP, dimana pada sisi *server* seluruh kode program dijalankan oleh *server*.

#### 2.7 HTML

*Hyper Text Markup Language* (HTML) adalah sebuah bahasa pemrograman yang digunakan untuk membuat sebuah halaman web. HTML dapat menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dalam format *hypertext* sederhana. Disebut *hypertext* karena *script* pada HTML dapat dibuat agar sebuah teks menjadi link yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik teks tersebut. [7]

HTML berawal dari bahasa SGML (*Standard Generalized Markup Language*) yang penulisannya disederhanakan. HTML dapat dibaca oleh berbagai macam *platform* dan merupakan bahasa pemrograman yang fleksibel serta dapat digabungkan dengan bahasa pemrograman lain seperti PHP, JSP, dan *JavaScript*.

#### 2.8 CSS

CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan pada sebuah *website*. Pada umumnya CSS digunakan untuk memformat halaman web yang ditulis dengan HTML atau XHTML. [8]

CSS diciptakan guna memisahkan antara konten utama dengan tampilan dokumen yang termasuk *layout*, *font* dan juga warna. Pemisahan ini berguna untuk meningkatkan daya akses konten pada *web*, kemudian menyediakan lebih banyak fleksibilitas dan juga kontrol dalam spesifikasi dari sebuah karakteristik pada sebuah tampilannya. CSS memungkinkan untuk mengurangi kesulitan dalam penulisan kode dan juga struktur dari konten, misalnya dalam teknik *tableless* pada desain *web*.

#### 2.9 HTML

Firebase adalah sebuah platform yang menyediakan beberapa service/layanan untuk memudahkan developer membangun aplikasi Web, Android, dan iOS maupun Unity. [9] Firebase juga merupakan salah satu *platform* yang dikembangkan oleh Google dan menyediakan layanan *real-time database* menggunakan protocol seperti HTTP dan MQTT. Di dalam firebase, seluruh komunikasi ditransmisikan di jalur yang telah di enkripsi. Firebase juga mendukung berbagai macam perangkat seperti Arduino, ESP, Android, Raspberry Pi serta berbagai macam bahasa pemrograman. Selain itu, firebase menggunakan API (*Application Programming Interface*) agar pembuat dapat melakukan sinkronisasi antara komponen yang digunakan dengan aplikasi yang dibuat.

#### 2.10 MOS (Mean Opinion Score)

*Mean Opinion Score* adalah metode yang digunakan untuk mengukur kinerja dan mengetahui kualitas sistem dari sisi pengguna. Pengujian ini menggunakan kuesioner yang memiliki beberapa pertanyaan untuk dijawab oleh responden. Setiap pertanyaan yang dijawab oleh responden terdapat *range* bobot penilaian yang nanti digunakan untuk mencari kesimpulan skor hasil kinerja dan kualitas sistem. Berikut bobot nilai kuesioner dan rumus yang digunakan untuk menghitung MOS (*Mean Opinion Score*).

Tabel 2.1 Bobot Nilai Kuesioner

MOS	Keterangan	Bobot Nilai
SS	Sangat Setuju	5
S	Setuju	4
C	Cukup	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Persamaan  $mean\ p_i$  digunakan untuk mendapatkan hasil rata – rata jawaban yang diberikan responden pada setiap atribut pertanyaan. Adapun rumus untuk mendapatkan nilai rata – rata jawaban yang diberi oleh responden.

$$mean\ p_i = \frac{\sum p_i}{n} \quad (2.1)$$

Keterangan :

$mean\ p_i$  = rata - rata skor setiap atribut pertanyaan

$\sum p_i$  = jumlah skor kali bobot nilai setiap atribut pertanyaan

n = jumlah responden

Persamaan MOS digunakan untuk mencari total skor rata – rata yang diberikan oleh responden pada seluruh atribut pertanyaan. Sehingga dapat digunakan persamaan MOS sebagai berikut.

$$MOS = \frac{\sum_{i=1}^k mean\ p_i}{k} \quad (2.2)$$

Keterangan :

MOS = total skor rata – rata seluruh atribut pertanyaan

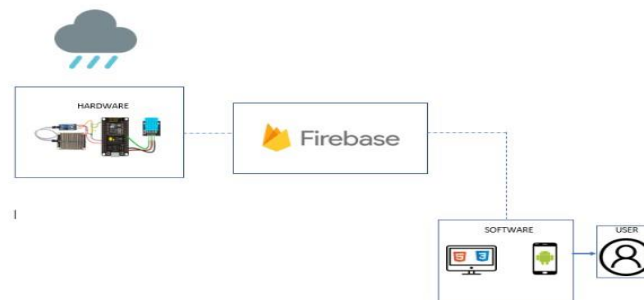
$mean\ p_i$  = rata - rata skor setiap atribut pertanyaan

k = jumlah atribut pertanyaan [10]

### 3. Perancangan Alat dan Sistem

#### 3.1 Deskripsi Proyek Akhir

Pada bagian ini akan dijelaskan mengenai arsitektur perancangan aplikasi pendeteksi hujan di daerah Ciganitri berbasis web dan android dengan menggunakan perangkat mikrokontroler yang terdiri dari *nodeMCU*, *rainfall sensor*, dan DHT11 (*temperature & humidity sensor*).



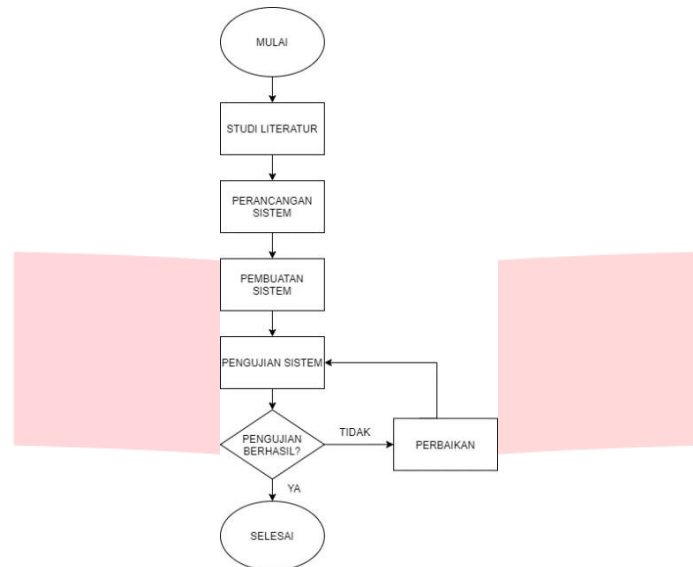
Gambar 3. 1 Model Sistem Pendeteksi Hujan Berbasis Web dan Android

Gambar 3.1 merupakan model sistem dari aplikasi pendeteksi hujan berbasis web dan android. Cuaca dan suhu nantinya akan di deteksi oleh masing-masing sensor yang digunakan (*rainfall* dan *temperature sensor*) pada nodemcu yang terhubung ke internet atau wifi. Kemudian, nodemcu akan disinkronisasikan dengan firebase menggunakan API key, agar data yang diterima sensor tersebut dapat ditampilkan melalui website yang telah di

hosting, lalu aplikasi android yang akan dibuat nantinya menampilkan data yang sama dengan data yang ada di website, aplikasi tersebut dibuat agar mempermudah pengguna untuk melakukan *monitoring* cuaca di tempat yang akan dituju oleh pengguna.

### 3.2 Proses Pengerjaan Proyek Akhir

Proses pengerjaan ini dilakukan berbagai tahapan sebagai berikut :



Gambar 3.2 *Flow Chart* Pengerjaan Proyek Akhir

Tahap pengerjaan proyek akhir dimulai dari studi literatur yang dilakukan untuk memulai pengerjaan proyek akhir dengan menelusuri berbagai sumber tulisan yang pernah dibuat sebelumnya. Selain itu, hal tersebut juga bertujuan untuk mencari referensi teori yang relevan dan terkait dengan pengerjaan proyek akhir melalui buku, jurnal, laporan penelitian agar mendapatkan referensi yang relevan dengan perumusan masalah.

Perancangan sistem merupakan tahapan selanjutnya setelah melakukan studi literatur dalam membuat sistem pendeteksi hujan. Perancangan sistem dilakukan untuk mengetahui apa saja yang akan dibuat atau ditampilkan pada sistem informasi. Perancangan sistem ini meliputi penentuan peletakan alat, penentuan komponen dan perancangan *website* dan aplikasi yang berbasis sistem operasi android.

Langkah selanjutnya adalah melakukan pembuatan sistem. Pembuatan sistem dilakukan berdasarkan perancangan yang dilakukan sebelumnya. Di bagian *hardware* dimulai dengan membuat rangkaian alat pendeteksi hujan serta mendesain maket yang menyesuaikan dengan rangkaian alat tersebut, kemudian membuat sistem informasi berupa *website* dan aplikasi berbasis sistem operasi android, dan yang terakhir di dalam tahap pembuatan adalah melakukan sinkronisasi dari rangkaian alat yang telah dibuat ke Firebase sebagai *real-time database*-nya agar hasil pengukuran dari alat dapat ditampilkan ke dalam *website* dan aplikasi.

Setelah melakukan pembuatan sistem secara menyeluruh, tahapan selanjutnya yaitu melakukan pengujian terhadap sistem pendeteksi hujan sesuai lokasi penempatannya. Pengujian dilakukan untuk mengetahui kinerja alat dari masing-masing bagian dengan mengharapkan hasil data pengujian yang valid dan untuk mengetahui fungsionalitasnya sudah sesuai dengan yang diharapkan atau belum. Jika belum mendapatkan hasil sesuai dengan yang diharapkan, maka akan dilakukan peninjauan kembali serta perbaikan pada sistem baik di sisi *hardware* maupun *software*.

### 3.3 Perancangan Sistem

Perancangan sistem merupakan sebuah tahapan yang hasilnya akan digunakan untuk melakukan pembuatan sistem. Perancangan sistem pendeteksi hujan meliputi perencanaan peletakan sensor untuk menentukan wilayah mana yang akan dilakukan pemantauan, kemudian menentukan apa saja komponen dan *platform* yang digunakan dalam pembuatan sistem di bagian *hardware* dan *software*, serta menentukan apa saja informasi yang akan ditampilkan ke pengguna di dalam *website* dan aplikasi berbasis sistem operasi android. Setelah memiliki semua aspek yang dibutuhkan, pembuatan sistem baru dapat dimulai.

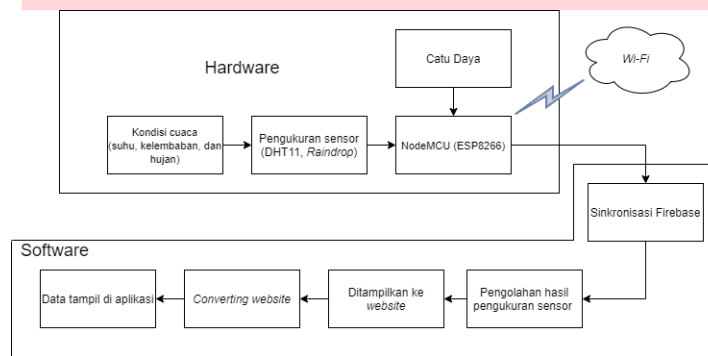
Tahap pertama perancangan Proyek Akhir ini adalah dengan melakukan perencanaan peletakan sensor di wilayah yang sudah ditentukan dan terindikasi cukup sering mengalami peristiwa hujan lokal. Dengan didasari hasil *survey* dengan beberapa mahasiswa yang tinggal di wilayah Ciganitri, penulis menentukan wilayah jalan Ciganitri sebagai wilayah tempat melakukan pemantauan.



Gambar 3.3 Lokasi Penempatan Alat Pendeteksi Hujan

Tahapan selanjutnya yaitu pemilihan komponen *hardware* dan *software* yang akan digunakan sebagai alat pendeteksi hujan dan media penyampaian informasi kepada pengguna. Di bagian *hardware*, penulis memilih perangkat mikrokontroler NodeMCU yang membutuhkan koneksi *Wi-Fi* sebagai media komunikasinya. Dengan bantuan sensor DHT11 sebagai sensor pengukur suhu dan *Raindrop* Sensor sebagai sensor pendeteksi hujan. Kemudian di bagian *software* penulis menggunakan fitur *real-time database* milik Google Firebase sebagai perantara sistem informasi dan alat pendeteksi hujan di bagian *hardware*.

Berikut gambar blok diagram secara umum dan menyeluruh dari sistem pendeteksi hujan.

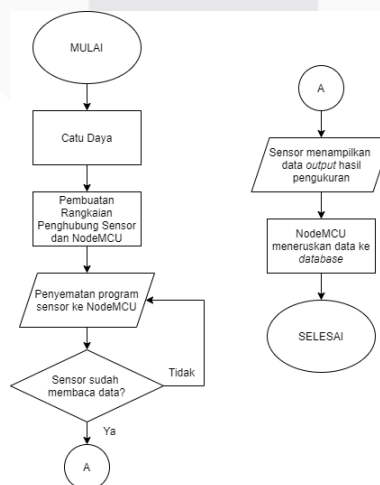


Gambar 3.4 Blok Diagram Sistem Aplikasi Pendeteksi Hujan

Gambar 3.4 menjelaskan secara umum bahwa perancangan sistem Proyek Akhir ini menggunakan dua sistem, yaitu *hardware* miktokontroler NodeMCU yang berfungsi untuk mengatur dan mengolah data-data yang didapatkan dari sensor. Kemudian, di sisi *software* dilakukan sinkronisasi menggunakan koneksi *Wi-Fi* atau internet antara NodeMCU dengan Firebase yang memiliki fitur bernama *real-time database*. Setelah itu dilakukan pengolahan data yang masuk di *real-time database* untuk disesuaikan dengan kebutuhan data yang akan ditampilkan ke dalam *website* dan aplikasi.

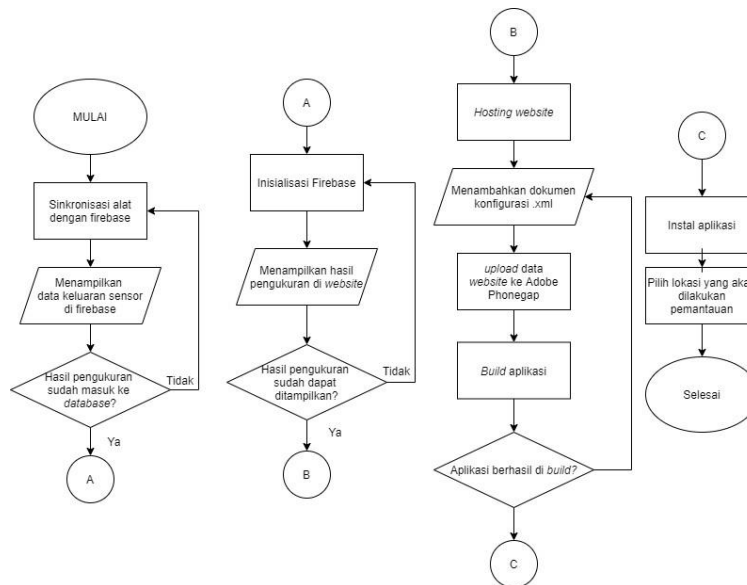
**3.4 Flowchart Sistem**

Komponen *hardware* yang digunakan yaitu meliputi perangkat mikrokontroler dan sensor yang digunakan. Komponen *hardware* ini berfungsi untuk membaca kondisi cuaca dan kemudian menjadikan masukan tersebut sebagai data yang akan dilanjutkan menuju ke Firebase *real-time database* pada bagian *software* dari sistem ini.



Gambar 3. 1 Flow Chart Bagian Hardware dari Alat Pendeteksi Hujan

Kemudian sistem di bagian *software* dimulai dari proses sinkronisasi NodeMCU dengan Firebase sebagai *database* yang berfungsi sebagai perantara untuk data masukan yang dihasilkan dari pengukuran sensor.



Gambar 3. 2 Flow Chart Software dari Aplikasi Pendeteksi Hujan

Setelah itu, agar hasil pengukuran sensor yang sudah masuk ke Firebase dapat ditampilkan ke dalam *web* dan aplikasi Android, dibutuhkan inisialisasi dengan menambahkan beberapa baris sintaks JavaScript pada *script* HTML dari *website* yang telah dibuat.

```
var firebaseConfig = {
  apiKey: "AIzaSyCi-4yEBoI_OFtu7ZYTtjEpbFu1M0VxGQ",
  authDomain: "cobapa-2a722.firebaseio.com",
  databaseURL: "https://cobapa-2a722.firebaseio.com",
  projectId: "cobapa-2a722",
  storageBucket: "cobapa-2a722.appspot.com",
  messagingSenderId: "465122199667",
  appId: "1:465122199667:web:7c638ea9dc0a49dcf3c206",
  measurementId: "G-CZWLQ931SB"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
firebase.analytics();
```

Gambar 3. 3 Script Inisialisasi Firebase

### 3.5 Perangkat Keras

#### 3.5.1 NodeMCU

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

#### 3.5.2 DHT11

Sensor DHT11 adalah modul sensor yang berfungsi untuk mensensing objek suhu dan kelembapan yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler. Sensor DHT11 pada umumnya memiliki fitur kalibrasi nilai pembacaan suhu dan kelembapan yang cukup akurat. Penyimpanan data kalibrasi tersebut terdapat pada memori program OTP yang disebut juga dengan nama koefisien kalibrasi.

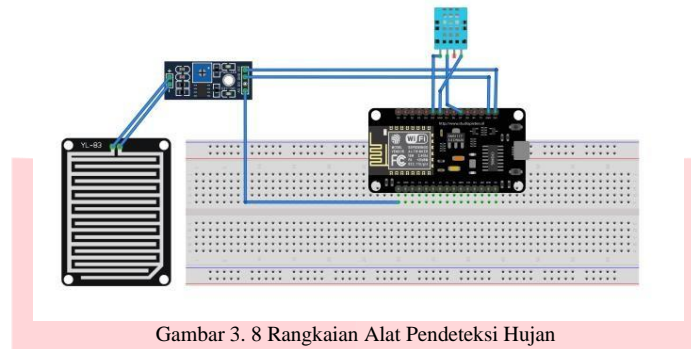
#### 3.5.3 Raindrop Sensor

Sensor hujan adalah jenis sensor yang berfungsi untuk mendeteksi terjadinya hujan atau tidak, yang dapat difungsikan dalam segala macam aplikasi dalam kehidupan sehari – hari. Dipasaran sensor ini dijual dalam bentuk module sehingga hanya perlu menyediakan kabel jumper untuk dihubungkan ke mikrokontroler atau Arduino. Prinsip kerja dari modul sensor ini yaitu pada saat ada air hujan turun dan mengenai panel sensor

maka akan terjadi proses elektrolisis oleh air hujan. Dan karena air hujan termasuk dalam golongan cairan elektrolit yang dimana cairan tersebut akan menghantarkan arus listrik.

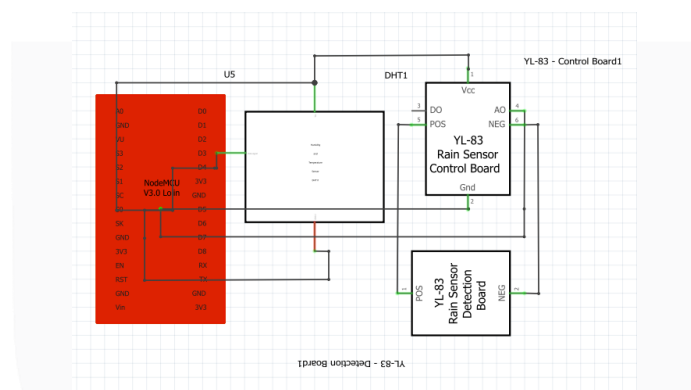
### 3.6 Rangkaian Schematic Alat Pendeteksi Hujan

Rangkaian *schematic* alat dibuat dengan tujuan memudahkan proses perangkaian komponen-komponen yang digunakan agar tidak terjadi kesalahan dalam perakitan. Berikut adalah rangkaian *schematic* alat pendeteksi hujan.



Gambar 3. 8 Rangkaian Alat Pendeteksi Hujan

Rangkaian alat di atas menunjukkan beberapa *port* pada perangkat NodeMCU yang digunakan dalam perancangan alat pendeteksi hujan. Berikut adalah gambar rangkaian *schematic* alat pendeteksi hujan beserta tabel penjelasan fungsi dari masing-masing *port* pada perangkat NodeMCU yang digunakan dalam perancangan alat pendeteksi hujan.



Gambar 3. 9 Rangkaian Schematic Alat Pendeteksi Hujan

Tabel 3.1 Penggunaan Port NodeMCU (ESP8266)

No	Port	Keterangan
1	A0	Digunakan untuk menerima data analog dari <i>Controlling Board</i> sensor <i>Raindrop</i>
2	D7	Digunakan untuk menerima data digital dari DHT11
3	3v3	Digunakan untuk mencatu semua sensor yang terhubung dengan NodeMCU
4	GND	Digunakan sebagai <i>ground</i> untuk semua sensor yang terhubung dengan NodeMCU



### 3.7 Perangkat Lunak

Komponen *software* digunakan pada proses perancangan alat, *website* dan aplikasi dari mulai penyematian *sketch* program pada NodeMCU, hingga data yang berasal dari *database* tersebut akan diterima oleh pengguna sebagai indikator untuk melakukan pemantauan dua kondisi cuaca yang telah direkam menggunakan sensor pada perangkat NodeMCU. Proyek akhir ini menggunakan Firebase sebagai *real-time database*-nya. Firebase berfungsi untuk menerima nilai dari data yang telah dibaca oleh sensor dan kemudian akan diteruskan dan ditampilkan ke laman *web*. Laman *web* tersebut dapat menampilkan nilai data hasil rekaman dari sensor jika API (Application Programming Interface) *key* dari NodeMCU telah disematkan kedalam *source code* dari *website* yang telah dirancang dengan menggunakan HTML. Lalu setelah API *key* disematkan, terdapat sintaks JavaScript khusus yang berfungsi untuk menampilkan nilai data hasil dari rekaman sensor. HTML dan CSS berfungsi untuk mengatur tampilan dari *website* yang telah dirancang seperti contohnya transisi dan pembuatan *widget* dari masing-masing halaman serta membuat *navigation bar*.

Selanjutnya, setelah *website* menampilkan nilai data dari hasil pembacaan sensor, maka langkah terakhir yaitu melakukan perancangan dari aplikasi android dapat menggunakan *platform* tambahan, seperti Phonegap, React Native, Gradle, MIT App Inventor dan aplikasi lainnya. Pada dasarnya, Phonegap adalah sebuah *framework* dari Adobe yang mampu membangun aplikasi berdasarkan *website*.

#### 3.7.1 Visual Studio Code

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

#### 3.7.2 Arduino IDE

Pada proses perancangan proyek akhir ini, Arduino IDE digunakan sebagai *software* yang berfungsi untuk menyematkan *sketch* ke dalam NodeMCU yang sudah terhubung dengan sensor agar sensor tersebut dapat membaca objek sesuai dengan fungsinya masing-masing.

#### 3.7.3 Firebase

Pada proses perancangan proyek akhir ini, salah satu fitur dari Firebase yang digunakan adalah *real-time database*-nya. Fitur tersebut memungkinkan aplikasi pendeteksi hujan bisa diakses langsung oleh pengguna. Aplikasi juga dapat menampilkan data hasil keluaran dari sensor secara lokal ketika tidak ada akses internet. Kemudian jika sudah terhubung kembali, fitur tersebut akan melakukan sinkronisasi data secara otomatis.

Berikut ini tampilan dari *database* dari aplikasi pendeteksi hujan.

(1)	(2)	(3)
<p>CGN01</p> <ul style="list-style-type: none"> <li>Humidity: 27</li> <li>Raindrop: 1024</li> <li>Temperature: 61</li> </ul>	<p>CGN02</p> <ul style="list-style-type: none"> <li>Humidity: 26</li> <li>Raindrop: 1024</li> <li>Temperature: 67</li> </ul>	<p>CGN03</p> <ul style="list-style-type: none"> <li>Humidity: 26</li> <li>Raindrop: 700</li> <li>Temperature: 90</li> </ul>

Gambar 3. 10 Tampilan *database* Aplikasi Pendeteksi Hujan

Gambar 3.13 adalah tampilan *database* dari tiga lokasi peletakan sensor yang diberi kode pada masing-masing alatnya, berikut penjelasannya:

1. Tampilan *Child* Suhu dalam Fahrenheit, kelembaban, dan kondisi cuaca di Kos Wisma Purnama 1
2. Tampilan *Child* Suhu dalam Fahrenheit, kelembaban, dan kondisi cuaca di Kos Violet
3. Tampilan *Child* Suhu dalam Fahrenheit, kelembaban, dan kondisi cuaca di Universitas Telkom

data hasil keluaran dari masing-masing sensor, hingga mengelola tampilan dari *website* dan aplikasi pendeteksi hujan.

## 4. Pengujian Alat dan Sistem

### 4.1 Deskripsi Simulasi Sistem

Simulasi sistem dimulai dari implementasi sistem aplikasi pendeteksi hujan di wilayah ciganitri yang rawan terjadi peristiwa hujan lokal, meliputi tiga titik lokasi yaitu kos Wisma Purnama Putra, kos Violet dan Universitas Telkom. Variabel yang di ukur meliputi suhu, kelembaban dan kondisi cuaca di tiga lokasi tersebut. Pengukuran dilakukan dua kali. Yaitu satu kali dengan rentang pengukuran tiap satu jam, dan satu kali dengan rentang jam pengukuran acak. Kegiatan pengukuran menggunakan aplikasi Accuweather sebagai acuan dari tingkat akurasi pengukuran sensor.

### 4.2 Pengujian Fungsionalitas Aplikasi

Pengujian ini dilakukan untuk mengetahui dan memastikan semua aplikasi pada proyek akhir yang sebelumnya telah dirancang serta dilakukan pengembangan dapat berjalan dengan baik. Pengujian dilakukan pada alat pendeteksi hujan serta seluruh fitur dari aplikasi *website* maupun aplikasi android dari sistem pendeteksi hujan.

#### 4.2.1 Pengujian Fungsionalitas Aplikasi Web

Pengujian dilakukan untuk mengetahui serta memastikan semua fitur yang terdapat pada aplikasi *website* sudah berjalan dengan baik sesuai fungsinya masing-masing. Tahap pengujian ini dilakukan dengan cara menjalankan semua fitur yang ada pada sistem tersebut.

Tabel 4.1 Tabel Pengujian Fungsionalitas Website

No	Nama Pengujian	Aksi	Hasil yang diharapkan	Hasil Pengujian	Status
1	Melihat tampilan halaman beranda <i>website</i>	Klik beranda pada menu <i>sidebar</i>	Menampilkan halaman beranda	Menampilkan halaman beranda	Berhasil
2	Transisi pergantian <i>caption</i> pada carousel di halaman beranda	Transisi otomatis atau klik tanda arah di bagian kanan dan kiri carousel	Transisi <i>caption</i> otomatis dan manual	Transisi <i>caption</i> otomatis dan manual	Berhasil
3	Melihat halaman pemantauan hujan di kos Wisma Purnama Putra	Pilih dan klik lokasi pemantauan kos Wisma Purnama Putra pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar kos Wisma Purnama Putra	Menampilkan kondisi detail cuaca di sekitar kos Wisma Purnama Putra	Berhasil
4	Melihat halaman pemantauan hujan di kos Violet	Pilih dan klik lokasi pemantauan kos Violet pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar kos Violet	Menampilkan kondisi detail cuaca di sekitar kos Violet	Berhasil
5	Melihat halaman pemantauan hujan di sekitar Universitas Telkom	Pilih dan klik lokasi pemantauan Universitas Telkom pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar Universitas Telkom	Menampilkan kondisi detail cuaca di wilayah sekitar Universitas Telkom	Berhasil
6	Melihat suhu dan waktu terkini pada grafik	Arahkan <i>cursor</i> ke titik paling kanan dari grafik perubahan suhu	Menampilkan suhu dan waktu terkini pada grafik	Menampilkan suhu dan waktu terkini pada grafik	Berhasil
7	Melihat titik tertinggi dan terendah pengukuran suhu di grafik	Arahkan <i>cursor</i> ke titik puncak dan titik terendah dari grafik	Menampilkan suhu tertinggi dan terendah pada grafik	Menampilkan suhu tertinggi dan terendah pada grafik	Berhasil
8	Mengubah <i>range</i> pengukuran menjadi satu menit	Klik <i>button</i> 1M pada grafik perubahan suhu	Menampilkan grafik hasil pengukuran dalam <i>range</i> satu menit	Menampilkan grafik hasil pengukuran dalam <i>range</i> satu menit	Berhasil

9	Mengubah <i>range</i> pengukuran menjadi lima menit	Klik <i>button</i> 5M pada grafik perubahan suhu	Menampilkan grafik hasil pengukuran dalam <i>range</i> lima menit	Menampilkan grafik hasil pengukuran dalam <i>range</i> lima menit	Berhasil
---	---	--	---	---	----------

#### 4.2.1 Pengujian Fungsionalitas Aplikasi Android

Pengujian dilakukan untuk mengetahui serta memastikan semua fitur yang terdapat pada aplikasi android sudah berjalan dengan baik sesuai fungsinya masing-masing. Tahap pengujian ini dilakukan dengan cara menjalankan semua fitur yang ada pada sistem tersebut.

Tabel 4.2 Pengujian Fungsionalitas Aplikasi Android

No	Nama Pengujian	Aksi	Hasil yang diharapkan	Hasil Pengujian	Status
1	Menampilkan dan menyembunyikan <i>sidebar</i>	Klik <i>toggle bar</i> di bagian atas halaman	<i>Sidebar</i> terlihat dan tidak terlihat	<i>Sidebar</i> terlihat dan tidak terlihat	Berhasil
2	Melihat tampilan halaman beranda <i>website</i>	Klik <i>toggle bar</i> kemudian klik beranda pada menu <i>sidebar</i>	Menampilkan halaman beranda	Menampilkan halaman beranda	Berhasil
3	Transisi pergantian <i>caption</i> pada carousel di halaman beranda	Transisi otomatis atau klik tanda arah di bagian kanan dan kiri carousel	Transisi <i>caption</i> otomatis dan manual	Transisi <i>caption</i> otomatis dan manual	Berhasil
4	Melihat halaman pemantauan hujan di kos Wisma Purnama Putra	Pilih dan klik lokasi pemantauan kos Wisma Purnama Putra pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar kos Wisma Purnama Putra	Menampilkan kondisi detail cuaca di sekitar kos Wisma Purnama Putra	Berhasil
5	Melihat halaman pemantauan hujan di kos Violet	Pilih dan klik lokasi pemantauan kos Violet pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar kos Violet	Menampilkan kondisi detail cuaca di sekitar kos Violet	Berhasil
6	Melihat halaman pemantauan hujan di sekitar Universitas Telkom	Pilih dan klik lokasi pemantauan Universitas Telkom pada <i>sidebar</i>	Menampilkan kondisi detail cuaca di sekitar Universitas Telkom	Menampilkan kondisi detail cuaca di wilayah sekitar Universitas Telkom	Berhasil
7	Melihat suhu dan waktu terkini pada grafik	Klik titik paling kanan dari grafik perubahan suhu	Menampilkan suhu dan waktu terkini pada grafik	Menampilkan suhu dan waktu terkini pada grafik	Berhasil
8	Melihat titik tertinggi dan terendah pengukuran suhu di grafik	Klik titik puncak dan titik terendah dari grafik	Menampilkan suhu tertinggi dan terendah pada grafik	Menampilkan suhu tertinggi dan terendah pada grafik	Berhasil
9	Mengubah <i>range</i> pengukuran menjadi satu menit	Klik <i>button</i> 1M pada grafik perubahan suhu	Menampilkan grafik hasil pengukuran dalam <i>range</i> satu menit	Menampilkan grafik hasil pengukuran dalam <i>range</i> satu menit	Berhasil
10	Mengubah <i>range</i> pengukuran menjadi lima menit	Klik <i>button</i> 5M pada grafik perubahan suhu	Menampilkan grafik hasil pengukuran dalam <i>range</i> lima menit	Menampilkan grafik hasil pengukuran dalam <i>range</i> lima menit	Berhasil

#### 4.3 Pengujian Performansi Website

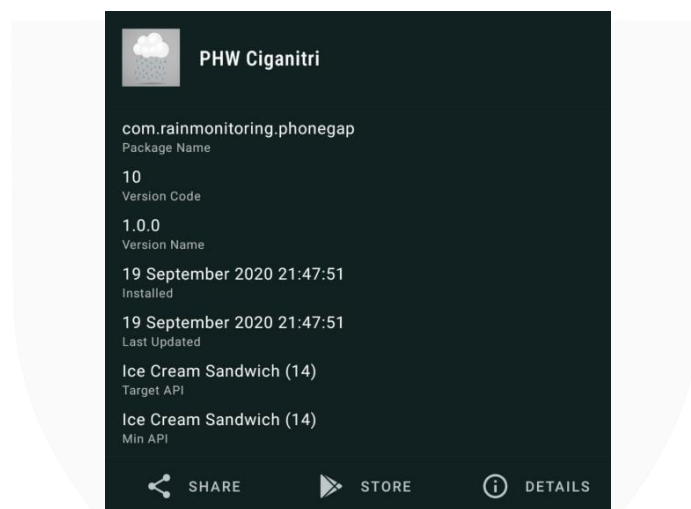
Pengujian ini dilakukan untuk mengetahui performansi dari website yang telah dikembangkan. Website tersebut akan diuji menggunakan suatu ekstensi dari Web Browser Google Chrome yang bernama Google Lighthouse. Faktor-faktor performansi yang diuji pada website dan hasilnya dapat dilihat pada tabel berikut:

Tabel 4.3 Tabel Pengujian Performansi Website

Indikator	Keterangan	Hasil (detik)
<i>First Contentful Paint</i>	Waktu yang dibutuhkan oleh teks dan gambar pertama berhasil ditampilkan	4.2
<i>Speed Index</i>	Waktu yang dibutuhkan untuk sebuah halaman web terisi oleh konten	4.2
<i>Largest Contentful Paint</i>	Waktu yang dibutuhkan oleh teks dan gambar yang berukuran paling besar untuk ditampilkan	7.8
<i>Time to Interactive</i>	Waktu yang dibutuhkan halaman untuk digunakan secara interaktif dengan pengguna	7.2
<i>Total Blocking Time</i>	Jumlah waktu dari indikator <i>First Contentful Paint</i> dan <i>Time to Interactive</i>	0
<i>Cumulative Layout Shift</i>	Tenggat waktu pergeseran antar elemen yang terdapat pada halaman web	0

#### 4.4 Pengujian Kompatibilitas Android

Pengujian ini dilakukan menggunakan aplikasi AppChecker yang dapat diunduh melalui Google Play Store. Pengujian ini berfungsi untuk mengetahui spesifikasi sistem operasi minimum yang dibutuhkan untuk menginstal aplikasi pendeteksi hujan.



Gambar 4.1 Spesifikasi Sistem Operasi Minimum yang Dibutuhkan Untuk Menginstal Aplikasi

Berdasarkan Gambar 4.2, pengujian kompatibilitas aplikasi android dapat diinstal pada versi android yang ditargetkan ketika melakukan perancangan dan pengembangan aplikasi. Versi android yang dibutuhkan untuk menginstal aplikasi pendeteksi hujan ini adalah android versi 4.0 atau dapat juga disebut android *Ice Cream Sandwich* dengan level API 14.

#### 4.5 Pengujian Alat Pendeteksi Hujan

Pengujian ini dilakukan agar dapat mengetahui kinerja dari alat yang telah dirancang sebelumnya apakah alat tersebut sudah dapat bekerja sesuai dengan fungsinya dan sesuai dengan apa yang ditargetkan, serta berdasarkan hasil pengujian yang diperoleh dapat mengetahui kelebihan dan kekurangan dari alat pendeteksi hujan secara menyeluruh. Pengujian ini dilakukan di waktu dan tempat yang sama pada saat melakukan perencanaan peletakan alat, yaitu di Kos Wisma Purnama Putra Ciganitri, Kos Violet Ciganitri, dan Universitas Telkom.

#### 4.5.1 Rekapitulasi Hasil Pengujian Alat di Sekitar Kos Wisma Purnama Ciganitri

Pengujian alat yang diletakkan di wilayah ini memperhatikan jarak antar satu alat pendeteksi dengan alat yang lain. Pengujian dilakukan dalam waktu satu hari. Variabel yang menjadi objek pengujian alat pendeteksi di wilayah ini meliputi suhu, kelembaban, dan dilakukan pada dua kondisi cuaca yaitu pada saat kondisi hujan dan tidak hujan. Berikut adalah rekapitulasi hasil pengujian alat pendeteksi hujan untuk di wilayah Kos Wisma Purnama Putra.

Tabel 4.4 Rekapitulasi Hasil Pengujian Pertama dengan Rentang 5 Jam

Waktu (WIB)	Suhu (°C)	Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
01.22	22	-	-	71	83	Tidak Hujan
02.22	21	20	5%	69	84	Tidak Hujan
03.22	21	20	5%	69	84	Tidak Hujan
04.22	21	20	5%	69	81	Tidak Hujan
05.22	20	19	5%	67	80	Tidak Hujan
06.22	22	19	15%	71	73	Tidak Hujan

Pengujian pertama dilakukan dengan durasi 6 jam. Pada rentang waktu tersebut, wilayah sekitar kos Wisma Purnama Putra sedang tidak mengalami hujan. Pada jam pertama, sensor DHT11 melakukan kalibrasi sehingga belum bisa memperoleh nilai persentase kesalahan relatif. Hasil pengukuran sensor DHT11 untuk pengukuran variabel suhu dan kelembaban menunjukkan suhu terendah terjadi pada pukul 05.22 WIB yaitu 20° Celcius, sedangkan untuk suhu tertinggi yaitu pada pukul 01.22 WIB dan pukul 06.22 WIB dengan nilai suhu 22° Celcius. Persentase kesalahan relatif menunjukkan seberapa besar tingkat kesalahan hasil pengukuran sensor DHT11 terhadap hasil pengukuran yang dijadikan acuan. Jadi dengan adanya persentase kesalahan relatif, maka dapat diketahui tingkat akurasi pengukurannya dalam satuan persen (%).

Tabel 4.5 Rekapitulasi Hasil Pengujian Kedua pada Rentang Jam Tertentu

Waktu (WIB)	Deteksi Suhu DHT11 (°C)	Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
08.04	28	26	7%	82	63	Tidak Hujan
11.04	35	34.7	6%	94	52	Tidak Hujan
13.04	36	33	9%	97	49	Tidak Hujan
20.10	24	-	-	80	80	Gerimis

Pengujian kedua dilakukan dengan durasi 4 jam di rentang jam tertentu, yaitu pada pukul 08.04 WIB, 11.04 WIB, pukul 13.04 WIB, dan pukul 20.10 WIB. Di antara rentang waktu tersebut, tepatnya pada pukul 20.10 WIB, terpantau kondisi di wilayah sekitar kos Wisma Purnama Putra mengalami gerimis. Hasil pengukuran sensor DHT11 untuk pengukuran variabel suhu dan kelembaban menunjukkan suhu terendah terjadi pada pukul 20.10 WIB, yaitu 24° Celcius karena dalam rentang waktu tersebut, kondisi cuaca gerimis dan sudah memasuki malam hari. Sedangkan untuk suhu tertinggi yaitu pada pukul 13.04 WIB dengan nilai suhu 36° Celcius. Terjadi perbedaan yang cukup jauh pada hasil pengukuran suhu yang didapat antara pukul 20.10 WIB dan pukul 13.04 WIB. Suhu tinggi didapatkan karena sensor DHT11 terkena sinar matahari langsung saat proses pengukurannya.

#### 4.5.2 Rekapitulasi Hasil Pengujian Alat di Sekitar Kos Violet Ciganitri

Pengujian alat yang diletakkan di wilayah ini memperhatikan jarak antar satu alat pendeteksi dengan alat yang lain. Pengujian dilakukan dalam waktu satu hari. Variabel yang menjadi objek pengukuran alat pendeteksi di wilayah ini meliputi suhu, kelembaban, dan dilakukan pada dua kondisi cuaca yaitu pada saat

kondisi hujan dan tidak hujan. Berikut adalah rekapitulasi hasil pengujian alat pendeteksi hujan untuk di wilayah Kos Violet.

Tabel 4. 1 Rekapitulasi Hasil Pengujian Pertama dengan Rentang 5 Jam

Waktu (WIB)	Suhu (°C)	Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
01.22	23	-	-	73	81	Tidak Hujan
02.22	23	20	15%	72	82	Tidak Hujan
03.22	22	20	10%	72	83	Tidak Hujan
04.22	23	20	15%	73	79	Tidak Hujan
05.22	22	19	15%	72	77	Tidak Hujan
06.22	23	19	21%	73	78	Tidak Hujan

Pengujian pertama dilakukan dengan durasi 6 jam. Pada rentang waktu tersebut, wilayah sekitar kos Violet sedang tidak mengalami hujan. Pada jam pertama, sensor DHT11 melakukan kalibrasi sehingga belum bisa memperoleh nilai persentase kesalahan relatif. Hasil pengukuran sensor DHT11 untuk pengukuran variabel suhu dan kelembaban menunjukkan suhu terendah terjadi pada pukul 05.22 WIB yaitu 20° Celcius, sedangkan untuk suhu tertinggi yaitu pada pukul 01.22 WIB dan pukul 06.22 WIB dengan nilai suhu 22° Celcius. Persentase kesalahan relatif menunjukkan seberapa besar tingkat kesalahan hasil pengukuran sensor DHT11 terhadap hasil pengukuran yang dijadikan acuan. Jadi dengan adanya persentase kesalahan relatif, maka dapat diketahui tingkat akurasi pengukurannya dalam satuan persen (%).

Tabel 4. 2 Rekapitulasi Hasil Pengujian Kedua Pada Rentang Jam Tertentu

Waktu (WIB)	Deteksi Suhu DHT11 (°C)	Deteksi Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
08.04	28	26	7%	82	66	Tidak Hujan
10.04	33	33	0%	91	48	Tidak Hujan
11.04	34	33	3%	92	44	Tidak Hujan
20.10	26	-	-	78	76	Hujan

Pengujian kedua dilakukan dengan durasi 4 jam di rentang jam tertentu, yaitu pada pukul 08.04 WIB, 10.04 WIB, pukul 11.04 WIB, dan pukul 20.10 WIB. Di antara rentang waktu tersebut, tepatnya pada pukul 20.10 WIB, terpantau kondisi di wilayah sekitar kos Violet mengalami hujan. Hasil pengukuran sensor DHT11 untuk pengukuran variabel suhu dan kelembaban menunjukkan suhu terendah terjadi pada pukul 20.10 WIB, yaitu 26° Celcius karena dalam rentang waktu tersebut, kondisi cuaca dalam keadaan hujan dan sudah memasuki malam hari. Sedangkan untuk suhu tertinggi yaitu pada pukul 11.04 WIB dengan nilai suhu 34° Celcius. Terjadi perbedaan yang cukup jauh pada hasil pengukuran suhu yang didapat antara pukul 20.10 WIB dan pukul 11.04 WIB. Suhu tinggi didapatkan karena sensor DHT11 terkena sinar matahari langsung saat proses pengukurannya.

#### 4.5.2 Rekapitulasi Hasil Pengujian Alat di Sekitar Universitas Telkom

Sama dengan pengujian di daerah yang lain, pengujian alat yang diletakkan di wilayah ini memperhatikan jarak antar satu alat pendeteksi dengan alat yang lain. Pengujian dilakukan dalam waktu satu hari. Variabel yang menjadi objek pengukuran alat pendeteksi di wilayah ini meliputi suhu, kelembaban, dan dilakukan pada dua kondisi cuaca yaitu pada saat kondisi hujan dan tidak hujan. Berikut adalah rekapitulasi hasil pengujian alat pendeteksi hujan untuk di daerah sekitar Universitas Telkom.

Tabel 4. 3 Rekapitulasi Hasil Pengujian Pertama dengan Rentang 5 Jam

Waktu (WIB)	Suhu (°C)	Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
01.22	22	-	-	71	79	Tidak Hujan
02.22	22	20	10%	71	81	Tidak Hujan
03.22	22	20	10%	71	78	Tidak Hujan
04.22	22	20	10%	71	77	Tidak Hujan
05.22	22	19	15%	71	77	Tidak Hujan
06.22	22	19	15%	71	77	Tidak Hujan

Pengujian pertama dilakukan dengan durasi 6 jam. Pada rentang waktu tersebut, wilayah sekitar kos Violet sedang tidak mengalami hujan. Hasil pengukuran dari sensor suhu DHT11 menunjukkan suhu di lingkungan tersebut relatif stabil. Pada jam pertama, sensor DHT11 melakukan kalibrasi sehingga belum bisa memperoleh nilai persentase kesalahan relatif. Persentase kesalahan relatif menunjukkan seberapa besar tingkat kesalahan hasil pengukuran sensor DHT11 terhadap hasil pengukuran yang dijadikan acuan. Jadi dengan adanya persentase kesalahan relatif, maka dapat diketahui tingkat akurasi pengukurannya dalam satuan persen (%).

Tabel 4. 4 Rekapitulasi Hasil Pengujian Kedua Pada Rentang Jam Tertentu

Waktu (WIB)	Deteksi Suhu DHT11 (°C)	Deteksi Suhu Aplikasi Accuweather (°C)	Kesalahan Relatif (%)	Suhu (°F)	Kelembaban (%)	Kondisi
08.15	28	28	0%	81	69	Tidak Hujan
09.24	29	32	9%	84	65	Tidak Hujan
10.33	31	33	6%	87	58	Tidak Hujan
22.08	-	-	-	-	-	-

Pengujian kedua untuk alat yang dipasang di wilayah Universitas Telkom dimulai dari pukul 08.15 WIB hingga pukul 10.33. Pada saat melakukan pengujian secara bersamaan, koneksi internet yang digunakan sebagai media komunikasi perangkat inti dari alat pendeteksi hujan sedang mengalami gangguan.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil perancangan, pengujian dan analisa yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Berdasarkan pengujian fungsionalitas, aplikasi *website* pendeteksi hujan wilayah Ciganitri berjalan dengan baik. Hal tersebut dibuktikan dengan pengujian fungsionalitas yang mendapatkan hasil keakuratan 100%, sehingga aplikasi *website* dapat memudahkan mahasiswa maupun masyarakat umum yang tinggal di wilayah Ciganitri untuk memantau kondisi cuaca di wilayah tersebut secara *real-time*.
2. Berdasarkan pengujian fungsionalitas, aplikasi android pendeteksi hujan wilayah Ciganitri berjalan dengan baik. Hal tersebut dibuktikan dengan pengujian fungsionalitas yang mendapatkan hasil keakuratan 100%, sehingga bagi mahasiswa maupun masyarakat umum yang tinggal di wilayah Ciganitri dan sedang ingin melakukan perjalanan ke wilayah sekitar Universitas Telkom maupun sebaliknya dapat melakukan pemantauan kondisi cuaca terlebih dahulu menggunakan aplikasi android pendeteksi hujan wilayah Ciganitri.
3. Hasil pengujian performansi *website* pendeteksi hujan menggunakan Google Lighthouse mendapatkan nilai 55. Hal ini dikarenakan beberapa faktor yang mempengaruhi kinerja *website*, salah satunya adalah konten yang disisipkan kedalam *website* tersebut. Akan tetapi, nilai yang dihasilkan menurut Google Lighthouse memiliki hasil performansi yang cukup baik jika dilihat dari rentang nilainya.
4. Pengujian kompatibilitas aplikasi android dilakukan dengan melakukan pengecekan pada aplikasi AppChecker yang dapat diunduh melalui Google *Play Store* untuk mengetahui batas minimum versi sistem operasi android yang dibutuhkan. Pengujian tersebut mendapatkan hasil bahwa spesifikasi minimum dari versi sistem operasi android yang dibutuhkan untuk melakukan instalasi aplikasi pendeteksi hujan adalah pada versi android 4.0 (*Ice Cream Sandwich*), sehingga perangkat yang memiliki versi android lebih lawas dari versi android 4.0 tidak dapat melakukan instalasi aplikasi pendeteksi hujan pada perangkatnya.
5. Hasil pengujian alat pendeteksi hujan dinilai cukup akurat karena perbedaan hasil pengukuran alat pendeteksi hujan dengan alat ukur yang digunakan oleh acuan pengukuran tidak berbeda jauh. Diketahui rata-rata kesalahan relatif sensor DHT11 adalah 8.35% jika dibandingkan dengan pengukuran acuan. Artinya kesalahan pengukuran pada sensor DHT11 relatif rendah.

### 5.2 Saran

Berdasarkan hasil kesimpulan proyek akhir ini, dapat disampaikan beberapa saran untuk pengembangan selanjutnya yaitu :

1. *Interface* aplikasi android agar dibuat lebih menarik.
2. Sistem catu dayanya dapat menggunakan *solar cell* atau baterai.
3. Melakukan pengembangan dengan cara menambahkan sensor yang digunakan sebagai alat ukur pada alat pendeteksi hujan agar variabel objek yang diukur lebih lengkap dan beragam seperti sensor kecepatan angin, tekanan udara.
4. Untuk aplikasi android dapat dikembangkan dengan menggunakan *framework* dari *platform* lain seperti React Native dan Flutter agar bisa dikembangkan menggunakan sistem operasi iOS.
5. Menambahkan fitur *controlling* pada aplikasi android agar dapat terintegrasi dengan salah satu bagian dari IoT seperti *home automation*.
6. Memperluas cakupan pemantauan di wilayah lain yang masih sering mengalami peristiwa hujan lokal dengan menambahkan jumlah alat pendeteksi hujan yang dipasang pada lokasi strategis.

## DAFTAR PUSTAKA

- [1] V. Tsiatsis, S. Karnouskos, J. Holler, D. Boyle and C. Mulligan, *Internet of Things : Technologies and Applications for a New Age of Intelligence second edition*, Stockholm: Academic Press, 2018.
- [2] Y. Muslimin and R. Kartono, *Konsep, Praktek, dan Implementasi JavaScript Untuk Mahasiswa dan Programmer*, Jakarta: Logika, 2017.
- [3] Yuhefizar, H. Moduuto and R. Hidayat, *Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla (CMS) Edisi Revisi*, Jakarta: PT. Elex Media Komputindo, 2009.
- [4] W. Komputer, *Membangun Aplikasi Cross Platform Dengan Phonegap*, Jakarta: PT. Elex Media Komputindo, 2014.
- [5] R. Ananda, *40 Project Robotik dan Aplikasi Android*, Yogyakarta: Deepublish, 2018.



- [6] J. Enterprise, Pengenalan HTML dan CSS, Jakarta: PT. Elex Media Komputindo, 2016.
- [7] Koesheryatin, Suryana and Taryana, Aplikasi Internet Menggunakan HTML, CSS, dan JavaScript, Jakarta: PT. Elex Media Komputindo, 2014.
- [8] J. Karman, H. Mulyono and A. T. Martadinata, Sistem Informasi Geografis Berbasis Android Studi Kasus Aplikasi SIG Pariwisata, Yogyakarta: Deepublish, 2019.
- [9] S. Syaifullah, I. G. P. S. Wijaya and A. Y. Husodo, "Sistem Informasi Kepuasan Layanan Administrasi Akademik Berbasis IPA (Importance Performance Analysis) Studi Kasus Fakultas Teknik Universitas Mataram," *Journal of Computer Science and Informatics Engineering (J-Cosine)*, vol. 2, no. 1, pp. 37-43, 2018.
- [10] M. I. Aljundi and M. A. Akbar, Kotlin Zero To Hero : Membuat Aplikasi Android dengan Kotlin Cocok Untuk Pemula, Udacoding, 2018.