

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh (Kania dkk. 2024) dengan judul “Sistem Monitoring Deteksi Cuaca Berbasis Internet of Things (IoT)” bertujuan mengembangkan sistem monitoring cuaca pada lingkungan Kampus Universitas Banten Jaya untuk menyediakan informasi cuaca lokal yang belum tersedia pada website kampus. Penelitian ini menggunakan metode Waterfall yang meliputi tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Sistem dibangun menggunakan NodeMCU ESP8266 sebagai mikrokontroler, sensor DHT11 untuk mengukur suhu dan kelembapan udara, sensor LDR untuk mendeteksi intensitas cahaya, serta sensor raindrop untuk mendeteksi kondisi hujan. Data hasil pembacaan sensor dikirimkan ke server dan ditampilkan melalui website sehingga pengguna dapat memantau kondisi cuaca secara real-time. Pengujian sistem dilakukan menggunakan metode black box testing yang menunjukkan bahwa seluruh fungsi, mulai dari pembacaan sensor, pengiriman data ke server, hingga tampilan informasi pada website, berjalan sesuai dengan yang diharapkan. Penelitian ini berhasil menghasilkan sistem monitoring cuaca yang dapat dimanfaatkan oleh mahasiswa, dosen, dan staf kampus dalam memperoleh informasi kondisi cuaca secara aktual. Penelitian tersebut memiliki kesamaan dengan penelitian ini, yaitu sama-sama mengembangkan sistem monitoring cuaca berbasis Internet of Things pada lingkungan kampus. Namun, penelitian ini menggunakan ESP32, sensor BME680 dan BH1750, Firebase Realtime Database, aplikasi Android berbasis MIT App Inventor, serta penyimpanan riwayat cuaca sehingga informasi yang dihasilkan lebih lengkap.

Penelitian selanjutnya dilakukan oleh (Fatahillah dan Roza, 2022) dengan judul “Perancangan Monitoring Cuaca Berbasis IoT Tenaga Matahari Menggunakan Panel Surya 20 WP”. Penelitian ini bertujuan merancang sistem monitoring cuaca yang dapat diakses dari jarak jauh dengan memanfaatkan teknologi Internet of Things dan sumber energi dari panel surya. Sistem menggunakan Arduino Uno sebagai pengendali utama, NodeMCU ESP8266

sebagai modul komunikasi Wi-Fi, sensor DHT11 untuk mengukur suhu dan kelembapan udara, sensor hujan untuk mendeteksi curah hujan, serta panel surya 20 WP sebagai sumber catu daya. Data sensor dikalibrasi oleh Arduino Uno sebelum dikirimkan ke server Blynk melalui NodeMCU ESP8266 sehingga dapat dipantau menggunakan smartphone. Penelitian juga melakukan pengujian terhadap catu daya, sensor DHT11, sensor hujan, NodeMCU ESP8266, Arduino Uno, dan panel surya. Hasil pengujian menunjukkan bahwa keluaran catu daya berada pada kisaran 5,03-5,05 V sesuai kebutuhan sistem, sedangkan sensor hujan menghasilkan tegangan keluaran sebesar 4,93 V pada kondisi kering dan menurun hingga 0,98 V pada kondisi terendam air. Penelitian ini membuktikan bahwa sistem mampu melakukan monitoring cuaca secara real-time dengan memanfaatkan energi surya sebagai sumber daya utama. Meskipun demikian, sistem masih menggunakan dua mikrokontroler dan aplikasi Blynk sebagai media pemantauan. Berbeda dengan penelitian tersebut, penelitian ini menggunakan ESP32 sebagai mikrokontroler tunggal, Firebase Realtime Database sebagai media penyimpanan data, aplikasi Android berbasis MIT App Inventor, serta menambahkan fitur klasifikasi kondisi cuaca dan penyimpanan riwayat cuaca.

Penelitian berikutnya dilakukan oleh (Kotangon dkk, 2024) dengan judul “Perancangan dan Implementasi Sistem Monitoring Kondisi Cuaca Berbasis Internet of Things (IoT)”. Penelitian ini bertujuan membangun sistem monitoring kondisi cuaca yang dapat membantu pengguna memperoleh informasi cuaca melalui teknologi Internet of Things. Penelitian menggunakan metode Prototype dalam proses pengembangan sistem. Perangkat yang digunakan meliputi NodeMCU ESP8266 sebagai mikrokontroler, sensor DHT22 untuk mengukur suhu dan kelembapan udara, sensor hujan untuk mendeteksi kondisi hujan, serta aplikasi Blynk sebagai media monitoring pada smartphone. Sistem bekerja dengan menghubungkan NodeMCU ESP8266 ke jaringan internet, kemudian membaca data dari sensor dan mengirimkannya ke aplikasi Blynk secara real-time. Berdasarkan hasil pengujian perangkat lunak menggunakan black box testing, seluruh fungsi sistem dapat berjalan sesuai dengan kebutuhan. Namun, pada bagian kesimpulan penulis menyatakan bahwa sensor hujan belum bekerja secara optimal karena adanya perbedaan kebutuhan tegangan antara sensor hujan dan NodeMCU

ESP8266 yang menggunakan tegangan 3V, sedangkan sensor memerlukan tegangan 5V. Penelitian ini memberikan gambaran mengenai implementasi sistem monitoring cuaca berbasis IoT yang sederhana dan mudah diakses melalui smartphone. Penelitian yang dilakukan pada Proyek Akhir ini memiliki tujuan yang serupa, tetapi menggunakan ESP32 dengan sensor BME680, sensor BH1750, dan sensor hujan analog sehingga parameter yang dipantau lebih lengkap. Selain itu, penelitian ini memanfaatkan Firebase Realtime Database sebagai media penyimpanan data dan aplikasi Android berbasis MIT App Inventor sebagai antarmuka pengguna, serta menyediakan fitur klasifikasi kondisi cuaca berbasis skor tertimbang dengan baseline adaptif dan riwayat cuaca.

Penelitian yang dilakukan oleh Dian Rahmadani, Andie, dan Fauzi Yusa Rahman pada tahun 2020 dengan judul “Stasiun Cuaca Berbasis IoT Menggunakan NodeMCU” bertujuan untuk membangun sistem stasiun cuaca berbasis Internet of Things (IoT) yang mampu mengukur parameter meteorologi dan mengirimkan hasil pengukuran ke database sehingga dapat diakses oleh pengguna melalui website. Sistem menggunakan mikrokontroler NodeMCU yang terhubung dengan sensor DHT11 dan BMP180 untuk memperoleh data suhu, kelembapan, tekanan udara, titik embun, dan ketinggian. Data hasil pembacaan sensor diproses kemudian dikirimkan ke database MySQL untuk ditampilkan dalam bentuk grafik dan tabel melalui antarmuka web. Hasil penelitian menunjukkan bahwa sistem mampu melakukan monitoring cuaca secara real-time sehingga memudahkan pengguna dalam memperoleh informasi kondisi cuaca melalui jaringan internet.

Penelitian yang dilakukan oleh Kamal Abdurrohman, M. Irfan Farid, dan Bijanto pada tahun 2023 dengan judul “*Monitoring Data Untuk Sistem Stasiun Cuaca Berbasis IoT Menggunakan Firebase*” bertujuan untuk membangun sistem stasiun cuaca berbasis Internet of Things (IoT) yang mampu mencatat data cuaca secara real-time menggunakan Firebase sebagai media penyimpanan berbasis cloud. Sistem menggunakan NodeMCU ESP8266 dan sensor BME280 untuk memperoleh data suhu, kelembapan, dan tekanan udara. Data hasil pembacaan sensor dikirimkan ke Firebase Realtime Database menggunakan autentikasi pengguna, kemudian ditampilkan melalui antarmuka web dalam bentuk kartu, gauge, grafik, dan tabel sehingga dapat diakses oleh pengguna melalui internet.

Hasil penelitian menunjukkan bahwa sistem mampu melakukan pencatatan dan monitoring data cuaca secara real-time menggunakan Firebase dengan baik sehingga informasi cuaca dapat diakses secara cepat dan akurat.

Tabel 2. 1 Penelitian terdahulu

Peneliti & Tahun	Judul Penelitian	Metode	Kelemahan
Yudhistira (2021)	Perekam Data Intensitas Cahaya, Suhu, dan Kelembapan Udara Berbasis IoT (Internet of Things) Menggunakan ESP32 dan MQTT (Message Queuing Telemetry Transport)	Perancangan prototipe data logger menggunakan ESP32, BME680, sensor intensitas cahaya GA1A1S202WP, LCD 1602, dan MQTT melalui Wi-Fi. Pengukuran dilakukan berkala setiap 60 detik dan dikirim ke server.	Parameter masih terbatas pada suhu, kelembapan, dan intensitas cahaya; belum mencakup tekanan udara dan sensor hujan. Monitoring belum menggunakan aplikasi Android.
Fatahillah dan Roza (2022)	Perancangan Monitoring Cuaca Berbasis IoT Tenaga Matahari Menggunakan Panel Surya 20 WP	Menggunakan Arduino Uno sebagai pengendali, NodeMCU ESP8266 untuk Wi-Fi, DHT11, sensor hujan, panel surya 20 WP, dan Blynk sebagai media monitoring smartphone.	Masih menggunakan dua mikrokontroler dan Blynk. Parameter tekanan udara belum digunakan dan belum menggunakan ESP32 sebagai pengendali tunggal.
Dian Rahmadani,	Stasiun Cuaca Berbasis IoT Menggunakan NodeMCU	Menggunakan observasi dan penelitian pustaka. Sistem memakai NodeMCU, DHT11, BMP180, MySQL, PHP, HTML,	Monitoring masih berbasis website dan MySQL. Belum menggunakan aplikasi Android, Firebase Realtime Database,

Peneliti & Tahun	Judul Penelitian	Metode	Kelemahan
Andie, dan Fauzi Yusa Rahman (2020)		CSS, dan JavaScript. Data suhu, kelembapan, tekanan, titik embun, dan ketinggian ditampilkan melalui website.	serta sensor hujan sebagai bagian utama sistem.
Abdurrohman, Farid, dan Bijanto (2023)	Monitoring Data Untuk Sistem Stasiun Cuaca Berbasis IoT Menggunakan Firebase	Menggunakan metode Waterfall. NodeMCU ESP8266 dan BME280 mengirim data ke Firebase Realtime Database, kemudian data ditampilkan real-time melalui web dalam bentuk kartu, gauge, grafik, dan tabel.	Sudah menggunakan Firebase, tetapi masih menggunakan NodeMCU ESP8266 dan BME280 serta monitoring berbasis web. Belum menggunakan ESP32, BME680, sensor hujan, dan aplikasi Android.
Fahrel May Inzagi, Muhammad Amirudin, dan Ganjar Winasis	Sistem Monitoring Cuaca Berbasis IoT dengan Integrasi Mikrokontroler ESP32 dan Notifikasi Real-Time pada Penjemuran Kerupuk	Menggunakan metode Research and Development (R&D) dengan pendekatan kuantitatif. Sistem menggunakan ESP32, DHT11, LDR, sensor hujan, WhatsApp Chatbot, dan alarm speaker horn.	Fokus penelitian pada monitoring dan peringatan proses penjemuran kerupuk. Belum menggunakan tekanan udara serta belum menggunakan Firebase dan aplikasi Android sebagai media monitoring utama.

Peneliti & Tahun	Judul Penelitian	Metode	Kelemahan
(2025)			
Yudhistira (2021)	Perekam Data Intensitas Cahaya, Suhu, dan Kelembapan Udara Berbasis IoT (Internet of Things) Menggunakan ESP32 dan MQTT (Message Queuing Telemetry Transport)	Perancangan prototipe data logger menggunakan ESP32, BME680, sensor intensitas cahaya GA1A1S202WP, LCD 1602, dan MQTT melalui Wi-Fi. Pengukuran dilakukan berkala setiap 60 detik dan dikirim ke server.	Parameter masih terbatas pada suhu, kelembapan, dan intensitas cahaya; belum mencakup tekanan udara dan sensor hujan. Monitoring belum menggunakan aplikasi Android.
Fatahillah dan Roza (2022)	Perancangan Monitoring Cuaca Berbasis IoT Tenaga Matahari Menggunakan Panel Surya 20 WP	Menggunakan Arduino Uno sebagai pengendali, NodeMCU ESP8266 untuk Wi-Fi, DHT11, sensor hujan, panel surya 20 WP, dan Blynk sebagai media monitoring smartphone.	Masih menggunakan dua mikrokontroler dan Blynk. Parameter tekanan udara belum digunakan dan belum menggunakan ESP32 sebagai pengendali tunggal.
Dian Rahmadani, Andie, dan	Stasiun Cuaca Berbasis IoT Menggunakan NodeMCU	Menggunakan observasi dan penelitian pustaka. Sistem memakai NodeMCU, DHT11, BMP180, MySQL, PHP, HTML,	Monitoring masih berbasis website dan MySQL. Belum menggunakan aplikasi Android, Firebase Realtime Database,

Peneliti & Tahun	Judul Penelitian	Metode	Kelemahan
Fauzi Yusa Rahman (2020)		CSS, dan JavaScript. Data suhu, kelembapan, tekanan, titik embun, dan ketinggian ditampilkan melalui website.	serta sensor hujan sebagai bagian utama sistem.

2.2 Landasan Teori

2.2.1 *Internet of Things*

Internet of Things (IoT) merupakan konsep yang menghubungkan berbagai perangkat fisik ke jaringan internet sehingga perangkat tersebut dapat saling bertukar data tanpa memerlukan interaksi langsung antara manusia dengan manusia atau manusia dengan komputer. Setiap perangkat dalam sistem IoT umumnya dilengkapi dengan sensor, aktuator, serta modul komunikasi nirkabel yang memungkinkan pengumpulan data dari lingkungan sekitar untuk kemudian dikirim, disimpan, dan diolah pada suatu sistem terpusat, baik berupa server lokal maupun layanan cloud. Istilah *Internet of Things* pertama kali diperkenalkan pada akhir tahun 1990-an dan terus berkembang seiring dengan kemajuan teknologi mikrokontroler, sensor berbiaya rendah, serta ketersediaan jaringan internet nirkabel yang semakin luas. Konsep dasar IoT adalah menjadikan objek fisik sehari-hari sebagai bagian dari jaringan digital, sehingga kondisi objek tersebut dapat dipantau maupun dikendalikan dari jarak jauh secara real-time.

Arsitektur umum sistem IoT terdiri atas tiga lapisan utama. Lapisan pertama adalah lapisan perangkat (device layer atau perception layer), yang mencakup sensor dan aktuator yang bertugas mengumpulkan data fisik dari lingkungan, seperti suhu, kelembapan, cahaya, maupun gerakan. Lapisan kedua adalah lapisan jaringan (network layer), yang bertanggung jawab atas transmisi data dari perangkat menuju sistem pengolahan, baik menggunakan protokol Wi-Fi, seluler, LoRa, maupun Bluetooth. Lapisan ketiga adalah lapisan aplikasi (application layer), yang berfungsi mengolah dan menyajikan data kepada pengguna akhir melalui antarmuka seperti dashboard web maupun aplikasi mobile. Dalam praktiknya, komunikasi data pada sistem IoT umumnya memanfaatkan protokol ringan yang dirancang khusus untuk perangkat dengan sumber daya terbatas, seperti Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT), maupun Constrained Application Protocol (CoAP). Pemilihan protokol komunikasi disesuaikan dengan kebutuhan aplikasi, misalnya HTTP lebih sesuai untuk pengiriman data yang tidak terlalu sering namun membutuhkan struktur data yang jelas, sedangkan MQTT lebih sesuai untuk pengiriman data secara kontinu dengan

konsumsi bandwidth yang lebih rendah.. Ilustrasi IoT dapat dilihat pada gambar 2.1.



Gambar 2. 1 Internet Of Things

Seperti namanya, *Internet of Things* bergantung pada Internet sebagai konektivitas antara sensor atau perangkat yang akan saling berkomunikasi di cloud. Data dari sensor yang dikirim ke cloud akan diproses oleh software yang akan menentukan action selanjutnya. Action ini bisa berbentuk pengiriman alarm, penyesuaian jadwal, penutupan akses pada alat, atau lainnya.

2.2.2 ESP32

ESP32 adalah modul mikrokontroler *system-on-chip* (SoC) yang dikembangkan oleh Espressif Systems, dirancang khusus untuk aplikasi IoT dengan kebutuhan konektivitas nirkabel. ESP32 merupakan penerus dari seri ESP8266 dengan peningkatan pada aspek performa, jumlah antarmuka perifer, serta dukungan konektivitas ganda berupa Wi-Fi dan Bluetooth dalam satu chip. Secara arsitektur, ESP32 dilengkapi dengan prosesor dual-core Tensilica Xtensa LX6 yang dapat beroperasi hingga frekuensi 240 MHz, memori SRAM internal, serta mendukung penambahan memori eksternal melalui antarmuka SPI. Modul ini memiliki konektivitas Wi-Fi 802.11 b/g/n pada frekuensi 2,4 GHz serta Bluetooth Low Energy (BLE), sehingga mampu berfungsi sebagai perangkat akuisisi data sekaligus perangkat komunikasi ke jaringan internet dalam satu modul tanpa memerlukan modul tambahan.

ESP32 juga menyediakan berbagai antarmuka perifer, seperti Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver-Transmitter (UART), Analog-to-Digital Converter (ADC), serta Pulse

Width Modulation (PWM), yang memudahkan proses interkoneksi dengan berbagai jenis sensor maupun aktuator. Jumlah pin General Purpose Input Output (GPIO) yang relatif banyak turut memberikan fleksibilitas dalam merancang sistem dengan berbagai perangkat pendukung. Keunggulan ESP32 dibandingkan mikrokontroler konvensional terletak pada konektivitas Wi-Fi yang telah terintegrasi langsung pada chip, konsumsi daya yang relatif rendah dengan dukungan mode deep sleep, harga yang relatif terjangkau, serta dukungan terhadap berbagai bahasa pemrograman, termasuk C/C++ pada framework Arduino maupun MicroPython. Kombinasi karakteristik tersebut menjadikan ESP32 banyak dipilih sebagai unit pemroses utama pada berbagai aplikasi berbasis IoT, mulai dari sistem monitoring, sistem kendali jarak jauh, hingga perangkat wearable. Gambar 2.2 adalah bentuk fisik dari ESP32.



Gambar 2. 2 ESP32

2.2.3 Sensor BME680

Sensor BME680 merupakan sensor lingkungan digital yang dikembangkan oleh Bosch Sensortec untuk mengukur beberapa parameter lingkungan dalam satu perangkat, yaitu suhu, kelembapan relatif, tekanan udara, dan kadar gas. Sensor ini dirancang untuk aplikasi Internet of Things (IoT), perangkat bergerak, serta sistem pemantauan lingkungan yang membutuhkan ukuran komponen yang ringkas dengan konsumsi daya yang rendah. Integrasi beberapa parameter pengukuran dalam satu sensor menjadikan BME680 lebih efisien dibandingkan penggunaan beberapa sensor yang terpisah, sehingga mampu mengurangi kompleksitas rangkaian dan kebutuhan ruang pada perangkat elektronik. BME680 menggunakan beberapa elemen sensor yang bekerja secara bersamaan. Pengukuran suhu dilakukan menggunakan sensor temperatur digital, sedangkan kelembapan diukur menggunakan elemen sensor kapasitif. Tekanan udara diperoleh melalui sensor

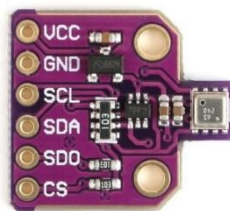
piezoresistif berakurasi tinggi, sementara kualitas udara diukur menggunakan sensor gas berbasis *Metal Oxide* (MOX). Sensor gas tersebut mampu mendeteksi perubahan senyawa organik volatil (*Volatile Organic Compounds/VOC*) dan berbagai gas lainnya melalui perubahan nilai resistansi. Nilai keluaran sensor gas tidak menunjukkan jenis gas tertentu, melainkan memberikan indikasi perubahan kualitas udara berdasarkan konsentrasi gas yang terdeteksi.

Komunikasi antara sensor BME680 dan mikrokontroler dapat dilakukan menggunakan protokol I2C maupun SPI. Dukungan terhadap kedua protokol tersebut memudahkan proses integrasi dengan berbagai platform mikrokontroler, termasuk ESP32. Selain itu, sensor ini memiliki tingkat akurasi yang baik, waktu respons yang cepat, serta konsumsi daya yang rendah sehingga banyak digunakan pada sistem pemantauan kualitas lingkungan, stasiun cuaca digital, perangkat rumah pintar, dan aplikasi IoT lainnya berikut adalah spesifikasi dari sensor BME680.

Tabel 2. 2 Spesifikasi BME680

No.	Parameter	Spesifikasi
1	Jenis Sensor	Sensor lingkungan terintegrasi (suhu, kelembapan, tekanan udara, dan gas)
2	Pengukuran Suhu	Jangkauan -40 °C hingga 85 °C
3	Akurasi Suhu	±0,5 °C
4	Pengukuran Kelembapan	Jangkauan 0% hingga 100% RH
5	Akurasi Kelembapan	±3% RH
6	Pengukuran Tekanan Udara	Jangkauan 300 hingga 1100 hPa
7	Akurasi Tekanan Udara	±1 hPa
8	Sensor Gas	Metal Oxide (MOX) Gas Sensor
9	Waktu Respons Sensor Gas	33%–63% (T33–T63)

10	Antarmuka Komunikasi	I ² C (hingga 3,4 MHz) atau SPI (hingga 10 MHz)
11	Tegangan Operasi	1,71 V hingga 3,6 V
12	Konsumsi Daya	0,09 mA (mode tidur/idle), 3,1 mA (mode pengukuran)
13	Dimensi	3,0 mm × 3,0 mm × 0,93 mm (<i>ukuran chip BME680</i>)



Gambar 2. 3 BME680

2.2.4 BH1750

BH1750 merupakan sensor digital yang digunakan untuk mengukur intensitas cahaya dengan satuan lux. Sensor ini dikembangkan oleh ROHM Semiconductor dan banyak digunakan pada aplikasi pengendalian pencahayaan otomatis, sistem rumah pintar, perangkat elektronik, serta sistem monitoring lingkungan. Berbeda dengan sensor cahaya analog seperti *Light Dependent Resistor* (LDR), BH1750 menghasilkan keluaran digital sehingga tidak memerlukan proses konversi *Analog-to-Digital Converter* (ADC) pada mikrokontroler. Hal tersebut menghasilkan pembacaan yang lebih stabil dan memiliki tingkat akurasi yang lebih baik.

Sensor BH1750 bekerja dengan mendeteksi jumlah cahaya yang diterima oleh elemen fotodiode, kemudian mengubahnya menjadi nilai intensitas cahaya dalam satuan lux melalui rangkaian konverter digital yang terdapat di dalam sensor. Nilai lux merupakan satuan internasional yang digunakan untuk menyatakan tingkat pencahayaan pada suatu permukaan. Sensor ini memiliki rentang pengukuran yang luas sehingga mampu digunakan pada kondisi pencahayaan rendah maupun tinggi. Komunikasi antara BH1750 dan mikrokontroler menggunakan protokol Inter-Integrated Circuit (I²C). Penggunaan antarmuka I²C

memungkinkan proses pertukaran data dilakukan hanya melalui dua jalur komunikasi, yaitu Serial Data (SDA) dan Serial Clock (SCL). Dengan demikian, jumlah pin yang digunakan menjadi lebih sedikit sehingga mempermudah proses perancangan perangkat keras dan integrasi dengan sensor lain yang juga menggunakan protokol I2C.



Gambar 2. 4 BH1750

2.2.5 Sensor Hujan

Sensor hujan merupakan sensor yang digunakan untuk mendeteksi keberadaan air pada permukaan sensor sehingga dapat diketahui kondisi hujan di suatu lokasi. Sensor ini banyak diterapkan pada sistem monitoring cuaca, sistem irigasi otomatis, rumah pintar (smart home), dan berbagai aplikasi Internet of Things. Prinsip kerja sensor hujan didasarkan pada perubahan hambatan atau konduktivitas listrik ketika permukaan sensor terkena air. Semakin banyak air yang mengenai permukaan sensor, semakin kecil nilai hambatan yang dihasilkan sehingga menyebabkan perubahan tegangan keluaran yang dapat dibaca oleh mikrokontroler.

Secara umum, sensor hujan terdiri atas papan sensor (rain detection plate) dan modul pengolah sinyal. Papan sensor berfungsi mendeteksi keberadaan air melalui jalur konduktor yang akan saling terhubung ketika terkena tetesan air, sedangkan modul pengolah sinyal mengubah perubahan tersebut menjadi sinyal analog maupun digital. Keluaran analog menghasilkan nilai yang berubah secara kontinu sesuai banyaknya air yang terdeteksi, sedangkan keluaran digital hanya menunjukkan dua kondisi, yaitu terdapat hujan atau tidak terdapat hujan berdasarkan nilai ambang (threshold) yang telah ditentukan. Kemampuan

mendeteksi keberadaan air dengan cepat menjadikan sensor hujan sebagai salah satu komponen penting pada sistem pemantauan lingkungan. Sensor ini banyak digunakan pada stasiun cuaca digital, sistem peringatan dini, sistem irigasi otomatis, serta berbagai aplikasi berbasis Internet of Things yang memerlukan informasi mengenai kondisi hujan secara real-time.



Gambar 2. 5 Sensor Hujan

2.2.6 Android

Android adalah sistem operasi berbasis kernel Linux yang dikembangkan untuk perangkat mobile, dan merupakan platform dengan pangsa pengguna terbesar di dunia untuk perangkat smartphone. *Android* bersifat *open-source* dan dikembangkan oleh Google, sehingga banyak digunakan oleh berbagai produsen perangkat mobile di seluruh dunia dengan berbagai variasi antarmuka yang disesuaikan oleh masing-masing produsen. Secara arsitektur, sistem operasi *Android* tersusun atas beberapa lapisan, mulai dari kernel Linux yang menangani manajemen proses, memori, dan driver perangkat keras; lapisan *Hardware Abstraction Layer* (HAL) yang menjembatani perangkat keras dengan sistem di atasnya; lapisan *Android Runtime* (ART) yang menjalankan aplikasi; hingga lapisan *Application Framework* yang menyediakan berbagai komponen dan layanan bagi aplikasi, seperti manajemen activity, content provider, dan notification manager.

Sistem operasi ini mendukung berbagai jenis aplikasi, mulai dari aplikasi hiburan, produktivitas, hingga aplikasi yang berfungsi sebagai antarmuka pemantauan (monitoring interface) bagi sistem berbasis IoT. Pengembangan aplikasi *Android* umumnya dilakukan menggunakan bahasa pemrograman Java atau Kotlin melalui *Android Studio* sebagai integrated development environment

(IDE) resmi, yang menyediakan berbagai perangkat bantu seperti emulator, debugger, dan pustaka Android SDK. Selain pendekatan berbasis kode tekstual tersebut, terdapat pula platform pengembangan visual berbasis blok (block-based programming) yang memungkinkan pembuatan aplikasi tanpa memerlukan penulisan kode secara langsung, sehingga lebih mudah dipelajari oleh pengembang pemula maupun kalangan pendidikan yang belum memiliki latar belakang pemrograman tekstual. Gambar 2.8 adalah logo android



Gambar 2. 6 Android

2.2.7 MIT App Inventor

MIT App Inventor adalah platform pengembangan aplikasi Android berbasis blok visual yang dikembangkan oleh Massachusetts Institute of Technology (MIT). Platform ini pada awalnya dikembangkan oleh Google Labs pada tahun 2010 dengan nama Google App Inventor, sebelum kemudian pengembangan dan pemeliharaannya diserahkan kepada MIT Center for Mobile Learning pada tahun 2012 dan berganti nama menjadi MIT App Inventor. Platform ini memungkinkan pengguna merancang antarmuka aplikasi (user interface) melalui mode Designer, yaitu tampilan drag-and-drop untuk menempatkan komponen visual seperti label, tombol, kotak teks, dan tampilan grafik pada layar aplikasi. Selanjutnya, logika program disusun menggunakan blok-blok visual pada mode Blocks, di mana setiap blok merepresentasikan suatu instruksi atau operasi, seperti pengambilan data, percabangan kondisi, maupun pengulangan, sehingga proses pengembangan aplikasi dapat dilakukan tanpa penulisan kode program secara tekstual.

Pendekatan pemrograman berbasis blok pada App Inventor mengadopsi konsep visual programming yang serupa dengan Scratch, di mana setiap blok

memiliki bentuk yang hanya dapat disatukan dengan blok lain yang kompatibel secara tipe data, sehingga meminimalkan kesalahan sintaks yang umum terjadi pada pemrograman tekstual. App Inventor menyediakan berbagai komponen bawaan (built-in components) yang dikelompokkan berdasarkan kategori, seperti komponen antarmuka pengguna (*User Interface*), komponen media, komponen sensor perangkat, serta komponen penyimpanan data (Storage), termasuk komponen untuk terhubung dengan layanan basis data cloud seperti Firebase (Firestore component). Komponen tersebut memungkinkan aplikasi membaca dan menulis data pada basis data tersebut tanpa memerlukan penulisan kode konfigurasi tambahan. Karakteristik ini menjadikan App Inventor banyak digunakan dalam pengembangan aplikasi monitoring maupun kendali perangkat berbasis IoT dengan waktu pengembangan yang relatif singkat, gambar 2.9 adalah logo MIT APP INVENTOR



Gambar 2. 7 MIT APP INVENTOR

2.2.8 *Firebase*



Gambar 2. 8 Firebase

Firestore Realtime Database adalah layanan basis data cloud berbasis NoSQL yang disediakan oleh Google sebagai bagian dari platform Firebase, dengan struktur data berbentuk JSON (JavaScript Object Notation) tree. Berbeda dengan basis data relasional yang menyimpan data dalam bentuk tabel, Firestore Realtime Database menyimpan seluruh data dalam satu struktur pohon (tree) besar yang dapat diakses melalui path tertentu, mirip dengan struktur direktori pada sistem berkas. Karakteristik utama layanan ini adalah kemampuan sinkronisasi data secara real-time ke seluruh perangkat yang terhubung, sehingga setiap perubahan data yang terjadi pada basis data akan langsung diperbarui pada semua klien yang sedang membaca data tersebut tanpa perlu melakukan permintaan (*request*) secara berulang atau melakukan proses polling. Mekanisme ini dicapai melalui koneksi WebSocket yang dipertahankan antara klien dan server Firestore.

Firestore Realtime Database menyediakan Representational State Transfer (REST) API yang memungkinkan perangkat mikrokontroler mengirim data menggunakan permintaan HTTP standar, sehingga tidak memerlukan pustaka klien khusus yang kompleks pada sisi perangkat. Selain Realtime Database, Firestore juga menyediakan Firestore, sebuah basis data NoSQL dengan model data berbasis koleksi dan dokumen yang menawarkan kueri lebih fleksibel, meskipun Realtime Database tetap banyak dipilih untuk aplikasi yang mengutamakan kesederhanaan struktur data dan latensi sinkronisasi yang rendah. Dari sisi keamanan, Firestore Realtime Database menyediakan mekanisme aturan keamanan (*security rules*) yang dapat dikonfigurasi untuk mengatur hak akses baca dan tulis pada setiap path data, sehingga pengembang dapat membatasi akses data hanya kepada perangkat atau pengguna yang telah terautentikasi. Karakteristik sinkronisasi real-time serta kemudahan integrasi ini menjadikan Firestore Realtime Database banyak digunakan sebagai penghubung komunikasi antara perangkat IoT dengan aplikasi klien, seperti aplikasi mobile, tanpa memerlukan server perantara tambahan.