

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Bagian ini menguraikan beberapa kajian dari penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan sistem yang akan dikembangkan. Tinjauan ini digunakan untuk mengkaji sejarah penyelesaian permasalahan, membantu pemilihan prosedur penelitian, serta mengidentifikasi posisi kebaruan dari sistem yang diusulkan.

Penelitian pertama dilakukan oleh (Hanani & Haryadi) yang merancang sistem hunian pintar (*smart home*) berbasis *Internet of Things (IoT)* dengan memanfaatkan teknologi interaksi suara (*voice command*) pada layanan *Google Assistant*. Penelitian tersebut dirancang untuk memecahkan masalah aksesibilitas bagi pengguna disabilitas atau lanjut usia agar dapat menghidupkan dan mematikan peralatan rumah tangga tanpa harus menjangkau sakelar fisik. Metode penelitian yang digunakan adalah menghubungkan *Google Assistant* dengan *Webhooks* melalui layanan *IFTTT (If This Than That)*, yang kemudian meneruskan instruksi berupa teks ke *platform* komputasi awan (*cloud*) untuk menggerakkan *mikrokontroler ESP32* dan *aktuator relay*.

Penelitian kedua oleh (Nugraha Dimiyati et al. 2025) berfokus pada pengembangan sistem pemantauan kualitas udara berbasis *IoT* yang divisualisasikan melalui aplikasi *Android*. Kajian ini memberikan solusi atas

kurangnya informasi terpadu yang cepat dan akurat mengenai deteksi polusi udara di lingkungan kawasan industri, yang mana sangat berisiko memicu gangguan pernapasan. Prosedur penelitian yang diimplementasikan adalah menggunakan *mikrokontroler ESP32* yang mengambil data fisik dari *sensor MQ-135* untuk mendeteksi gas berbahaya dan *sensor PMS5003* untuk mendeteksi partikel debu. Data pemantauan tersebut kemudian dikirimkan ke pangkalan data (*database*) melalui antarmuka *REST API* untuk ditampilkan secara aktual (*real-time*) kepada pengguna.

Penelitian ketiga dikemukakan oleh (Fajar A'la Udin et al. 2026) yang membangun sistem pemantauan kelembapan dan suhu udara berbasis *IoT* secara spesifik untuk skala ruang tertutup. Penelitian tersebut menjawab permasalahan terkait fluktuasi kondisi lingkungan mikro yang rentan menurunkan performa perangkat keras operasional serta memicu risiko kesehatan. Sistem tersebut dirancang menggunakan *mikrokontroler Arduino Uno* dan *sensor DHT11*, di mana data pembacaannya dikirimkan ke komputer dan diolah menggunakan bahasa pemrograman *Python-Flask* agar dapat dipantau terus-menerus melalui *dashboard* berbasis web.

Penelitian keempat oleh (Supiyandi et al. 2023) menerapkan sistem *smart home* berbasis *IoT* untuk mengendalikan peralatan elektronik secara terpusat. Penelitian ini menyelesaikan masalah pemborosan energi dan keterbatasan mobilitas manusia dengan mengimplementasikan metode kendali jarak jauh melalui ponsel pintar (*smartphone*) dan otomatisasi berbasis pewaktu (*timer*) menggunakan *mikrokontroler*.

Berdasarkan tinjauan terhadap prosedur dan cara menjawab permasalahan pada penelitian-penelitian di atas, dapat disimpulkan bahwa sebagian besar pengembangan sistem *IoT* masih berjalan secara terpisah pada satu aspek fungsionalitas tertentu. Sistem yang ada umumnya hanya berfokus pada fungsi otomatisasi perangkat keras, atau sebatas memantau parameter kualitas lingkungan tanpa adanya tindakan pengontrolan perangkat secara terpadu. Sebagai bentuk pengembangan, penelitian yang akan diusulkan ini merancang metode yang menggabungkan berbagai aspek tersebut ke dalam satu ekosistem yang utuh.

Perbedaan utama penelitian ini terletak pada integrasi beberapa jalur kendali dan pemantauan dalam satu prototipe. Sistem menggabungkan kendali manual melalui *Blynk*, kendali terjadwal, kendali otomatis berbasis sensor, dan kendali suara melalui *Google Assistant* atau *Google Home*. Berbeda dari penelitian yang menggunakan IFTTT, penelitian ini menggunakan notifikasi *Android* dan *MacroDroid* sebagai middleware. *MacroDroid* mencocokkan kata kunci perintah, kemudian mengirimkan *HTTP GET* ke *Blynk External API* untuk memperbarui virtual pin. Sistem juga menggabungkan *LCD* lokal, *dashboard Blynk*, notifikasi, dan data logging *Google Sheets*.

Tabel 2. 1 Perbandingan Penelitian Terkait

Penelitian	Fokus/teknologi	Keterbatasan	Pengembangan dalam penelitian ini
Hanani dan Hariyadi	<i>Google Assistant</i> dan <i>IoT</i> untuk kendali perangkat	Berfokus pada kendali suara	<i>MacroDroid</i> , <i>Blynk</i> , pemantauan sensor, dan data logging
Nugraha Dimiyati dkk.	Pemantauan kualitas udara berbasis <i>ESP32</i>	Tidak mengintegrasikan seluruh mode kendali	Pemantauan dan kendali perangkat dalam satu sistem
Fajar A'la Udin dan Fuatdi	Pemantauan suhu dan kelembapan	Parameter dan fungsi kendali terbatas	<i>DHT22</i> , <i>MQ-135</i> , <i>BME280</i> , <i>Blynk</i> , dan aktuator
Supiyandi dkk.	Smart home dan kendali jarak jauh	Tidak berfokus pada evaluasi kondisi lingkungan	Kendali manual, suara, jadwal, otomatis, dan pencatatan data

2.2 Landasan Teori

2.2.1 Internet of Things (IoT)

Internet of Things (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Konsep dasar dan prinsip kerja perangkat *IoT* adalah benda atau perangkat di dunia nyata diberikan identitas unik dan dilengkapi dengan modul nirkabel (*wireless*), sehingga dapat dikendalikan dan direpresentasikan dalam bentuk

data pada sebuah sistem komputer melalui *platform* komputasi awan (*cloud platform*) [10].

2.2.2 *Smart Room* dan *Smart Home*

Ruangan pintar (*smart room*) yang merupakan turunan dari konsep rumah pintar (*smart home*) adalah sistem yang dibangun dengan bantuan komputasi untuk mengintegrasikan dan mengendalikan perangkat perangkat di dalam ruangan secara otomatis dan efisien. Perkembangan teknologi ini bertujuan untuk membantu manusia dalam mengelola aktivitas sehari-hari, memberikan kemudahan kendali operasional, serta meningkatkan kenyamanan dan keamanan ruang [11].

2.2.3 Kualitas Udara dalam Ruangan

Kualitas udara dalam ruangan menjadi aspek yang sangat krusial dalam menjaga kesehatan dan tingkat produktivitas manusia. Lingkungan ruangan yang tampak bersih secara visual belum tentu memiliki udara yang sehat, karena sangat mungkin mengandung berbagai polutan tertutup. Polutan tersebut meliputi peningkatan kadar *karbon dioksida* (CO_2), *karbon monoksida* (CO), partikel debu, dan berbagai senyawa organik yang mudah menguap. Apabila kualitas udara yang buruk ini tidak terdeteksi, hal tersebut dapat memicu berbagai gangguan kesehatan seperti iritasi pada saluran pernapasan hingga penyakit kronis jika terpapar dalam jangka waktu yang panjang [12].

2.2.4 Mikro Kontroler ESP32

ESP32 merupakan sebuah modul *mikrokontroler* yang sangat ideal difungsikan sebagai peladen (*server*) mandiri pada sistem berbasis teknologi *IoT*. Perangkat ini memiliki keunggulan karena telah dilengkapi dengan modul komunikasi nirkabel bawaan, sehingga mampu memproses instruksi kendali sekaligus memantau kondisi lingkungan, lalu mengirimkan data pemantauan tersebut secara aktual (*real-time*) ke jaringan internet [13].

2.2.5 Asisten Pintar dan *Voice Command*

Google Assistant atau *Google Home* digunakan sebagai antarmuka perintah suara [14]. Pada penelitian ini, layanan tersebut tidak berkomunikasi langsung dengan *ESP32*. Perintah suara menghasilkan notifikasi pada perangkat *Android*. *MacroDroid* membaca notifikasi berdasarkan aplikasi sumber dan kata kunci yang ditentukan. Ketika kata kunci sesuai, *MacroDroid* menjalankan tindakan *HTTP GET* menuju *Blynk External API*. *Blynk* memperbarui virtual pin, kemudian *ESP32* menerima perubahan *datastream* dan mengubah status *relay*.

2.2.6 MacroDroid dan Notifikasi Android

MacroDroid merupakan aplikasi otomasi *Android* yang bekerja berdasarkan pemicu, tindakan, dan batasan [15]. Pemicu yang digunakan adalah notifikasi dari *Google Home* atau *Google Assistant* yang memuat kata kunci *LampuON*, *LampuOFF*,

KipasON, atau KipasOFF. Tindakannya berupa *HTTP GET* menuju endpoint *Blynk*. Keberhasilan mekanisme tersebut bergantung pada izin notifikasi, koneksi internet, proses latar belakang, dan konsistensi teks notifikasi.

2.2.7 *Blynk Cloud dan External API*

Blynk Cloud digunakan untuk menyimpan *datastream* dan menghubungkan *dashboard* dengan *ESP32* [16]. Layanan eksternal seperti *MacroDroid* dapat memperbarui virtual pin melalui *Blynk External API*. Permintaan memuat alamat server, token autentikasi, nomor virtual pin, dan nilai yang dikirim. Token merupakan kredensial sistem sehingga tidak ditampilkan secara lengkap di dalam laporan

2.2.8 *Komponen Sensor Lingkungan*

Pemantauan kualitas udara memerlukan beberapa perangkat *sensor* spesifik yang dikalibrasi untuk membaca kondisi fisik lingkungan, di antaranya:

1. *Sensor MQ-135*

Merupakan *sensor* gas yang difungsikan untuk memonitor tingkat kualitas udara [17]. *Sensor* ini memiliki sensitivitas untuk mendeteksi berbagai jenis gas berbahaya yang mencemari udara, seperti amonia (*NH₃*), *nitrogen oksida* (*NO_x*), *alkohol*, *benzena*, serta *karbon dioksida* (*CO₂*) .

2. *Sensor Suhu dan Kelembapan*

Kelembapan dan suhu udara merupakan parameter lingkungan yang memiliki pengaruh langsung terhadap kenyamanan fisik serta kestabilan perangkat elektronik di dalam ruang tertutup. Pendeteksian terhadap *fluktuasi* perubahan *termal* dan tingkat air di udara umumnya menggunakan komponen *sensor* seri *DHT*, seperti *DHT11* atau *DHT22* [18].

3. *Sensor Tekanan Udara (BME280)*

Sensor BME280 digunakan untuk mengukur tekanan atmosfer dan mendukung analisis kondisi lingkungan di dalam ruangan. Data tekanan udara ini dapat membantu sistem dalam menampilkan informasi kondisi cuaca sederhana dan mendeteksi perubahan lingkungan secara *real-time* [19].

2.2.9 Metode Pengembangan dan Pengujian Sistem

1. *Waterfall*

Metode *Waterfall* adalah model pengembangan sistem secara *linear* yang dimulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Metode ini sesuai untuk pengembangan sistem *smart room* berbasis *IoT* yang memerlukan tahapan terstruktur [20].

2. *Blackbox Testing*

Blackbox Testing adalah metode pengujian sistem yang fokus pada *input-output* tanpa memperhatikan kode internal. Metode ini

digunakan untuk memastikan seluruh fungsi *sensor*, *aktuator*, dan aplikasi *IoT* bekerja sesuai spesifikasi [21].