

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi *Internet of Things* (IoT) berdampak besar pada berbagai sektor kehidupan, termasuk di lingkungan akademik dan tempat tinggal. IoT menghubungkan perangkat untuk bertukar data secara otomatis tanpa campur tangan manusia. Sistem ini mendukung pengendalian dan pemantauan perangkat melalui jaringan dengan model komunikasi *client-server*. Dalam model ini, perangkat IoT bertindak sebagai *client* dan sistem *cloud* sebagai *server* (Erlangga Prasetya et al., 2022). Penggunaan sistem *cloud* dalam pengembangan IoT bertujuan untuk menambah fleksibilitas, efisiensi, dan aksesibilitas. Namun, muncul tantangan terkait keamanan data, terutama ketika data disimpan di *cloud* pihak ketiga seperti Blynk yang berisiko bocor atau disalahgunakan.

Berdasarkan pengamatan peneliti, layanan *cloud* pihak ketiga seperti Blynk menyimpan data pengguna di *server* mereka. Hal ini menimbulkan risiko kebocoran data jika terjadi serangan pada *server* Blynk. Oleh karena itu, peneliti memutuskan membangun sistem *cloud* sendiri agar data tetap aman dan terkendali. Oleh karena itu, diperlukan sistem autentikasi yang ketat. Pengguna *dashboard* wajib memiliki token JWT (*JSON Web Token*) untuk mengakses API. Setiap perangkat IoT juga diberikan token unik saat registrasi. Token ini harus disertakan dalam setiap pengiriman data melalui protokol CoAP. *Server* akan memeriksa keabsahan token sebelum menyimpan data. Sistem kontrol akses juga diterapkan untuk membatasi pengguna yang berwenang dalam mengakses data dan mengoperasikan perangkat.

Penelitian ini merancang dan membangun sistem *cloud* IoT mandiri berbasis PaaS yang diuji pada lingkungan lokal sebagai bukti konsep (*proof of concept*) sebelum direkomendasikan untuk implementasi skala lebih luas. Sistem ini dirancang untuk melepaskan ketergantungan pada layanan *cloud* pihak ketiga sekaligus memberi kendali penuh atas data dan keamanan. Privasi dan keamanan data tetap terjaga selama proses pengiriman. Erwin & Pratama (2023) menguji sistem fuzzy dalam penelitian berjudul "Rancang bangun sistem monitoring suhu dan kelembaban ruang *server* berbasis IoT menggunakan Arduino pada PT Bintaro Serpong Damai". Penelitian tersebut masih bergantung pada aplikasi Blynk. Aplikasi itu hanya mengirim notifikasi suhu dan terbatas dalam fleksibilitas

pengelolaan data. Untuk mengatasi ketergantungan pada pihak ketiga, penelitian ini mengembangkan sistem *cloud* dengan metode PaaS dan protokol CoAP. Tujuan penelitian ini adalah membuat komunikasi *client-server* menjadi lebih optimal. Sistem mengelola dan memantau perangkat IoT secara efisien melalui implementasi metode PaaS dengan CoAP. Sistem ini memanfaatkan sumber daya jaringan internal yang terkontrol.

Sistem *cloud* internal berjalan di jaringan lokal tanpa bergantung pada pihak ketiga. PaaS memudahkan pemantauan data, deteksi gangguan, serta efisiensi daya. CoAP digunakan sebagai protokol komunikasi karena mendukung metode GET, PUT, POST, dan DELETE dalam mengelola sumber daya di jaringan (Shelby et al., 2014). Namun, aspek keamanan CoAP masih perlu ditingkatkan karena keamanan data tetap menjadi tantangan dalam komunikasi CoAP.

Pengguna menyimpan data dalam kapasitas besar dan mengaksesnya dengan mudah melalui sistem *cloud*. Pengguna tidak perlu bergantung pada media penyimpanan fisik. Salah satu protokol umum dalam komunikasi IoT adalah MQTT. Protokol ini mengadopsi pola *publish/subscribe*. Keunggulan pola ini adalah sifat *decouple*. Pengirim dan penerima pesan tidak berkomunikasi secara langsung. Komunikasi berlangsung melalui broker MQTT (Permatasari & Dhika, 2018). Dengan pendekatan PaaS, rancangan sistem *cloud* IoT ini menyediakan infrastruktur yang skalabel, fleksibel, dan terkontrol. Rancangan ini berfokus pada pengembangan sistem *cloud* IoT tanpa bergantung pada layanan pihak ketiga. Layanan PaaS yang dibangun dapat dikontrol sepenuhnya.

Sistem ini menerapkan dua lapis autentikasi. Lapis pertama adalah token JWT (*JSON Web Token*) untuk pengguna *dashboard*. Lapis kedua adalah token unik untuk setiap perangkat IoT. Kedua token ini diverifikasi oleh *server* sebelum data diproses. Rancangan ini meningkatkan efisiensi pemanfaatan sumber daya dan memperkuat aspek keamanan dalam komunikasi IoT.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka perumusan masalah dalam pembuatan proyek akhir ini adalah

- 1) Bagaimana merancang dan membangun sistem *cloud* IoT berbasis protokol CoAP pada infrastruktur jaringan lokal yang mampu menerima, memproses, dan menyimpan data dari perangkat sensor secara real-time?
- 2) Bagaimana mengimplementasikan *dashboard* web yang menyediakan fitur manajemen perangkat, visualisasi data sensor, notifikasi, otomatisasi, penjadwalan, backup restore, dan pembangkitan kode client universal untuk berbagai jenis sensor dan mikrokontroler?
- 3) Bagaimana kinerja sistem *cloud* IoT yang dibangun dalam aspek latensi, throughput, dan keandalan pengiriman data?

1.3 Batasan Masalah

Agar pembahasan materi dalam penelitian ini optimal dan terarah dalam memenuhi hasil yang diharapkan, maka ditetapkan beberapa batasan masalah sebagai berikut:

- 1) Protokol komunikasi yang digunakan antara perangkat IoT dan *server cloud* adalah Constrained Application Protocol (CoAP).
- 2) "Sistem *cloud* diuji secara lokal (localhost) pada laptop peneliti dalam tahap pengembangan (development phase) sebagai bukti konsep (proof of concept), dengan hasil pengujian digunakan sebagai dasar rekomendasi untuk implementasi pada skala yang lebih luas.
- 3) Perangkat IoT yang didukung dalam penelitian ini adalah mikrokontroler ESP8266 dan ESP32.
- 4) Jenis sensor yang diuji dalam penelitian ini adalah soil moisture dalam dua varian yaitu analog dan digital, serta motion untuk mendeteksi gerakan. Secara teknis, sistem telah mendefinisikan 10 jenis sensor dalam basis data, namun pengujian baru dilakukan pada ketiga jenis sensor tersebut, yaitu dua varian soil moisture dan satu motion.
- 5) Selain itu, sistem menerapkan otorisasi berbasis peran dengan dua tingkat hak akses. Admin memiliki akses penuh terhadap seluruh fitur sistem, termasuk manajemen pengguna seperti menambah, mengedit, dan menghapus akun, serta backup dan restore *database*. Sedangkan user hanya dapat mengakses fitur

operasional seperti pemantauan data sensor, manajemen perangkat, notifikasi, dan pengaturan pribadi, tanpa akses ke manajemen pengguna.

- 6) Keamanan data dijamin melalui infrastruktur jaringan lokal (on-premise) dengan mekanisme autentikasi berbasis token pada setiap pengiriman data dari perangkat ke *server*.
- 7) Fungsionalitas sistem mencakup:
 - a. Penerimaan dan penyimpanan data sensor melalui protokol CoAP
 - b. Pemantauan data secara real-time melalui *dashboard* web
 - c. Manajemen perangkat (registrasi, penghapusan, dan detail perangkat)
 - d. Notifikasi berbasis ambang batas nilai sensor
 - e. Pembangkitan kode client yang dapat diunduh pengguna
 - f. Fitur otomatisasi terbatas pada aturan sederhana (automation rules) berdasarkan kondisi nilai sensor yang telah ditentukan.
- 8) Sistem menerapkan otorisasi berbasis peran (role-based access control) dengan dua tingkat hak akses:
 - a. Admin: memiliki akses penuh terhadap seluruh fitur sistem, termasuk manajemen pengguna.
 - b. User: hanya dapat mengakses fitur operasional tanpa manajemen pengguna.

1.4 Tujuan Dan Manfaat

Tujuan dari penelitian ini adalah sebagai berikut:

- 1) Merancang dan membangun sistem *cloud* IoT berbasis protokol CoAP yang mampu menerima, memproses, dan menyimpan data sensor secara real-time.
- 2) Mengimplementasikan *dashboard* web dengan fitur manajemen perangkat, visualisasi data sensor, notifikasi berbasis ambang batas, otomatisasi, penjadwalan, backup restore, dan pembangkitan kode client universal.
- 3) Mengukur kinerja sistem *cloud* IoT yang dibangun dalam aspek latensi, throughput, dan keandalan pengiriman data.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

- 1) Data yang dihasilkan oleh perangkat IoT dapat dikelola secara mandiri dalam lingkungan pengembangan (development environment), sehingga mengurangi risiko kebocoran data yang sering terjadi pada layanan *cloud* pihak ketiga.
- 2) Pengguna dapat memantau dan mengelola perangkat IoT melalui *dashboard*

web yang dapat diakses dari perangkat yang terhubung dalam satu jaringan.

- 3) Penelitian ini menghasilkan sistem yang tidak bergantung pada layanan *cloud* komersial, sehingga biaya operasional jangka panjang dapat ditekan.
- 4) Kode client yang dihasilkan bersifat universal dan dilengkapi panduan, sehingga memudahkan mahasiswa dan peneliti lain dalam mengembangkan sistem serupa.
- 5) Sistem ini dapat menjadi referensi bagi pengembangan infrastruktur IoT di lingkungan akademik yang mengutamakan keamanan data dan kemandirian sistem.

1.6 Metode Penelitian

Metode penelitian yang digunakan dalam pembuatan proyek akhir ini adalah sebagai berikut:

1.6.1 Studi Literatur

Tahap awal penelitian dilakukan dengan mengumpulkan dan mempelajari referensi berupa buku, artikel ilmiah, jurnal, dan dokumentasi teknis yang berkaitan dengan:

- 1) Konsep Internet of Things dan arsitektur sistem *cloud* IoT
- 2) Protokol CoAP (Constrained Application Protocol) beserta format pesan dan mekanisme kerjanya
- 3) Pemrograman mikrokontroler ESP8266 dan ESP32 menggunakan Platform IO
- 4) Pengembangan aplikasi web dengan Node.js, Express.js, dan *WebSocket*
- 5) Perancangan *database* menggunakan SQLite

1.6.2 Konsultasi dan Diskusi

Kegiatan konsultasi dan diskusi dilakukan dengan dosen pembimbing untuk mendapatkan arahan mengenai:

- 6) Perumusan masalah yang sesuai dengan kemampuan peneliti
- 7) Metodologi pengembangan sistem yang tepat
- 8) Evaluasi hasil pengujian dan analisis kinerja sistem
- 9) Penyusunan laporan proyek akhir yang sistematis
- 10) Perancangan Sistem

1.6.3 Perancangan Sistem

Tahap perancangan sistem meliputi:

- 1) Perancangan Arsitektur Sistem terdiri dari:

- a. Merancang diagram blok yang menggambarkan alur data dari perangkat IoT ke *server*
- b. Menentukan komponen-komponen sistem beserta interaksinya
- 2) Perancangan *Database*
Merancang struktur tabel untuk menyimpan data perangkat, data sensor, log aktivitas, dan konfigurasi sistem
- 3) Perancangan Antarmuka Pengguna
 - a. Merancang tata letak *dashboard* yang menampilkan statistik, daftar perangkat, dan visualisasi data
 - b. Merancang halaman manajemen perangkat dan formulir registrasi
- 4) Perancangan Komunikasi CoAP
 - a. Merancang format pesan CoAP yang akan digunakan untuk komunikasi antara client dan *server*
 - b. Menentukan mekanisme autentikasi menggunakan token

1.6.4 Implementasi Sistem

Tahap implementasi dilakukan dengan menulis kode program untuk:

- 1) Backend Server (Node.js)
 - a. Implementasi REST API untuk manajemen perangkat dan data sensor
 - b. Implementasi listener *User Datagram Protocol* (UDP) untuk menerima pesan CoAP dari perangkat IoT
 - c. Implementasi *WebSocket* untuk data real-time ke *dashboard*
- 2) *Database* (SQLite)
Implementasi query untuk menyimpan dan mengambil data perangkat serta histori sensor
- 3) Frontend *Dashboard* (HTML, CSS, JavaScript)
 - a. Implementasi antarmuka *dashboard* dengan fitur statistik, daftar perangkat, dan modal detail
 - b. Implementasi fitur download client code dengan form generator
- 4) Firmware Perangkat IoT (Platform I/O)
Implementasi kode client ESP8266/ESP32 untuk membaca sensor dan mengirim data via CoAP.

1.6.5 Pengujian Sistem

Tahap pengujian dilakukan dengan skenario sebagai berikut:

- 1) Pengujian Registrasi Perangkat

- a. Memastikan perangkat baru dapat didaftarkan melalui *dashboard*
- b. Memastikan token unik tergenerate untuk setiap perangkat
- 2) Pengujian Komunikasi CoAP
 - a. Menguji pengiriman data dari ESP8266 ke *server* menggunakan protokol CoAP
 - b. Menguji validasi token pada setiap pengiriman data
 - c. Menguji penolakan data dari perangkat tidak terdaftar
- 3) Pengujian *Dashboard*
 - a. Menguji tampilan statistik dan daftar perangkat
 - b. Menguji fitur detail perangkat dan grafik historis
 - c. Menguji fitur penghapusan perangkat
- 4) Pengujian Download Client Code
 - a. Menguji pembangkitan kode client berdasarkan konfigurasi yang diisi pengguna
 - b. Menguji hasil download file dengan ekstensi .txt.

1.6.6 Analisis Hasil

Tahap analisis meliputi evaluasi terhadap:

- 1) Keberhasilan implementasi protokol CoAP untuk komunikasi data sensor
- 2) Fungsionalitas seluruh fitur *dashboard* yang telah dikembangkan
- 3) Kemudahan pengguna dalam menyesuaikan kode client untuk berbagai jenis sensor
- 4) Kendala yang dihadapi selama pengembangan dan solusi yang diterapkan

1.7 Kontribusi Penelitian

Adapun kontribusi penelitian dalam proyek ini adalah:

- 1) Pengembangan sistem *cloud* IoT mandiri tanpa ketergantungan pada pihak ketiga
- 2) Implementasi protokol CoAP untuk komunikasi efisien pada perangkat terbatas
- 3) Sistem autentikasi dua lapis JWT ((*JSON Web Token*) + Token Device))
- 4) Fitur pembangkitan kode client universal untuk berbagai sensor
- 5) *Dashboard* real-time dengan *WebSocket* dan visualisasi *Chart.js*

1.8 Sistematika Penelitian

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan hasil penelitian terdahulu yang relevan dan landasan

teori yang diperlukan untuk merancang sistem.

BAB III PERANCANGAN

Bab ini menjelaskan pendekatan penelitian, lingkup penelitian, perancangan arsitektur sistem, komponen sistem, diagram alir, implementasi sistem, serta metode pengujian dan evaluasi sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil implementasi *server*, hasil implementasi client IoT, data pengujian fungsionalitas, data pengujian kinerja, data pengujian keamanan, data pengujian interoperabilitas, dan analisis terhadap seluruh hasil yang diperoleh.

BAB V KESIMPULAN

Bab ini berisi kesimpulan dari seluruh penelitian dan saran untuk pengembangan lebih lanjut