

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi saat ini menuntut organisasi dan perusahaan untuk memiliki sistem informasi yang efisien, fleksibel, serta mampu beradaptasi terhadap perubahan kebutuhan bisnis yang dinamis [1]. Salah satu aspek penting dalam organisasi adalah sistem pengelolaan sumber daya manusia atau *Human Resource* (HR), yang berperan dalam mengatur berbagai proses seperti perekrutan, manajemen data karyawan, absensi, penggajian, hingga evaluasi kinerja. Seiring bertambahnya jumlah karyawan dan kompleksitas proses bisnis, sistem HR konvensional sering kali menghadapi kendala dalam hal kinerja, skalabilitas, dan pemeliharaan [2].

Sebagian besar sistem HR yang digunakan saat ini masih menerapkan arsitektur *monolitik*, yaitu pendekatan di mana seluruh komponen sistem seperti modul kepegawaian, penggajian, dan absensi digabung menjadi satu aplikasi besar [3]. Pendekatan ini mudah diimplementasikan pada tahap awal pengembangan, namun akan menimbulkan berbagai permasalahan ketika sistem berkembang menjadi semakin besar dan kompleks. Masalah umum yang sering muncul antara lain waktu pengembangan yang lama, kesulitan dalam menambahkan fitur baru, serta risiko tinggi ketika melakukan pembaruan kode karena setiap perubahan dapat memengaruhi seluruh sistem [4]. Selain itu, sistem monolitik memiliki keterbatasan dalam hal skalabilitas, karena seluruh komponen harus ditingkatkan secara bersamaan tanpa dapat memisahkan beban kerja berdasarkan fungsinya.

Untuk mengatasi permasalahan tersebut, konsep arsitektur *microservices* menjadi salah satu solusi yang semakin populer dalam dunia pengembangan perangkat lunak modern [5]. *Microservices architecture* merupakan pendekatan yang membagi sistem besar menjadi beberapa layanan kecil (*services*) yang berjalan secara independen dan saling berkomunikasi melalui protokol ringan seperti *REST API* [6]. Setiap layanan memiliki tanggung jawab tertentu, misalnya layanan penggajian hanya menangani proses perhitungan gaji, sementara layanan absensi hanya berfokus pada pencatatan kehadiran karyawan. Dengan pemisahan tanggung jawab tersebut, sistem menjadi lebih mudah dikelola, diuji, dan dikembangkan tanpa mengganggu layanan lain.

Penerapan arsitektur *microservices* pada sistem HR dapat memberikan berbagai keuntungan, seperti peningkatan fleksibilitas, kecepatan pengembangan, dan kemampuan sistem dalam menyesuaikan diri terhadap perubahan kebutuhan organisasi [7]. Misalnya, ketika perusahaan ingin menambahkan fitur baru seperti sistem cuti daring atau evaluasi kinerja otomatis, pengembang cukup menambahkan layanan baru tanpa harus mengubah keseluruhan sistem. Selain itu, setiap layanan dapat dikembangkan menggunakan teknologi yang paling sesuai dengan kebutuhan masing-masing modul, sehingga sistem menjadi lebih adaptif terhadap perkembangan teknologi [8].

Salah satu pendekatan komunikasi antar layanan yang paling umum digunakan dalam arsitektur *microservices* adalah *Representational State Transfer Application Programming Interface* atau *REST API* [9]. *REST API* menggunakan prinsip komunikasi berbasis HTTP dengan format data seperti JSON yang ringan dan mudah dipahami. Dengan *REST API*, setiap layanan dapat saling bertukar data melalui permintaan dan tanggapan HTTP (*request-response*) tanpa perlu mengetahui detail implementasi dari layanan lain. Pendekatan ini tidak hanya memudahkan integrasi antar layanan, tetapi juga memungkinkan pengembang untuk membangun sistem yang terdistribusi dengan tingkat keterikatan rendah (*low coupling*) [6].

Dalam penelitian ini, teknologi yang digunakan untuk membangun sistem *microservices* adalah *Node.js*. *Node.js* merupakan lingkungan eksekusi berbasis JavaScript yang dirancang dengan arsitektur *event-driven* dan mekanisme *non-blocking I/O*, sehingga sangat efisien dalam menangani banyak permintaan secara bersamaan [10]. Keunggulan ini membuat *Node.js* sangat cocok untuk membangun sistem *microservices* yang membutuhkan komunikasi antar layanan secara intensif. Selain itu, *Node.js* memiliki ekosistem pustaka yang luas, termasuk *Express.js*, yang merupakan kerangka kerja ringan untuk membangun *REST API* dengan cepat dan terstruktur.

Dengan memanfaatkan arsitektur *microservices* dan teknologi *Node.js*, sistem HR dapat dikembangkan secara modular dan terdistribusi [11]. Setiap layanan, seperti manajemen karyawan, absensi, dan penggajian, dapat dikembangkan dan di-*deploy* secara terpisah. Hal ini tidak hanya mempermudah proses pengembangan, tetapi juga meningkatkan skalabilitas sistem karena setiap layanan dapat dijalankan pada server yang berbeda sesuai kebutuhan beban kerja. Selain itu, arsitektur ini juga mendukung penerapan *Continuous Integration* dan *Continuous Deployment* (CI/CD), yang memungkinkan pembaruan sistem

dilakukan secara cepat dan aman tanpa mengganggu layanan lain [12].

Dalam konteks PT Annisaa Putri Kaligayam, yang merupakan institusi layanan kesehatan di bidang medis dan sedang berada dalam tahap pengembangan dari klinik utama menuju klinik pratama hingga rumah sakit tipe D, kebutuhan akan sistem *Human Resource* (HR) yang terintegrasi dan andal menjadi semakin krusial. Proses transformasi tersebut berdampak langsung pada peningkatan jumlah tenaga medis dan non-medis, kompleksitas pengelolaan jadwal kerja berbasis *shift*, pencatatan absensi, serta proses penggajian yang harus menyesuaikan dengan kebutuhan operasional dan regulasi yang lebih ketat.

Pada kondisi sebelumnya, PT Annisaa Putri Kaligayam belum memiliki sistem informasi HR berbasis teknologi informasi. Seluruh proses pengelolaan data kepegawaian, absensi, dan penggajian masih dilakukan secara manual, seperti pencatatan menggunakan dokumen fisik serta pengolahan data dengan bantuan aplikasi perkantoran sederhana. Pendekatan manual ini berpotensi menimbulkan berbagai permasalahan, antara lain keterlambatan pemrosesan data, tingginya risiko kesalahan pencatatan, duplikasi data, serta kesulitan dalam menghasilkan laporan yang akurat dan tepat waktu.

Seiring dengan rencana pengembangan institusi menuju rumah sakit tipe D, pendekatan manual dalam pengelolaan HR menjadi tidak lagi memadai untuk menunjang kebutuhan operasional yang semakin kompleks. Oleh karena itu, dibutuhkan sistem HR berbasis teknologi informasi yang mampu mendukung pengelolaan data secara terpusat, akurat, serta mudah dikembangkan sesuai dengan pertumbuhan organisasi.

Dalam penelitian ini, sistem HR dirancang dan dibangun dengan mengadopsi arsitektur *microservices* sejak tahap awal pengembangan. Pemilihan arsitektur ini bertujuan untuk menyediakan fondasi sistem yang modular, fleksibel, dan mudah dikembangkan, sekaligus menghindari keterbatasan yang umum ditemui pada sistem terpusat (*monolithic*) ketika sistem mengalami pertumbuhan skala dan kompleksitas. Setiap fungsi utama HR dipisahkan ke dalam layanan mandiri, seperti layanan manajemen karyawan, layanan absensi, dan layanan penggajian, yang saling berkomunikasi melalui *Representational State Transfer Application Programming Interface* (REST API).

Dengan pendekatan tersebut, sistem HR dapat dikembangkan secara bertahap tanpa mengganggu layanan lainnya. Selain itu, setiap layanan dapat dipelihara dan ditingkatkan secara independen sesuai dengan kebutuhan operasional institusi.

Dengan demikian, penerapan arsitektur *microservices* berbasis *REST API* menggunakan *Node.js* pada sistem HR PT Annissaa Putri Kaligayam diharapkan mampu mendukung proses transformasi digital institusi secara berkelanjutan. Sistem yang dikembangkan tidak hanya menjawab kebutuhan operasional saat ini, tetapi juga siap untuk dikembangkan lebih lanjut seiring dengan peningkatan skala dan kompleksitas layanan kesehatan sebagai rumah sakit tipe D di masa mendatang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, sistem pengelolaan sumber daya manusia yang masih menggunakan arsitektur monolitik sering menghadapi berbagai kendala, seperti kesulitan dalam pengembangan fitur baru, keterbatasan dalam skalabilitas, serta kompleksitas tinggi dalam pemeliharaan sistem. Oleh karena itu, perlu dilakukan penelitian untuk menerapkan arsitektur *microservices* yang lebih modular dan fleksibel menggunakan *Node.js* dengan pendekatan *REST API* sebagai mekanisme komunikasi antar layanan.

Agar penelitian ini memiliki arah yang jelas, maka rumusan masalah yang akan dibahas dapat dirinci sebagai berikut:

1. Bagaimana merancang arsitektur sistem HR berbasis *microservices* yang modular dan terdistribusi?
2. Bagaimana proses implementasi arsitektur *microservices* menggunakan *Node.js* dengan komunikasi antar layanan berbasis *REST API*?
3. Bagaimana cara mengintegrasikan berbagai layanan seperti manajemen karyawan, absensi, dan penggajian dalam satu sistem yang saling berinteraksi secara efisien?
4. Bagaimana hasil evaluasi terhadap kinerja dan skalabilitas sistem HR setelah menerapkan arsitektur *microservices* dibandingkan dengan pendekatan monolitik secara konseptual?

1.3 Batasan Permasalahan

Agar penelitian ini memiliki fokus yang jelas dan dapat diselesaikan sesuai dengan waktu serta sumber daya yang tersedia, maka perlu ditetapkan beberapa

batasan masalah. Adapun batasan permasalahan dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada penerapan arsitektur *microservices* dengan menggunakan *Node.js* sebagai bahasa pemrograman utama.
2. Komunikasi antar layanan dalam sistem *microservices* dibatasi hanya menggunakan *REST API* sebagai protokol pertukaran data.
3. Layanan yang dikembangkan dalam sistem HR meliputi modul manajemen data karyawan, absensi, dan penggajian, tanpa mencakup seluruh aspek kompleks sistem HR secara menyeluruh.
4. Penelitian ini tidak membahas perbandingan kinerja antara *REST API*, *gRPC*, maupun *GraphQL*.
5. Pengujian sistem hanya difokuskan pada aspek fungsionalitas dasar dan integrasi antar layanan, tanpa melakukan pengujian beban (*load testing*) secara mendalam.
6. Implementasi sistem dilakukan dalam lingkungan pengujian lokal atau *development environment*, bukan pada infrastruktur produksi berskala besar.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan sistem pengelolaan sumber daya manusia (HR) berbasis arsitektur *microservices* dengan memanfaatkan *Node.js* dan *REST API*. Adapun tujuan spesifik dari penelitian ini adalah sebagai berikut:

1. Merancang arsitektur sistem HR yang terdistribusi dan modular dengan pendekatan *microservices*.
2. Mengimplementasikan komunikasi antar layanan menggunakan *REST API* sebagai media pertukaran data.
3. Membangun beberapa layanan utama dalam sistem HR, seperti manajemen data karyawan, absensi, dan penggajian, yang dapat berjalan secara independen namun tetap saling terintegrasi.

4. Melakukan pengujian terhadap fungsionalitas dan integrasi antar layanan untuk memastikan sistem berjalan sesuai rancangan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun praktis bagi berbagai pihak yang terlibat dalam bidang pengembangan perangkat lunak dan manajemen sistem informasi. Adapun manfaat penelitian ini adalah sebagai berikut:

1. Manfaat Akademis

Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang arsitektur perangkat lunak, khususnya dalam penerapan arsitektur *microservices* menggunakan *Node.js* dan *REST API*. Hasil penelitian ini dapat menjadi referensi bagi mahasiswa atau peneliti lain yang ingin mempelajari konsep dan implementasi *microservices* dalam konteks sistem informasi.

2. Manfaat Teknis

Menjadi acuan dalam merancang sistem berbasis *microservices* yang modular, terdistribusi, dan mudah dikembangkan. Implementasi ini juga memberikan gambaran nyata mengenai penerapan *REST API* sebagai mekanisme komunikasi antar layanan dalam sistem HR.

3. Manfaat Praktis

Memberikan solusi alternatif bagi organisasi atau perusahaan yang ingin melakukan migrasi dari sistem monolitik menuju arsitektur *microservices*, khususnya dalam konteks sistem HR. Dengan penerapan ini, diharapkan pengelolaan data karyawan, absensi, dan penggajian dapat dilakukan secara lebih efisien, terukur, dan mudah diintegrasikan.

4. Manfaat Pengembangan Sistem

Menjadi dasar pengembangan sistem HR yang lebih fleksibel dan skalabel di masa mendatang, serta mendorong penerapan arsitektur modern yang sesuai dengan tren industri perangkat lunak saat ini.

1.6 Sistematika Penulisan

Sistematika penelitian ini disusun untuk memberikan gambaran umum mengenai alur penyusunan laporan penelitian, agar pembaca dapat memahami isi dan arah penelitian secara terstruktur. Adapun sistematika penulisan dalam laporan ini adalah sebagai berikut:

1. Bab I – Pendahuluan

Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan. Bab ini bertujuan untuk memberikan gambaran umum mengenai alasan dilakukannya penelitian dan arah penelitian yang akan dilakukan.

2. Bab II – Tinjauan Pustaka

Bab ini membahas teori-teori yang mendukung penelitian, termasuk konsep arsitektur *microservices*, prinsip kerja *REST API*, teknologi *Node.js*, serta studi literatur mengenai penerapan arsitektur *microservices* pada sistem HR atau sistem informasi lainnya yang relevan. Bab ini juga mencakup penelitian-penelitian terdahulu yang menjadi acuan dalam penelitian ini.

3. Bab III – Metodologi Penelitian

Bab ini menjelaskan metode penelitian yang digunakan, termasuk tahapan penelitian, rancangan sistem, kebutuhan perangkat lunak dan perangkat keras, perancangan arsitektur *microservices*, serta pendekatan implementasi menggunakan *Node.js* dan *REST API*. Bab ini juga mencakup diagram alur penelitian dan langkah-langkah pengujian sistem.

4. Bab IV – Implementasi dan Pengujian Sistem

Bab ini membahas proses implementasi sistem HR berbasis arsitektur *microservices*, mulai dari konfigurasi lingkungan pengembangan, pembuatan layanan (*services*) utama seperti manajemen karyawan, absensi, dan penggajian, hingga integrasi antar layanan. Selain itu, dijelaskan pula hasil pengujian fungsionalitas dan integrasi antar layanan untuk memastikan sistem berjalan sesuai rancangan.

5. Bab V – Analisis dan Pembahasan

Bab ini berisi analisis terhadap hasil implementasi sistem yang telah dibangun, termasuk evaluasi terhadap fleksibilitas, skalabilitas, serta

kemudahan pengelolaan sistem berbasis arsitektur *microservices*. Pembahasan juga mencakup kelebihan dan kekurangan sistem berdasarkan hasil pengujian dan pengamatan.

6. **Bab VI - Kesimpulan dan Saran**

Bab ini menyajikan kesimpulan yang diperoleh dari hasil penelitian serta memberikan saran-saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut, baik dari segi teknis maupun konseptual.

