

## IMPLEMENTASI METODE MICROSERVICE UNTUK PENGEMBANGAN SISTEM INFORMASI KARYAWAN BERBASIS WEB

Nazwa Hanifah<sup>1</sup>, Rahma Santika<sup>2</sup> and, Yusuf Maulana<sup>3</sup>, and Saprudin<sup>4</sup>

<sup>1,2,3</sup>Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten, Indonesia, 15310  
e-mail: [1nzwahnfh5@gmail.com](mailto:nzwahnfh5@gmail.com), [2rahmasantika064@gmail.com](mailto:rahmasantika064@gmail.com), [3maulsr9@gmail.com](mailto:maulsr9@gmail.com),  
[4dosen00845@unpam.ac.id](mailto:dosen00845@unpam.ac.id)

### *Abstract*

*This study discusses the implementation of the microservice method for the development of a webbased employee information system at the Meteorology, Climatology, and Geophysics Agency (BMKG). In the current digital era, organizations are striving to improve operational efficiency and effectiveness through information technology, with microservices architecture offering a modular and flexible approach. The main objective of this project is to design and implement a system that streamlines employee attendance and leave application processes, integrating these features into independent yet interconnected microservices. The system utilizes Next.js for the frontend and FastAPI for the backend, ensuring responsive access, real-time data management, and enhanced security. The research methodology includes needs analysis, system architecture design, system development, and comprehensive testing. The results demonstrate that the microservice-based system significantly improves the efficiency of managing employee attendance and leave, providing real-time monitoring and easier data management for BMKG. The study concludes that adopting microservices architecture in employee information systems can enhance scalability, flexibility, and maintainability, meeting the evolving needs of the organization*

*Keywords: employee information system, fastAPI, microservices, next.js, web application*

### *Abstrak*

Penelitian ini membahas implementasi metode microservice untuk pengembangan sistem informasi karyawan berbasis web pada Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Di era digital saat ini, organisasi berupaya meningkatkan efisiensi dan efektivitas operasional melalui teknologi informasi, dengan *arsitektur microservices* sebagai pendekatan yang modular dan fleksibel. Tujuan utama proyek ini adalah merancang dan mengimplementasikan sistem yang mempermudah proses absensi dan pengajuan cuti karyawan, dengan integrasi fitur-fitur tersebut ke dalam microservice yang independen namun saling terhubung. Sistem dikembangkan menggunakan *Next.js* untuk frontend dan *FastAPI* untuk backend, sehingga memberikan akses yang responsif, pengelolaan data secara *real-time*, serta keamanan yang lebih baik. Metodologi penelitian meliputi analisis kebutuhan, perancangan arsitektur sistem, pengembangan sistem, dan pengujian menyeluruh. Hasil penelitian menunjukkan bahwa sistem berbasis *microservice* secara signifikan meningkatkan efisiensi pengelolaan absensi dan cuti karyawan, memberikan pemantauan *real-time*, serta memudahkan manajemen data di lingkungan BMKG. Penelitian ini menyimpulkan bahwa penerapan *arsitektur microservice* pada sistem informasi karyawan dapat meningkatkan skalabilitas,

fleksibilitas, dan kemudahan pemeliharaan, sehingga mampu memenuhi kebutuhan organisasi yang terus berkembang.

Kata Kunci: sistem informasi karyawan, FastAPI, microservice, Next.js, aplikasi web

## 1. PENDAHULUAN

Dalam era digital saat ini, organisasi dituntut untuk terus meningkatkan efisiensi dan efektivitas operasional melalui pemanfaatan teknologi informasi yang tepat guna. Salah satu pendekatan yang semakin banyak diadopsi adalah *arsitektur microservice*, yang menawarkan modularitas, fleksibilitas, serta kemudahan integrasi antar layanan dalam pengembangan sistem informasi. Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) sebagai lembaga pemerintah yang memiliki jumlah pegawai besar menghadapi tantangan dalam pengelolaan data absensi dan pengajuan cuti karyawan secara efektif dan realtime.

Sistem informasi karyawan yang masih bersifat monolitik seringkali mengalami kendala dalam skalabilitas, pengembangan fitur baru, serta pemeliharaan sistem secara berkelanjutan. Permasalahan lain yang dihadapi adalah kurangnya layanan yang terpisah dan mandiri, sehingga proses pengelolaan absensi dan cuti menjadi kurang efisien serta pengambilan keputusan sering terlambat akibat pemantauan data yang belum optimal.

Penerapan metode *microservice* dalam pengembangan sistem informasi karyawan di BMKG bertujuan untuk menciptakan solusi yang memudahkan proses absensi dan pengajuan cuti secara *online*, serta memberikan kemudahan bagi manajemen dalam memantau secara *realtime*. Dengan membangun sistem berbasis *web* menggunakan *Next.js* pada *frontend* dan *FastAPI* pada *backend*, diharapkan sistem yang dikembangkan mampu memberikan akses yang responsif, keamanan data yang lebih baik, serta kemudahan dalam pengelolaan dan pengembangan sistem ke depannya.

Penelitian ini akan membahas proses perancangan dan implementasi sistem informasi karyawan berbasis web dengan pendekatan *microservice* di lingkungan BMKG, serta

mengevaluasi kinerja dan manfaatnya dalam mendukung kebutuhan organisasi yang dinamis

## 2. LANDASAN TEORI

### a. *Microservice*

*Microservice* adalah arsitektur pengembangan perangkat lunak yang membagi sistem besar menjadi layanan-layanan kecil yang independen dan berfungsi secara spesifik sesuai tugasnya. Setiap layanan bekerja secara otonom dan berinteraksi melalui API berbasis protokol *HTTP/REST*. Keuntungan utama dari *arsitektur microservice* adalah skalabilitas, fleksibilitas pengembangan dan pemeliharaan, serta ketahanan terhadap kegagalan bagian tertentu. Sistem *microservice* cocok digunakan dalam pengembangan aplikasi berbasis web yang memerlukan skalabilitas tinggi dan perubahan yang cepat.

### b. *Entity Relationship Diagram (ERD)* dan Relasi Tabel

ERD adalah diagram yang menunjukkan struktur data dan hubungan antar entitas dalam sistem. Dalam konteks pengelolaan absensi dan cuti, entitas utama adalah Karyawan, Absensi, dan Cuti. Relasi antar entitas biasanya adalah *oneto-many* (1:N), di mana satu karyawan dapat memiliki banyak data absensi dan pengajuan cuti.

### c. *UML (Unified Modeling Language)*

UML digunakan untuk memvisualisasikan desain sistem secara lengkap, seperti:

- 1) *Use Case Diagram* untuk menggambarkan interaksi pengguna dengan sistem.
- 2) *Sequence Diagram* yang menunjukkan bagaimana komponen berinteraksi seiring waktu, misalnya proses pengajuan cuti dari pengguna sampai data tersimpan dan notifikasi terkirim.

- 3) *Activity Diagram* yang memperlihatkan alur langkah-langkah kegiatan pengguna dan sistem, misalnya proses pengisian formulir, validasi data, dan pemberian status persetujuan

#### d. Perancangan

Perancangan dalam konteks sistem berbasis arsitektur *microservice* adalah proses merancang atau membuat rancangan sistem yang terdiri dari beberapa layanan mandiri (*independent services*) yang dapat dikembangkan, dikelola, dan diterapkan secara independen satu sama lain. Pada tahap perancangan, dilakukan identifikasi komponen-komponen utama yang akan menjadi servis mandiri, perancangan model arsitektur *microservices*, serta pembuatan diagram seperti UML (Use Case, Activity, Sequence, dan Class Diagram), ERD, dan perancangan database untuk tiap layanan. Tujuan perancangan adalah memastikan sistem yang dibangun modular, efisien, *scalable*, dan sesuai dengan kebutuhan pengguna

#### e. Sistem

Sistem adalah sekumpulan elemen, komponen, atau variabel yang saling berhubungan, berinteraksi, dan bekerja sama dalam satu kesatuan untuk mencapai tujuan tertentu. Sistem dapat terdiri dari manusia, mesin, prosedur, dokumen, data, atau elemen lain yang terorganisasi dan saling bergantung satu sama lain

#### f. Sistem Informasi

Sistem informasi adalah kumpulan elemen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan data sehingga menjadi informasi yang berguna bagi pengambilan keputusan di organisasi. Dalam konteks absensi dan pengajuan cuti, sistem ini memudahkan proses pencatatan kehadiran, serta persetujuan cuti secara digital, sehingga lebih efisien,

akurat, dan mudah diakses kapan pun dibutuhkan.

#### g. Node.js

*Node.js* adalah *runtime environment* untuk *JavaScript* yang memungkinkan pengembangan aplikasi *server-side* dengan performa tinggi dan *non-blocking I/O*, sangat cocok untuk membangun API yang cepat dan *scalable*.

#### h. JavaScript

*JavaScript* merupakan bahasa pemrograman inti untuk pengembangan web modern, baik di sisi client maupun server, dan digunakan hampir di seluruh *website* untuk membuat aplikasi lebih interaktif dan dinamis.

#### i. Bootstrap

*Bootstrap* adalah *framework* CSS yang digunakan untuk mempercepat pengembangan antarmuka web yang *responsif* dan konsisten. *Bootstrap* menyediakan komponen siap pakai seperti grid, tombol, form, dan navigasi, sehingga tampilan aplikasi menjadi menarik, ringan, dan mudah diadaptasi ke berbagai ukuran layar perangkat.

#### j. Next.js sebagai Framework Frontend

*Next.js* adalah *framework* berbasis *React* yang mendukung *server-side rendering* (SSR) dan *static site generation* (SSG), sehingga aplikasi web lebih optimal, cepat, dan *SEO-friendly*. *Next.js* memudahkan integrasi dengan *API backend*, mendukung *code splitting*, dan memberikan pengalaman pengguna yang lebih baik melalui performa dan keamanan yang tinggi.

#### k. FastAPI sebagai Framework Backend

*FastAPI* adalah *framework web* berbasis *Python* yang digunakan untuk membangun *RESTful API* dengan performa tinggi dan sintaks yang sederhana. *FastAPI* mendukung *asynchronous programming*,

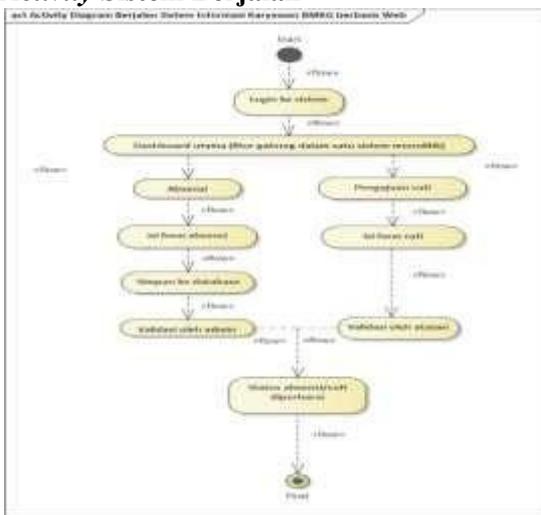
validasi otomatis, serta dokumentasi API yang terintegrasi, sehingga mempercepat proses pengembangan *backend modern*.

### 3. ANALISA DAN PERANCANGAN

#### 3.1 Analisa sistem

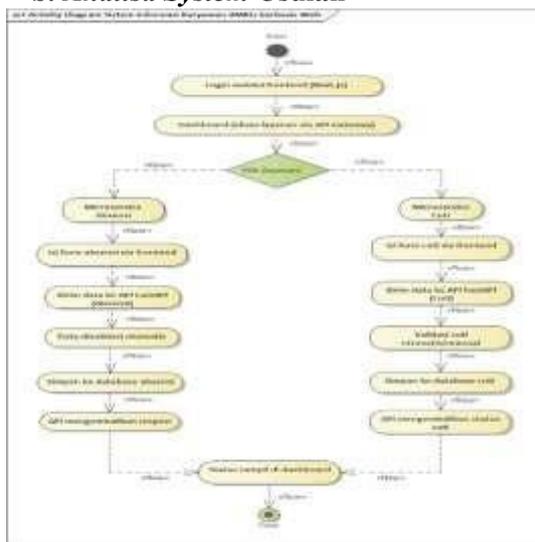
Analisis sistem dilakukan untuk memahami kondisi serta proses bisnis yang berlangsung saat ini. Tujuannya adalah mengevaluasi kebutuhan sistem yang akan dikembangkan agar sesuai dengan kebutuhan pengguna.

##### a. Activity System Berjalan



Gambar 1 Activity Sistem Berjalan

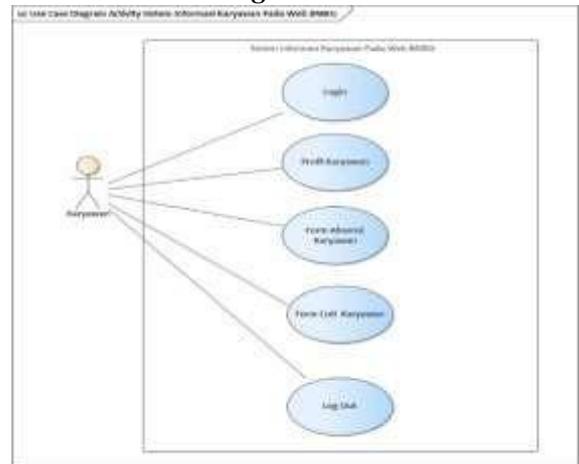
##### b. Analisa System Usulan



Gambar 2 Analisa System Usulan

### 3.2 Perancangan Sistem

#### c. Use Case Diagram

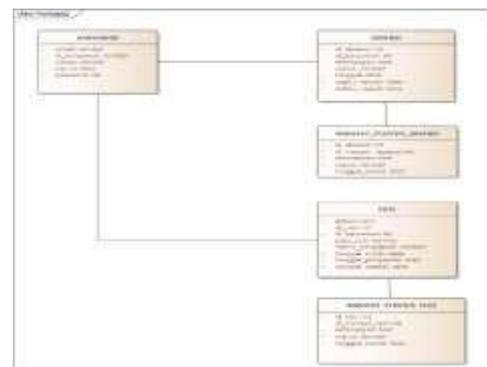


Gambar 3 Use Case Diagram

Aktor (*Use case diagram* digunakan untuk mendeskripsikan cakupan sistem dan kebutuhan fungsional yang akan dikembangkan)

#### d. Entity Relationship Diagram (ERD)

ERD menggambarkan hubungan one-tomany antar entitas (misalnya satu anggota bisa punya banyak catatan absensi).



Gambar 4 ERD

## 4. IMPLEMENTASI DAN TESTING

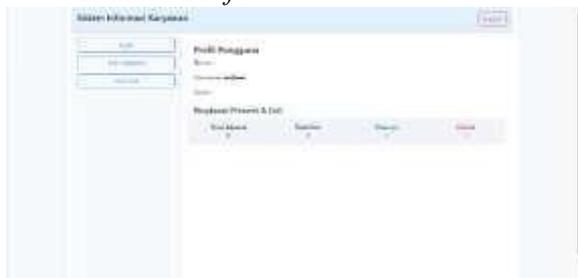
### 4.1 Implementasi

#### 1. Halaman Login



Gambar 5 Halaman Login

#### 2. Menu Profil



Gambar 7 Menu Profil

#### 3. From Absensi



Gambar 8 From Absensi

#### 4. Tabel Absensi

Total Absensi	Total Cuti
11	7

Gambar 9 Tabel Absensi

#### 5. From Cuti



Gambar 10 From Cuti

#### 6. Tabel Cuti

Total Cuti	Disetujui	Ditolak
7	1	1

Gambar 11 Tabel Cuti

**4.2 Testing**

No	Halaman	Skenario	Test Case	Hasil yang di Harapkan	Hasil Pengujian	Kesimpulan
1.	Login	Pengguna login dengan username dan password yang benar	Masukkan username dan password valid, lalu klik tombol login	Pengguna berhasil masuk ke dashboard utama	Pengguna diarahkan ke dashboard utama	Berhasil
		Pengguna login dengan username atau password salah	Masukkan username salah atau password salah, lalu klik tombol login	Muncul pesan error "Username atau Password salah"	Pesan error muncul sesuai ekspektasi	Gagal
2.	Profil	Pengguna mengakses halaman profil	Klik menu "Profil" setelah login	Data diri pengguna ditampilkan lengkap	Data tampil sesuai database	Berhasil
3.	Absensi	Pengguna melakukan absensi	Klik tombol "Absen Masuk"	Data absensi tersimpan dan muncul notifikasi berhasil	Data berhasil tersimpan dan notifikasi tampil	Berhasil
		Pengguna mengisi form absensi dengan data kosong	Klik tombol "Absen Masuk" tanpa mengisi form	Muncul pesan peringatan "Data tidak boleh kosong"	Validasi berjalan dan pesan tampil	Berhasil

4.	Cutu	Pengguna mengajukan cutu dengan data lengkap	Isi formulir pengajuan cutu lalu klik tombol “Ajukan”	Permintaan cutu tersimpan dan menunggu persetujuan	Data cutu tersimpan, status menunggu persetujuan	Berhasil
		Pengguna mengajukan cutu tanpa mengisi semua kolom wajib	Klik tombol “Ajukan” tanpa mengisi kolom keterangan	Muncul pesan peringatan “Harap lengkapi data pengajuan cutu”	Validasi berhasil dan sistem menolak pengajuan	Berhasil
5.	Logout	Pengguna logout dari sistem	Klik menu “Logout”	Pengguna kembali ke halaman login	Pengguna diarahkan ke halaman login setelah logout	Berhasil

## 5. KESIMPULAN

Penelitian ini menunjukkan bahwa implementasi metode microservice dalam pengembangan Sistem Informasi Karyawan Berbasis Web di BMKG berhasil mengatasi permasalahan utama yang ada pada sistem sebelumnya. Arsitektur microservice yang diterapkan terbukti lebih efisien, aman, dan andal dalam mendukung proses absensi dan pengajuan cuti karyawan. Pemisahan layanan menjadi subsistem Absensi dan Pengajuan Cuti memungkinkan setiap layanan berjalan secara independen namun tetap terintegrasi dengan baik melalui API, sehingga meningkatkan skalabilitas dan mempermudah pengembangan fitur baru.

Penggunaan teknologi Next.js pada frontend dan FastAPI pada backend memudahkan pengelolaan data secara real-time, mempercepat proses pengajuan cuti, serta memudahkan monitoring absensi oleh manajemen. Hasil

## DAFTAR PUSTAKA

- [1] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
- [2] Richardson, C. (2019). Microservices Patterns: With examples in Java. Manning Publications.
- [3] Tiangolo, S. (2021). FastAPI Documentation. Diakses dari: <https://fastapi.tiangolo.com/>
- [4] Vercel. (2023). Next.js Documentation. Diakses dari: <https://nextjs.org/docs>
- [5] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education Limited.
- [6] Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.
- [7] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- [8] Sari, A. (2021). Perancangan Sistem Absensi Berbasis Web Menggunakan Microservice. Jurnal Teknologi dan Sistem Informasi, 9(2), 134–140.
- [9] Budi, R. (2020). Implementasi REST API dengan FastAPI untuk Sistem Pengajuan Cuti Digital. Jurnal Informatika dan

- Komputasi, 7(1), 88–95 [10] Dewi, L. (2022). Pengembangan Sistem Manajemen Kepegawaian Berbasis Web dengan Next.js. Prosiding Seminar Nasional Teknologi Informasi, 5(1), 101–110.
- [11] Putra, F. (2019). Arsitektur Microservice dalam Sistem Informasi Organisasi. Jurnal Rekayasa Sistem dan Teknologi Informasi, 6(3), 45–52.
- [12] Chonoles, M. J., & Schardt, J. A. (2003). UML 2 for Dummies. Wiley Publishing.
- [13] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
- [14] Wahana Komputer. (2020). Panduan Lengkap Pemrograman Web Menggunakan JavaScript dan Node.js. Andi Publisher.
- [15] W3Schools. (2024). Bootstrap Tutorial. Diakses dari: <https://www.w3schools.com/bootstrap/>