BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Integrasi Kecerdasan Buatan (*Artificial Intelligence*/AI) ke dalam sistem produksi modern telah berkembang dari sekadar prototipe penelitian menjadi komponen inti pada berbagai aplikasi dunia nyata. Namun, transisi dari pengembangan model ke tahap *deployment production* menghadapi tantangan signifikan, dengan studi menunjukkan bahwa sebagian besar model AI gagal mencapai tahap *production* dan digunakan oleh *end users* [1, 2]. Implementasi AI pada lingkungan produksi menuntut infrastruktur *backend* yang mampu menyajikan model secara efisien dan stabil dalam skala besar [3]. Dalam arsitektur berbasis *microservices*, penyajian model AI menjadi bagian penting dari alur pemrosesan data yang memerlukan orkestrasi sumber daya komputasi yang matang untuk menjamin keberlanjutan layanan [4].

Salah satu domain kritis yang memanfaatkan AI adalah bidang medis, khususnya diagnosis kanker, di mana akurasi dan kecepatan respons sistem menjadi faktor penentu keberhasilan deteksi dini. Penerapan *machine learning* untuk diagnosis kanker telah menunjukkan hasil yang menjanjikan, dengan model *deep learning* berbasis *weakly supervised learning* mencapai akurasi hingga 97,3% dalam klasifikasi citra kanker paru-paru [5]. Sistem diagnosis kanker berbasis AI seperti AIRA (*AI Research of Oncology*) mengintegrasikan berbagai algoritma *machine learning* termasuk *Support Vector Machine* (SVM), *Logistic Regression* (LR), *Random Forest*, *Multi-Layer Perceptron* (MLP), XGBoost, dan *Voting Classifier* untuk menyediakan diagnosis pada berbagai jenis kanker seperti prostat, payudara, kandung kemih, dan adenokarsinoma paru [6]. Namun, implementasi sistem diagnosis kanker berbasis web yang melayani banyak pengguna secara konkuren menghadapi tantangan infrastruktur *backend* yang signifikan, terutama ketika proses inferensi model AI yang intensif secara komputasi harus diselesaikan dalam waktu respons yang dapat diterima untuk aplikasi medis *real-time*.

Proses inferensi AI pada sistem diagnosis kanker sering kali menjadi *bottleneck* utama dalam sistem *backend* karena sifatnya yang intensif secara komputasi, terutama ketika harus memproses citra medis beresolusi tinggi atau menganalisis data genomik kompleks. *Backend* modern dituntut untuk

memenuhi persyaratan latensi rendah dan *throughput* tinggi agar mampu menangani permintaan pengguna secara simultan, yang menjadi sangat kritis dalam konteks aplikasi medis di mana penundaan dapat mempengaruhi keputusan klinis. Ketika arsitektur *backend* tidak dioptimalkan dengan baik, peningkatan konkurensi pengguna dapat menyebabkan lonjakan waktu respons dan penurunan stabilitas sistem. Studi kasus pada sistem pemrosesan citra menunjukkan bahwa pendekatan arsitektur konvensional dapat mengalami waktu respons hingga 3.7 detik per *request*, yang tidak dapat diterima untuk aplikasi *real-time* [7, 8].

Salah satu penyebab utama permasalahan tersebut adalah penggunaan model *request-response* sinkron yang bersifat *blocking*. Pada pendekatan ini, klien harus menunggu hingga seluruh proses inferensi selesai sebelum menerima respons HTTP. Studi performa menunjukkan bahwa skenario *blocking* dapat membatasi kapasitas sistem dan meningkatkan latensi secara signifikan pada beban tinggi, dengan pendekatan *asynchronous* terbukti mampu meningkatkan *throughput* hingga 75% dan mengurangi penggunaan memori hingga 68% dibandingkan model sinkron tradisional [9]. Ketergantungan pada alur sinkron ini menyulitkan sistem untuk melakukan skalabilitas dinamis dan meningkatkan risiko kegagalan layanan, terutama pada kondisi beban tinggi dengan konkurensi pengguna yang masif.

Berbagai penelitian telah mengeksplorasi pendekatan arsitektural untuk mengatasi masalah tersebut, seperti penerapan pemrosesan *asynchronous* dan penggunaan *message broker* guna mendekopel komunikasi antar layanan [10, 11]. Penggunaan *dedicated worker processes* dengan *message queue* terbukti mampu mengurangi waktu respons API hingga 70–85%, dengan peningkatan *throughput* mencapai 370% pada kasus pemrosesan citra yang diproses secara *asynchronous* [7]. Selain itu, strategi *caching* terbukti efektif dalam mengurangi redundansi komputasi dengan menyimpan hasil inferensi sebelumnya sehingga dapat menurunkan latensi respons hingga 71.8% pada aplikasi dengan *traffic* tinggi [12].

Meskipun demikian, sebagian besar penelitian berfokus pada optimisasi infrastruktur *cloud* secara umum, platform *serverless*, atau implementasi pada *environment* berbasis Java, dan belum secara spesifik mengkaji optimisasi *backend* inferensi pada arsitektur hibrida yang menggabungkan API *Gateway* berbasis Node.js dengan layanan inferensi FastAPI berbasis Python [3, 13, 9]. Penelitian sebelumnya juga belum mengeksplorasi secara komprehensif kombinasi pemrosesan *asynchronous* dengan *caching* Redis untuk *inference serving* dalam konteks arsitektur *microservices* hibrida, khususnya untuk aplikasi diagnosis medis

yang memerlukan akurasi tinggi dan latensi rendah.

Berbagai platform diagnosis kanker berbasis AI telah dikembangkan secara komersial dan diterapkan di institusi medis global, dengan fokus utama pada peningkatan akurasi diagnostik dan deteksi biomarker. PathAI menyediakan AISight Dx, sebuah platform manajemen citra patologi digital yang telah memperoleh persetujuan FDA 510(k) untuk digunakan dalam diagnosis primer di lingkungan klinis serta sertifikasi CE mark di Eropa [14, 15]. Paige.AI mengembangkan Paige Prostate, sistem deteksi kanker prostat berbasis citra histopatologi digital yang dilatih menggunakan arsip digital Memorial Sloan Kettering Cancer Center, memperoleh otorisasi FDA melalui jalur *De Novo* setelah studi klinis menunjukkan peningkatan deteksi kanker sebesar 7,3% dibandingkan diagnosis manual oleh patolog [16, 17]. Lunit menyediakan rangkaian solusi AI untuk onkologi, mencakup INSIGHT CXR dan INSIGHT MMG untuk deteksi dini kanker berbasis *medical imaging* yang telah memperoleh persetujuan FDA dan CE mark serta diintegrasikan ke lebih dari 130 fasilitas kesehatan di Jepang, serta SCOPE untuk analisis biomarker presisi pada jaringan kanker [18, 19]. Platform-platform ini secara konsisten berfokus pada optimasi algoritma model AI untuk meningkatkan sensitivitas dan spesifisitas deteksi, dengan investasi besar pada pengembangan model *deep learning* yang mampu mengidentifikasi pola subvisual pada citra histopatologi.

Berbeda dengan pendekatan tersebut, penelitian ini tidak berfokus pada peningkatan akurasi model AI, melainkan pada optimasi infrastruktur *backend* untuk meningkatkan *throughput* sistem dan mengurangi latensi layanan tanpa mengubah model yang telah tervalidasi secara klinis. Keterbatasan platform komersial yang ada terletak pada kemampuan *scalability* ketika harus melayani volume permintaan tinggi secara simultan aspek yang sangat kritis untuk *deployment* di negara berkembang dengan rasio patolog rendah dan volume pasien tinggi. Studi terbaru menunjukkan bahwa satu dari tiga kasus kanker di dunia tidak terdiagnosis, dengan angka tersebut melampaui 60% di sebagian wilayah negara berkembang, sebagian besar disebabkan oleh kekurangan tenaga diagnostik seperti radiolog dan patolog yang diproyeksikan mencapai 16 juta orang secara global hingga 2050 [20]. Kesenjangan ini menegaskan bahwa sistem diagnosis berbasis AI harus mampu menangani beban konkuren yang masif untuk memberikan dampak klinis yang signifikan di wilayah dengan keterbatasan sumber daya medis. Penelitian ini mengeksplorasi kombinasi *asynchronous processing* dan *result caching* dalam konteks arsitektur *hybrid* Node.js–FastAPI yang belum

pernah dikaji secara komprehensif dalam literatur, dengan tujuan memungkinkan sistem AIRA untuk *scale up* melayani lebih banyak pasien tanpa degradasi performa *enabling factor* penting untuk aksesibilitas layanan diagnosis kanker di wilayah dengan infrastruktur kesehatan yang terbatas namun kebutuhan layanan kesehatan yang tinggi.

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengkaji dan mengimplementasikan optimisasi *backend* inferensi AI melalui pemisahan alur respons HTTP dan eksekusi inferensi menggunakan pemrosesan *asynchronous*, serta penerapan mekanisme *caching* Redis pada arsitektur hibrida Node.js–FastAPI dalam konteks sistem AIRA (*AI Research of Oncology*). AIRA merupakan platform diagnosis kanker berbasis web yang mengintegrasikan delapan model *machine learning* untuk empat jenis kanker (prostat, payudara, kandung kemih, dan adenokarsinoma paru) dengan rencana ekspansi ke jenis kanker lainnya [6]. Diharapkan pendekatan optimasi ini dapat meningkatkan stabilitas sistem, menurunkan latensi, dan meningkatkan *throughput* dibandingkan pendekatan sinkron konvensional, dengan target peningkatan performa yang sebanding dengan penelitian sebelumnya, sehingga mampu melayani kebutuhan diagnosis kanker berbasis AI yang memerlukan respons cepat dan andal untuk mendukung keputusan klinis.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi *asynchronous processing* dan *caching* pada implementasi AIRA?
2. Berapa perbandingan peningkatan performa model arsitektur dengan menerapkan *asynchronous processing* dan *caching* pada sistem *backend inference*?

1.3 Batasan Permasalahan

Untuk menjaga fokus penelitian, batasan permasalahan dalam penelitian ini adalah sebagai berikut:

1. Penelitian difokuskan pada optimasi *backend inference* AI tanpa melakukan perubahan pada struktur maupun parameter model AI pada website AIRA.
2. Sistem *backend* yang diteliti menggunakan arsitektur *microservices* dengan *API Gateway* berbasis Node.js dan layanan *inference* berbasis FastAPI.
3. Mekanisme *caching* dibatasi pada penyimpanan hasil *inference* menggunakan Redis sebagai *in-memory cache*.
4. Evaluasi performa dilakukan menggunakan metrik latensi, *throughput*, dan tingkat keberhasilan permintaan melalui pengujian beban (*load testing*).
5. Penelitian ini tidak membahas optimasi pada tingkat infrastruktur *cloud* maupun perangkat keras.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan mekanisme *asynchronous processing* dan *caching* pada sistem AIRA dalam arsitektur *backend inference* berbasis *microservices*.
2. Mengevaluasi peningkatan performa sistem *backend inference* setelah penerapan *asynchronous processing* dan *caching* berdasarkan metrik latensi, *throughput*, dan tingkat keberhasilan permintaan.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi ilmiah dalam bidang optimasi arsitektur *backend inference* AI pada sistem berbasis *microservices*.
2. Menjadi referensi bagi penelitian selanjutnya terkait peningkatan performa sistem AI tanpa melakukan modifikasi pada model *machine learning*.
3. Memberikan panduan praktis bagi pengembang dalam merancang sistem *backend inference* AI yang efisien dan skalabel.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini disusun sebagai berikut:

- **Bab 1 Pendahuluan**
  Bab ini berisi latar belakang masalah, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.
- **Bab 2 Tinjauan Pustaka**
  Bab ini membahas landasan teori yang mendukung penelitian, meliputi konsep *backend inference* AI, arsitektur *microservices*, *asynchronous processing*, mekanisme *caching*, serta penelitian terkait.
- **Bab 3 Metodologi Penelitian**
  Bab ini menjelaskan metode penelitian yang digunakan, termasuk perancangan arsitektur sistem, implementasi optimasi, skenario pengujian, serta metode evaluasi performa.
- **Bab 4 Hasil dan Pembahasan**
  Bab ini memaparkan hasil dan pembahasan terkait implementasi sistem serta hasil pengujian performa sebelum dan sesudah optimasi.
- **Bab 5 Kesimpulan dan Saran**
  Bab ini menyajikan kesimpulan dari penelitian dan saran untuk pengembangan lebih lanjut.